МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Национальный исследовательский Нижегородский государственный
университет им. Н.И. Лобачевского

В.Н. Якимов

ОСНОВЫ АНАЛИЗА БИОМЕДИЦИНСКИХ И ЭКОЛОГИЧЕСКИХ ДАННЫХ В СРЕДЕ R

ЧАСТЬ 2

Учебное пособие

Рекомендовано ученым советом Института биологии и биомедицины для студентов ННГУ, обучающихся по направлениям подготовки 06.03.01 «Биология», 05.03.06 «Экология и природопользование», 30.05.01 «Медицинская биохимия», 30.05.02 «Медицинская биофизика», 30.05.03 «Медицинская кибернетика»

Нижний Новгород
2019

УДК 57.087.1:519.2(075.8)
ББК 28с:22.172я73
Я 45

Я 45 Якимов В.Н. ОСНОВЫ АНАЛИЗА БИОМЕДИЦИНСКИХ И ЭКОЛОГИЧЕСКИХ ДАННЫХ В СРЕДЕ R. ЧАСТЬ 2: Учебное пособие – Н. Новгород: Нижегородский госуниверситет, 2019. – 169 с.

Рецензенты: д.б.н. **Шитиков В.К.**
д.б.н. **Воденеев В.А.**
д.ф.-м.н., д.б.н. **Иудин Д.И.**

Настоящее учебное пособие имеет целью ознакомление студентов с основами синтаксиса языка программирования R, а также с основами работы в интегрированной среде разzработки RStudio; изучение основных понятий, принципов и методов анализа данных в биомедицинских и экологических исследованиях; приобретение практических навыков анализа данных в среде вычислений R.

Учебное пособие предназначено для студентов очной формы обучения, обучающихся по направлениям подготовки 06.03.01 «Биология», 05.03.06 «Экология и природопользование», 30.05.01 «Медицинская биохимия», 30.05.02 «Медицинская биофизика», 30.05.03 «Медицинская кибернетика».

Ответственный за выпуск:
председатель методической комиссии ИББМ ННГУ
к.б.н. Водеенеева Е.Л.

УДК 57.087.1:519.2(075.8)
ББК 28с:22.172я73

© Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского, 2019
© В.Н. Якимов, 2019

Оглавление

Глава 1. Вероятности и распределения 5
1.1. Определения вероятности 5
1.2. Распределение вероятностей 8
1.3. Семейства функций для работы с распределениями 10
Глава 2. Описание данных 17
2.1. Описание количественных переменных 19
2.2. Описание качественных переменных 24
Глава 3. Нормальное распределение и диагностика отклонений 30
3.1. Свойства нормального распределения 30
3.2. Визуальные средства диагностики отклонений 34
3.3. Критерии согласия 38
Глава 4. Простейшие гипотезы: одно- и двухвыборочные сравнения 44
4.1. Критерий Стьюдента для сравнения среднего с эталоном 44
4.2. Критерий Стьюдента для сравнения двух выборочных средних 50
4.3. Критерий Стьюдента для сравнения связанных выборок 52
4.4. Непараметрические аналоги: критерии Уилкоксона и Манна-Уитни 55
Глава 5. Сравнение нескольких групп: однофакторный дисперсионный анализ и критерий Крускала-Уоллиса 59
5.1. Основы дисперсионного анализа 59
5.2. Реализация дисперсионного анализа в R 61
5.3. Предположения дисперсионного анализа 63
5.4. Дисперсионный анализ в группах с гетерогенной дисперсией 66
5.5. Непараметрический аналог: критерий Крускала-Уоллиса 67
Глава 6. Множественные сравнения 69
6.1. Поправки Бонферрони и Холма 70
6.2. Критерий Тьюки 73
6.3. Критерий Ньюмана-Кейлса 75
6.4. Множественные сравнения с учетом зависимости данных 78
6.5. Доверительные интервалы и визуализация результатов 83
6.6. Сравнения с контрольной группой 88
6.7. Непараметрические аналоги 92
Глава 7. Простая регрессия и корреляция 98
7.1. Основы регрессионного анализа 98
7.2. Построение и анализ линейной модели с помощью функции lm() 100
7.3. Остатки и прогнозируемые значения 103
7.4. Прогнозы и доверительные интервалы 105

7.5. Предположения регрессионного анализа и их диагностика 108
7.6. Корреляционный анализ 114
Глава 8. Анализ комплекса признаков 118
8.1. Анализ матрицы корреляций 118
8.2. Множественная регрессия: подбор линейной модели 124
8.3. Пошаговые процедуры поиска оптимальной модели 128
Глава 9. Статистика нечисловых данных 135
9.1. Сравнение доли с эталоном 135
9.2. Сравнение выборочных долей 138
9.3. Критерий χ^2 для сравнения с теоретическим распределением 141
9.4. Анализ зависимости между категориальными переменными 143
Глава 10. Элементы многомерного анализа 146
10.1. Многомерный дисперсионный анализ 146
10.2. Анализ главных компонент 151
10.3. Линейный дискриминантный анализ 158
Список рекомендуемой литературы 167
Программное обеспечение и Интернет-ресурсы 168

Глава 1. Вероятности и распределения

Современный анализ данных основан на использовании аппарата математической статистики – раздела математики, который является своего рода практическим расширением теории вероятностей. Для правильного понимания методов анализа данных, которые будут описаны во второй половине настоящего пособия, необходимо хотя бы кратко ознакомиться с минимальным набором понятий теории вероятностей.

1.1. Определения вероятности

Исторически теория вероятностей развивалась как теория азартных игр – карты, рулетка, кости, банальная игра в орлянку... Так или иначе, все эти игры сводятся к так называемой классической схеме. Для того, чтобы ситуация ей соответствовала, необходимо выполнение следующих условий:

(1) в результате опыта может произойти одно из конечного набора элементарных исходов (например, при игре в кости могут выпасть цифры от 1 до 6, в карточных играх может быть выбрана одна из 52 карт и т.д.);

(2) элементарные исходы равновозможны.

В рамках классической схемы применяется классическое определение вероятности: вероятностью события А называется отношение числа *m* благоприятствующих этому событию элементарных исходов к общему числу *n* элементарных исходов:

$$P(A) = \frac{m}{n}.$$

Например, для нахождения вероятности выпадения черного на рулетке нужно посчитать число благоприятствующих событию элементарных исходов (число черных секторов, $m = 18$) и отнести его к общему числу элементарных исходов ($n = 37$).

Другое определение вероятности является статистическим. Пусть проведена серия из *n* независимых опытов и событие А произошло *m* раз. Число *m* называется натуральной частотой события А, а его относительной частотой называется отношение

$$W_n(A) = \frac{m}{n}.$$

Число *P(A)*, к которому стремится относительная частота события А при росте числа опытов *n*, называется статистической вероятностью события А:

$$P(A) = \lim_{n\to\infty} W_n(A).$$

В силу случайного характера событий при небольшом числе опытов возможен существенный разброс, однако при росте числа опытов относительная частота стабилизируется и приближается к статистической вероятности.

Опыты в рамках классической схемы легко смоделировать в среде R. Пусть у нас имеется вектор, элементами которого являются элементарные исходы некоторого опыта, тогда для моделирования этого опыта можно воспользоваться функцией `sample()`, которая осуществляет случайную выборку. Смоделируем игру в кости:

```
> sample(x = 1:6, size = 20, replace = TRUE)
 [1] 2 4 6 4 3 6 1 2 3 2 1 4 5 5 4 3 6 5 6 1
```

Первым аргументом функции передается вектор элементарных исходов (для игральной кости это набор чисел от 1 до 6), аргумент `size` задает число опытов (или, другими словами, – объем выборки, которую мы хотим извлечь из имеющейся совокупности), логический аргумент `replace` определяет, следует ли возвращать уже выбранное значение в исходную совокупность (в нашем случае при каждом новом броске возможно выпадение любого числа, поэтому значения возвращаются). Таким образом, мы смоделировали 20 бросков кости и получили 20 исходов. Обратимся теперь к вопросу о вероятности того, что при броске кости выпадет более 4 очков. Элементарными исходами, благоприятствующими этому событию, являются выпадения пятерки и шестерки, поэтому классическое определение вероятности дает нам $P = 2/6 = 1/3$. Убедимся в этом с использованием статистического определения вероятности и моделирования:

```
> (result <- sample(1:6, size = 30, replace = TRUE))
 [1] 1 2 3 3 2 6 5 1 5 5 1 3 5 4 2 3 2 1 4 4 5 5 2 3 3 5 4 1 4 3
> (outcome <- result > 4)
 [1] FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE
[11] FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[21]  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
> (m <- cumsum(outcome))
 [1] 0 0 0 0 0 1 2 2 3 4 4 4 5 5 5 5 5 5 5 5 6 7 7 7 7 8 8 8 8 8
```

Мы смоделировали 30 бросков кости и записали результаты в вектор `result`. Мы интересуемся определенным событием, поэтому вектор результатов преобразуем в логический вектор исходов `outcome`, в котором значение `TRUE` означает, что выпало более 4 очков. Для расчета натуральной частоты `m` мы воспользовались функцией `cumsum()`, которая считает кумулятивную сумму, то есть сумму элементов вектора от начала до данной позиции (первый элемент кумулятивной суммы соответствует первому элементу исходного вектора,

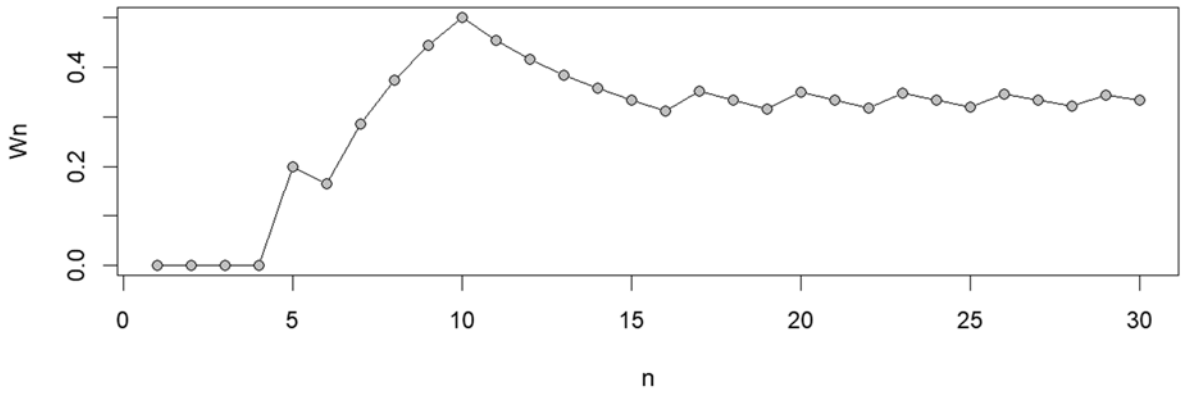

Рис. 1. Зависимость относительной частоты от числа опытов (n = 30)

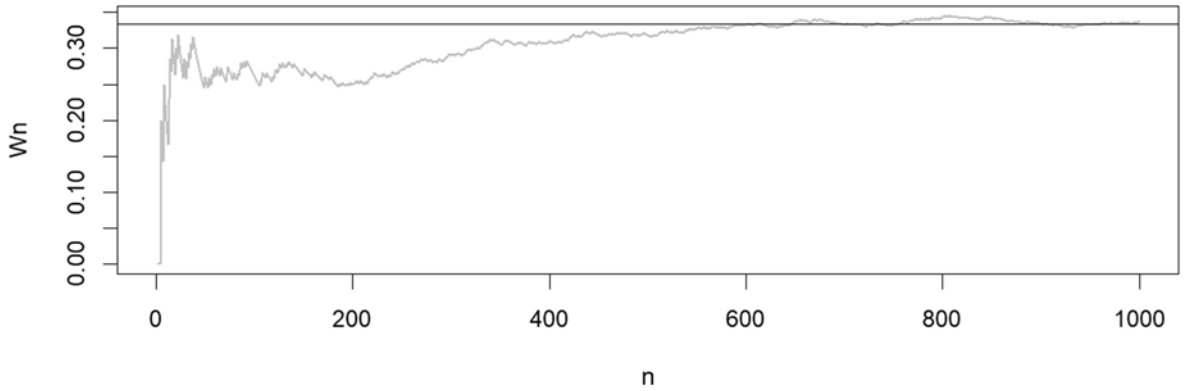

Рис. 2. Зависимость относительной частоты от числа опытов (n = 1000)

второй элемент – сумме первых двух элементов, третий – сумме первых трех элементов и так далее).

```
> n <- 1:30
> Wn <- m / n
```

Число опытов n – это просто последовательность чисел от 1 до 30, а относительная частота легко рассчитывается делением одного вектора на другой. Посмотрим, как ведет себя относительная частота в зависимости от числа опытов (Рис. 1):

```
> plot(n, Wn, type = "o", pch = 21, bg = "grey")
```

Мы видим, что после нескольких «неудачных» исходов интересующее нас событие начинает появляться в серии и его относительная частота начинает флуктуировать в диапазоне примерно от 0.2 до 0.4. Необходимо помнить, что мы моделируем последовательность случайных событий, поэтому при каждом новом повторении нашего опыта (при каждом повторном запуске приведенного кода), мы будем получать несколько иные результаты, однако общая тенденция останется неизменной.

Повторим моделирование, но теперь существенно увеличим количество опытов в серии (Рис. 2):

```
> result <- sample(1:6, size = 1000, replace = TRUE)
> outcome <- result > 4
> m <- cumsum(outcome)
> n <- 1:1000
> Wn <- m / n
> plot(n, Wn, type = "l", lwd = 2, col = "grey")
> abline(h = 1/3)
```

При росте числа опытов `n` относительная частота `Wn` флуктуирует все ближе и ближе к теоретическому значению 1/3 (изображено на графике горизонтальной прямой).

1.2. Распределение вероятностей

Случайной величиной называется величина, которая принимает то или иное значение в результате опыта со случайным исходом. Примерами случайных величин из мира азартных игр могут быть число очков, выпавших на игральной кости или размер выигрыша на рулетке при той или иной ставке. Случайными величинами можно считать результаты практически любых научных измерений.

Рассмотрим классическую схему опытов Бернулли. Пусть имеется простой опыт, в результате которого происходит либо не происходит некое событие A с известной вероятностью *P(A)*; если событие происходит, то говорят об успехе испытания. Примерами такого опыта могут быть подбрасывание монеты, в результате которого орел может выпасть (успех, $P = 1/2$ в случае идеальной монеты) либо нет, а также бросок кости, в результате которого может выпасть более 4 очков ($P = 1/3$) либо нет. Вероятность успеха обозначают $P(A) = p$, а вероятность неудачи обозначают q. Поскольку в результате опыта успех либо неудача обязательно произойдут, сумма их вероятностей равна единице: $p + q = 1$. Пусть проводится серия из n простых опытов и мы следим за числом успехов в этой серии. Вероятность m успехов в серии из n опытов определяется по формуле Бернулли:

$$P_n(m) = C_m^n \cdot p^m \cdot (1-p)^{n-m} = \frac{n!}{m!\cdot(n-m)!} \cdot p^m \cdot q^{n-m}.$$

Число успехов в схеме Бернулли можно рассматривать в качестве случайной величины X. Эта случайная величина может принимать значения от 0 до n. Совокупность вероятностей $P(X = m) = P_n(m)$ для m от 0 до n называется законом распределения вероятностей случайной величины X. Это распределение известно как биномиальное распределение, поскольку величины C_m^n фигурируют

в качестве коэффициентов при разложении в бином Ньютона степени суммы двух слагаемых.

Распределение вероятностей случайной величины полностью описывает ее поведение. Распределение вероятностей может быть задано в виде совокупности отдельных вероятностей в случае дискретной случайной величины с конечным числом значений. Непрерывные случайные величины могут принимать бесконечное число значений, поэтому вероятность принять какое-либо одно конкретное значение равна нулю. Для описания распределения вероятностей непрерывных случайных величин требуется иной подход.

Универсальной формой задания распределения вероятностей, пригодной и для дискретных, и для непрерывных случайных величин, является функция распределения – вероятность того, что случайная величина примет значение, меньшее либо равное заданному значению *x*:

$$F(x) = P(X \le x).$$

Еще одной формой задания распределения вероятностей, применяемой для описания непрерывных случайных величин, является плотность распределения – вероятность попасть в бесконечно малую окрестность данной точки, отнесенная к размеру этой окрестности. Плотность вероятности непрерывных случайных величин в определенном смысле аналогична вероятностям конкретных значений дискретных случайных величин. Функция *F(x)* и плотность *f(x)* распределения связаны следующими соотношениями:

$$f(x) = \frac{dF(x)}{dx},$$

$$F(x) = \int_{-\infty}^{x} f(x)dx.$$

Наиболее распространенным, известным и важным распределением является так называемое нормальное распределение или распределение Гаусса. Плотность вероятности нормального распределения определяется следующим выражением:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

где μ и σ – параметры распределения, соответствующие среднему и стандартному отклонению.

Функция нормального распределения аналитической формы не имеет и определяется через интеграл плотности распределения. Аналогичным образом, многие другие применяемые в теории вероятностей и математической статистике непрерывные распределения не могут быть выражены в простой

аналитической форме. Например, плотность знаменитого *t*-распределения Стьюдента может быть выражена только через гамма-функцию:

$$f(x)=\frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}},$$

где ν – параметр распределения, называемый числом степеней свободы, а гамма-функция определяется как

$$\Gamma(t)=\int_0^\infty x^{t-1}e^{-x}dx.$$

Для работы с такими распределениями до конца прошлого века использовались таблицы распределений, в которых были приведены значения плотности и функции распределения в конкретных точках для конкретных значений параметров распределений, либо квантили (см. ниже). В современную эпоху цифровых технологий расчет этих величин не представляет особой сложности, поскольку существуют специальные функции, в том числе и в среде R.

1.3. Семейства функций для работы с распределениями

В базовых пакетах среды R реализованы функции для большого числа распределений, используемых в теории вероятностей и математической статистике. Для каждого из распределений существует 4 функции:

`d`dist (от density) – для расчета плотности распределения либо вероятностей дискретных значений;

`p`dist (от probability) – для расчета функции распределения;

`q`dist (от quantile) – для расчета квантилей распределения;

`r`dist (от random) – для генерации псевдослучайных чисел, подчиняющихся данному распределению.

Во всех случаях вместо `dist` необходимо подставить обозначение соответствующего распределения (`binom` – биномиальное, `norm` – нормальное и т.д., см. Таблицу 1 в конце настоящего раздела).

Рассчитаем распределение вероятностей случайной величины, которая соответствует числу успехов в серии из 30 опытов с вероятностью успеха 1/3. Такая случайная величина подчиняется биномиальному распределению и может

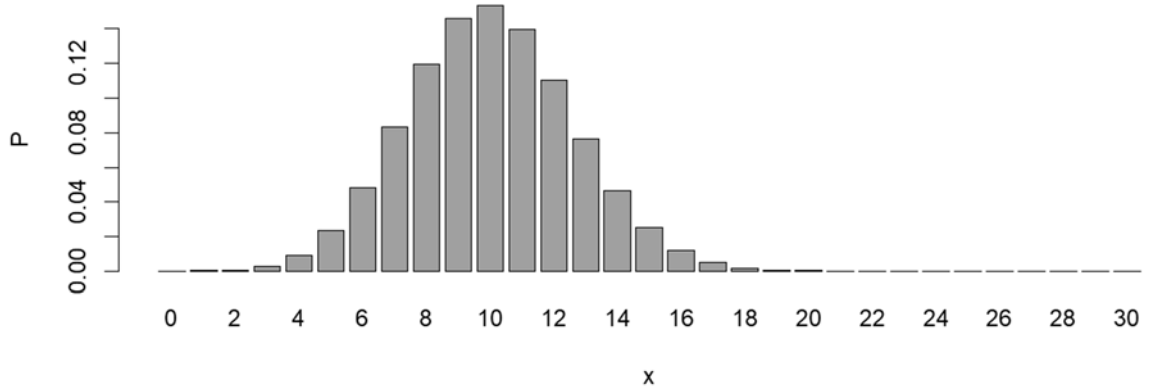

Рис. 3. Биномиальное распределение вероятностей

принимать значения от 0 до 30. Для расчета вероятностей этих значений воспользуемся функцией `dbinom()`:

```
> pbin <- dbinom(0:30, size = 30, prob = 1/3)
```

Первый аргумент функций для расчета плотности распределения (или в случае дискретных распределений – вероятностей значений, как в данном случае) всегда представляет собой вектор значений случайной величины, для которых мы хотим рассчитать плотность либо вероятность. Затем следуют специфичные для каждого распределения аргументы, соответствующие параметрам данного распределения. В случае функции `dbinom()` это аргументы `size` (число опытов в серии, целое число) и `prob` (вероятность успеха, число от 0 до 1). Они не имеют значений по умолчанию, поэтому их использование обязательно.

Отобразим полученные вероятности в виде столбчатой диаграммы, которая строится с помощью функции `barplot()`. Первым аргументом эта функция получает высоты столбцов, аргумент `names` определяет подписи под столбцами.

```
> barplot(pbin, names = 0:30, xlab = "x", ylab = "P", col = "grey")
```

Мы видим (Рис. 1) практически симметричное распределение, причем наиболее вероятное значение ($x = 10$) соответствует произведению числа опытов на вероятность успеха. Чем больше отклонение от этого значения, тем меньше соответствующие вероятности. Поскольку мы рассчитали вероятности всех возможных значений случайной величины, то их сумма должна быть равна единице, в чем легко убедиться:

```
> sum(pbin)
[1] 1
```

Зная распределение случайной величины, можно решать задачи, связанные с вероятностями тех или иных событий. Например, можно рассчитать вероятность того, что в серии из 30 опытов с вероятностью успеха 1/3 будет

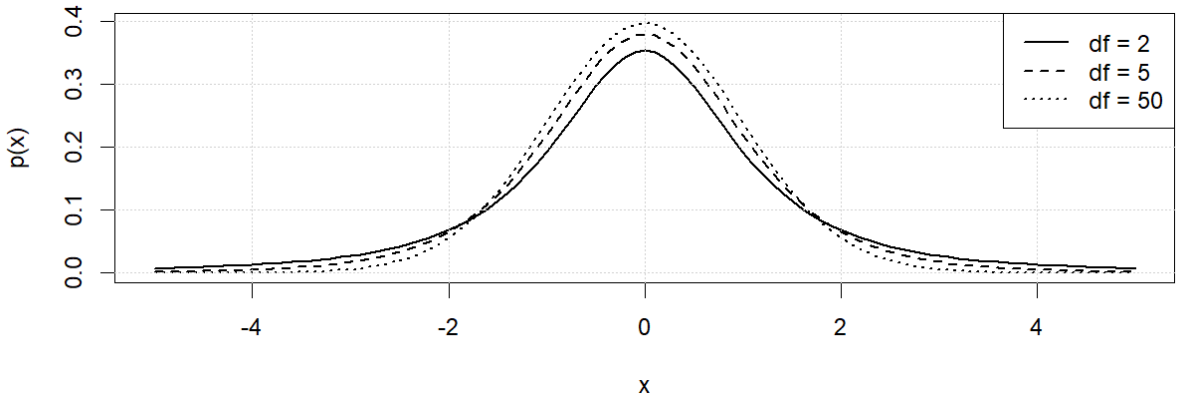

Рис. 4. Плотность распределения Стьюдента

получено не менее 15 успехов, для этого нужно просуммировать вероятности значений от 15 до 30, то есть последних 16 элементов вектора `pbin`:

```
> sum(pbin[16:31])
[1] 0.04348228
```

Перейдем теперь к рассмотрению непрерывных распределений. Отложив подробное рассмотрение нормального распределения до главы 3, познакомимся с функциями для работы с распределениями на примере *t*-распределения Стьюдента. Для работы с этим распределением служат функции `dt()`, `pt()`, `qt()` и `rt()`. Распределение Стьюдента характеризуется одним параметром, который называется числом степеней свободы (degrees of freedom, часто сокращается до df). Построим график плотности распределения для трех контрастных степеней свободы:

```
> x <- seq(-5, 5, by = 0.1)
> tden2 <- dt(x, df = 2)
> tden5 <- dt(x, df = 5)
> tden50 <- dt(x, df = 50)
> plot(x, tden2, type = "l", lwd = 2,
+      ylim = range(c(tden2, tden5, tden50)),
+      ylab = "f(x)")
> lines(x, tden5, lwd = 2, lty = 2)
> lines(x, tden50, lwd = 2, lty = 3)
> grid()
> legend("topright", legend = c("df = 2", "df = 5", "df = 50"),
+        lwd = 2, lty = 1:3)
```

Сначала мы создали вектор значений `x`, для которых рассчитали плотность с помощью функции `dt()`, при этом использовали обязательный аргумент `df`. Затем построили график, на который добавили сетку и легенду (Рис. 2). Мы видим, что распределение Стьюдента – одновершинное, симметричное вокруг нуля. Это означает, что наиболее вероятное значение – 0, и чем дальше от ноля, тем менее вероятно данное значение, причем отклонения в положительную и

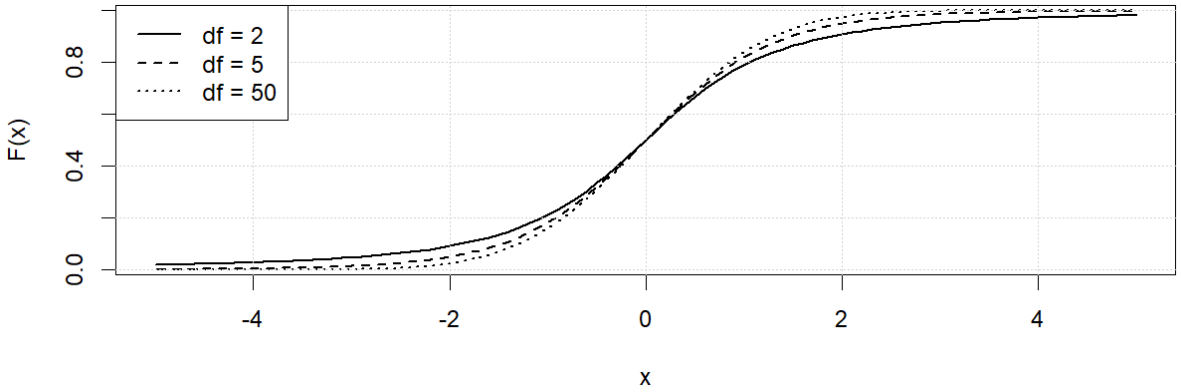

Рис. 5. Функция распределения Стьюдента

отрицательную сторону равновероятны. Число степеней свободы определяет форму распределения: чем меньше степеней свободы, тем более вероятны уклонения от центра распределения.

Построим теперь аналогичный график для функций распределения, значения которых рассчитываются функцией `pt()`:

```
> pden2 <- pt(x, df = 2)
> pden5 <- pt(x, df = 5)
> pden50 <- pt(x, df = 50)
> plot(x, pden2, type = "l", lwd = 2, ylab = "F(x)")
> lines(x, pden5, lwd = 2, lty = 2)
> lines(x, pden50, lwd = 2, lty = 3)
> grid()
> legend("topleft", legend = c("df = 2", "df = 5", "df = 50"),
+        lwd = 2, lty = 1:3)
```

Полученный график (Рис. 5) прекрасно демонстрирует основные свойства функции распределения: (1) $0 \le F(x) \le 1$, поскольку $F(x)$ является вероятностью; (2) $F(x_2) \ge F(x_1)$, если $x_2 > x_1$, то есть $F(x)$ неубывающая функция; (3) $F(-\infty) = 0$ и (4) $F(\infty) = 1$. Сами графики функции распределения обычно малоинформативны. Чаще всего функция распределения используются для вычисления вероятности попадания в заданный интервал. Пусть мы имеем случайную величину, подчиняющуюся распределению Стьюдента с 10 степенями свободы. Вероятность того, что величина примет значение, меньшее некоторого a, равна значению функции распределения в этой точке $F(a)$ (это следует из определения функции распределения), например $P(X < 1)$ вычисляется как

```
> pt(1, df = 10)
[1] 0.8295534
```

Чтобы получить вероятность того, что величина примет значение, большее a, нужно вычесть $F(a)$ из единицы, например $P(X > 2)$ вычисляется как

```
> 1 - pt(2, df = 10)
[1] 0.03669402
```

Наконец, чтобы вычислить вероятность попадания в интервал (*a, b*) нужно вычислить разность значений функции распределения на границах интервала, например $P(-0.5 < X < 0.1)$ вычисляется как

```
> pt(0.1, df = 10) - pt(-0.5, df = 10)
[1] 0.2248928
```

Квантилем распределения называется такое значение, которое случайная величина не превышает с заданной вероятностью. *p*-квантиль представляет собой такое значение, для которого $P(X < x) = p$, то есть он является решением уравнения $F(x) = p$. В этом смысле функция распределения и квантиль распределения являются взаимнообратными функциями. Продолжим экспериментировать с распределением Стьюдента с 10 степенями свободы. Как найти значение, которое превышают только 5 % значений случайной величины? Это такое значение, меньше которого 95 % значений, то есть это 0.95-квантиль этого распределения:

```
> qt(0.95, df = 10)
[1] 1.812461
```

Традиционные таблицы распределений почти всегда содержат квантили для заданного набора уровней. Таблица содержит граничное значение, которое критериальная статистика должна превысить для того, чтобы считаться значимой на данном уровне. Это делалось в утилитарных целях для облегчения расчетов. Сегодня мы можем легко вычислять вероятности соответствующих значений критериальных статистик с использованием функций распределения (подробнее см. гл. 4).

Квантили также используются для построения доверительных интервалов, анализа мощности критериев, а также при визуальной диагностике отклонений от распределения в виде так называемых квантильных графиков.

Функции для генерации случайных чисел, подчиняющихся заданному распределению, используются чаще всего в целях моделирования тех или иных ситуаций. В строгом смысле слова эти числа не являются случайными, поскольку для их генерации используются так называемые генераторы последовательностей псевдо-случайных чисел, которые рано или поздно начинают повторяться, однако этот период повтора настолько велик, что в практических целях можно рассматривать результаты как случайные. Первым аргументом функций-генераторов является количество чисел, которое необходимо сгенерировать, затем следуют аргументы, описывающие параметры распределения. Для примера сгенерируем несколько наборов случайных чисел из распределения Стьюдента с 10 степенями свободы:

```
> rt(10, df = 10)
 [1]  1.0674795 -0.3942987  0.3781185  1.0538709 -0.2046322
 [6]  1.3048241 -1.5513326 -0.7348046 -0.7193443  0.6008737
> rt(10, df = 10)
 [1]  0.12229444  0.87684016  0.01421381  0.15405672 -1.01278735
 [6]  2.33520065  0.33009645  0.20287011 -0.93832650  1.58506767
> rt(10, df = 10)
 [1]  0.4099280  0.5598465  0.6507501 -1.6888523  1.0383576
 [6] -0.1681005  0.2874757 -0.5367275  1.3807147 -0.3209681
```

Получаемые числа являются выборками из интересующего нас распределения, причем каждый раз эти числа будут разными.

При генерации случайных чисел из дискретных распределений результаты будут соответствовать имеющимся ограничениям. Например, при генерации биномиально распределенных чисел в выборках будут присутствовать исключительно целые числа от 0 до числа опытов:

```
> rbinom(20, size = 30, prob = 1/3)
 [1] 12  9  8 12  9 17 14 10  9  7 13  9 14 10 14 10  9  9 10 10
> rbinom(20, size = 30, prob = 1/3)
 [1] 12 10 11 12  7  8 13 12 10 11  5  9  2  7 10 10 11 11  8 11
```

Здесь сгенерированы результаты 40 серий по 30 опытов с вероятностью успеха 1/3.

Таблица 1

Основные распределения вероятностей

Название	**Обозначение в R**	**Аргументы функций**
Дискретные		
Биномиальное	`binom`	`size` – число испытаний в серии `prob` – вероятность успеха в отдельном испытании
Геометрическое	`geom`	`prob` – вероятность успеха в отдельном испытании
Гипер-геометрическое	`hyper`	`m` – число белых шариков в урне `n` – число черных шариков в урне `k` – число шариков, вынимаемых из урны
Мультиномиальное	`multinom`	`size` – число объектов, которые распределяются по K ячейкам `prob` – вектор неотрицательных чисел длины K, определяющий вероятности K классов (нормализуется до суммирования в единицу)

Название	Обозначение в R	Аргументы функций
Отрицательное биномиальное	`nbinom`	`size` – число успешных испытаний, которое должно быть достигнуто в серии (либо дисперсия распределения при альтернативной параметризации) `prob` – вероятность успеха в отдельном испытании `mu` – среднее при альтернативной параметризации
Пуассона	`pois`	`lambda` – вектор средних (неотрицательных)
Непрерывные		
Бета	`beta`	`shape1`, `shape2` – параметры формы `ncp` – параметр нецентральности
Коши	`cauchy`	`location` – параметр положения `scale` – параметр разброса
χ^2	`chisq`	`df` – число степеней свободы `ncp` – параметр нецентральности
Экспоненциальное	`exp`	`rate` – вектор скоростей процессов
F Фишера	`f`	`df1`, `df2` – числа степеней свободы (допускается `Inf`) `ncp` – параметр нецентральности
Гамма	`gamma`	`shape` – параметр формы `scale` – параметр масштаба `rate` – альтернативный способ параметризации масштаба
Логнормальное	`lnorm`	`meanlog` – среднее в логарифмическом масштабе `sdlog` – стандартное отклонение в логарифмическом масштабе
Нормальное	`norm`	`mean` – среднее `sd` – стандартное отклонение
Стьюдента	`t`	`df` – число степеней свободы `ncp` – параметр нецентральности
Равномерное	`unif`	`min` – минимальное значение `max` – максимальное значение
Вейбулла	`weibull`	`shape` – параметр формы `scale` – параметр масштаба

Глава 2. Описание данных

Любой анализ данных начинается с их описания. В настоящей главе будут рассмотрены традиционные средства описательной статистики и их реализация в среде R.

В качестве основного рабочего примера мы будем использовать набор данных по морфологической изменчивости медоносных пчел (*Apis mellifera* L.). Эти данные были любезно предоставлены А.А. Брагазиным. Основной целью исследования было изучение межпородной изменчивости пчел и возможности ее применения в экологическом мониторинге. На первом этапе исследования стояла задача выявления отличий между тремя породами пчел (*A. m. carnica*, *A. m. caucasica*, *A. m. mellifera*) по стандартному комплексу морфологических признаков, используемых в апиологии, который включает шесть количественных признаков (длина хоботка – proboscis, длина третьего тергита – tergit.length, ширина третьего тергита – tergit.width, длина крыла – wing.length, ширина крыла – wing.width, кубитальный индекс – cubital.index) и один качественный (дискоидальное смещение – discoidal.shift, может быть положительным, отрицательным и нейтральным). Для характеристики пород были отобраны и промерены выборки чистопородных пчел (для *A. m. carnica* и *A. m. caucasica* по 30 особей из 8 семей, всего по 240 особей, для *A. m. mellifera* – 114 особей без указания семейной принадлежности).

Начнем знакомство с этими данными.

```
> load("apis.rda")
```

Данные сохранены в файле apis.rda в виде фрейма `apis`, который содержит 594 наблюдения (строки) и 9 переменных. С названиями переменных и общей структурой данных удобно ознакомиться с помощью функции `str()`:

```
> str(apis)
'data.frame':    594 obs. of  9 variables:
 $ proboscis      : num  5.44 5.66 5.35 5.98 5.32 ...
 $ tergit.length  : num  2.47 2.41 2.33 2.31 2.26 ...
 $ tergit.width   : num  4.62 4.81 4.92 4.86 4.88 ...
 $ wing.width     : num  3.18 3.15 3.25 3.19 3.2 ...
 $ wing.length    : num  9.65 9.46 9.41 9.67 9.47 ...
 $ cubital.index  : num  45.1 38.1 33 31.5 36 ...
 $ discoidal.shift: chr  "positive" "positive" "positive" "positive"
 $ family         : chr  "fam.24a" "fam.24a" "fam.24a" "fam.24a" ...
 $ breed          : chr  "carnica" "carnica" "carnica" "carnica" ...
```

Функция выводит в консоль массу полезной информации. В первой строке она сообщает, что мы имеем дело с фреймом данных размерностью 594 × 9.

Далее построчно перечисляются переменные, для каждой указывается тип данных (`num` – числовой, `int` – целочисленный, `chr` – текстовый, `logi` – логический, `cplx` – комплексный, `Factor` – фактор; для факторов указывается также число градаций), затем следуют нескольких первых наблюдений (не более 10), что дает возможность получить представление о характере данных. Мы видим, что первые семь переменных соответствуют морфологическим признакам (см. выше), а две последние текстовые переменные указывают семейную (`family`) и породную (`breed`) принадлежность.

Для первого взгляда на данные можно также воспользоваться функциями `head()` и `tail()`, которые выводят в консоль несколько (по умолчанию – 6, но можно задать иное значение аргумента `n`) первых либо последних наблюдений фрейма. Например,

```
> head(apis, n = 3)
  proboscis tergit.length tergit.width wing.width wing.length
1    5.4359        2.4718       4.6191     3.1798      9.6507
2    5.6580        2.4082       4.8142     3.1538      9.4639
3    5.3531        2.3314       4.9191     3.2533      9.4100
  cubital.index discoidal.shift  family   breed
1       45.0913        positive fam.24a carnica
2       38.1251        positive fam.24a carnica
3       32.9677        positive fam.24a carnica
> tail(apis, n  = 3)
    proboscis tergit.length tergit.width wing.width wing.length
592    6.2684        2.4360       5.2152     3.3101      9.5065
593    6.4540        2.4492       5.1715     3.3192      9.5747
594    6.4005        2.3990       4.9833     3.2012      9.5561
    cubital.index discoidal.shift  family     breed
592       61.1744        positive fam.mel mellifera
593       49.6503        positive fam.mel mellifera
594       56.3695        negative fam.mel mellifera
```

Мы видим, что в начале фрейма находятся данные по пчелам породы *A. m. carnica* 24-й семьи, а в конце – по пчелам породы *A. m. mellifera*.

Можно также воспользоваться инструментом просмотра объектов, который запускается функцией `View()` (при работе в RStudio можно кликнуть на иконку в виде таблицы справа от названия объекта на вкладке рабочего пространства):

```
> View(apis)
```

На панели источников откроется новая вкладка с названием объекта (в нашем случае apis), которая покажет его текущее состояние в табличной форме. Этот инструмент служит исключительно для просмотра, никакие изменения здесь производить нельзя.

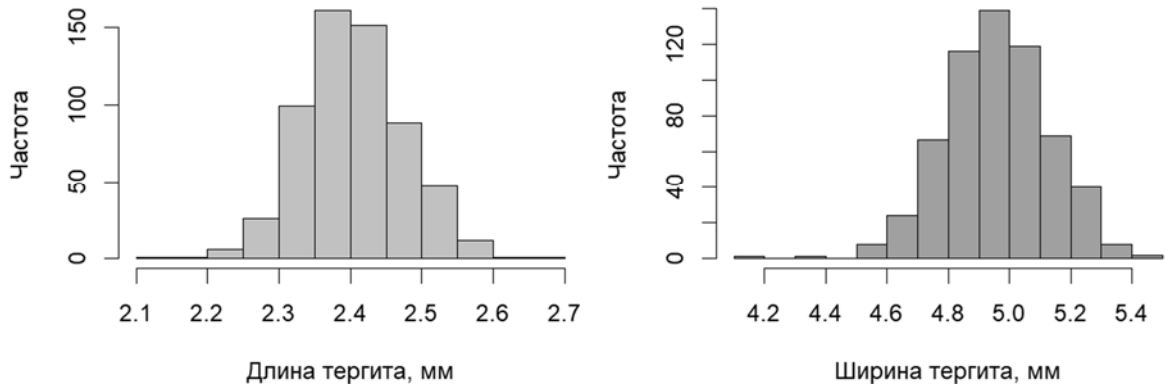

Рис. 6. Гистограммы распределения длины и ширины тергита

После первичного знакомства со структурой фрейма приступим к описанию морфологической изменчивости пчел. Сразу же оговоримся, что в этой главе мы не будем обращать внимание на породную и семейную принадлежность пчел. Таким образом, наша задача заключается в описании морфологии пчел по выборке из 594 особей.

2.1. Описание количественных переменных

Традиционно рассмотрение описательной статистики начинается с мер положения и разброса, которые являются одновременно простейшими и наиболее важными статистиками. Мы же рекомендуем всегда начинать с визуального представления данных.

Основными средствами визуализации данных являются гистограммы и так называемые диаграммы размахов (они же – ящики с усами, box-and-whiskers plots). Гистограмма отображает распределение непрерывной переменной в виде столбчатой диаграммы, разделяя диапазон значений переменной на заданное число непересекающихся отрезков и отображая частоту внутри каждого отрезка по оси ординат. Основной функцией для создания гистограмм в R является `hist()`, это графическая функция высокого уровня, создающая новый график и обладающая многими стандартными графическими аргументами. Построим гистограммы распределения значений длины и ширины тергита (результат представлен на Рис. 6):

```
> par(mfrow = c(1,2), mar = c(5.1, 4.1, .5, .5))
> hist(apis$tergit.length, col = "grey", main = "",
+      xlab = "Длина тергита, мм", ylab = "Частота")
> hist(apis$tergit.width, col = "grey", main = "",
+      xlab = "Ширина тергита, мм", ylab = "Частота")
```

Каждый столбик гистограммы соответствует интервалу значений непрерывной переменной, который служит его основанием, а высота столбика отражает частоту этого класса (число наблюдений, попадающих в данный интервал). В частности, при построении первой гистограммы диапазон значений длины тергита был разбит на 12 интервалов по 0.05 мм (от 2.1 до 2.15, от 2.15 до 2.2 и т.д.), для каждого интервала была рассчитана частота (по 1 в первых двух интервалах, 6 в третьем и т.д.), а результаты отображены в виде столбчатой диаграммы. Гистограмма служит прекрасным средством визуализации распределения данных: мы видим, какие значения принимают наши данные, а также сколько наблюдений имеют те или иные значения. Длина и ширина тергита пчел имеют симметричное одновершинное (унимодальное) распределение, поскольку большая часть наблюдений попадает в центральный класс, по мере удаления от которого число наблюдений снижается. Такое распределение типично для морфологических данных.

Функция `hist()` по умолчанию сама выбирает число и ширину интервалов, однако при необходимости можно воспользоваться аргументом `breaks`, которому можно передать требуемое число интервалов, либо вектор, содержащий границы интервалов. Рассмотрим применение этого аргумента на примере гистограммы распределения значений кубитального индекса (результат представлен на Рис. 7):

```
> hist(apis$cubital.index, col = "grey", main = "", ylab = "Частота",
+      xlab = "Кубитальный индекс", breaks = 25)
> hist(apis$cubital.index, col = "grey", main = "", ylab = "Частота",
+      xlab = "Кубитальный индекс", breaks = seq(10, 80, by = 5))
```

В первом случае при вызове функции `hist()` мы указали, что необходимо построить гистограмму на основе 25 интервалов, во втором – передали через аргумент `breaks` вектор границ интервалов (последовательность от 10 до 80 с шагом 5). При задании `breaks = n` мы получим приблизительно `n` интервалов, поскольку алгоритм оптимизирует вид гистограммы. Полный контроль достигается при задании вектора границ интервалов, как во втором случае, можно даже задавать интервалы неравной длины.

Обратите внимание, что вид гистограммы напрямую зависит от числа интервалов. Если задать слишком малое число интервалов, можно получить малоинформативный график из нескольких столбцов, если же сделать слишком дробные интервалы, то появится шум, как в случае первой гистограммы для кубитального индекса (в целом одновершинное распределение распадается на множество локальных пиков). Оптимальное число интервалов для построения гистограммы приходится подбирать для каждого набора данных.

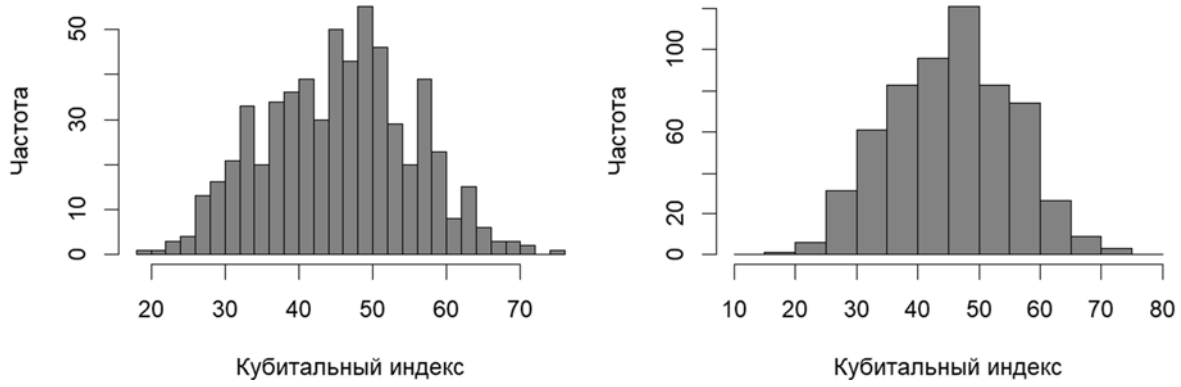

Рис. 7. Гистограммы распределения кубитального индекса

Диаграммы размахов отображают в графической форме основные описательные статистики: минимум, максимум и квартили. Диаграммы размахов строятся функцией `boxplot()`, которая создает новый график:

```
> boxplot(apis$tergit.length, ylab = "Длина тергита, мм", col = "grey")
> boxplot(apis$tergit.width, ylab = "Ширина тергита, мм", col = "grey")
```

Медиана – это такое значение, выше которого ровно половина наблюдений (ниже, соответственно, тоже), на диаграмме размахов медиана отображается в виде жирной горизонтальной черты. Вокруг медианы строится прямоугольник (он же – ящик), границы которого являются нижним и верхним квартилями – это такие значения, меньше (нижний) либо больше (верхний) которых насчитывается 25 % наблюдений. Таким образом, внутри ящика сосредоточена половина выборки. За пределами ящика располагаются «усы» – отрезки, отображающие разброс данных. Если минимум и максимум попадают в пределы полутора межквартильных расстояний (размер ящика) от медианы, то они и отображаются в качестве границы усов (см. верхний ус для ширины тергита на Рис. 8). Если какие-либо значения выходят за этот предел, то они отображаются отдельно в виде точек, а границей уса становится последнее значение, попавшее в этот предел (именно такую ситуацию мы видим на диаграммах размаха длины и ширины тергита, Рис. 8).

После того, как мы получили визуальное представление о характере распределения данных, можно переходить к вычислению описательных статистик. Меры положения описывают центр распределения и таким образом характеризуют типичное значение выборки. Основными мерами положения являются среднее и медиана, для векторов они вычисляются функциями `mean()` и `median()` соответственно:

```
> mean(apis$tergit.length)
[1] 2.403259
```

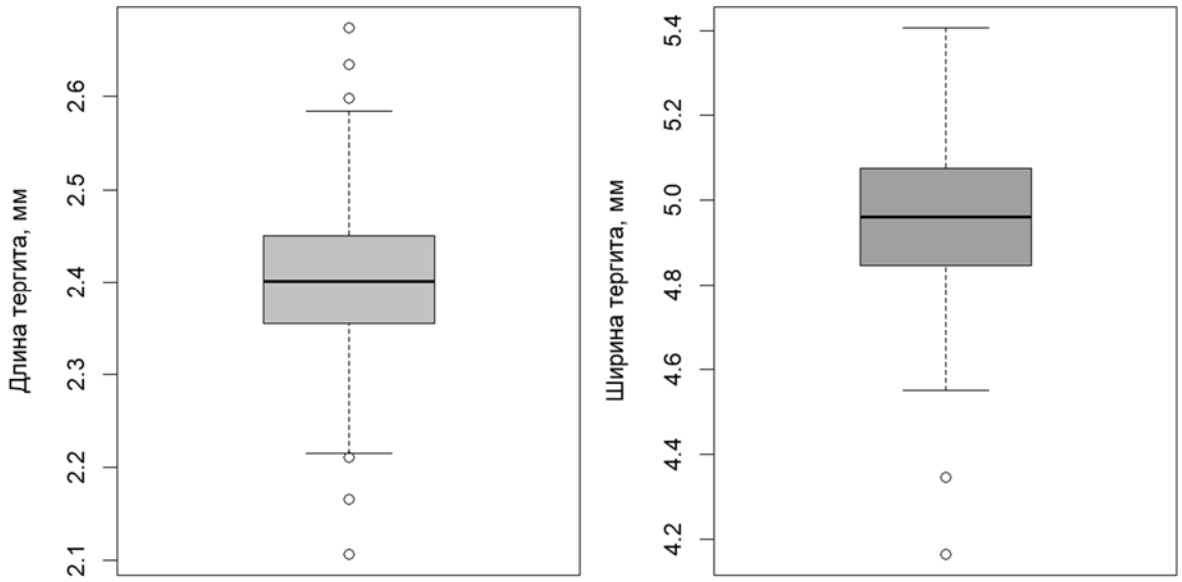

Рис. 8. Диаграммы размаха длины и ширины тергита

```
> mean(apis$tergit.width)
[1] 4.960209
> median(apis$tergit.length)
[1] 2.4004
> median(apis$tergit.width)
[1] 4.96045
```

Функция `mean()` вычисляет простое арифметическое среднее (сумма значений, отнесенная к числу значений).

Меры разброса характеризуют изменчивость наблюдений в пределах выборки. Простейшими мерами разброса являются минимальное и максимальное значения, которые можно вычислять с помощью функций `min()`, `max()` и `range()`, последняя вычисляет оба предела и возвращает их в виде вектора из двух элементов:

```
> min(apis$tergit.length)
[1] 2.1069
> max(apis$tergit.length)
[1] 2.6744
> range(apis$tergit.length)
[1] 2.1069 2.6744
```

Наиболее распространенной мерой разброса является выборочная дисперсия:

$$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1},$$

где x_i – i-е выборочное значение, $\bar{x}$ – арифметическое среднее, n – объем выборки (число наблюдений). Она вычисляется функцией `var()`:

```
> var(apis$tergit.length)
[1] 0.005291149
```

При вычислении дисперсии отклонения от среднего возводятся в квадрат, поэтому итоговая величина имеет размерность квадрата исходной размерности (в случае длины тергита – квадратные миллиметры). Наряду с дисперсией часто используется стандартное отклонение (квадратный корень из дисперсии), лишенное этого недостатка. Стандартное отклонение вектора можно вычислить с помощью функции `sd()`:

```
> sd(apis$tergit.length)
[1] 0.07274029
```

Дисперсия и стандартное отклонение чувствительны к отдельным значениям, сильно уклоняющимся от среднего. Более устойчивыми мерами разброса являются межквартильное расстояние (inter-quartile range, разница между верхним и нижним квартилями) и медианное отклонение (median absolute deviation, медианное отклонение от медианы), вычисляемые функциями `IQR()` и `mad()`:

```
> IQR(apis$tergit.length)
[1] 0.09545
> mad(apis$tergit.length)
[1] 0.07072002
```

Для более подробного описания распределения данных служат выборочные квантили – это такие значения, которые отсекают определенную долю (или процент) выборки. Медиана (отсекает 50 %), квартили (отсекают 25 % и 75 %), а также минимум (0 %) и максимум (100 %) являются частными случаями квантилей, именно они вычисляются функцией `quantile()` по умолчанию:

```
> quantile(apis$tergit.length)
      0%      25%      50%      75%     100%
2.106900 2.354625 2.400400 2.450075 2.674400
```

Функция возвращает именованный вектор (каждому элементу задано имя); именование можно отменить с помощью аргумента логического типа `names` (для отмены нужно использовать значение `FALSE`). Для задания нестандартных квантилей можно воспользоваться аргументом `probs`, через который передается вектор долей. Например, если нам необходимы 5-й и 95-й квантили:

```
> quantile(apis$tergit.length, probs = c(0.05, 0.95))
      5%      95%
2.292940 2.530645
```

Мы рассмотрели множество функций, используемых для вычисления описательных статистик выборочных данных, представленных в виде вектора. В случае, если в обрабатываемом векторе содержатся отсутствующие данные типа NA, то результатом тоже будет NA, что во многих случаях нежелательно. Для того чтобы при расчете отсутствующие значения просто игнорировались, необходимо использовать аргумент na.rm (от англ. remove NA), который нужно установить в значение TRUE.

Помимо отдельных функций, рассчитывающих описательные статистики по отдельности, в R существует много удобных функций, позволяющих рассчитывать наиболее распространенные статистики одновременно и для нескольких переменных. В частности, функция summary() рассчитывает стандартные квантили и среднее:

```
> summary(apis)
   proboscis     tergit.length    tergit.width
 Min.   :5.007   Min.   :2.107   Min.   :4.166
 1st Qu.:5.661   1st Qu.:2.355   1st Qu.:4.846
 Median :6.341   Median :2.400   Median :4.960
 Mean   :6.280   Mean   :2.403   Mean   :4.960
 3rd Qu.:6.901   3rd Qu.:2.450   3rd Qu.:5.075
 Max.   :7.340   Max.   :2.674   Max.   :5.406
   wing.width     wing.length    cubital.index
 Min.   :2.239   Min.   :8.655   Min.   :19.93
 1st Qu.:3.142   1st Qu.:9.268   1st Qu.:38.12
 Median :3.203   Median :9.386   Median :45.73
 Mean   :3.202   Mean   :9.389   Mean   :45.44
 3rd Qu.:3.276   3rd Qu.:9.519   3rd Qu.:52.31
 Max.   :3.493   Max.   :9.826   Max.   :74.57
 discoidal.shift       family             breed
 Length:594         Length:594         Length:594
 Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character
```

Для числовых переменных рассчитываются квантили и среднее, для текстовых переменных указывается только объем выборки, для логических переменных указывается число значений TRUE и NA.

2.2. Описание качественных переменных

Для описания качественных данных применяются частоты и доли. Частота представляет собой число наблюдений в выборке, имеющих данное значение

качественной переменной. Для расчета частот в R служит функция `table()`. Рассчитаем частоты дискоидального смещения:

```
> (f <- table(apis$discoidal.shift))
negative  neutral positive
     108       94      392
```

Функция возвращает таблицу сопряженности – специальный объект класса `table`:

```
> class(f)
[1] "table"
```

В рассмотренном примере таблица сопряженности у нас одномерная – это просто вектор натуральных частот значений дискоидального смещения пчел. Предположим, что нас интересуют частоты дискоидального смещения не в целом по выборке, а в зависимости от породной принадлежности, в этом случае необходимо построить двумерную таблицу сопряженности:

```
> (f <- table(apis$discoidal.shift, apis$breed))
           carnica caucasica mellifera
  negative       2        87        19
  neutral       13        64        17
  positive     225        89        78
```

Строкам таблицы сопряженности соответствуют уникальные значения переменной, которая указана первой при вызове функции `table()`, столбцам – значения второй переменной. В общем случае число переменных не ограничено. Объект класса `table` представляет собой многомерный массив с именованными измерениями. В ячейках массива содержатся частоты комбинаций значений соответствующих переменных. В частности, полученная выше таблица демонстрирует, что в нашей выборке есть 2 пчелы породы *A. m. carnica*, имеющих отрицательное дискоидальное смещение, 87 пчел породы *A. m. caucasica*, имеющих отрицательное дискоидальное смещение и так далее. Сумма по строкам соответствует частотам значений дискоидального смещения, а сумма по столбцам – частотам пород. Для их вычисления на основе уже имеющейся таблицы сопряженности можно воспользоваться функцией `margin.table()`:

```
> margin.table(f, margin = 1)
negative  neutral positive
     108       94      392
> margin.table(f, margin = 2)
  carnica caucasica mellifera
      240       240       114
```

Аргумент `margin` определяет номер измерения таблицы сопряженности, по которому необходимо просуммировать частоты, первым измерением считаются строки, вторым – столбцы и т.д.

При работе с таблицами сопряженности часто более информативно выражать данные не в виде частот, а в виде долей или пропорций. Преобразовать таблицу сопряженности в доли можно с помощью функции `prop.table()`:

```
> prop.table(f)
              carnica   caucasica   mellifera
  negative 0.003367003 0.146464646 0.031986532
  neutral  0.021885522 0.107744108 0.028619529
  positive 0.378787879 0.149831650 0.131313131
```

По умолчанию функция `prop.table()` пересчитывает значения частот в доли от общего объема выборки. Если умножить полученные значения на 100, мы получим проценты:

```
> prop.table(f) * 100
              carnica  caucasica  mellifera
  negative  0.3367003 14.6464646  3.1986532
  neutral   2.1885522 10.7744108  2.8619529
  positive 37.8787879 14.9831650 13.1313131
```

Так, 37.87 % выборки составляют пчелы породы *A. m. carnica*, имеющие положительное дискоидальное смещение. Иногда полезно получить доли (либо проценты) не от всей выборки, а внутри категорий одной из рассматриваемых переменных. В этом случае необходимо использовать аргумент `margin` при вызове функции `prop.table()`:

```
> prop.table(f, margin = 2) * 100
              carnica  caucasica  mellifera
  negative  0.8333333 36.2500000 16.6666667
  neutral   5.4166667 26.6666667 14.9122807
  positive 93.7500000 37.0833333 68.4210526
```

Полученная таблица демонстрирует, что 93.75 % пчел породы *A. m. carnica* имеют положительное дискоидальное смещение, 5.41 % нейтральное и только 0.83 % имеют отрицательное дискоидальное смещение. Совсем иное соотношение наблюдается для породы *A. m. caucasica*, где распределение частот гораздо ближе к равномерному.

Для визуализации качественных данных можно воспользоваться столбчатыми диаграммами. Новая столбчатая диаграмма создается функцией `barplot()`, которая обладает рядом стандартных графических аргументов, а также аргументами `height` (первый аргумент, задает высоту столбцов), `width` (задает ширину столбцов) и `space` (задает промежутки между столбцами). Логический аргумент `horiz` (по умолчанию `FALSE`) определяет, нужно ли строить

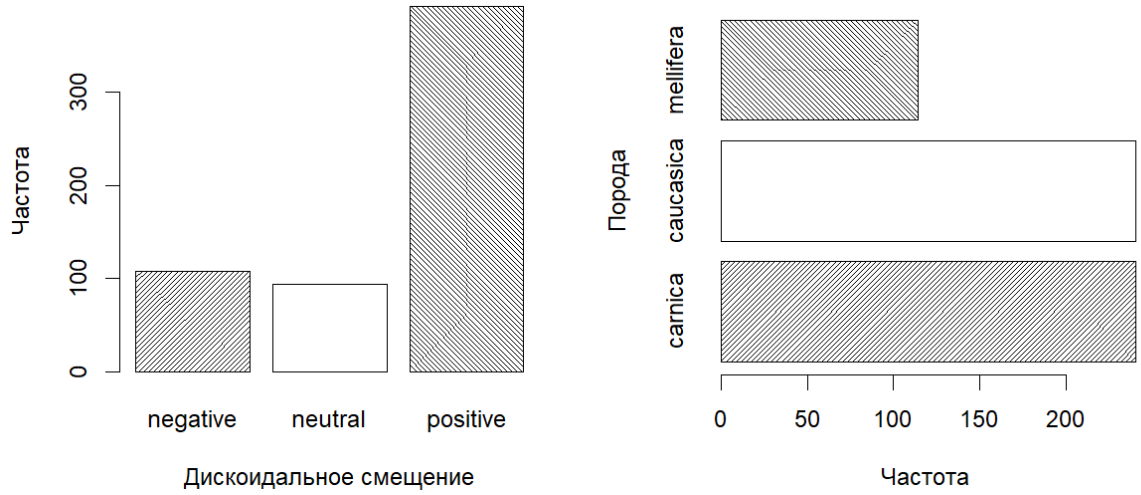

Рис. 9. Простые столбчатые диаграммы

диаграмму горизонтально. Построим столбчатые диаграммы с частотами пород и дискоидального смещения (Рис. 9):

```
> barplot(table(apis$discoidal.shift),
+         density = c(20, 0, 20), angle = c(45, 0, -45),
+         xlab = "Дискоидальное смещение", ylab = "Частота")
> barplot(table(apis$breed), horiz = TRUE,
+         density = c(20, 0, 20), angle = c(45, 0, -45),
+         xlab = "Частота", ylab = "Порода")
```

Сначала строится одномерная таблица сопряженности, которая передается аргументу `height` (это первый аргумент, поэтому его можно не именовать) функции `barplot()`, которая «умеет» работать с данными в формате `table`, в частности – автоматически формируются подписи столбцов. При построении второго графика мы использовали аргумент `horiz = TRUE`, в результате построена горизонтальная диаграмма.

Если аргументу `height` передать двумерную таблицу сопряженности, то будет построена составная столбчатая диаграмма, а если при этом использовать специальный логический аргумент `beside = TRUE`, то будет построена столбчатая диаграмма с группировкой. В этих случаях группировка будет осуществляться по той переменной, которая соответствует столбцам таблицы сопряженности, если необходимо изменить группирующую переменную, то таблицу сопряженности нужно предварительно транспонировать с помощью функции `t()`. Рассмотрим пример сложных столбчатых диаграмм, построенных на основе двумерной таблицы сопряженности `f`:

```
> barplot(f, xlab = "Порода", ylab = "Частота",
+         density = c(20, 0, 20), angle = c(45, 0, -45),
+         legend = rownames(f), args.legend = list(bty = "n"))
```

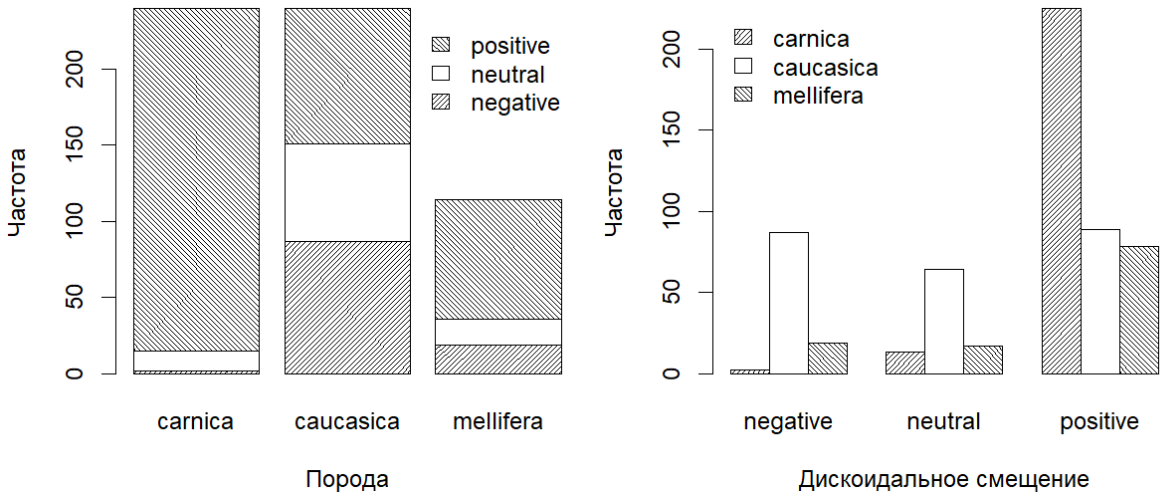

Рис. 10. Сложные столбчатые диаграммы

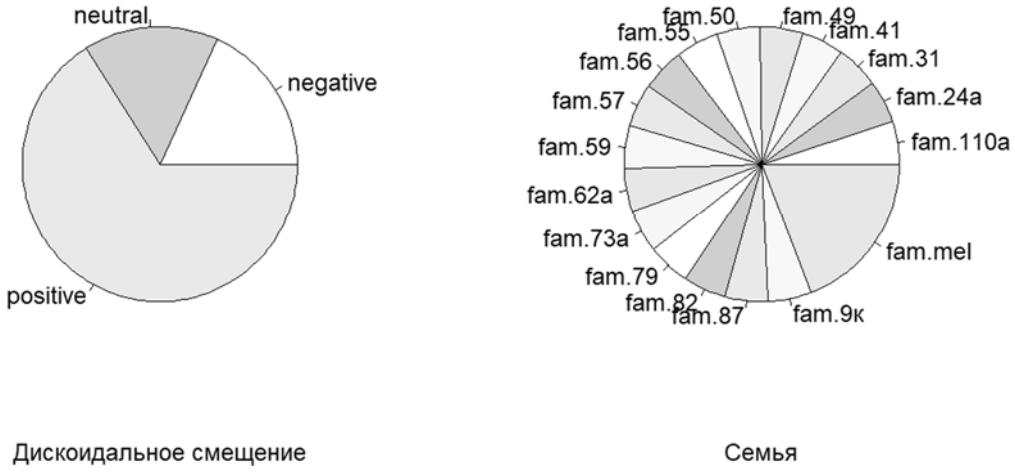

Рис. 11. Круговые диаграммы

```
> barplot(t(f), beside = TRUE, density = c(20,0,20), ylab = "Частота",
+       angle = c(45, 0, -45),  xlab = "Дискоидальное смещение",
+  legend = colnames(f), args.legend = list(x = "topleft", bty = "n"))
```

В первом случае (Рис. 10, слева) построена составная диаграмма, каждому столбику соответствует столбец таблицы сопряженности (в данном случае – породы пчел), а сегменты внутри столбиков соответствуют строкам таблицы сопряженности (в данном случае – значениям дискоидального смещения). Мы также добавили на график легенду с помощью аргумента `legend`, которому передан вектор значений дискоидального смещения, для создания которого мы воспользовались атрибутом `rownames()` нашей таблицы сопряженности.

Во втором случае (Рис. 10, справа) мы транспонировали таблицу сопряженности (поэтому теперь данные сгруппированы не по породам, а по значениям дискоидального смещения) и воспользовались аргументом `beside`,

поэтому теперь данные представлены в виде сгруппированных столбиков. В качестве значений для легенды мы теперь воспользовались атрибутом `colnames()` таблицы сопряженности, а для контроля положения легенды мы использовали аргумент `args.legend()`, через который можно в виде списка передавать параметры, определяющие вид легенды. В частности, мы воспользовались аргументом положения `x`, поскольку по умолчанию легенда размещается в верхнем правом углу и на нашем графике она наложилась бы на высокий столбик положительного дискоидального смещения.

Помимо столбчатых диаграмм для визуализации качественных данных можно использовать круговые диаграммы, которые создаются функцией `pie()`:

```
> pie(table(apis$discoidal.shift), xlab = "Дискоидальное смещение")
> pie(table(apis$family), xlab = "Семья")
```

В данном примере (Рис. 11) построены круговые диаграммы распределения значений дискоидального смещения и семейной принадлежности. Обратите внимание, что функция `pie()` является обычной графической функцией и поддерживает множество обычных графических аргументов. Несмотря на большую популярность круговых диаграмм, вместо них рекомендуется использовать столбчатые диаграммы, поскольку людям легче сравнивать длины, чем площади.

Глава 3. Нормальное распределение и диагностика отклонений

Симметричные одновершинные распределения колоколообразной формы чрезвычайно широко распространены в статистике. Настолько широко, что описывающее их распределение принято называть нормальным. Помимо широкого эмпирического распространения, нормальное распределение имеет важнейшее теоретическое значение: согласно центральной предельной теореме именно такое распределение имеет сумма большого числа независимых одинаково распределенных случайных величин.

Важнейшее практическое значение имеет вопрос о том, можно ли считать выборку подчиняющейся нормальному распределению. Значительная часть методов традиционной математической статистики исходит из предположения о том, что данные распределены нормально и проверяет гипотезы, касающиеся параметров нормального распределения. Такие методы называются параметрическими. Строго говоря, параметрическими считаются любые методы, подразумевающие какое-либо распределение исходных данных, не только нормальное, однако на практике исследователь обычно имеет дело с методами, основанными на нормальном распределении. В противоположность им существуют методы, называемые непараметрическими, которые не делают предположений о характере распределения исходных данных – единственное предположение заключается в том, что сравниваемые совокупности имеют одно и то же распределение. Параметрические методы в целом обладают большей мощностью (разрешающей способностью) по сравнению с непараметрическими, однако в некоторых ситуациях использование параметрической статистики для анализа данных, отклоняющихся от нормального распределения, может привести к неверным выводам. Поэтому одним из первых этапов при анализе данных является диагностика отклонений от нормального распределения, которая помогает решить, какие методы дальнейшего анализа следует применять.

3.1. Свойства нормального распределения

Нормальное распределение имеет два параметра: среднее μ и стандартное отклонение σ. Среднее определяет положение распределения на числовой оси, а

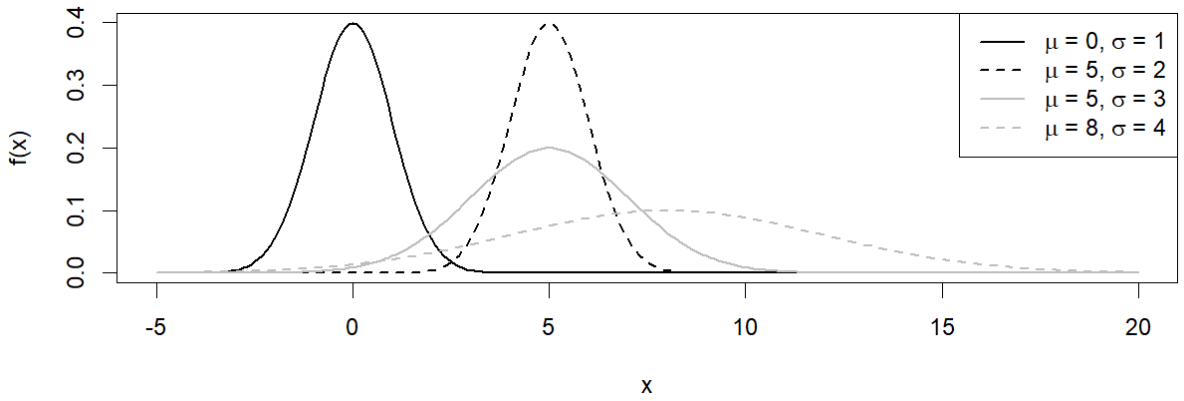

Рис. 12. Плотность нормального распределения

стандартное отклонение – разброс вокруг среднего. Проиллюстрируем свойства параметров графиками плотности распределения (Рис. 12):

```
> x <- seq(-5, 20, by = .1)
> nden1 <- dnorm(x)
> nden2 <- dnorm(x, mean = 5, sd = 1)
> nden3 <- dnorm(x, mean = 5, sd = 2)
> nden4 <- dnorm(x, mean = 8, sd = 4)
> plot(x, nden1, type = "l", lwd = 2, ylab = "f(x)")
> lines(x, nden2, lwd = 2, lty = 2)
> lines(x, nden3, lwd = 2, col = "grey")
> lines(x, nden4, lwd = 2, col = "grey", lty = 2)
> legend("topright", legend = c(expression(mu~"= 0," ~ sigma ~ "= 1"),
+                               expression(mu ~ "= 5," ~ sigma ~ "= 1"),
+                               expression(mu ~ "= 5," ~ sigma ~ "= 2"),
+                              expression(mu ~ "= 8," ~ sigma ~ "= 4")),
+  lwd = 2, col = c("black"," black","grey ","grey"), lty = c(1,2,1,2))
```

Плотность нормального распределения рассчитывается функцией `dnorm()`, а параметры распределения задаются аргументами `mean` (по умолчанию 0) и `sd` (по умолчанию 1). Увеличение μ ведет к смещению распределения вправо. Увеличение σ уменьшает концентрацию распределения в окрестностях среднего, за счет чего график плотности «прижимается» к числовой оси, поскольку должно сохраняться основное свойство плотности распределения вероятностей – площадь под ее графиком всегда равна единице (или в символьной форме: $\int_{-\infty}^{\infty} f(x)dx = 1$).

Нормальное распределение с нулевым средним и единичным стандартным отклонением (это параметры по умолчанию в R) называется стандартным нормальным распределением и часто обозначается как z-распределение. К этому распределению стремится распределение Стьюдента при росте числа степеней

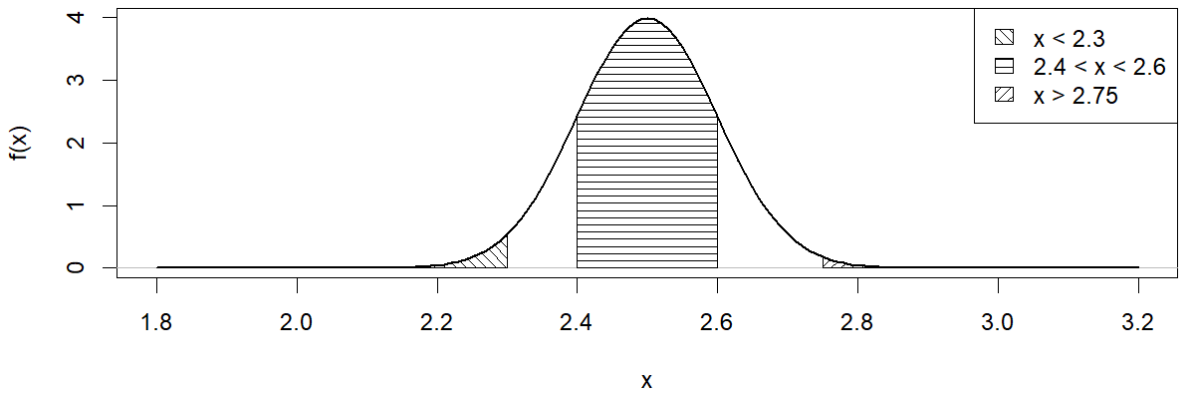

Рис. 13. К расчету вероятностей попадания в интервал

свободы. Любое нормальное распределение может быть приведено к стандартному путем так называемого *z*-преобразования:

$$z = \frac{x - \mu}{\sigma},$$

то есть, если из каждого значения вычесть среднее и разность отнести к стандартному отклонению, преобразованные данные будут иметь стандартное нормальное распределение.

Рассмотрим пример использования нормального распределения для решения вероятностных вопросов, связанных с экспериментальными данными. Согласно литературным данным средняя длина третьего тергита пчел породы *A. m. mellifera* составляет 2.5 мм, а стандартное отклонение – 0.1 мм. Исходя из этих данных, какова ожидаемая в выборке доля пчел, имеющих длину тергита (1) меньше 2.3 мм, (2) больше 2.75 мм, (3) от 2.4 до 2.6 мм? Изобразим интересующие нас интервалы на графике плотности распределения (Рис. 13):

```
> x <- seq(1.8, 3.2, by = 0.01)
> nden <- dnorm(x, mean = 2.5, sd = 0.1)
> plot(x, nden, type = "l", lwd = 2, ylab = "f(x)")
> abline(h = 0, col = "grey")
> # первый интервал
> int1 <- x <= 2.3
> polygon(c(1.8, x[int1], 2.3), c(0, nden[int1], 0),
+           density=10, angle= -45)
> # второй интервал
> int2 <- x >= 2.4 & x <= 2.6
> polygon(c(2.4, x[int2], 2.6), c(0, nden[int2], 0),
+           density= 10, angle = 0)
> # третий интервал
> int3 <- x >= 2.75
> polygon(c(2.75, x[int3], 3.2), c(0, nden[int3], 0),
+           density=10, angle= 45)
> # легенда
> legend("topright",legend = c("x < 2.3", "2.4 < x < 2.6", "x > 2.75"),
+         density = 10, angle = c(-45, 0, 45))
```

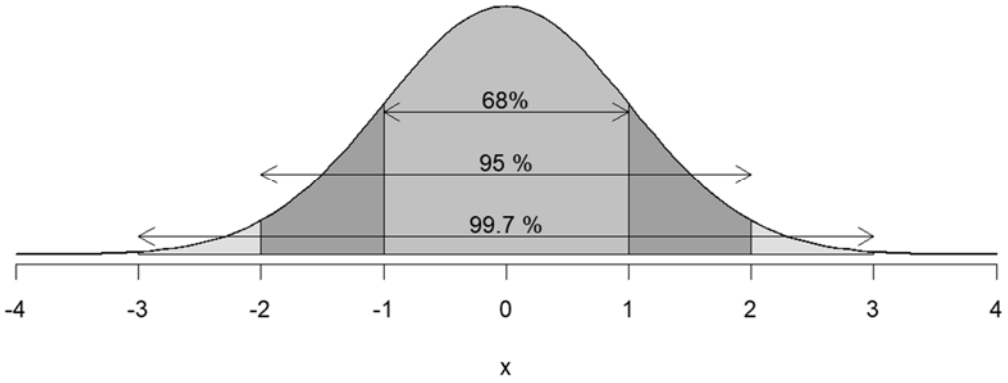

Рис. 14. «Правило *n* сигм»

Для штриховки областей мы нарисовали многоугольники функцией `polygon()`. Нас интересует площадь заштрихованных областей. Для ее нахождения необходимо воспользоваться функцией нормального распределения *F(x)*, для расчета которой служит функция `pnorm()`:

```
> pnorm(2.3, mean = 2.5, sd = 0.1)
[1] 0.02275013
> pnorm(2.6, mean = 2.5, sd = 0.1) - pnorm(2.4, mean = 2.5, sd = 0.1)
[1] 0.6826895
> 1 - pnorm(2.75, mean = 2.5, sd = 0.1)
[1] 0.006209665
```

Итак, зная только параметры нормального распределения, мы выяснили, что в типичной выборке пчел породы *A. m. mellifera* следует ожидать около 2.3 % пчел с длиной тергита менее 2.3 мм, 68 % пчел с длиной тергита от 2.4 до 2.6 мм и всего лишь немногим более 0.6 % пчел с длиной тергита, превышающей 2.75 мм.

Можно сформулировать вопрос другого типа: в каких пределах следует ожидать нахождение 95 % типичной выборки пчел? Поскольку мы работаем с симметричным нормальным распределением, смысл вопроса заключается в нахождении квантилей, которые отсекают по (100 % – 95 %) / 2 сверху и снизу распределения, то есть квантилей уровня 0.025 и 0.975. Для их нахождения нужно воспользоваться функцией `qnorm()`:

```
> qnorm(0.025, mean = 2.5, sd = 0.1)
[1] 2.304004
> qnorm(0.975, mean = 2.5, sd = 0.1)
[1] 2.695996
```

Итак, 95 % типичной выборки пчел породы *A. m. mellifera* попадает в интервал от 2.304 до 2.696 мм по длине тергита.

Масштаб изменчивости нормального распределения задается стандартным отклонением, поэтому полезно выражать отклонения от среднего в стандартных отклонениях (то есть, по сути, применять *z*-преобразование). В качестве практического правила широко применяется утверждение, известное как «правило *n* сигм»: 68 % значений отклоняются от среднего менее чем на стандартное отклонение, 95 % – менее чем на 2 стандартных отклонения, 99.7 % – менее чем на 3 стандартных отклонения. В справедливости этого правила легко убедиться графически (Рис. 14):

```
> x <- seq(-4, 4, by = 0.1)
> nden <- dnorm(x)
> plot(x, nden, type = "l", lwd = 2, axes = F, ylab = "")
> axis(1, at = -4:4, labels = TRUE)
> int1 <- x >= -3 & x <= 3
> polygon(c(-3,x[int1],3), c(0,nden[int1],0), col = "wheat")
> int2 <- x >= -2 & x <= 2
> polygon(c(-2,x[int2],2), c(0,nden[int2],0), col = "limegreen")
> int3 <- x >= -1 & x <= 1
> polygon(c(-1,x[int3],1), c(0,nden[int3],0), col = "skyblue")
> arrows(x0 = c(-1,-2,-3),y0 = c(.23,.13,.03),x1 = c(1,2,3),code = 3)
> text(x = 0, y = c(.25, .15, .05), labels = c("68%","95 %","99.7 %"))
```

Рассчитать точные доли значений, попадающих в соответствующие интервалы, также не составляет труда:

```
> pnorm(1) - pnorm(-1)
[1] 0.6826895
> pnorm(2) - pnorm(-2)
[1] 0.9544997
> pnorm(3) - pnorm(-3)
[1] 0.9973002
```

3.2. Визуальные средства диагностики отклонений

Для визуальной диагностики соответствия данных нормальному распределению используются два типа графиков: гистограммы эмпирического распределения с наложенной плотностью нормального распределения и так называемые квантильные графики.

Рассмотрим применение этих графиков на примере анализа распределения длины тергита и длины хоботка в первой семье пчел породы *A. m. carnica*, для чего выберем данные по этой семье в отдельный фрейм:

```
> load("apis.rda")
> fam1 <- subset(apis, subset = family == family[1])
```

Построим теперь гистограмму распределения длины тергита в этой семье:

```
> opar <- par(mfrow = c(2, 2), mar = c(5, 4, 2, 1))
> hist(fam1$tergit.length, main = "(a)", freq = FALSE, col = "grey",
+      xlab = "Длина тергита, мм", ylab = "Частота")
```

Обратите внимание, что при построении этой гистограммы мы воспользовались логическим аргументом `freq` и передали ему значение `FALSE`. Этот параметр определяет, нужно ли отображать на графике натуральные частоты (вариант по умолчанию) либо плотности вероятности (для значения `FALSE`). Плотность вероятности рассчитывается как отношение натуральной частоты к ширине интервала, при ее использовании общая площадь столбцов гистограммы становится равной единице.

Для анализа соответствия распределения нормальному наложим на полученную гистограмму график плотности нормального распределения (Рис. 15a):

```
> x <- seq(min(fam1$tergit.length), max(fam1$tergit.length),len=1000)
> y <- dnorm(x, mean(fam1$tergit.length), sd(fam1$tergit.length))
> lines(x, y, col = "darkblue", lwd = 2)
```

При идеальном соответствии эмпирического и теоретического распределений график плотности теоретического распределения пересекает вершины столбцов гистограммы посередине. На нашем графике мы видим такое пересечение для третьего и пятого столбцов, тогда как остальные столбцы отклоняются от графика плотности нормального распределения в большую (второй столбец) или меньшую стороны (четвертый столбец).

Построим теперь аналогичную гистограмму распределения длины хоботка (Рис. 15b):

```
> hist(fam1$proboscis, main = "(b)", freq = FALSE, col = "darkgrey",
+      xlab = "Длина хоботка, мм", ylab = "Частота")
> x <- seq(min(fam1$proboscis), max(fam1$proboscis), length = 1000)
> y <- dnorm(x, mean = mean(fam1$proboscis), sd = sd(fam1$proboscis))
> lines(x, y, col = "darkgreen", lwd = 2)
```

Распределение длины хоботка имеет совершенно иной характер. Оно сильно асимметричное и включает два сильно уклоняющихся значения (последние столбцы около 7 мм). Высота первого столбца соответствует плотности нормального распределения, тогда как остальные сильно отклоняются от него в большую или меньшую сторону.

Квантильный график строится функцией `qqnorm()`. Построим квантильный график для распределения длины тергита (Рис. 15c):

```
> qqnorm(fam1$tergit.length, main = "(c)", pch = 21, bg = "grey")
```

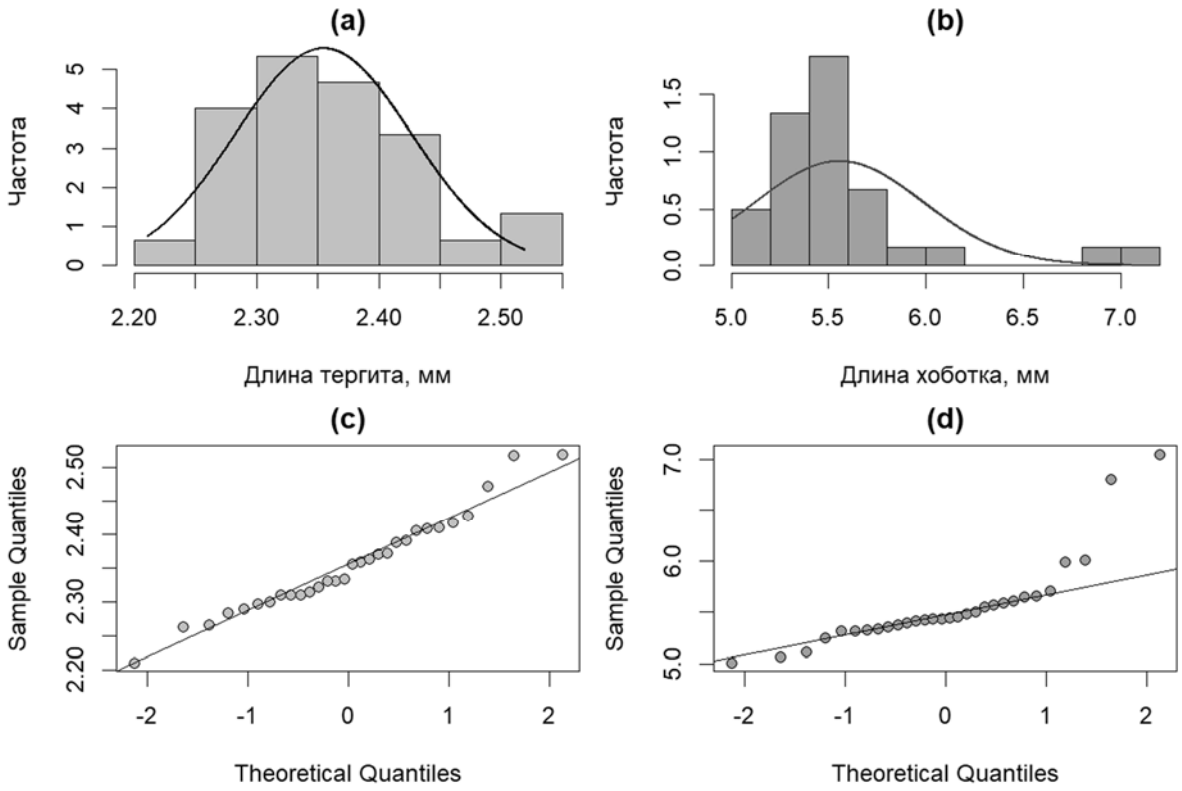

Рис. 15. Визуальный анализ нормальности распределения длины тергита и хоботка

Точки на этом графике соответствуют наблюдениям. По ординате отложены выборочные значения (выборочные квантили, sample quantiles), по абсциссе же отложены теоретические квантили (theoretical quantiles) – значения, меньше которых такая же доля значений в стандартном нормальном распределении. При соответствии теоретического и эмпирического распределений квантильный график ложится на прямую линию. Для удобства визуального сравнения эту линию можно поместить на график с использованием функции `qqline()`:

```
> qqline(fam1$tergit.length)
```

Видно, что точки хорошо соответствуют прямой, за исключением нескольких максимальных значений. Обратимся теперь к квантильному графику для распределения длины хоботка (Рис. 15d):

```
> qqnorm(fam1$proboscis, main = "(d)", pch = 21, bg = "darkgrey")
> qqline(fam1$proboscis)
> par(opar)
```

Картина во многом аналогична распределению длины хоботка: центр распределения хорошо ложится на прямую линию, но несколько значений в верхней части сильно отклоняются от нее. Степень отклонения здесь гораздо больше по сравнению с длинами тергита.

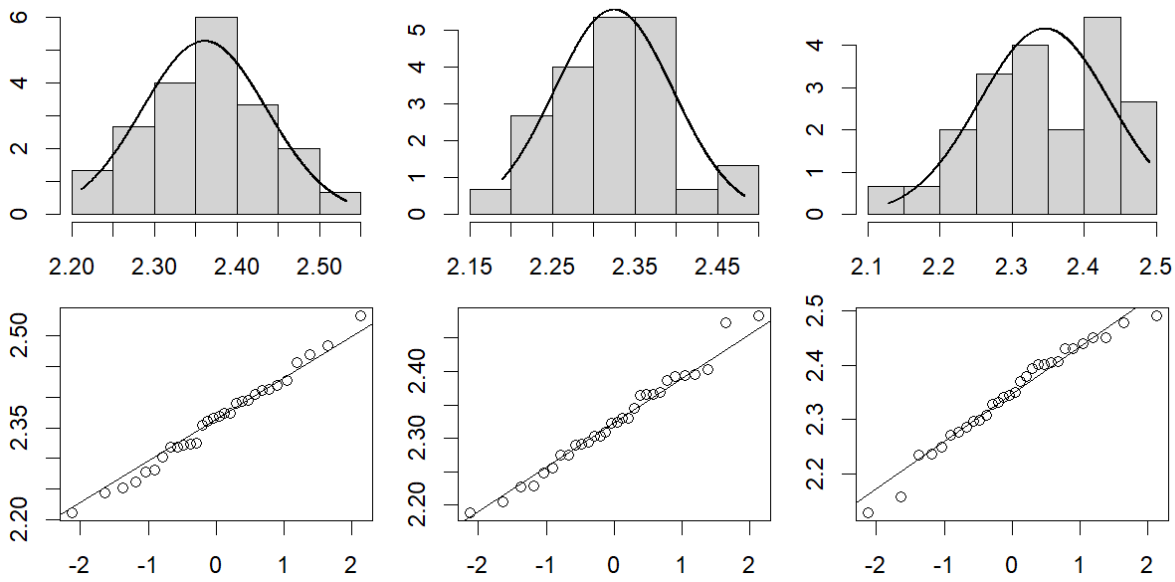

Рис. 16. Анализ нормальности искусственных выборок

Графические средства диагностики отклонений от нормального распределения требуют наличия определенного опыта работы с графиками этого типа. Основной вопрос заключается в следующем: насколько характерны те или иные отклонения для типичной выборки из нормального распределения? Мы видели, что распределение длины тергита в одной выборке пчел похоже на нормальное, но при этом все же отклоняется от него. Длины хоботка в той же выборке отклоняются еще сильнее. Но насколько те или иные отклонения существенны? Или такого рода отклонения типичны для выборок из нормального распределения? Чтобы получить какое-то представление об этом вопросе воспользуемся статистическим моделированием, то есть сгенерируем несколько выборок из нормального распределения аналогичного объема и построим для них гистограммы и квантильные графики.

```
> opar <- par(mfcol = c(2,3), mar = c(2,2,1,1))
> for (ii in 1:3)
+ {
+     sam <- rnorm(dim(fam1)[1], mean = mean(fam1$tergit.length),
+                  sd = sd(fam1$tergit.length))
+     hist(sam, main = "", freq = FALSE, col = "lightgrey")
+     x <- seq(min(sam), max(sam), length = 1000)
+     y <- dnorm(x, mean = mean(sam), sd = sd(sam))
+     lines(x, y, lwd = 2)
+     qqnorm(sam, main = "")
+     qqline(sam)
+ }
> par(opar)
```

Полученные таким образом графики для трех искусственных выборок представлены на Рис. 16. Графики построены по данным, соответствующим параметрам распределения длины тергита, и нас интересует масштаб отклонений от нормального распределения. Сопоставив их с графиками для эмпирического распределения длины тергита (Рис. 15a и 15c) легко убедиться, что отмеченные нами несоответствия теоретическим кривым вполне типичны для выборок из нормального распределения аналогичного объема. Сопоставление же с графиками для распределения длины хоботка (Рис. 15b и 15d) убеждает, что масштаб отклонений этого распределения от нормальности не характерен для искусственных выборок.

3.3. Критерии согласия

Гистограммы и квантильные графики служат для разведочного анализа. Они не дают строгих количественных оснований для принятия либо отклонения гипотезы о нормальном характере распределения данных. Мы рассмотрели два контрастных случая: длина тергита слабо отклоняется от нормальности, длина хоботка отклоняется сильно. Но возможен полный диапазон переходных случаев. Где пролегает граница? Однозначного ответа на этот вопрос не существует, но его можно перевести в вероятностную плоскость с помощью критериев согласия.

Критерии согласия предназначены для количественного сопоставления эмпирического распределения с теоретическим. Существуют универсальные критерии, пригодные для сравнения с распределением любого типа (например, критерий Колмогорова-Смирнова), но нас больше всего интересует сравнение с нормальным распределением, для чего предназначены несколько специальных критериев, в частности критерии Шапиро-Уилка, Лиллиефорса и Андерсона-Дарлинга.

Все критерии согласия построены по одному принципу. Нуль-гипотезой служит предположение о соответствии распределений. По эмпирическим данным рассчитывается некая критериальная статистика (показатель), для которой известно распределение в случае выполнения нуль-гипотезы. Рассмотрим всю совокупность возможных выборок из нормального распределения определенного объема. Для каждой из этих выборок можно рассчитать статистику критерия согласия. Распределение этой статистики показывает, какие значения встречаются часто, а какие редко. Зная это распределение и рассчитав эмпирическое значение критериальной статистики

можно оценить вероятность того, что в случайной выборке из нормального распределения будет получено такое же (или еще более экстремальное) значение, как в анализируемой выборке. Если эта вероятность велика, нуль-гипотеза принимается и выборка считается соответствующей нормальному распределению. Если же эта вероятность мала, нуль-гипотезу отвергают и считают выявленным отклонение от нормальности. Вопрос о том, какую вероятность следует считать достаточно малой для отбрасывания нуль-гипотезы решается конвенционально: в биологии таким критическим уровнем считается 5 %.

Наиболее распространенным критерием для диагностики нормальности эмпирического распределения является критерий Шапиро-Уилка. Он реализован в функции `shapiro.test()`:

```
> shapiro.test(fam1$tergit.length)

 Shapiro-Wilk normality test

data:  fam1$tergit.length
W = 0.9649, p-value = 0.4098
```

Мы применили критерий Шапиро-Уилка к распределению длины тергита в первой семье пчел. Функции передан числовой вектор значений выборки. После вызова функции в консоль выведен отчет о результатах применения критерия. Первая строка содержит название критерия (Shapiro-Wilk normality test – критерий Шапиро-Уилка для нормальности), затем указано, к каким данным он применен, последняя строка выводит значение критериальной статистики W и p-значение. Последнее представляет собой вероятность получить значение W, большее чем 0.9649, в случайной выборке из нормального распределения. На основании этой вероятности обычно и принимается решение, можно ли считать нашу выборку соответствующей нормальному распределению: если $p > 0.05$, то нет оснований отклонять нуль-гипотезу, если же $p < 0.05$, то отклонения от нормальности считаются выявленными. В случае анализа распределения длины тергита $p = 0.4098$, что говорит о том, что отклонения нашей выборки от теоретического распределения вполне типичны для случайных выборок из него (встречаются примерно в 40 % выборок из нормального распределения аналогичного объема). Таким образом, можно считать, что длина тергита в первой семье пчел распределена нормально.

Применим критерий Шапиро-Уилка к распределению длины хоботка:

```
> shapiro.test(fam1$proboscis)

 Shapiro-Wilk normality test

data:  fam1$proboscis
W = 0.7521, p-value = 9.845e-06
```

Значение критериальной статистики *W* меньше, значительно меньше и вероятность *p*. Значение вероятности свидетельствует о том, что имеющиеся в выборке отклонения от нормального распределения крайне маловероятны для случайных выборок из него, поэтому можно считать установленным фактом происхождение нашей выборки из какого-то другого распределения.

Помимо критерия Шапиро-Уилка для анализа нормальности распределения используются критерии Лиллиефорса и Андерсона-Дарлинга. Эти критерии отсутствуют в стандартных пакетах среды R, но их реализация содержится в пакете `nortest` (от англ. normality testing). Перед применением описанных ниже функций не забудьте установить этот пакет (выражением `install.packages("nortest")` либо вручную через графический интерфейс, см. раздел 2.7 части 1) и загрузить его для использования в текущей сессии выражением `library(nortest)`. Критерий Лиллиефорса реализован в функции `lillie.test()`, а критерий Андерсона-Дарлинга – в функции `ad.test()`. Применим критерии Лиллиефорса и Андерсона-Дарлинга для анализа нормальности распределения длины тергита и хоботка в первой семье пчел:

```
> library(nortest)
> lillie.test(fam1$tergit.length)

 Lilliefors (Kolmogorov-Smirnov) normality test

data:  fam1$tergit.length
D = 0.1189, p-value = 0.3437

> ad.test(fam1$tergit.length)

 Anderson-Darling normality test

data:  fam1$tergit.length
A = 0.3788, p-value = 0.384
```

Форма вызова и структура вывода результата аналогична функции `shapiro.test()`. Оба критерия не выявляют отклонений распределения длины тергита от нормальности ($p > 0.05$).

```
> lillie.test(fam1$proboscis)

 Lilliefors (Kolmogorov-Smirnov) normality test

data:  fam1$proboscis
D = 0.239, p-value = 0.0001337
> ad.test(fam1$proboscis)

 Anderson-Darling normality test

data:  fam1$proboscis
A = 2.5124, p-value = 1.641e-06
```

Для длины хоботка критерии Лиллиефорса и Андерсона-Дарлинга выявляют отклонения от нормальности ($p < 0.05$).

Таким образом, все три критерия согласия в рассмотренных случаях дали одинаковые результаты. Хотя численные *p*-значения различались, качественный вывод сохранялся неизменным: для длины тергита значимых отклонений от нормальности нет, а для длины хоботка – есть. Так происходит далеко не всегда. Иногда встречаются ситуации, когда один из критериев согласия выявляет отклонения от нормальности, тогда как другие – нет. Дело в том, что отклонения от нормальности могут иметь разный характер и критерии согласия имеют разную чувствительность к отклонениям разного типа. Именно поэтому для выявления отклонений от нормальности рекомендуется применять сразу несколько критериев согласия. Иногда такую комплексную процедуру называют «батареей тестов». Если хотя бы один из них срабатывает, то отклонения от нормальности следует считать установленными.

Для удобной реализации батареи тестов на отклонения от нормальности в R можно написать специальную функцию, например:

```
> normbat <- function(x)
+ {
+     shapiro.test(x)$p.value > .05 & lillie.test(x)$p.value > .05 &
+         ad.test(x)$p.value > .05
+ }
```

Функция `normbat()` принимает в качестве аргумента числовой вектор `x` и применяет к нему критерии Шапиро-Уилка, Лиллиефорса и Андерсона-Дарлинга. Полученные в результате применения критериев *p*-значения сравниваются с пороговым уровнем 0.05 и, в случае отрицательного диагноза хотя бы одного из тестов, возвращается значение `FALSE`, в противном случае возвращается значение `TRUE`. В качестве примера применим эту функцию для

диагностики отклонений от нормальности длины и ширины крыла в первой выборке пчел:

```
> normbat(fam1$wing.width)
[1] TRUE
> normbat(fam1$wing.length)
[1] TRUE
```

Полученные оба раза значения `TRUE` свидетельствуют об отсутствии значимых отклонений от нормальности.

Результаты анализа нормальности распределения всех количественных признаков пчел во всех семьях представлены в таблице 2. Единицы соответствуют выявлению отклонений с помощью батареи тестов из критериев Шапиро-Уилка, Лиллиефорса и Андерсона-Дарлинга, нули – отсутствию значимых отклонений. Отклонения от нормальности хотя бы в одной семье выявлены для всех признаков, однако имеется значительный контраст между длиной хоботка и остальными признаками. Длина хоботка распределена не нормально во всех семьях, а для остальных признаков отклонения выявлены менее чем в половине семей. В дальнейшем анализе мы будем полагать нормальным распределение всех признаков за исключением длины хоботка. Соответственно длина хоботка будет использоваться для демонстрации непараметрических методов, а остальные признаки (чаще всего – длина тергита) – для демонстрации параметрических методов.

Таблица 2

Результаты анализа нормальности распределения количественных морфологических признаков в семьях пчел
(1 – отклонения значимы, 0 – отклонения не выявлены)

Порода	**Семья**	**Признаки**					
		Длина хоботка	Длина тергита	Ширина тергита	Ширина крыла	Длина крыла	Кубитальный индекс
A.m.carnica	fam.110a	1	0	0	1	0	0
	fam.24a	1	0	0	0	0	0
	fam.31	1	0	1	1	1	0
	fam.41	1	0	1	1	1	0
	fam.49	1	1	0	0	0	0
	fam.50	1	0	0	0	0	0
	fam.55	1	0	0	0	0	1
	fam.56	1	0	0	0	0	1
A.m.caucasica	fam.57	1	0	0	1	1	0
	fam.59	1	0	0	0	0	0
	fam.62a	1	0	0	1	0	0
	fam.73a	1	0	0	1	0	0
	fam.79	1	0	0	0	1	1
	fam.82	1	0	1	0	0	0
	fam.87	1	1	1	1	1	0
	fam.9к	1	0	0	0	0	0
A.m.mellifera	fam.mel	1	0	0	1	1	0
Всего		**17**	**2**	**4**	**8**	**6**	**3**

Глава 4. Простейшие гипотезы: одно- и двухвыборочные сравнения

Начиная с этой главы мы переходим к описанию конкретных статистических методов, которые часто используются при анализе биомедицинских и экологических данных. Основное внимание будет уделяться не теоретическим основам различных методов, а деталям их практической реализации в среде R. В подавляющем большинстве случаев пользователю нет необходимости вникать в механику метода: почему используется именно такая критериальная статистика (часто даже нет необходимости знать формулу расчета), почему эта статистика имеет определенное распределение, почему необходим именно этот набор предпосылок. Для успешного применения статистического метода необходим следующий минимальный набор сведений: для чего применяется метод, какие ограничения (требования к данным) при его применении, каковы основные результаты и характер их интерпретация, техника реализации расчетов в R (как должны быть организованы данные, какие функции и в каком порядке нужно применять, какова структура вывода). Именно на них будет сосредоточено основное внимание.

Мы начнем с описания группы методов, которые позволяют сравнивать выборочные средние с конкретным значением (эталоном) и между собой в случае двух выборок.

4.1. Критерий Стьюдента для сравнения среднего с эталоном

Главной предпосылкой для применения критерия Стьюдента является предположение о том, что анализируемая выборка происходит из генеральной совокупности, которая имеет нормальное распределение. Критерий Стьюдента – это типичный параметрический метод.

Пусть мы имеем значения $x_1, x_2, \ldots, x_n$, каждое из которых является независимой реализацией случайной величины, имеющей распределение $N(\mu, \sigma^2)$, то есть нормальное распределение со средним μ и дисперсией σ^2. Мы хотим проверить гипотезу $\mu = \mu_0$. Выборочными оценками неизвестных параметров μ и σ являются среднее $\bar{x}$ и стандартное отклонение s, однако, вследствие случайной природы нашей выборки, мы никогда не сможем оценить параметры точно, то есть всегда существует некая ошибка.

Стандартной ошибкой среднего называется величина

$$s_{\bar{x}} = \frac{\sigma}{\sqrt{n}}.$$

Она представляет собой стандартное отклонение оценок среднего в случайных выборках объемом n из нормального распределения со средним μ и дисперсией σ^2. Это означает, что если мы будем повторять эксперимент множество раз и рассчитывать среднее значение для каждой новой выборки объема n, то распределение этих средних значений будет значительно уже исходного распределения (и тем уже, чем больше объем выборки n).

Смоделируем описанную ситуацию в R. Сгенерируем выборку из нормального распределения и рассчитаем среднее.

```
> mu <- 9
> sigma <- 2
> n <- 10
> x <- rnorm(n, mu, sigma)
> (m <- mean(x))
[1] 8.6661
```

Выборочное среднее несколько отличается от известного генерального среднего ($\mu = 9$). Можно сгенерировать еще одну выборку и убедиться в том, что будет получено немного отличающееся значение выборочного среднего:

```
> x <- rnorm(n, mu, sigma)
> (m[2] <- mean(x))
[1] 9.391798
```

Этот процесс можно продолжать. Суть вопроса заключается в том, насколько велик разброс выборочных средних вокруг истинного генерального среднего. Воспользуемся циклом и сгенерируем большое число выборочных средних:

```
> for (ii in 3:1000) m[ii] <- mean(rnorm(n, mu, sigma))
```

Рассмотрим теперь их распределение и рассчитаем его параметры:

```
> mean(m)
[1] 9.004806
> sd(m)
[1] 0.6412352
> hist(m, main = "", col = "wheat")
```

Среднее выборочных средних оказывается очень близко к истинному среднему μ, но при этом существует разброс, который характеризуется стандартным отклонением $s_{\bar{x}}$, существенно меньшим истинного стандартного отклонения σ генеральной совокупности. Гистограмма выборочных средних свидетельствует о том, что их распределение близко к нормальному, в чем легко убедиться и с применением критериев согласия.

Стандартная ошибка среднего характеризует разброс выборочных средних вокруг истинного генерального среднего, и она является оценкой стандартного отклонения этого распределения. Убедимся в этом численно:

```
> (sx <- sigma / sqrt(n))
[1] 0.6324555
```

Теоретическое значение стандартной ошибки среднего очень близко к стандартному отклонению смоделированной нами совокупности выборочных средних. Истинное значение параметров генеральной совокупности практически никогда неизвестно, поэтому при работе с реальными данными стандартные ошибки рассчитываются на основании выборочных оценок, то есть как $s/\sqrt{n}$.

Исходная гипотеза, которую мы проверяем, сформулирована как $\mu = \mu_0$. То есть мы хотим знать, отличается ли среднее генеральной совокупности, из которой происходит наша выборка (μ), от некого эталонного значения (μ_0). У нас есть выборочное среднее и мы можем оценить его разброс вокруг истинного среднего. Тогда можно рассчитать вероятность того, что наша выборка происходит из распределения со средним μ_0.

Продолжим наш модельный пример. Мы выяснили, что средние значения выборок объемом n подчиняются нормальному распределению со средним μ и стандартным отклонением $\sigma/\sqrt{n}$. Рассмотрим теперь величину

$$t = \frac{\bar{x} - \mu}{s_{\bar{x}}}.$$

Величина t характеризует отклонение выборочного среднего от генерального среднего и подчиняется стандартному нормальному распределению. Чем больше абсолютное значение t, тем менее вероятно его появление в случайной выборке. Например, в первой нашей искусственной выборке было получено среднее значение 8.6661, которому соответствует

```
> (t <- (m[1] - mu) / sx)
[1] -0.5279418
```

Можно рассчитать вероятность получения такого же или еще более низкого значения t с помощью функции стандартного нормального распределения:

```
> pnorm(t)
[1] 0.2987699
```

Однако выборочное среднее может отклоняться от истинного среднего как в большую, так и в меньшую сторону. Нас интересует именно величина отклонения, поэтому мы должны интересоваться вероятностью получить еще более экстремальное отклонение, чем данное. Поэтому к уже рассчитанной вероятности получения более низких значений нужно прибавить вероятность получения значений больших, чем $-t$. Ввиду симметричности стандартного

нормального распределения полученную ранее вероятность можно просто удвоить:

```
> pnorm(t) + 1 - pnorm(-t)
[1] 0.5975397
> pnorm(t) * 2
[1] 0.5975397
```

Итак, примерно 60 % выборок из нормального распределения характеризуются такими же отклонениями выборочного среднего от генерального среднего, как и наша выборка, или еще большими. То есть наша выборка является типичной (что совершенно неудивительно, поскольку мы сами ее сгенерировали).

Предположим теперь, что мы имеем дело с эмпирическими данными. У нас есть нуль-гипотеза, что они происходят из нормального распределения со средним μ_0 и дисперсией σ^2. Мы можем рассчитать выборочное среднее, его стандартную ошибку и величину отклонения t. Тогда можно рассчитать вероятность получения такого же или еще более экстремального отклонения в условиях нуль-гипотезы. Если эта вероятность мала, то, скорее всего, наша гипотеза не соответствует действительности и ее следует отвергнуть. Напротив, если эта вероятность велика, нуль-гипотезу следует принять. Предельная вероятность, начиная с которой мы будем отвергать нуль-гипотезу, называется *уровнем значимости* и обозначается α. В биологии и экологии принято использование 5 %-го уровня значимости, в других дисциплинах используются также 1 %-ный и 0.1 %-ный уровни значимости.

Мы только что сконструировали критерий для сравнения выборочного среднего с эталонным значением. Единственная тонкость заключается в том, что наш критерий годится только для ситуаций, когда известно истинное стандартное отклонение генеральной совокупности σ, чего практически никогда не бывает. Обычно стандартное отклонение оценивается по выборочным данным. В такой ситуации статистика t подчиняется не стандартному нормальному распределению, а распределению Стьюдента с $n - 1$ степенями свободы, что не представляет собой проблемы, поскольку с этим распределением мы тоже можем работать. Разновидности методов, в которых критериальная статистика имеет распределение Стьюдента, называются критериями Стьюдента (t-test).

Обратимся теперь к конкретному эмпирическому примеру. Согласно литературным данным средняя длина третьего тергита пчел породы *A. m. mellifera* составляет 2.45 мм. Выясним, соответствует ли имеющаяся в

нашем распоряжении выборка пчел этой породы данному стандарту? Сформулируем рабочие гипотезы:

$$H_0: \mu = 2.45,$$
$$H_1: \mu \neq 2.45.$$

Нуль-гипотеза обозначается символом H_0. Противоположное утверждение называется альтернативной гипотезой и обозначается символом H_1.

```
> load("apis.rda")
> mel <- subset(apis, subset = breed == "mellifera")
```

Мы выбрали все данные по породе *A. m. mellifera* в отдельный фрейм `mel`. Рассчитаем необходимые для применения критерия статистики – среднее `m`, стандартную ошибку среднего `se.m` и отклонение `t`:

```
> (m <- mean(mel$tergit.length))
[1] 2.469166
> (se.m <- sd(mel$tergit.length) / sqrt(dim(mel)[1]))
[1] 0.00604139
> (t <- (m - 2.45) / se.m)
[1] 3.172414
```

Теперь можно рассчитать вероятность получения такого же или более экстремального отклонения в условиях нуль-гипотезы:

```
> pt(-abs(t), df = dim(mel)[1] - 1) * 2
[1] 0.001947129
```

Таким образом, мы можем отвергнуть нуль-гипотезу на 5 %-ном уровне значимости ($p < 0.05$): имеющиеся отклонения слишком велики, чтобы считаться случайными.

В приведенном примере мы сами рассчитали критериальную статистику t и соответствующее ей p-значение. Однако проще было воспользоваться специальной функцией `t.test()`, которая позволяет применить критерий Стьюдента и получить тот же самый результат:

```
> t.test(mel$tergit.length, mu = 2.45)

 One Sample t-test

data:  mel$tergit.length
t = 3.1724, df = 113, p-value = 0.001947
alternative hypothesis: true mean is not equal to 2.45
95 percent confidence interval:
 2.457197 2.481135
sample estimates:
mean of x
 2.469166
```

В данном случае функции передан числовой вектор, а также значение mu, с которым нужно произвести сравнение. Структура вывода типична для функций, реализующих статистические критерии в R. Первая строка содержит название применяемого критерия (One Sample t-test – критерий Стьюдента для одной выборки). Обратите внимание, что функция сама определила, какой из вариантов критерия Стьюдента использовать, основываясь на характере входящих данных. Вторая строка описывает данные, к которым применен критерий (data: mel$tergit.length). Третья строка выдает значение критериальной статистики (t), число степеней свободы (df) и *p*-значение (p-value). Обратите внимание, что здесь фигурирует *p*-значение, которое мы получили самостоятельно чуть раньше. Четвертая строка содержит формулировку альтернативной гипотезы (true mean is not equal to 2.45 – истинное среднее не равно 2.45). Далее следует 95 %-ный доверительный интервал – это такой интервал, который покрывает истинное среднее с заданной вероятностью. В данном случае, можно утверждать, что неизвестное нам истинное среднее с вероятностью 0.95 содержится в интервале от 2.457 до 2.481 мм – это так называемая интервальная оценка истинного среднего. Доверительный интервал не покрывает значение, с которым мы производили сравнение, что не удивительно, поскольку мы обнаружили отличия на аналогичном уровне значимости. Последний элемент вывода содержит оценку параметра: выборочное среднее (mean of x) – это точечная оценка истинного среднего.

Функция t.test() имеет множество аргументов, три из которых имеют отношение к одновыборочным критериям. Мы уже использовали аргумент mu, который определяет среднее значение для нуль-гипотезы, по умолчанию mu = 0. Аргумент conf.level определяет вероятность для построения доверительного интервала. Например, для получения 99 % доверительного интервала нужно указать conf.level = 0.99. Аргумент alternative определяет тип проверяемой гипотезы. Он может принимать значения «two.sided» (двусторонняя, используется по умолчанию), «greater» (больше) или «less» (меньше). Выше мы рассмотрели применение так называемой двусторонней альтернативной гипотезы, когда отклонения от среднего возможны в обе стороны. В некоторых ситуациях рассматриваются отклонения только в одну сторону, поэтому при расчете вероятности получения имеющихся или более экстремальных отклонений учитывается только один хвост распределения, а не два, соответственно эта вероятность в два раза меньше. При необходимости использовать одностороннюю альтернативную гипотезу, через аргумент alternative нужно указать соответствующее направление возможных отклонений (больше или меньше).

4.2. Критерий Стьюдента для сравнения двух выборочных средних

Двухвыборочный критерий Стьюдента проверяет гипотезу о том, что две выборки происходят из распределений с одинаковыми средними.

Пусть мы имеем две выборки $x_{11}, x_{12}, \ldots, x_{1n}$ и $x_{21}, x_{22}, \ldots, x_{2n}$, которые происходят из распределений $N(\mu_1, \sigma_1^2)$ и $N(\mu_2, \sigma_2^2)$. Проверяемые гипотезы формулируются как

$$H_0: \mu_1 = \mu_2,$$
$$H_1: \mu_1 \neq \mu_2.$$

Критериальная статистика t рассчитывается как отношение разности выборочных средних к стандартной ошибке этой разности:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{s_{\bar{x}_1 - \bar{x}_2}},$$

а стандартная ошибка разности в случае равенства объемов выборок определяется выражением

$$s_{\bar{x}_1 - \bar{x}_2} = \sqrt{s_{x_1}^2 + s_{x_2}^2}.$$

Существует две разновидности критерия Стьюдента, различающиеся предположением о равенстве дисперсий в сравниваемых группах. «Классический» вариант предполагает такое равенство, при его использовании статистика t в условиях нуль-гипотезы имеет распределение Стьюдента с $n_1 + n_2 - 2$ степенями свободы. Альтернативный вариант называется модификацией Уэлча, он не требует равенства дисперсий в сравниваемых группах. В этом случае распределение t-статистики может быть аппроксимировано распределением Стьюдента со средневзвешенным числом степеней свободы, рассчитываемым на основе выборочных дисперсий и объемов выборки, причем это число часто оказывается нецелым.

Оба варианта обычно дают очень близкие результаты, если разница в дисперсиях и объемах выборки не слишком велика. Тем не менее, модификация Уэлча не требует предварительного сравнения дисперсий, поэтому рассматривается как менее трудоемкая. Именно она используется в функции `t.test()` по умолчанию.

В качестве примера рассмотрим применение критерия Стьюдента для сравнения длины тергита в первых двух семьях пчел породы *A. m. carnica*:

```
> t.test(x = apis$tergit.length[1:30], y = apis$tergit.length[31:60])
 Welch Two Sample t-test

data:  apis$tergit.length[1:30] and apis$tergit.length[31:60]
t = -2.9673, df = 56.95, p-value = 0.004386
```

```
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.09974301 -0.01936365
sample estimates:
mean of x mean of y
 2.354863  2.414417
```

Первые два аргумента функции `t.test()` имеют имена `x` и `y` (их можно опустить) и представляют собой числовые вектора. В приведенном примере мы воспользовались прямым индексированием по номеру и получили необходимые данные непосредственно из фрейма `apis`. Структура вывода аналогична выводу для одновыборочного критерия; отличия заключаются в том, что выводится доверительный интервал для разности средних, а точечные оценки средних приводятся для двух выборок. Низкое *p*-значение (< 0.05) указывает на то, что нуль-гипотезу следует отвергнуть, то есть обнаружены статистически значимые отличия по длине тергита между семьями. Обратите также внимание на нецелое значение числа степеней свободы (`df = 56.95`).

Вызов функции `t.test()` с использованием двух векторов считается несколько старомодным. Более современным считается передача этой функции объекта класса формула (`formula`):

```
> t.test(tergit.length ~ family, data = apis[1:60, ])

 Welch Two Sample t-test

data:  tergit.length by family
t = -2.9673, df = 56.95, p-value = 0.004386
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.09974301 -0.01936365
sample estimates:
mean in group fam.24a mean in group fam.62a
             2.354863              2.414417
```

Формула имеет вид `tergit.length ~ family`. Она означает, что длина тергита (`tergit.length`) моделируется в зависимости (~) от семейной принадлежности (`family`). Слева от тильды стоит имя числовой переменной, справа – имя фактора с двумя уровнями, которые используются для разделения числовой переменной на выборки. Для того, чтобы функции было понятно, где искать имена переменных, аргументу `data` передается соответствующий фрейм данных. Поскольку мы применяем критерий Стьюдента, который не может одновременно сравнивать более двух выборок, мы передали только часть фрейма `apis`, в которой и содержатся данные по первым двум семьям. Такой вызов функции более удобен при работе с фреймами.

Результат применения критерия не изменился, но немного поменялся формат описания данных и групповых средних в выводе.

Чтобы получить классический вариант критерия Стьюдента, описанный во всех учебниках, нужно воспользоваться аргументом `var.equal` (равенство дисперсий) и установить его в значение `TRUE`:

```
> t.test(tergit.length ~ family, data = apis[1:60,], var.equal = TRUE)

 Two Sample t-test

data:  tergit.length by family
t = -2.9673, df = 58, p-value = 0.004359
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.09972724 -0.01937943
sample estimates:
mean in group fam.24a mean in group fam.62a
             2.354863              2.414417
```

Обратите внимание, что изменились только *p*-значение и доверительный интервал (да и те незначительно), а число степеней свободы стало целым числом (30 + 30 – 2 = 58).

4.3. Критерий Стьюдента для сравнения связанных выборок

Критерии для связанных выборок применяются в случаях, когда имеются два или несколько измерений одной и той же экспериментальной единицы. Типичная ситуация, когда возникают связанные данные – изучение одной и той же группы объектов во времени, например до и после изучаемого воздействия. Само это воздействие может быть как экспериментальным, так и вполне естественным. Например, в экологии часто изучаются одни и те же объекты (местообитания, сообщества) в разные сезоны года.

Пусть имеются две выборки, для которых необходимо сравнить средние, но при этом нужно учесть тот факт, что выборки не являются независимыми. Технически при этом вычисляется разница значений в каждой паре и задача сводится к одновыборочному критерию для сравнения выборки разностей с нулем.

Рассмотрим применение анализа связанных данных на примере исследования структуры сообщества макрозообентоса р. Кудьмы, которое было проведено в 2012–2013 гг. На 16 станциях в разные сезоны отбирались стандартные пробы зообентоса дночерпателем. Для каждой пробы определялся

видовой состав, число особей и биомасса каждого из обнаруженных видов. На основе этих данных каждая проба характеризовалась рядом показателей. Загрузим данные и рассмотрим их структуру.

```
> load("bent.rda")
> str(bent)
'data.frame':   64 obs. of  7 variables:
 $ S      : int  16 31 25 5 5 11 10 24 13 5 ...
 $ biomass: num  86.63 34.32 17.15 0.48 0.16 ...
 $ density: int  3080 17960 4280 440 320 1040 1120 9240 1920 1520 ...
 $ H      : num  2.91 3.59 3.71 1.97 2.25 2.92 2.57 2.67 3.19 0.97 ...
 $ C      : num  0.18 0.14 0.12 0.3 0.22 0.18 0.24 0.3 0.15 0.62 ...
 $ site   : Factor w/ 16 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10..
 $ date   : Factor w/ 4 levels "Jul12","May13",..: 1 1 1 1 1 1 1 1 1 1 ...
```

Данные сохранены в виде фрейма `bent`, который содержит 7 переменных, 5 из которых являются числовыми характеристиками проб – видовое богатство (`S`), биомасса (`biomass`), плотность особей (`density`), индекс разнообразия Шеннона (`H`), а также индекс доминирования Симпсона (`C`). Последние две переменные являются факторами: `site` – станция отбора проб, `date` – дата отбора. Станции обозначены номерами от 1 до 16, а даты – текстовыми значениями, соответствующими отбору проб в июле 2012 года, мае, июле и октябре 2013 года:

```
> levels(bent$date)
[1] "Jul12" "May13" "Jul13" "Oct13"
```

Совокупности проб с 16 станций в каждый период времени можно рассматривать как выборки из сообщества макрозообентоса р. Кудьма. Таким образом, имеющиеся данные позволяют провести анализ сезонной динамики различных показателей структуры сообщества.

Рассмотрим анализ межгодовой изменчивости видового богатства в пробах зообентоса. Необходимо сравнить две выборки проб, отобранных в июле 2012 и июле 2013 года. Видовое богатство в имеющихся выборках не отклоняется от нормального распределения, поэтому для анализа можно использовать параметрические методы, в частности – критерий Стьюдента.

```
> t.test(x = bent$S[bent$date == "Jul12"],
+        y = bent$S[bent$date == "Jul13"])

 Welch Two Sample t-test

data:  bent$S[bent$date == "Jul12"] and bent$S[bent$date == "Jul13"]
t = 1.852, df = 22.605, p-value = 0.07714
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.5171387  9.2754720
```

```
sample estimates:
mean of x mean of y
 14.06667   9.68750
```

Для обращения к выборкам за интересующие нас даты мы воспользовались индексацией с помощью логического выражения. Критерий Стьюдента в модификации Уэлча значимых отличий не выявил ($p > 0.05$).

Однако базовая версия критерия Стьюдента полагает наблюдения в выборках независимыми. В нашем случае такой независимости нет: пробы отбирались на одних и тех же станциях, поэтому выборки можно считать связанными. Для учета зависимости между данными и применяется модификация критерия Стьюдента для связанных выборок. Чтобы провести такой анализ, в функции `t.test()` необходимо воспользоваться аргументом `paired` и установить его в значение `TRUE`:

```
> t.test(x = bent$S[bent$date == "Jul12"],
+        y = bent$S[bent$date == "Jul13"], paired = TRUE)

 Paired t-test

data:  bent$S[bent$date == "Jul12"] and bent$S[bent$date == "Jul13"]
t = 2.2268, df = 14, p-value = 0.04288
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.1572245 8.3761088
sample estimates:
mean of the differences
               4.266667
```

Обратите внимание, что при выводе результатов в консоль поменялось название критерия (`Paired t-test`), существенно изменилось число степеней свободы (`df = 14`, для связанных выборок оно равно числу пар за вычетом единицы, в нашем случае число пар равно 15, потому что отсутствуют проба с 13 станции за июль 2012 года, соответственно эта станция исключена из анализа), выборочная оценка приведена только для разности средних (`mean of the differences`). Вариант критерия для связанных выборок выявил значимые отличия среднего видового богатства ($p < 0.05$, а доверительный интервал для разности средних не покрывает ноль).

Абсолютно идентичные результаты можно получить, если применить критерий Стьюдента для сравнения с нулем разности значений видового богатства в каждой паре:

```
> t.test(bent$S[bent$date == "Jul12"] -
+         bent$S[bent$date == "Jul13"], mu = 0)

 One Sample t-test

data:  bent$S[bent$date == "Jul12"] - bent$S[bent$date == "Jul13"]
t = 2.2268, df = 14, p-value = 0.04288
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 0.1572245 8.3761088

sample estimates:
mean of x
 4.266667
```

При анализе связанных выборок необходимо быть уверенным, что в передаваемых функции `t.test()` векторах наблюдения, соответствующие одним и тем же объектам, следуют в одинаковом порядке. В нашем случае это так, поскольку во фрейме `bent` данные по сезонам организованы в блоки, внутри которых сохраняется порядок следования станций.

4.4. Непараметрические аналоги: критерии Уилкоксона и Манна-Уитни

Критерий Стьюдента в различных его формах достаточно устойчив к отклонениям от нормальности, однако для полной уверенности в результатах анализа лучше отказаться от его использования в случае сомнений относительно характера распределения данных. Существуют непараметрические аналоги этого критерия, которые не требуют определенного типа распределения исходных данных. Обычно эти методы основаны на замене исходных данных их рангами в объединенной совокупности.

Непараметрическим аналогом одновыборочного критерия Стьюдента является критерий Уилкоксона, оценивающий положение выборки по сравнению с эталоном μ_0. При его применении эталонное значение μ_0 вычитается из каждого элемента выборки, полученные разности ранжируются без учета знака (то есть перед ранжированием берутся абсолютные значения), а затем суммируются ранги отрицательных или положительных значений. Если бы распределение данных было симметрично относительно μ_0, то критериальная статистика представляла бы собой сумму значений выборки из набора чисел от 1 до n, каждое из которых может быть выбрано с вероятностью 0.5.

Распределение этой статистики может быть рассчитано точно, по крайней мере в принципе. При больших объемах выборки расчет этого распределения становится вычислительно сложен, но оно может быть аппроксимировано нормальным распределением.

Критерий Уилкоксона реализован в функции `wilcox.test()`, ее применение аналогично функции `t.test()`:

```
> wilcox.test(mel$tergit.length, mu = 2.45)

 Wilcoxon signed rank test with continuity correction

data:  mel$tergit.length
V = 4407.5, p-value = 0.001405
alternative hypothesis: true location is not equal to 2.45
```

Здесь мы повторили анализ отклонений длины тергита в нашей выборке пчел породы *A. m. mellifera* от породного стандарта, который составляет 2.45 мм. Структура вывода аналогична: название теста, описание данных, результаты, формулировка альтернативной гипотезы. Отсутствуют точечная и интервальная оценки параметров, поскольку применен непараметрический метод. Результат анализа не изменился: отклонение от породного стандарта статистически значимо.

Анализ, проведенный в гл. 4.3, показал, что длина тергита подчиняется нормальному распределению, поэтому у нас не было острой необходимости применять для сравнения с породным стандартом именно непараметрический критерий. В таких случаях лучше применять параметрические критерии, поскольку они более эффективны в смысле выявления статистически значимых отличий.

Обратимся к анализу длины хоботка, средняя длина которого у пчел породы *A. m. mellifera* согласно литературным данным составляет 6.35 мм.

```
> wilcox.test(mel$proboscis, mu = 6.35)

 Wilcoxon signed rank test with continuity correction

data:  mel$proboscis
V = 3087, p-value = 0.5911
alternative hypothesis: true location is not equal to 6.35
```

На сей раз отклонений от породного стандарта не обнаружено: сумма рангов V существенно меньше (по сравнению с анализом длины тергита), а p-значение сильно превышает уровень значимости ($p > 0.05$).

Критерий Уилкоксона чувствителен к проблеме совпадающих значений в выборке. В таких случаях при расчете рангов эти значения получают одинаковый

средний ранг, в результате чего расчет точного распределения критериальной статистики становится чрезвычайно сложным и он не реализован в функции `wilcox.test()`. Для использования точного распределения нужно воспользоваться аргументом `exact` и передать ему значение `TRUE`. По умолчанию используется аппроксимация нормальным распределением, при этом применяется так называемая поправка на непрерывность, отменить которую можно с использованием аргумента `correct = FALSE`.

Непараметрическим аналогом критерия Стьюдента для сравнения двух выборочных средних является критерий Манна-Уитни. Этот вариант критерия в литературе также часто называется критерием Уилкоксона для двух выборок, поэтому в R он реализован в функции `wilcox.test()`.

При его применении значения признака заменяются на ранги в объединенной выборке, а критериальная статистика представляет собой сумму рангов для одной из выборок (обычно – первой). Проблема сводится к выборке объемом n_1 из совокупности чисел от 1 до $n_1 + n_2$. Распределение суммы этих чисел может быть вычислено точно, а в случае больших объемов может быть аппроксимировано нормальным распределением. Строго говоря, данный критерий сравнивает не средние значения (поскольку они не вычисляются), а оценивает положение выборок друг относительно друга, однако смысловое содержание анализа аналогично сравнению средних.

Вызов функции `wilcox.test()` аналогичен вызову функции `t.test()`. В качестве примера проанализируем отличие средней длины хоботка в первых двух семьях пчел породы *A. m. carnica*:

```
> wilcox.test(proboscis ~ family, data = apis[1:60, ])

 Wilcoxon rank sum test

data:  proboscis by family
W = 232, p-value = 0.001032
alternative hypothesis: true location shift is not equal to 0
```

Длина хоботка в первых двух семьях статистически значимо отличается ($p > 0.05$), также как и длина тергита (см. разд. 13.2).

В заключение рассмотрим вариант критерия Уилкоксона для связанных выборок. По сути он представляет собой одновыборочный критерий Уилкоксона для сравнения с нулем, который применяется к разностям значений в парах. Для его применения нужно воспользоваться аргументом `paired` функции `wilcox.test()`, которому нужно передать значение `TRUE`.

В качестве примера обратимся к анализу межгодовых отличий в средней биомассе в выборках зообентоса (см. гл. 4.3). Биомасса бентоса в выборках за

разные сезоны отклоняется от нормального распределения, поэтому для сравнения двух выборок за июль 2012 и 2013 года мы воспользуемся критерием Уилкоксона для связанных выборок:

```
> wilcox.test(x = bent$biomass[bent$date == "Jul12"],
+ y = bent$biomass[bent$date == "Jul13"], paired = TRUE)

 Wilcoxon signed rank test

data:  bent$biomass[bent$date == "Jul12"] and
bent$biomass[bent$date == "Jul13"]
V = 82, p-value = 0.2293
alternative hypothesis: true location shift is not equal to 0
```

Критерий Уилкоксона отличий не обнаруживает ($p > 0.05$).

Таким образом, функции `t.test()` и `wilcox.test()` в значительной степени аналогичны, что упрощает их применение.

Глава 5. Сравнение нескольких групп: однофакторный дисперсионный анализ и критерий Крускала-Уоллиса

Критерии Стьюдента и Манна-Уитни могут использоваться для сравнения средних в двух группах. В практике анализа данных очень часто возникают ситуации, когда исследователь одновременно имеет дело с множеством выборок. Прежде чем производить попарные сравнения выборок между собой необходимо провести совместный анализ всей совокупности данных. Для этого предназначен однофакторный дисперсионный анализ и его непараметрический аналог – критерий Крускала-Уоллиса.

5.1. Основы дисперсионного анализа

Начнем рассмотрение дисперсионного анализа с введения системы обозначений, которая традиционно используется для его описания. Пусть x_{ij} обозначает наблюдение j в группе i, то есть например x_{53} – третье наблюдение в пятой группе. $\bar{x}_i$ – среднее в группе i, $\bar{x}_\bullet$ – общее среднее (среднее по всем наблюдениям во всех группах).

Каждое наблюдение может быть разложено на компоненты:

$$x_{ij} = \bar{x}_\bullet + (\bar{x}_i - \bar{x}_\bullet) + (x_{ij} - \bar{x}_i),$$

представляющие собой общее среднее, отклонение группового среднего от общего и отклонение данного наблюдения от группового среднего.

Такое разложение соответствует следующей теоретико-вероятностной модели:

$$X_{ij} = \mu + \alpha_i + \varepsilon_{ij},\ \varepsilon_{ij} \sim N(0, \sigma^2).$$

Здесь случайная величина X_{ij} моделируется как сумма общего среднего μ, отклонения группового среднего α_i и отклонения ε_{ij} от группового среднего, которое интерпретируется как случайная ошибка, подчиняющаяся нормальному распределению с нулевым средним и дисперсией σ^2. Нуль-гипотеза о том, что средние в группах не отличаются, эквивалентно утверждению:

$$H_0: \alpha_1 = \alpha_2 = \ldots = \alpha_k,$$

где k – общее число групп. Обратите также внимание, что случайные ошибки ε_{ij} должны быть независимыми и иметь одинаковую дисперсию во всех группах.

Рассмотрим теперь суммы квадратов отклонений от общего и групповых средних. Величина

$$SS_w = \sum_i \sum_j (x_{ij} - \bar{x}_i)^2$$

называется внутригрупповой суммой квадратов (**w**ithin-group **s**um of **s**quares). Величина

$$SS_b = \sum_i \sum_j (\bar{x}_i - \bar{x}_\bullet)^2 = \sum_i n_i (\bar{x}_i - \bar{x}_\bullet)^2$$

называется межгрупповой суммой квадратов (**b**etween-group **s**um of **s**quares). Можно показать, что

$$SS_w + SS_b = SS_t = \sum_i \sum_j (x_{ij} - \bar{x}_\bullet)^2 ,$$

где SS_t – общая сумма квадратов (**t**otal **s**um of **s**quares). Таким образом, общая изменчивость раскладывается на две компоненты: изменчивость групповых средних и случайную изменчивость внутри групп. В таких случаях говорят, что группировка объясняет часть общей изменчивости.

Очевидно, что группировка, определяемая важным влияющим фактором, должна объяснять значительную часть общей изменчивости. Однако даже случайная группировка данных будет «объяснять» какую-то часть общей изменчивости просто в силу случайных отклонений и того факта, что суммы квадратов всегда положительны. Вопрос заключается в том, насколько мала должна быть доля объясненной изменчивости, чтобы считать ее случайной.

Рассмотренные суммы квадратов аналогичны сумме квадратов отклонений от среднего, которая рассчитывается при вычислении дисперсии. Аналогом же дисперсии в нашем случае являются так называемые средние квадраты (**m**ean **s**quares):

$$MS_w = \frac{SS_w}{N - k},$$

$$MS_b = \frac{SS_b}{k - 1},$$

где N – общее число наблюдений во всех группах.

Внутригрупповой средний квадрат MS_w рассчитывается путем суммирования отклонений в каждой группе и их усреднения, поэтому он является оценкой σ^2. В отсутствие эффекта группировки (то есть в условиях нуль-гипотезы) межгрупповой средний квадрат MS_b также будет являться оценкой σ^2. Если же средние в группах систематически отличаются (то есть группировка информативна), то межгрупповой средний квадрат MS_b будет больше. Таким образом, для выявления отличий между средними можно использовать сравнение двух оценок дисперсии σ^2. Именно поэтому рассматриваемый метод называется дисперсионным анализом, хотя его целью является сравнение средних.

Критериальной статистикой в дисперсионном анализе служит отношение

$$F = \frac{MS_b}{MS_w},$$

которое в условиях нуль-гипотезы должно быть близко к единице. Вследствие случайной изменчивости определенный разброс возможен, он описывается F-распределением Фишера с $k-1$ и $N-k$ степенями свободы. В условиях альтернативной гипотезы, которая формулируется как неравенство хотя бы одной пары групповых средних, отношение F будет больше единицы, поэтому нуль-гипотеза отвергается (на 5 % уровне значимости), если оно превышает квантиль уровня 0.95 соответствующего F-распределения. Обратите внимание, что этот критерий односторонний, поскольку отклонение в меньшую сторону относительно единицы означает, что разница между эмпирическими групповыми средними очень мала, что не соответствует альтернативной гипотезе.

5.2. Реализация дисперсионного анализа в R

Дисперсионный анализ в R реализован в функции `aov()`. Проведем дисперсионный анализ длины тергита во всех восьми семьях пчел породы *A. m. carnica*. Напомним, что по этой породе имеются 8 выборок по 30 пчел из каждой семьи. Нас будет интересовать наличие межсемейной изменчивости.

```
> load("apis.rda")
> carn <- subset(apis, subset = breed == "carnica")
> model <- aov(carn$tergit.length ~ carn$family)
```

Мы загрузили данные по пчелам, выбрали данные по породе *A. m. carnica* в отдельный фрейм `carn` и использовали функцию `aov()` для создания модели дисперсионного анализа. Функции передается объект класса формула, которая описывает необходимую нам схему анализа. В данном случае формула предельно простая. Она определяет, что мы хотим проанализировать длину тергита (`tergit.length`) в зависимости от (~) семейной принадлежности (`family`). Справа от тильды стоит имя анализируемой переменной, это должна быть числовая переменная. Слева от тильды стоит имя группирующей переменной, которая задает распределение наблюдений по группам. Группирующая переменная должна быть фактором (в крайнем случае – текстовой переменной, которую функция самостоятельно преобразует в фактор). Если группирующая переменная будет иметь числовой формат (такое может случиться, если в качестве идентификаторов групп использовать их порядковые номера), то будет

построена линейная модель для регрессионного анализа, а не для дисперсионного анализа.

В приведённой выше формуле использованы имена переменных вместе с названием фрейма, в котором они содержатся (фактически функции переданы вектора). Такая нотация делает вызов функций громоздким, а сами формулы – плохо читаемыми. Удобнее использовать только названия переменных, а фрейм, в котором необходимо искать имена переменных, фигурирующих в формуле, указывать в аргументе `data`. Например, следующий вызов функции `aov()` эквивалентен приведенному выше, но лаконичнее и удобнее на практике:

```
> model <- aov(tergit.length ~ family, data = carn)
```

Функция `aov()` возвращает специальный объект класса «aov», который представляет собой сложно структурированный список из 13 элементов. Для извлечения результатов дисперсионного анализа из линейной модели нужно воспользоваться функцией `anova()`:

```
> anova(model)
Analysis of Variance Table

Response: tergit.length
           Df  Sum Sq  Mean Sq F value    Pr(>F)
family      7 0.22428 0.032039   7.742 2.056e-08 ***
Residuals 232 0.96011 0.004138
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

В консоль выведена так называемая таблица дисперсионного анализа, о чем свидетельствует заголовок в первой строке вывода. В следующей строке отображено имя переменной отклика (`Response`) – той переменной, которая анализируется. Затем следует собственно таблица анализа, в данном случае состоящая из двух строк, которые соответствуют двум источникам изменчивости. В первой строке описана межгрупповая изменчивость. Поскольку группировка задается факторной переменной `family`, именно ее имя фигурирует в таблице. Вторая строка описывает внутригрупповую изменчивость, которая иногда называется остаточной изменчивостью. Остатками (`Residuals`) в случае дисперсионного анализа называются отклонения от групповых средних. В столбцах представлены степени свободы (`Df`, degrees of freedom), суммы квадратов (`Sum Sq`), средние квадраты (`Mean Sq`), отношение средних квадратов (`F value`) и вероятность получить имеющееся или более экстремальное отношение средних квадратов в условиях нуль-гипотезы (`Pr(>F)`).

В данном случае мы получили очень сильно отличающееся от единицы отношение F (7.742), которое очень маловероятно в условиях нуль-гипотезы

($p = 2.056 \cdot 10^{-8}$), поэтому нуль-гипотезу можно отвергнуть и принять альтернативную. В таблице дисперсионного анализа традиционно используется система обозначения статистически значимых эффектов с использованием звездочек и точек. Три звездочки в конце первой строки означают выявление эффекта семейной принадлежности на 0.1 % - ном уровне значимости ($p < 0.001$). Расшифровка обозначений приведена в последней строке вывода.

5.3. Предположения дисперсионного анализа

Дисперсионный анализ – это параметрический метод, в основе которого лежат следующие предположения:

1) независимость данных в группах;
2) нормальность распределения остатков;
3) гомогенность дисперсии.

Независимость данных в группах означает, что наблюдения не должны быть каким-либо образом связаны между собой. Это требование касается не только межгрупповой независимости, но и внутригрупповой. В нашем примере с анализом межсемейной изменчивости пчел выделяемые группы соответствуют семьям и никакой связи между отдельными пчелами в разных семьях нет. То же самое относится и к связям отдельных пчел внутри семьи – пчелы отбирались случайным образом.

Одной из целей исследования пчел являлось выявление межпородных отличий. С этой целью можно сразу провести дисперсионный анализ каждого из признаков в трех группах, соответствующих породам. В этом случае независимость наблюдений между группами очевидна. Но вот насколько можно считать независимыми отдельных пчел внутри групп? Данные, представляющие породы *A. m. carnica* и *A. m. caucasica*, представлены восемью семьями. Полной независимости данных внутри этих групп нет. Пчелы из отдельных семей могут оказаться ближе друг к другу по морфологии, чем к другим пчелам. Таким образом, прямой дисперсионный анализ межпородной морфологической изменчивости пчел без учета межсемейной изменчивости может оказаться некорректным. Именно поэтому в нашем примере мы начали с анализа межсемейной изменчивости пчел внутри конкретной породы. Если бы этот анализ не выявил межсемейных отличий, у нас были бы основания к объединению всех пчел породы *A. m. carnica* в единую выборку и проведению анализа, который описан в настоящем абзаце, однако отличия выявлены.

Теоретико-вероятностная модель дисперсионного анализа подразумевает, что отклонения наблюдений от средних (остатки) подчиняются нормальному распределению. Обратите внимание, что принципиально важно нормальное распределение именно остатков, а не самих данных. В случае однофакторного дисперсионного анализа, который мы рассматриваем, это требование эквивалентно требованию нормального распределения внутри групп, однако в иных формах линейных моделей (в частности – в регрессионном анализе, который будет рассмотрен в следующей главе), такая эквивалентность не обязательна и данные могут быть распределены как угодно, лишь бы сохранялась нормальность распределения остатков в модели.

При анализе характера распределения данных необходимо обратить внимание на следующий аспект: нельзя анализировать все исходные данные одновременно. В этом случае мы рискуем получить смесь нормальных распределений с отличающимися средними, что эквивалентно мультимодальному распределению (с несколькими пиками), которое будет диагностироваться всеми критериями согласия. Из этой ситуации есть два выхода. Можно анализировать характер распределения для каждой группы в отдельности, что мы и проделали в гл. 3, убедившись, что длина тергита внутри семей подчиняется нормальному распределению. Однако лучше анализировать распределение остатков, то есть отклонений от групповых средних. В этом случае все группы анализируются одновременно, что ведет к увеличению мощности анализа.

Вектор остатков содержится в модели дисперсионного анализа, которую возвращает функция `aov()`. Получить к нему доступ можно либо напрямую по имени элемента, либо воспользовавшись специальной функцией `resid()`, которая извлекает остатки из объектов статистических моделей разного типа.

```
> res <- model$residuals
> res <- resid(model)
```

Вектор res далее анализируется на предмет соответствия нормальному распределению с использованием критериев согласия и соответствующих функций (см. гл. 3):

```
> library(nortest)
> lillie.test(res)

Lilliefors (Kolmogorov-Smirnov) normality test

data:  res
D = 0.034591, p-value = 0.6906
```

Последнее требование – гомогенность дисперсии. Теоретико-вероятностная модель дисперсионного анализа подразумевает, что распределение остатков в группах полностью идентично, включая дисперсию этого распределения. Требование гомогенности дисперсии (иногда это свойство также называют гомоскедастичностью) очень важно. В случае большого разброса дисперсий в группах (особенно в случае неравного объема групп), распределение критериальной статистики может сильно измениться, что приведет к неадекватным результатам анализа. Таким образом, перед проведением параметрического дисперсионного анализа необходимого убедиться в гомогенности дисперсий.

Критерии Бартлетта, Левена и Флигнера-Киллина, используемые для анализа гомогенности дисперсий, в определенном смысле аналогичны однофакторному дисперсионному анализу: они проверяют нуль-гипотезу о равенстве дисперсий одновременно во всех группах.

Критерий Бартлетта реализован в функции `bartlett.test()`, ее вызов осуществляется с помощью формулы:

```
> bartlett.test(tergit.length ~ family, data = carn)

 Bartlett test of homogeneity of variances

data:  tergit.length by family
Bartlett's K-squared = 8.5853, df = 7, p-value = 0.2838
```

Поскольку $p > 0.05$, критерий Бартлетта не выявляет значимых отличий дисперсии длины тергита в 8 семьях пчел породы *A. m. carnica*. Критерий Бартлетта является параметрическим и он сильно чувствителен к отклонениям от нормального распределения. Меньшей чувствительностью к отклонениям обладает критерий Левена. Он отсутствует в стандартных пакетах среды R. Для его использования необходимо установить и загрузить пакет `car` (от англ. **c**ompanion to **a**pplied **r**egression – средства для прикладной регрессии), где критерий Левена реализован в функции `leveneTest()`:

```
> library(car)
> leveneTest(tergit.length ~ family, data = carn)
Levene's Test for Homogeneity of Variance (center = median)
       Df F value Pr(>F)
group   7  1.2721  0.265
      232
Warning message:
In leveneTest.default(y=y, group=group, ...): group coerced to factor.
```

Результаты выданы в виде небольшой таблички и они соответствуют результатам критерия Бартлета: дисперсии гомогенны. Обратите внимание на

предупреждение, выведенное в консоль после результатов. Оно информирует о том, что группирующая переменная `family` была преобразована в фактор.

Критерий Флигнера-Киллина является непараметрическим, он реализован в функции `fligner.test()`. Эта функция менее внимательна к пользователю и не умеет преобразовывать текстовые группирующие переменные в фактор, в результате чего могут происходить ошибки, как например в нашем случае:

```
> fligner.test(tergit.length ~ family, data = carn)
Error in fligner.test.default(c(2.4718,2.4082,2.3314,2.3108, 2.2631,:
  all group levels must be finite
```

Чтобы избежать этой ошибки можно в явном виде преобразовать нашу текстовую переменную `family` в фактор, это можно сделать прямо в конструкции формулы:

```
> fligner.test(tergit.length ~ factor(family), data = carn)

 Fligner-Killeen test of homogeneity of variances

data:  tergit.length by factor(family)
Fligner-Killeen:med chi-squared = 9.5569, df = 7, p-value = 0.2151
```

Итак, результаты всех трех критериев гомогенности дисперсий совпадают: гетерогенности дисперсии длины тергита не выявлено. Более того – очень близки и соответствующие *p*-значения.

Критерий Бартлетта является наиболее мощным из трех описанных критериев гомогенности дисперсий, то есть он чаще выявляет истинные отличия, но он чувствителен к отклонениям от нормальности, поэтому при прочих равных рекомендуется применять критерий Левена. Однако критерий Левена сильно уступает в мощности критерию Бартлетта в случае сравнения дисперсий в двух выборках. Критерий Флигнера-Киллина является непараметрическим, его рекомендуется использовать в случае необходимости анализа изменчивости в группах, сильно отклоняющихся от нормального распределения.

5.4. Дисперсионный анализ в группах с гетерогенной дисперсией

В случае выявления значимых отличий дисперсии в группах, применять стандартный параметрический дисперсионный анализ нельзя, поскольку он чувствителен к такого рода отклонениям. Если целью является анализ средних значений в нескольких группах, существует альтернативный метод, который является обобщением критерия Стьюдента в модификации Уэлча (эта модификация также не требует равенства дисперсий) для нескольких групп.

Этот вид анализа реализован в функции `oneway.test()` (однофакторный критерий):

```
> oneway.test(tergit.length ~ family, data = carn)

 One-way analysis of means (not assuming equal variances)

data:  tergit.length and family
F = 8.4605, num df = 7.00, denom df = 99.31, p-value = 4.332e-08
```

Итоговый вывод относительно изменчивости средних не изменился: длина тергита у пчел породы *A. m. carnica* в разных семьях значимо отличается.

Несмотря на то, что этот вид анализа не требует гомогенности дисперсий, он все же является параметрическим методов и основывается на предположении о нормальном распределении данных во всех группах. Если же имеются значимые отклонения от нормальности, следует использовать непараметрические методы.

5.5. Непараметрический аналог: критерий Крускала-Уоллиса

Непараметрическим аналогом однофакторного дисперсионного анализа для одновременного сравнения средних в нескольких группах является критерий Крускала-Уоллиса. Также как и при использовании критерия Уилкоксона для сравнения двух групп, данные заменяются на их ранги в объединенной совокупности. Затем производится расчет внутригрупповой суммы квадратов на основе рангов и ранговых средних в группах. Распределение критериальной статистики для небольшого числа групп небольшого объема можно рассчитать точно, либо можно использовать специально разработанные аппроксимации.

Критерий Крускала-Уоллиса реализован в функции `kruskal.test()`. Используем ее для анализа межсемейной изменчивости длины хоботка пчел породы *A. m. carnica*. Напомним, что в гл. 3 выявлены значимые отклонения распределения этого признака от нормальности, поэтому применять параметрический дисперсионный анализ в этом случае нельзя.

```
> kruskal.test(proboscis ~ factor(family), data = carn)

 Kruskal-Wallis rank sum test

data:  proboscis by factor(family)
Kruskal-Wallis chi-squared = 27.8714, df = 7, p-value = 0.000232
```

При вызове функции используется формула. Обратите внимание, что мы использовали явное преобразование переменной `family` в фактор. Результаты приведены в последней строке вывода. Поскольку $p < 0.05$, можно констатировать выявление межсемейных отличий.

Глава 6. Множественные сравнения

После проведения однофакторного дисперсионного анализа в той или иной форме (включая непараметрический критерий Крускала-Уоллиса), неизбежно возникает вопрос о том, какие именно группы отличаются между собой. Разумеется, этот вопрос возникает только в случае выявления отличий, поскольку отвергнутая нуль-гипотеза в дисперсионном анализе означает, что среднее хотя бы одной группы отличается от всех остальных, но не указывает какая именно (и вполне возможно, что все группы характеризуются уникальными средними). Для выявления отличий между конкретными группами используются так называемые попарные сравнения, для которых обычно используется критерий Стьюдента. При этом возникает проблема множественных сравнений.

Дело в том, что в конструкции любого критерия заложена возможность совершить ошибку – отвергнуть нуль-гипотезу, когда она верна, то есть обнаружить несуществующие отличия. Такие ошибки называются ошибками первого рода и их минимизация является приоритетом при разработке методов статистического анализа. Вероятностью совершения ошибки первого рода при отбрасывании нуль-гипотезы является *p*-значение при каждом применении любого статистического критерия. Сама же возможность ошибки является неустранимым следствием случайной природы данных.

При использовании попарных сравнений исследователь применяет один и тот же критерий к одним и тем же данным множество раз. Предположим, мы используем 5 % - ный уровень значимости, то есть мы считаем допустимой вероятность ошибки первого рода при каждом сравнении в 0.05. Однако, повторяя процедуру анализа, мы рискуем совершить ошибку первого рода в каждом из сравнений, поэтому общая вероятность ошибки возрастает:

$$P = 1 - (1 - \alpha)^m,$$

где α – уровень значимости, m – число попарных сравнений. Так, в нашем примере с анализом межсемейной изменчивости пчел породы *A. m. carnica* можно провести $m = 8 \cdot (8 - 1)/2 = 28$ попарных сравнений. Тогда вероятность совершения ошибки первого рода хотя бы в одном случае составит

```
> 1 - (1 - .05) ^ 28
[1] 0.7621731
```

Для контроля ошибки первого рода при проведении множественных сравнений предназначены специальные методы: поправки к уровню значимости либо специальные критерии.

6.1. Поправки Бонферрони и Холма

Наиболее простым и известным методом контроля ошибки первого рода при проведении множественных сравнений является поправка Бонферрони. Этот метод исходит из так называемого неравенства Бонферрони (в общем виде оно утверждает, что вероятность объединения событий не больше суммы их вероятностей), которое при небольшом числе сравнений m можно считать приблизительным равенством:

$$P = 1 - (1 - \alpha)^m < \alpha \cdot m .$$

Тогда при использовании критерия в каждом из сравнений нужно руководствоваться не исходным уровнем значимости α, а уровнем α/m. Либо, что то же самое, перед принятием решения относительно нуль-гипотезы полученное p-значение нужно помножить на m. Например, если при проведении 3 попарных сравнений были получены p-значения 0.015, 0.02 и 0.04, то на 5 % - ном уровне значимости нуль-гипотезу следует отклонить только в первом случае ($0.015 \cdot 3 = 0.045 < 0.05$).

Чтобы избавить пользователя от необходимости применять функцию `t.test()` к каждой паре данных (например, с использованием цикла), в R реализована специальная функция для множественных сравнений `pairwise.t.test()`. Используем ее для проведения попарных сравнений длины тергита в 8 семьях пчел породы *A. m. carnica*:

```
> load("apis.rda")
> carn <- subset(apis, subset = breed == "carnica")
> pairwise.t.test(carn$tergit.length, carn$family,
+                    p.adjust.method = "bonferroni", pool.sd = FALSE)

Pairwise comparisons using t tests with non-pooled SD

data:  carn$tergit.length and carn$family

        fam.110a fam.24a fam.31  fam.41  fam.62a fam.73a fam.87
fam.24a 0.05525  -       -       -       -       -       -
fam.31  0.19823  1.00000 -       -       -       -       -
fam.41  0.03384  1.00000 1.00000 -       -       -       -
fam.62a 1.00000  0.12282 0.40108 0.10331 -       -       -
fam.73a 1.00000  0.00026 0.00065 6e-05   1.00000 -       -
fam.87  0.05972  1.00000 1.00000 1.00000 0.16081 0.00012 -
fam.9к  1.00000  0.01490 0.05211 0.00761 1.00000 1.00000 0.01397

P value adjustment method: bonferroni
```

Функция pairwise.t.test() не работает с формулами, поэтому при ее вызове первым аргументом передается вектор данных, а вторым – вектор группирующей переменной. Аргумент p.adjust.method = "bonferroni" указывает, поправку какого типа необходимо использовать (в данном случае – поправку Бонферрони), а аргумент pool.sd определяет, нужно ли использовать при сравнениях общую оценку дисперсии из дисперсионного анализа. Первая строка вывода сообщает, что проведены попарные сравнения с использованием критерия Стьюдента без объединенной оценки дисперсии, вторая строка описывает проанализированные данные. Затем следует таблица, в которой представлены *p*-значения с поправкой Бонферрони, строки и столбцы таблицы соответствуют выборкам (семьям пчел). При умножении *p*-значений на число сравнений могут быть получены значения, превышающие единицу, не соответствующие определению вероятности, поэтому в таблице такие значения заменены на единицы. Тип поправки указан в последней строке вывода. Для подсчета числа выявленных значимых попарных отличий можно воспользоваться следующей конструкцией:

```
> sum(pairwise.t.test(carn$tergit.length, carn$family,
+         p.adjust.method = "bonferroni",
+         pool.sd = FALSE)$p.value < .05, na.rm = TRUE)
[1] 8
```

Функция pairwise.t.test() возвращает объект класса «pairwise.htest», который представляет собой список, в элементе которого с именем p.value и содержится таблица *p*-значений. Поэтому к таблице результатов можно обратиться с помощью оператора $ (обращение к элементу списка по имени). Результатом сравнения этой таблицы с уровнем значимости будет таблица логических значений, содержащая значения TRUE в позициях, соответствующих отклонению нуль-гипотезы. При суммировании такой таблицы значения TRUE будут преобразованы в единицы, поэтому результат покажет нам, сколько раз была отклонена нуль-гипотеза. При суммировании нужно использовать аргумент na.rm = TRUE, поскольку в исходной таблице содержались отсутствующие значения NA (это сделано для предотвращения дублирования).

Итак, в результате множественных сравнений с использованием критерия Стьюдента и поправки Бонферрони обнаружено 8 отличий.

Поправка Бонферрони была одним из первых предложенных средств контроля ошибки первого рода при множественных сравнениях. Несмотря на простоту в реализации, это очень консервативный метод. Вследствие ужесточения уровня значимости, отличия между средними должны быть очень

велики, чтобы быть обнаруженными. Поэтому поправку Бонферрони не рекомендуется применять, когда число сравнений превышает 7-8.

Для смягчения излишней консервативности поправки Бонферрони разработаны специальные методы, наиболее распространенным из которых является поправка Холма (иногда также называемая поправкой Бонферрони-Холма). Этот метод основан на следующей процедуре. Все полученные в результате множественных сравнений *p*-значения упорядочиваются от меньшего к большему. К первому *p*-значению из этого упорядоченного ряда применяется поправка Бонферрони, то есть оно умножается на число сравнений *m* (либо сравнивается с α/m). Второе же значение умножается не на *m*, а на $m - 1$, третье – на $m - 2$ и так далее.

Для использования поправки Холма аргументу `p.adjust.method` функции `pairwise.t.test()` нужно передать значение «`holm`» (это значение по умолчанию, поэтому в принципе этот аргумент можно опустить):

```
> pairwise.t.test(carn$tergit.length, carn$family,
+                 p.adjust.method = "holm", pool.sd = FALSE)

 Pairwise comparisons using t tests with non-pooled SD

data:  carn$tergit.length and carn$family

        fam.110a fam.24a fam.31  fam.41  fam.62a fam.73a fam.87
fam.24a 0.03749  -       -       -       -       -       -
fam.31  0.09911  1.00000 -       -       -       -       -
fam.41  0.02538  1.00000 1.00000 -       -       -       -
fam.62a 1.00000  0.07018 0.18621 0.06272 -       -       -
fam.73a 1.00000  0.00025 0.00058 6e-05   1.00000 -       -
fam.87  0.03839  1.00000 1.00000 1.00000 0.08615 0.00012 -
fam.9к  1.00000  0.01171 0.03722 0.00653 1.00000 1.00000 0.01148

P value adjustment method: holm
```

При внимательном рассмотрении новой таблицы можно видеть, что минимальное *p*-значение не изменилось (сравнение семей `fam.41` и `fam.73a`, пересечение пятой строки и четвертого столбца), а остальные *p*-значения, не равные 1, поменялись в меньшую сторону, что привело к обнаружению значимых отличий для нескольких пар семей. В частности, выявлены отличия между семьей `fam.110` и семьями `fam.24a` и `fam.87`, которые при использовании поправки Бонферрони были незначимы. Общее число значимых попарных отличий достигло одиннадцати:

```
> sum(pairwise.t.test(carn$tergit.length, carn$family,
+                     p.adjust.method = "holm",
+                     pool.sd = FALSE)$p.value < .05, na.rm = T)
[1] 11
```

6.2. Критерий Тьюки

Если дисперсионный анализ выявил отличия средних в совокупности групп и необходимо провести сравнения между всеми возможными парами групп, можно воспользоваться критерием Тьюки. Критериальной статистикой при каждом попарном сравнении служит нормированная разность средних, аналогичная используемой в критерии Стьюдента:

$$q = \frac{\bar{x}_1 - \bar{x}_2}{s_{\bar{x}_1 - \bar{x}_2}}.$$

Отличие заключается в том, что ошибка разности средних $s_{\bar{x}_1 - \bar{x}_2}$ вычисляется не на основе выборочных дисперсий, а на основе общей оценки дисперсии во всех сравниваемых группах. Такой оценкой является межгрупповой средний квадрат MS_w из дисперсионного анализа. Статистика q в условиях нуль-гипотезы подчиняется так называемому распределению стьюдентизированных размахов (studentized range distribution), параметрами которого являются общее число степеней свободы $N - k$ (N – общий объем всех выборок) и число групп k.

Критерий Тьюки реализован в функции `TukeyHSD()` (Tukey Honest Significant Differences – критерий подлинно значимых отличий). Эта функция принимает объект класса «aov», который обычно является результатом ранее проведенного дисперсионного анализа.

```
> model <- aov(tergit.length ~ family, data = carn)
> (hsd <- TukeyHSD(model))
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = tergit.length ~ family, data = carn)

$family
                          diff           lwr           upr     p adj
fam.24a-fam.110a -0.055676667 -0.1064836959 -0.0048696374 0.0207269
fam.31-fam.110a  -0.043823333 -0.0946303626  0.0069836959 0.1477050
fam.41-fam.110a  -0.053136667 -0.1039436959 -0.0023296374 0.0332123
fam.62a-fam.110a  0.003876667 -0.0469303626  0.0546836959 0.9999981
fam.73a-fam.110a  0.023900000 -0.0269070293  0.0747070293 0.8381923
fam.87-fam.110a  -0.050073333 -0.1008803626  0.0007336959 0.0566048
fam.9к-fam.110a   0.007260000 -0.0435470293  0.0580670293 0.9998588
fam.31-fam.24a    0.011853333 -0.0389536959  0.0626603626 0.9965402
```

```
fam.41-fam.24a    0.002540000 -0.0482670293  0.0533470293 0.9999999
fam.62a-fam.24a   0.059553333  0.0087463041  0.1103603626 0.0096127
fam.73a-fam.24a   0.079576667  0.0287696374  0.1303836959 0.0000799
fam.87-fam.24a    0.005603333 -0.0452036959  0.0564103626 0.9999756
fam.9к-fam.24a    0.062936667  0.0121296374  0.1137436959 0.0046968
fam.41-fam.31    -0.009313333 -0.0601203626  0.0414936959 0.9992629
fam.62a-fam.31    0.047700000 -0.0031070293  0.0985070293 0.0832029
fam.73a-fam.31    0.067723333  0.0169163041  0.1185303626 0.0015925
fam.87-fam.31    -0.006250000 -0.0570570293  0.0445570293 0.9999487
fam.9к-fam.31     0.051083333  0.0002763041  0.1018903626 0.0476889
fam.62a-fam.41    0.057013333  0.0062063041  0.1078203626 0.0160069
fam.73a-fam.41    0.077036667  0.0262296374  0.1278436959 0.0001572
fam.87-fam.41     0.003063333 -0.0477436959  0.0538703626 0.9999996
fam.9к-fam.41     0.060396667  0.0095896374  0.1112036959 0.0080724
fam.73a-fam.62a   0.020023333 -0.0307836959  0.0708303626 0.9296999
fam.87-fam.62a   -0.053950000 -0.1047570293 -0.0031429707 0.0286393
fam.9к-fam.62a    0.003383333 -0.0474236959  0.0541903626 0.9999992
fam.87-fam.73a   -0.073973333 -0.1247803626 -0.0231663041 0.0003467
fam.9к-fam.73a   -0.016640000 -0.0674470293  0.0341670293 0.9738242
fam.9к-fam.87     0.057333333  0.0065263041  0.1081403626 0.0150312
```

Функция возвращает объект специального класса «TukeyHSD», основу которого составляет список из единственного элемента – матрицы размером $m \times 4$, название которой соответствует названию группирующей переменной в дисперсионном анализе. Мы сохранили этот объект с именем `hsd`. В консоль выводится название критерия, описание данных, а также таблица результатов. В строках таблицы содержатся результаты попарных сравнений: разница средних (`diff`), нижняя и верхняя границы доверительного интервала этой разницы (`lwr` и `upr`), а также соответствующее *p*-значение (`p adj`). Именно последний столбец обычно представляет интерес, поскольку на основании *p*-значений принимается либо отвергается нуль-гипотеза о равенстве средних:

```
> sum(hsd[[1]][, 4] < 0.05)
[1] 13
```

Итак, в результате проведенных множественных сравнений длины тергита в 8 семьях пчел породы *A. m. carnica* с использованием критерия Тьюки значимые отличия выявлены в 13 парах.

Объект класса «TukeyHSD» можно использовать для визуализации результатов, просто передав его функции `plot()`:

```
> par(mar = c(5, 8, 4, 2))
> plot(hsd, las = 1, cex.axis = 0.8)
```

Создается график (Рис. 17), на котором отображены доверительные интервалы для разности средних. Если эти доверительные интервалы не покрывают ноль (выделен на графике вертикальной штриховой линией), разница между средними статистически значима. Пары групп отображаются в виде подписей к вертикальной оси, для удобства отображения мы расположили эти

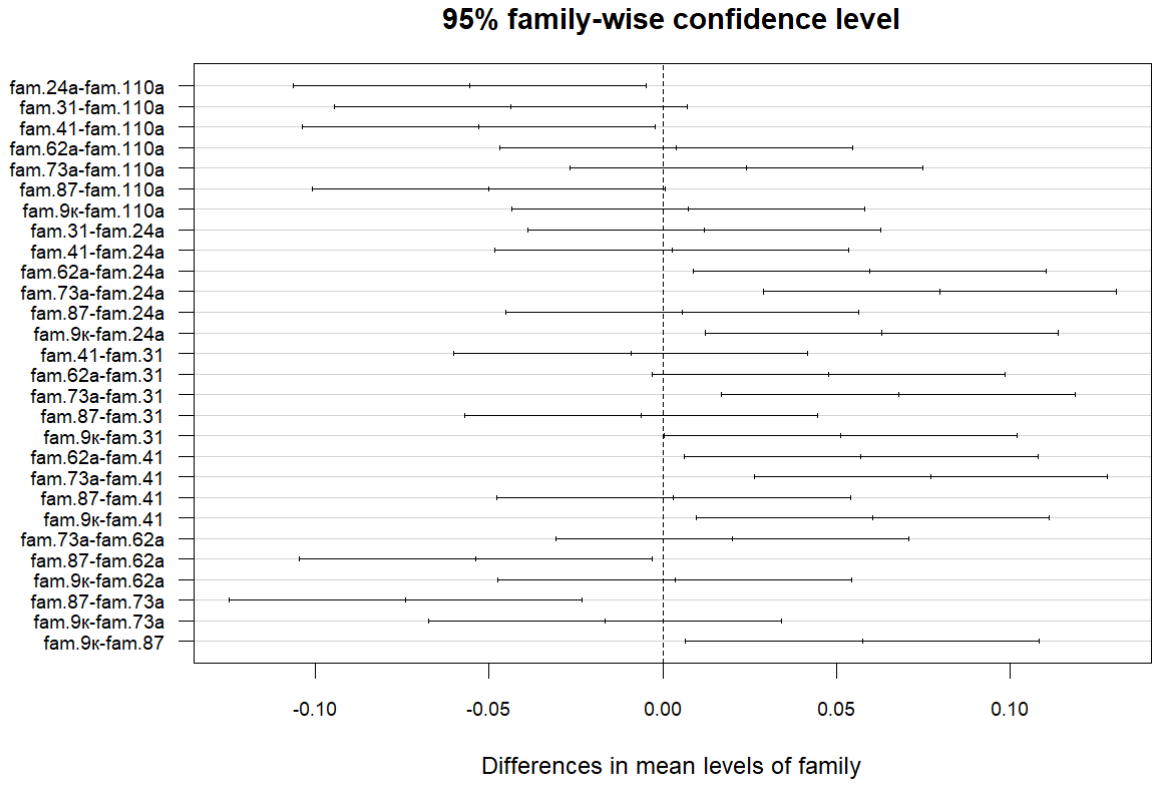

Рис. 17. График доверительных интервалов разностей

подписи перпендикулярно оси (`las = 1`), немного уменьшили размер шрифта для подписей оси (`cex.axis = 0.8`), а также перед созданием графика увеличили пространство слева от графика (`par(mar = c(5, 8, 4, 2))`). К сожалению, нет возможности изменить заголовок и подпись горизонтальной оси.

6.3. Критерий Ньюмана-Кейлса

Критерий Ньюмана-Кейлса (также иногда называемый критерием Стьюдента-Ньюмана-Кейлса) является версией критерия Тьюки, делающей его еще более либеральным. В этом критерии используется та же самая критериальная статистика q и распределение стьюдентизированных размахов, что и в критерии Тьюки.

Однако перед расчетом значений q и вычислением p-значений проводится процедура ранжирования групповых средних: они упорядочиваются от меньшего к большему. Попарные равнения проводятся в определённом порядке. Сначала сравниваются наибольшее среднее с наименьшим, то есть 1-е с k-м, затем k-е со 2, далее с 3-м и так далее до (k–1)-го. Затем предпоследнее (k–1)-е среднее сравнивается с 1-м, 2-м и так далее пока не будут перебраны все пары средних. При этом считается, что вывод, полученный при сравнении любой

пары, можно распространить на весь диапазон групповых средних, который заключен между ними. Так, если не обнаружено отличий между 3-м и 6-м средними в упорядоченном ряду, то можно не проводить сравнения 3-го с 5-м и 4-м, 6-го с 4-м и 5-м, а также 4-го с 5-м. Поэтому при расчете *p*-значения используется распределение стьюдентизированных размахов с числом степеней свободы $N - k$ и числом групп, которое задается так называемым интервалом сравнения – числом позиций между сравниваемыми средними в упорядоченном ряду. Напомним, что при использовании критерия Тьюки всегда используется число имеющихся групп. Таким образом, критерий Ньюмана-Кейлса и критерий Тьюки дадут одинаковые результаты только при сравнении наиболее контрастных групп, во всех остальных случаях критерий Ньюмана-Кейлса будет менее строгим.

Считается, что критерий Ньюмана-Кейлса излишне либерален и склонен выявлять ненадежные отличия, поэтому в стандартных пакетах среды R он отсутствует. Однако можно найти его реализации во множестве дополнительных пакетов, в частности в пакете `mutoss`, который перед применением необходимо установить и загрузить. Критерий Ньюмана-Кейлса реализован в функции `snk()` (от Student-Newman-Keuls), которая принимает формулу. Также обязательно использовать аргумент `alpha` (уровень значимости), который не имеет значения по умолчанию:

```
> (nk <- snk(tergit.length ~ family, data = carn, alpha = .05))
#---Student-Newman-Keuls (1927;1939;1952) rejective Multiple Test Procedure

#---Attention: The SNK test controls the FWER only in the WEAK sense

$Ordered.Means
    Sample Size    Means    Variance
1  fam.24a   30 2.354863 0.005221589
2   fam.41   30 2.357403 0.003706725
3   fam.87   30 2.360467 0.003675457
4   fam.31   30 2.366717 0.003788293
5 fam.110a   30 2.410540 0.003601543
6  fam.62a   30 2.414417 0.006862218
7   fam.9к   30 2.417800 0.003572506
8  fam.73a   30 2.434440 0.002678898

$Variances
      Overall  df
1 0.004138404 232
```

```
$SNK
         Comparison   Diff Statistic  Adj.P Rejected Layer
1   fam.73a-fam.24a 0.0796    6.7753  1e-04     TRUE     1
2    fam.9к-fam.24a 0.0629    5.3586 0.0036     TRUE     2
3    fam.73a-fam.41 0.0770    6.5591  1e-04     TRUE     2
4   fam.62a-fam.24a 0.0596    5.0705 0.0054     TRUE     3
5     fam.9к-fam.41 0.0604    5.1423 0.0045     TRUE     3
6    fam.73a-fam.87 0.0740    6.2982  2e-04     TRUE     3
7  fam.110a-fam.24a 0.0557    4.7404 0.0083     TRUE     4
8    fam.62a-fam.41 0.0570    4.8542 0.0063     TRUE     4
9     fam.9к-fam.87 0.0573    4.8815 0.0059     TRUE     4
10   fam.73a-fam.31 0.0677    5.7661  6e-04     TRUE     4
11   fam.31-fam.24a 0.0119    1.0092 0.8916    FALSE     5
12  fam.110a-fam.41 0.0531    4.5242 0.0085     TRUE     5
13   fam.62a-fam.87 0.0540    4.5934 0.0073     TRUE     5
14    fam.9к-fam.31 0.0511    4.3493 0.0125     TRUE     5
15 fam.73a-fam.110a 0.0239    2.0349 0.4764    FALSE     5
16   fam.87-fam.24a 0.0056    0.4771 0.9392    FALSE     6
17    fam.31-fam.41 0.0093    0.7930  0.841    FALSE     6
18  fam.110a-fam.87 0.0501    4.2633  0.008     TRUE     6
19   fam.62a-fam.31 0.0477    4.0613 0.0124     TRUE     6
20  fam.9к-fam.110a 0.0073    0.6181 0.9001    FALSE     6
21  fam.73a-fam.62a 0.0200    1.7048 0.4511    FALSE     6
22   fam.41-fam.24a 0.0025    0.2163 0.8786    FALSE     7
23    fam.87-fam.41 0.0031    0.2608 0.8539    FALSE     7
24    fam.31-fam.87 0.0063    0.5321 0.7071    FALSE     7
25  fam.110a-fam.31 0.0438    3.7312 0.0089     TRUE     7
26 fam.62a-fam.110a 0.0039    0.3301 0.8156    FALSE     7
27   fam.9к-fam.62a 0.0034    0.2881 0.8388    FALSE     7
28   fam.73a-fam.9к 0.0166    1.4168 0.3175    FALSE     7
```

При использовании этой функции всегда выдается предупреждение о том (см. строку вывода `Attention`), что критерий Ньюмана-Кейлса контролирует общую вероятность ошибки первого рода только в слабом смысле, то есть этот критерий крайне либерален.

Функция `snk()` возвращает список из трех элементов. Первый элемент `Ordered.Means` содержит описательную статистику (имя выборки – объем – среднее – дисперсия), причем выборки упорядочены по среднему значению. Второй элемент `Variances` содержит общую оценку дисперсии (`Overall`) и число степеней свободы (`df`). Последний элемент `SNK` является фреймом с основными результатами: сравниваемые пары (`Comparison`), разница средних (`Diff`), критериальная статистика *q* (`Statistic`), *p*-значение (`Adj.P`), отвергнута ли нуль-гипотеза (`Rejected`), уровень (`Layer`).

Критерий Ньюмана-Кейлса выявил отличия в 16 парах семей пчел породы *A. m. carnica* по длине тергита:

```
> sum(nk$SNK$Rejected)
[1] 16
```

6.4. Множественные сравнения с учетом зависимости данных

Описанные выше методы множественных сравнений не учитывают взаимозависимость данных. Когда одни и те же выборки участвуют в сравнениях неоднократно, результаты сравнений теряют независимость друг от друга. Пусть проведено сравнение выборок A и B на основе одной из модификаций критерия Стьюдента и получено некое значение критериальной статистики t_1 и соответствующее p-значение. Если после этого провести сравнение выборок B и C, то результаты будут связаны с итогами первого сравнения просто потому, что половина данных, участвующих в двух сравнениях, совпадает. Таким образом, полученное во втором сравнении значение t_2 не является независимым от t_1. Поправки к p-значениям и семейство представленных методов, основанных на распределении стьюдентизированных размахов, взаимозависимость данных не учитывают, из-за чего в какой-то степени уменьшается их мощность.

Ряд современных параметрических методов множественных сравнений позволяет учесть взаимозависимость данных. При проведении множества сравнений решение о принятии гипотез принимается на основании совокупности критериальных статистик (их число равно числу сравнений). В условиях нуль-гипотезы эта совокупность описывается многомерным распределением Стьюдента, которое учитывает зависимость данных. Возможность работы с этим распределением появилась относительно недавно с развитием вычислительной техники.

В среде R современные методы множественных сравнений реализованы в пакете `multcomp`. Основная функция этого пакета – `glht()` – принимает объект, содержащий описание статистической модели, допускающей возможность множественных сравнений. В относительно простых случаях, которые мы будем рассматривать, это будет объект класса «`aov`», который создается одноименной функцией. При работе с пакетом `multcomp` принципиально важно, чтобы группирующие переменные хранились в виде факторов, а не текстовых переменных, иначе функции будут выдавать ошибки. Второй объект, который обязательно необходимо передать этой функции через аргумент `linfct`, специфицирует сравнения, которые необходимо провести. Функция `glht()` поддерживает несколько способов спецификации сравнений, в том числе с использованием матрицы линейных контрастов и различных видов текстовых описаний, однако в стандартных ситуациях удобнее воспользоваться специальной функцией `mcp()`. Функция `glht()` возвращает объект класса «`glht`», содержащий спецификацию модели. Рассмотрим конкретный пример и создадим

модель для множественных сравнений 8 семей пчел породы *A. m. carnica* по длине тергита.

```
> library(multcomp)
> carn$family <- factor(carn$family)
> model <- aov(tergit.length ~ family, data = carn)
```

Перед работой с функциями пакета `multcomp` мы заново создали модель дисперсионного анализа `model`, предварительно преобразовав группирующую переменную `family` в фактор.

```
> model.mc <- glht(model, linfct = mcp(family = "Tukey"))
```

Рассмотрим формат применения функции `mcp()`. Для каждого фактора, фигурирующего в модели в качестве независимой переменной, указывается матрица контрастов или ее текстовое описание. Имена передаваемых функции `mcp()` аргументов, должны соответствовать именам используемых в модели факторов, в нашем случае таким фактором является семейная принадлежность `family`.

Если мы хотим провести все возможные попарные сравнения групповых средних аналогично рассмотренному выше критерию Тьюки, то для спецификации этого типа множественных сравнений используется текстовая константа «`Tukey`», которая и передается вместе с аргументом `family`. Полученные итоги множественных сравнений были сохранены в объекте `model.mc`, который представляет собой список из 7 элементов, описывающих модель выполненных сравнений. Если распечатать этот объект в консоли, то будет выведен список с оценками межгрупповых разностей средних для каждой сравниваемой пары.

Проведение полного анализа и расчет *p*-значений осуществляется с помощью функции `summary()`, которой необходимо передать объект класса «`glht`», а также дополнительную спецификацию типа критерия, который необходимо использовать. По умолчанию используется так называемая одноэтапная процедура с использованием многомерного распределения Стьюдента.

```
> summary(model.mc)

 Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts

Fit: aov(formula = tergit.length ~ family, data = carn)
```

```
Linear Hypotheses:
                         Estimate Std. Error t value Pr(>|t|)
fam.24a - fam.110a == 0 -0.055677   0.016610  -3.352   0.0206 *
fam.31 - fam.110a == 0  -0.043823   0.016610  -2.638   0.1478
fam.41 - fam.110a == 0  -0.053137   0.016610  -3.199   0.0329 *
fam.62a - fam.110a == 0  0.003877   0.016610   0.233   1.0000
fam.73a - fam.110a == 0  0.023900   0.016610   1.439   0.8382
fam.87 - fam.110a == 0  -0.050073   0.016610  -3.015   0.0564 .
fam.9к - fam.110a == 0   0.007260   0.016610   0.437   0.9999
fam.31 - fam.24a == 0    0.011853   0.016610   0.714   0.9965
fam.41 - fam.24a == 0    0.002540   0.016610   0.153   1.0000
fam.62a - fam.24a == 0   0.059553   0.016610   3.585    <0.01 **
fam.73a - fam.24a == 0   0.079577   0.016610   4.791    <0.01 ***
fam.87 - fam.24a == 0    0.005603   0.016610   0.337   1.0000
fam.9к - fam.24a == 0    0.062937   0.016610   3.789    <0.01 **
fam.41 - fam.31 == 0    -0.009313   0.016610  -0.561   0.9993
fam.62a - fam.31 == 0    0.047700   0.016610   2.872   0.0825 .
fam.73a - fam.31 == 0    0.067723   0.016610   4.077    <0.01 **
fam.87 - fam.31 == 0    -0.006250   0.016610  -0.376   0.9999
fam.9к - fam.31 == 0     0.051083   0.016610   3.075   0.0477 *
fam.62a - fam.41 == 0    0.057013   0.016610   3.432   0.0161 *
fam.73a - fam.41 == 0    0.077037   0.016610   4.638    <0.01 ***
fam.87 - fam.41 == 0     0.003063   0.016610   0.184   1.0000
fam.9к - fam.41 == 0     0.060397   0.016610   3.636    <0.01 **
fam.73a - fam.62a == 0   0.020023   0.016610   1.205   0.9297
fam.87 - fam.62a == 0   -0.053950   0.016610  -3.248   0.0286 *
fam.9к - fam.62a == 0    0.003383   0.016610   0.204   1.0000
fam.87 - fam.73a == 0   -0.073973   0.016610  -4.454    <0.01 ***
fam.9к - fam.73a == 0   -0.016640   0.016610  -1.002   0.9738
fam.9к - fam.87 == 0     0.057333   0.016610   3.452   0.0154 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Adjusted p values reported -- single-step method)
```

Основные результаты выводятся в виде таблицы, в первом столбце которой описываются проверяемые нуль-гипотезы в форме разностей. Так, выражение `fam.24a - fam.110a == 0`, означает нуль-гипотезу о равенстве средних в семьях с идентификаторами `fam.24a` и `fam.110a` (разность средних равна нулю). Далее следуют столбцы с оценками разностей (`Estimate`), стандартными ошибками разностей (`Std. Error`), значениями критериальной статистики *t* (`t value`), а также *p*-значениями (`Pr(>|t|)`). *p*-значения учитывают эффект множественных сравнений и могут сравниваться с необходимым уровнем значимости.

При работе с многомерным распределением Стьюдента используются алгоритмы численного интегрирования с применением рандомизации. В итоге рассчитанные *p*-значения будут немного отличаться при каждом вызове функций, однако разброс значений обычно невелик (в районе 4-5-го знака после запятой) и не влияет на итоги анализа.

Результаты анализа можно сохранить в отдельном объекте и извлечь необходимые для дальнейшего анализа данные.

```
> model.mc.res <- summary(model.mc)
> sum(model.mc.res$test$pvalues < 0.05)
[1] 13
```

Легко видеть, что результаты мало отличаются от итогов применения «классического» критерия Тьюки, который также обнаружил значимые отличия в 13 парах семей.

Необходимая мощность множественных сравнений не всегда достигается при использовании одноэтапной процедуры, проверяющей все нуль-гипотезы одновременно. Мощность теста можно повысить, используя так называемые пошаговые процедуры, из которых наиболее эффективным является метод Вестфолла. Проблема в применении этого метода заключается в том, что он требует проверки огромного количества промежуточных гипотез, число которых лавинообразно увеличивается с ростом числа сравниваемых выборок. В результате требуются серьезные вычислительные мощности и большой объем оперативной памяти компьютера. В качестве примера проведем попарные сравнения для 7 семей пчел (имеющаяся в нашем распоряжении техника не позволила провести сравнение для всех 8 семей). Для этого нам будет необходимо создать новые модели дисперсионного анализа и множественных сравнений.

```
> model7 <- aov(tergit.length ~ family, data = carn[1:210, ])
> model7.mc <- glht(model7, linfct = mcp(family = "Tukey"))
> (model7.mc.res <- summary(model7.mc,
+                           test = adjusted(type = "Westfall")))

Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts

Fit: aov(formula = tergit.length ~ family, data = carn[1:210, ])

Linear Hypotheses:
                       Estimate Std. Error t value Pr(>|t|)
fam.31 - fam.24a == 0   0.011853   0.016763   0.707  0.94472
fam.41 - fam.24a == 0   0.002540   0.016763   0.152  0.98432
fam.62a - fam.24a == 0  0.059553   0.016763   3.553  0.00475 **
fam.73a - fam.24a == 0  0.079577   0.016763   4.747  < 0.001 ***
fam.87 - fam.24a == 0   0.005603   0.016763   0.334  0.98432
fam.9к - fam.24a == 0   0.062937   0.016763   3.754  0.00312 **
fam.41 - fam.31 == 0   -0.009313   0.016763  -0.556  0.94472
fam.62a - fam.31 == 0   0.047700   0.016763   2.846  0.02311 *
fam.73a - fam.31 == 0   0.067723   0.016763   4.040  < 0.001 ***
fam.87 - fam.31 == 0   -0.006250   0.016763  -0.373  0.97536
fam.9к - fam.31 == 0    0.051083   0.016763   3.047  0.01461 *
fam.62a - fam.41 == 0   0.057013   0.016763   3.401  0.00531 **
fam.73a - fam.41 == 0   0.077037   0.016763   4.596  < 0.001 ***
fam.87 - fam.41 == 0    0.003063   0.016763   0.183  0.98432
```

```
fam.9к - fam.41 == 0    0.060397   0.016763   3.603  0.00357 **
fam.73a - fam.62a == 0  0.020023   0.016763   1.194  0.79864
fam.87 - fam.62a == 0  -0.053950   0.016763  -3.218  0.00727 **
fam.9к - fam.62a == 0   0.003383   0.016763   0.202  0.98432
fam.87 - fam.73a == 0  -0.073973   0.016763  -4.413  < 0.001 ***
fam.9к - fam.73a == 0  -0.016640   0.016763  -0.993  0.83241
fam.9к - fam.87 == 0    0.057333   0.016763   3.420  0.00488 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Adjusted p values reported -- Westfall method)
```

Обратите внимание, что при вызове функции `summary()` мы воспользовались аргументом `test`, которому передали функцию `adjusted()` с аргументом `type = "Westfall"`. Эта функция определяет метод для расчета *p*-значений, по умолчанию используется `test = adjusted(type = “single-step”))`, то есть одноэтапная процедура.

Проведем теперь попарные сравнения с использованием одноэтапной процедуры и сравним результаты.

```
> (model7.mc.res2 <- summary(model7.mc))

Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts

Fit: aov(formula = tergit.length ~ family, data = carn[1:210, ])

Linear Hypotheses:
                       Estimate Std. Error t value Pr(>|t|)
fam.31 - fam.24a == 0   0.011853   0.016763   0.707  0.99207
fam.41 - fam.24a == 0   0.002540   0.016763   0.152  1.00000
fam.62a - fam.24a == 0  0.059553   0.016763   3.553  0.00855 **
fam.73a - fam.24a == 0  0.079577   0.016763   4.747  < 0.001 ***
fam.87 - fam.24a == 0   0.005603   0.016763   0.334  0.99989
fam.9к - fam.24a == 0   0.062937   0.016763   3.754  0.00450 **
fam.41 - fam.31 == 0   -0.009313   0.016763  -0.556  0.99789
fam.62a - fam.31 == 0   0.047700   0.016763   2.846  0.07151 .
fam.73a - fam.31 == 0   0.067723   0.016763   4.040  0.00141 **
fam.87 - fam.31 == 0   -0.006250   0.016763  -0.373  0.99978
fam.9к - fam.31 == 0    0.051083   0.016763   3.047  0.04112 *
fam.62a - fam.41 == 0   0.057013   0.016763   3.401  0.01399 *
fam.73a - fam.41 == 0   0.077037   0.016763   4.596  < 0.001 ***
fam.87 - fam.41 == 0    0.003063   0.016763   0.183  1.00000
fam.9к - fam.41 == 0    0.060397   0.016763   3.603  0.00699 **
fam.73a - fam.62a == 0  0.020023   0.016763   1.194  0.89563
fam.87 - fam.62a == 0  -0.053950   0.016763  -3.218  0.02471 *
fam.9к - fam.62a == 0   0.003383   0.016763   0.202  0.99999
fam.87 - fam.73a == 0  -0.073973   0.016763  -4.413  < 0.001 ***
fam.9к - fam.73a == 0  -0.016640   0.016763  -0.993  0.95514
fam.9к - fam.87 == 0    0.057333   0.016763   3.420  0.01322 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
 (Adjusted p values reported -- single-step method)
```

Итак, пошаговая процедура Вестфолла обнаружила значимые отличия в 12 парах из 21, тогда как одноэтапная процедура обнаружила значимые отличия только в 11 парах. Для наглядности можно сравнить соответствующие *p*-значения:

```
> model7.mc.res2$test$pvalues > model7.mc.res$test$pvalues
 [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[14] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

Во всех случаях *p*-значения, получаемые путем одноэтапной процедуры, выше значений, получаемых методом Вестфолла. В результате в одном из попарных сравнений одноэтапная процедура значимых отличий не обнаружила ($p > 0.05$).

6.5. Доверительные интервалы и визуализация результатов

После знакомства с основными методами множественных сравнений рассмотрим вопрос о том, как можно визуализировать полученные результаты. Один из вариантов такой визуализации мы видели в разделе 16.2, когда строили график доверительных интервалов для разности средних на основе объекта класса «TukeyHSD». Аналогичные графики можно строить и с применением инструментов пакета multcomp.

Мы начнем наше рассмотрение с параллельных диаграмм размаха. Такой график дает качественное представление о характере распределения данных в совокупности рассматриваемых групп. Для его построения функции boxplot() нужно передать не отдельный вектор, как мы делали в разделе 2.1, а формулу. Справа от тильды указывается числовая переменная, для которой нужно построить диаграммы размахов, а слева – фактор, задающий разделение на категории (выборки). Построим параллельные диаграммы размаха для длины тергита в 8 семьях пчел породы *A. m. carnica*.

```
> par(mar = c(6, 4, 1, 1))
> boxplot(tergit.length ~ family, data = carn,
+         ylab = "Длина тергита, мм", las = 3,
+         col = "lightgrey")
> mtext("Семьи", side = 1, line = 5)
```

График представлен на Рис. 18. Мы внесли следующие модификации для лучшего визуального исполнения графика: нижнее поле увеличено до 6 строк (mar = c(6,4,1,1)), чтобы вместить идентификаторы семей, направление подписей отметок на осях изменено на вертикальное (las = 3), подпись горизонтальной оси добавлена с помощью функции mtext(), чтобы избежать

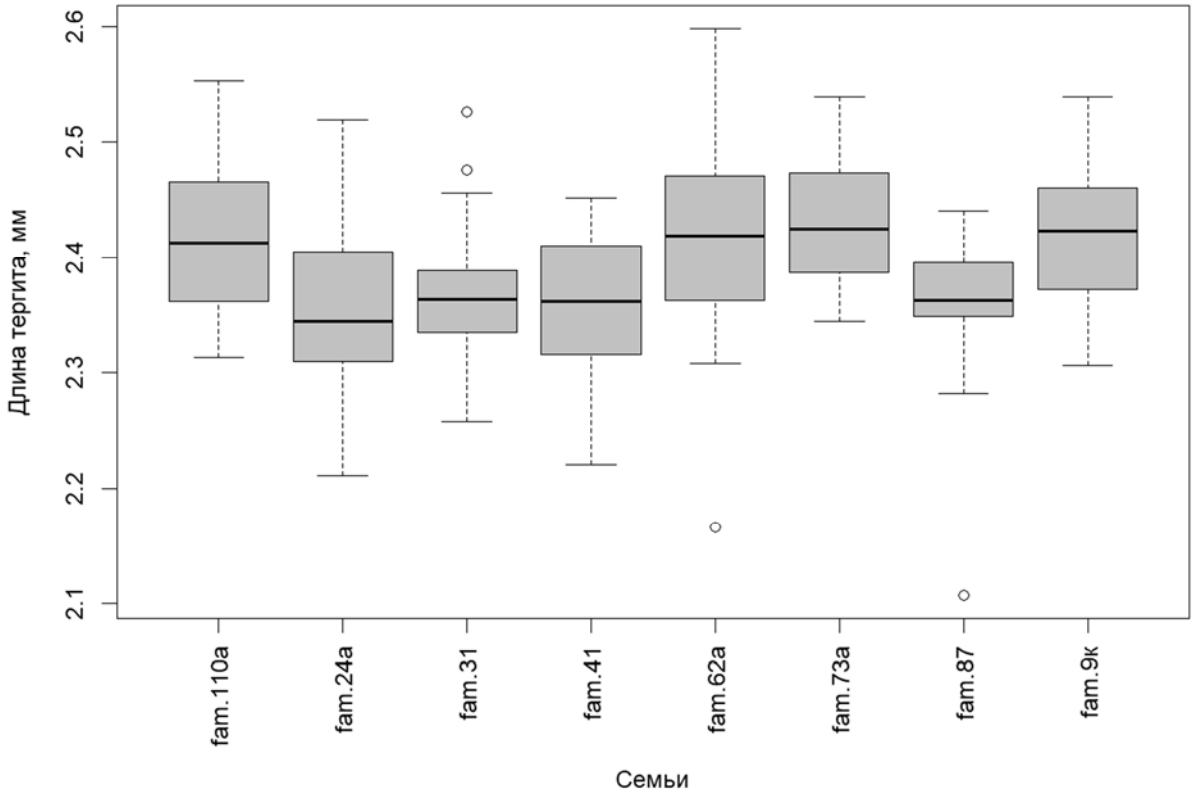

Рис. 18. Параллельные диаграммы размаха

пересечения с идентификаторами семей. Благодаря параллельному расположению диаграммы размаха можно получить примерное представление о том, как соотносятся выборки между собой. Так, хорошо видно, что семьи `fam.62a`, `fam.73a` и `fam.9к` практически не отличаются по длине тергита, чего не скажешь о семьях `fam.87` и `fam.9к`.

Параллельные диаграммы размаха не содержат информации о значимости отличий средних. Для ее визуализации лучше воспользоваться графиком доверительных интервалов для разности средних. Но сначала нужно рассчитать сами доверительные интервалы. Это делается с помощью функции `confint()`, которой передается объект, содержащий модель множественных сравнений. Построим 95 % доверительные интервалы для всех попарных сравнений 8 семей породы *A. m. carnica* по длине тергита.

```
> (model.mc.ci <- confint(model.mc, level = 0.95))

 Simultaneous Confidence Intervals

Multiple Comparisons of Means: Tukey Contrasts

Fit: aov(formula = tergit.length ~ family, data = carn)
```

```
Quantile = 3.0595
95% family-wise confidence level

Linear Hypotheses:
                          Estimate   lwr        upr
fam.24a - fam.110a == 0 -0.0556767 -0.1064954 -0.0048580
fam.31 - fam.110a == 0  -0.0438233 -0.0946420  0.0069954
fam.41 - fam.110a == 0  -0.0531367 -0.1039554 -0.0023180
fam.62a - fam.110a == 0  0.0038767 -0.0469420  0.0546954
fam.73a - fam.110a == 0  0.0239000 -0.0269187  0.0747187
fam.87 - fam.110a == 0  -0.0500733 -0.1008920  0.0007454
fam.9к - fam.110a == 0   0.0072600 -0.0435587  0.0580787
fam.31 - fam.24a == 0    0.0118533 -0.0389654  0.0626720
fam.41 - fam.24a == 0    0.0025400 -0.0482787  0.0533587
fam.62a - fam.24a == 0   0.0595533  0.0087346  0.1103720
fam.73a - fam.24a == 0   0.0795767  0.0287580  0.1303954
fam.87 - fam.24a == 0    0.0056033 -0.0452154  0.0564220
fam.9к - fam.24a == 0    0.0629367  0.0121180  0.1137554
fam.41 - fam.31 == 0    -0.0093133 -0.0601320  0.0415054
fam.62a - fam.31 == 0    0.0477000 -0.0031187  0.0985187
fam.73a - fam.31 == 0    0.0677233  0.0169046  0.1185420
fam.87 - fam.31 == 0    -0.0062500 -0.0570687  0.0445687
fam.9к - fam.31 == 0     0.0510833  0.0002646  0.1019020
fam.62a - fam.41 == 0    0.0570133  0.0061946  0.1078320
fam.73a - fam.41 == 0    0.0770367  0.0262180  0.1278554
fam.87 - fam.41 == 0     0.0030633 -0.0477554  0.0538820
fam.9к - fam.41 == 0     0.0603967  0.0095780  0.1112154
fam.73a - fam.62a == 0   0.0200233 -0.0307954  0.0708420
fam.87 - fam.62a == 0   -0.0539500 -0.1047687 -0.0031313
fam.9к - fam.62a == 0    0.0033833 -0.0474354  0.0542020
fam.87 - fam.73a == 0   -0.0739733 -0.1247920 -0.0231546
fam.9к - fam.73a == 0   -0.0166400 -0.0674587  0.0341787
fam.9к - fam.87 == 0     0.0573333  0.0065146  0.1081520
```

Аргумент `level` определяет доверительную вероятность, для которой строятся интервалы, по умолчанию это 0.95, то есть в нашем случае можно было этот аргумент опустить. Результаты расчета сохранены в объект `model.mc.ci` и выведены в консоль. Для каждой пары выведены формулировка нуль-гипотезы, разность средних (`Estimate`), нижняя (`lwr`) и верхняя (`upr`) границы доверительного интервала.

Для построения графика доверительных интервалов разностей полученный объект нужно передать функции `plot()`. По умолчанию строятся доверительные интервалы черного цвета. Для улучшения читаемости графика мы добавим цвет: интервалы, покрывающие ноль, будут темно-зелеными, а интервалы, не покрывающие ноль (что указывает на значимость разности средних), – красными.

```
> index <- model.mc.ci$confint[,2] < 0 & model.mc.ci$confint[, 3] > 0
> col <- rep("grey", 28)
> col[!index] <- "black"
```

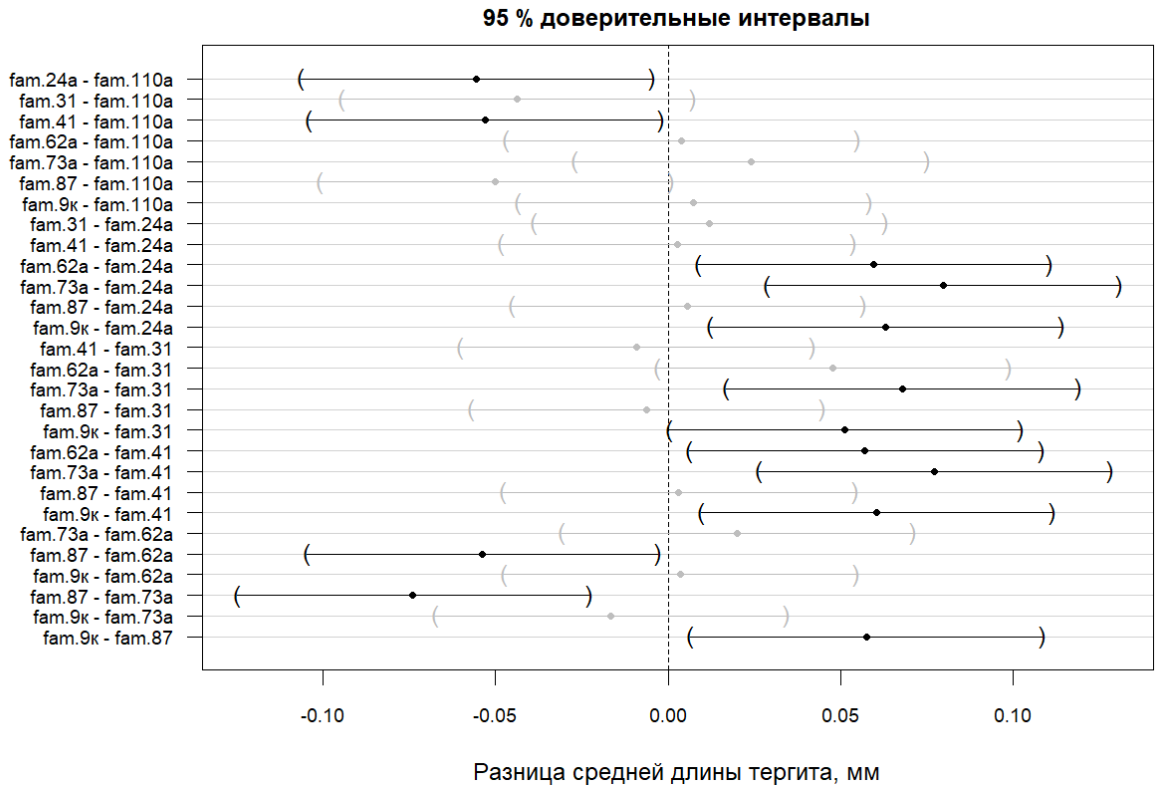

Рис. 19. График доверительных интервалов разностей

```
> opar <- par(mar = c(5, 8, 2, 2))
> plot(model.mc.ci, main = "95 % доверительные интервалы",
+         xlab = "Разница средней длины тергита, мм",
+         cex.axis = 0.8, cex.main = 1, col = col)
> par(opar)
```

Полученный график (Рис. 19) эквивалентен графику на Рис. 17, который получен с использованием стандартных пакетов среды R. Однако инструменты пакета `multcomp` позволяют модифицировать график доверительных интервалов разностей средних.

Еще один вид представления информации о значимости разности средних заключается в использовании так называемых компактных буквенных обозначений (compact letter display). Каждому уровню фактора назначаются буквенные кодировки, отражающие значимость разности средних. Группы, не отличающиеся значимо, получают общую букву. Таким образом, пары групп, не имеющих общей буквы, отличаются на используемом уровне значимости. Для создания компактных буквенных обозначений используется функция `cld()`, которой передается объект, содержащий модель множественных сравнений или же результаты функций `summary()` и `confint()`.

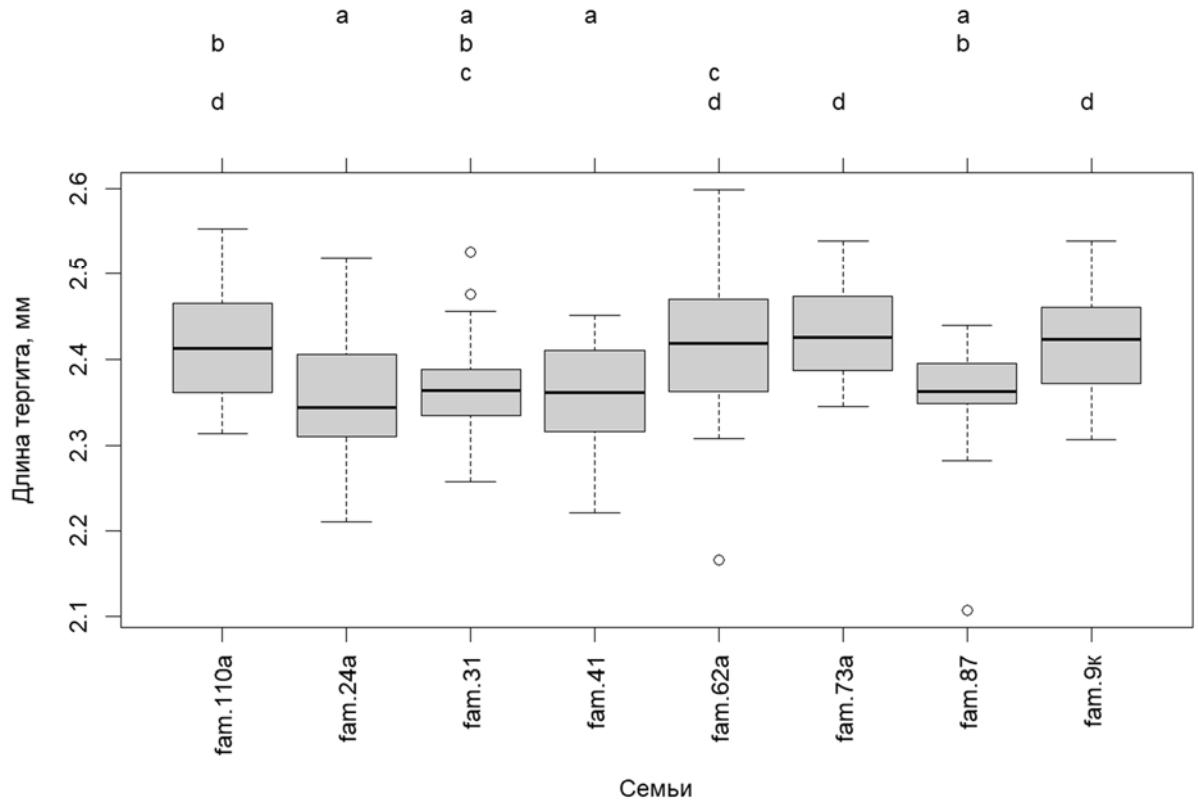

Рис. 20. Параллельные диаграммы размаха с компактными буквенными обозначениями

```
> (model.mc.cld <- cld(model.mc))
fam.110a  fam.24a   fam.31   fam.41  fam.62a  fam.73a   fam.87   fam.9к
    "bd"      "a"    "abc"      "a"     "cd"      "d"     "ab"      "d"
```

Функция возвращает сложно структурированный список, однако при его выводе в консоль отображаются назначенные каждой группе буквы. Для визуализации полученный объект нужно передать функции `plot()`, которая создаст параллельные диаграммы размаха для каждой группы, но сверху будут отображены компактные буквенные обозначения.

```
> opar <- par(mar = c(6, 4, 8, 2))
> plot(model.mc.cld, main = "", xlab = "", ylab = "Длина тергита, мм",
+      las = 3, col = "lightgrey")
> mtext("Семьи", side = 1, line = 5)
> par(opar)
```

Полученный график (Рис. 20) эквивалентен Рис. 18 (использовано то же самое оформление, см. выше), но теперь он содержит также буквенные обозначения, которые позволяют сразу понять, что, например, семьи `fam.24a`, `fam.31`, `fam.41` и `fam.87` между собой не отличаются, поскольку имеют общую

букву «а», а вот семьи `fam.110a` и `fam.24a` отличаются, поскольку общих букв не имеют.

6.6. Сравнения с контрольной группой

Многие экспериментальные планы не требуют полного набора попарных сравнений (когда каждая группа сравнивается с каждой). Часто встречаются схемы экспериментов, когда необходимо сравнить множество групп с одной – контрольной. Конечно, и в этой ситуации можно провести все попарные сравнения одним из описанных выше методов и выбрать только те результаты, которые относятся к сравнениям с контрольной группой. Однако при таком подходе вследствие большого общего числа сравнений мощность анализа будет невелика (чем больше число сравнений, тем строже требования во всех методах).

В таких ситуациях применяется критерий Даннетта, в своей классической форме являющийся модификацией критерия Тьюки. Мы же рассмотрим его современную реализацию, учитывающую зависимость попарных сравнений.

Рассмотрим применение этой схемы попарных сравнений на примере уже многократно проанализированных данных по длине тергита в 8 семьях пчел породы *A. m. carnica*. Формально у нас нет контрольной группы, однако мы будем считать, что задачей исследования является не выявление внутрисемейной изменчивости, а поиск отклонений от выборки пчел породы *A. m. mellifera*. Таким образом, выборка пчел породы *A. m. mellifera* будет являться той группой, с которой мы будем проводить сравнения.

```
> carn.mel <- subset(apis, subset = breed == "carnica" |
+                                       breed == "mellifera" )
> carn.mel$family <- factor(carn.mel$family)
> model <- aov(tergit.length ~ family, data = carn.mel)
```

Во-первых, мы выбрали данные по двум интересующим нас породам в новый фрейм `carn.mel`, затем преобразовали переменную `family` нового фрейма в фактор, после чего создали новую модель дисперсионного анализа. Перед множественными сравнениями нужно убедиться в том, что семейная принадлежность влияет на длину тергита, то есть провести дисперсионный анализ:

```
> anova(model)
Analysis of Variance Table

Response: tergit.length
           Df Sum Sq  Mean Sq F value    Pr(>F)
family      8 0.7138 0.089225  21.522 < 2.2e-16 ***
Residuals 345 1.4303 0.004146
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Таблица дисперсионного анализа свидетельствует о том, что действие фактора `family` статистически значимо (`p < 2.2e-16`). Теперь можно переходить к процедуре множественных сравнений. Прежде всего нужно создать объект, содержащий спецификацию необходимого нам анализа. Делается это точно так же, как и в случае попарных сравнений, однако вместо «`Tukey`» нужно указать «`Dunnett`».

```
> (model.mc <- glht(model, linfct = mcp(family = "Dunnett")))

 General Linear Hypotheses

Multiple Comparisons of Means: Dunnett Contrasts

Linear Hypotheses:
                          Estimate
fam.24a - fam.110a == 0 -0.055677
fam.31 - fam.110a == 0  -0.043823
fam.41 - fam.110a == 0  -0.053137
fam.62a - fam.110a == 0  0.003877
fam.73a - fam.110a == 0  0.023900
fam.87 - fam.110a == 0  -0.050073
fam.9к - fam.110a == 0   0.007260
fam.mel - fam.110a == 0  0.058626
```

Мы сразу же вывели полученный объект в консоль. Видно, что создана модель с контрастами Даннетта, то есть проведено сравнение множества средних с контрольным, но в качестве контроля использована семья `fam.110a`. Так произошло потому, что контрольной считается та группа, которой соответствует первое значение в списке уровней фактора. Рассмотрим использованный нами фактор `family` подробнее. Мы получили его путем преобразования исходной текстовой переменной с помощью функции `factor()`. При такой процедуре все доступные значения текстовой переменной упорядочиваются по алфавиту, поэтому первым в списке уровней получившегося фактора стоит `fam.110a`. В этом можно убедиться с помощью функции `levels()`, которая выводит уровни значений фактора.

```
> levels(carn.mel$family)
[1] "fam.110a" "fam.24a"  "fam.31"   "fam.41"   "fam.62a"  "fam.73a"
[7] "fam.87"   "fam.9к"   "fam.mel"
```

Для создания нужной нам спецификации множественных сравнений необходимо изменить порядок уровней фактора `family`. Это можно сделать с помощью функции `relevel()` следующим образом.

```
> carn.mel$family <- relevel(carn.mel$family, ref = "fam.mel")
> levels(carn.mel$family)
[1] "fam.mel"  "fam.110a" "fam.24a"  "fam.31"   "fam.41"   "fam.62a"
[7] "fam.73a"  "fam.87"   "fam.9к"
```

При использовании функции `relevel()` аргументу `ref` передается значение уровня, который необходимо сделать базовым. Этот уровень получит единичный код и будет первым в списке уровней. После преобразования фактора модель дисперсионного анализа и модель множественных сравнений нужно создать заново.

```
> model <- aov(tergit.length ~ family, data = carn.mel)
> (model.mc <- glht(model, linfct = mcp(family = "Dunnett")))

 General Linear Hypotheses

Multiple Comparisons of Means: Dunnett Contrasts


Linear Hypotheses:
                         Estimate
fam.110a - fam.mel == 0 -0.05863
fam.24a - fam.mel == 0  -0.11430
fam.31 - fam.mel == 0   -0.10245
fam.41 - fam.mel == 0   -0.11176
fam.62a - fam.mel == 0  -0.05475
fam.73a - fam.mel == 0  -0.03473
fam.87 - fam.mel == 0   -0.10870
fam.9к - fam.mel == 0   -0.05137
```

Теперь модель специфицирована правильно, все сравнения предполагается делать с группой пчел породы *A. m. mellifera*. Обратите внимание, что все значения разностей отрицательны, что свидетельствует о том, что длина тергита в семьях пчел породы *A. m. carnica*, меньше, однако насколько значимы эти разности? Как и в случае схемы Тьюки есть две альтернативы: одношаговая процедура и пошаговая нисходящая процедура. Первая реализована в методе `summary()` по умолчанию.

```
 > summary(model.mc)

 Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Dunnett Contrasts

Fit: aov(formula = tergit.length ~ family, data = carn.mel)

Linear Hypotheses:
                        Estimate Std. Error t value Pr(>|t|)
fam.110a - fam.mel == 0 -0.05863    0.01321  -4.437  < 1e-04 ***
fam.24a - fam.mel == 0  -0.11430    0.01321  -8.651  < 1e-04 ***
fam.31 - fam.mel == 0   -0.10245    0.01321  -7.754  < 1e-04 ***
fam.41 - fam.mel == 0   -0.11176    0.01321  -8.459  < 1e-04 ***
fam.62a - fam.mel == 0  -0.05475    0.01321  -4.144 0.000343 ***
fam.73a - fam.mel == 0  -0.03473    0.01321  -2.628 0.066694 .
fam.87 - fam.mel == 0   -0.10870    0.01321  -8.227  < 1e-04 ***
fam.9к - fam.mel == 0   -0.05137    0.01321  -3.888 0.000966 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Adjusted p values reported -- single-step method)
```

Высоко значимые отличия обнаружены одноэтапной процедурой для всех семей за исключением fam.73a. Воспользуемся теперь пошаговой нисходящей процедурой. Для этого при вызове функции summary() аргументу test нужно передать функцию adjusted() с аргументом type = "free".

```
> summary(model.mc, test = adjusted(type = "free"))

 Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Dunnett Contrasts

Fit: aov(formula = tergit.length ~ family, data = carn.mel)

Linear Hypotheses:
                        Estimate Std. Error t value Pr(>|t|)
fam.110a - fam.mel == 0 -0.05863    0.01321  -4.437 5.03e-05 ***
fam.24a - fam.mel == 0  -0.11430    0.01321  -8.651 3.55e-15 ***
fam.31 - fam.mel == 0   -0.10245    0.01321  -7.754 5.04e-13 ***
fam.41 - fam.mel == 0   -0.11176    0.01321  -8.459 1.18e-14 ***
fam.62a - fam.mel == 0  -0.05475    0.01321  -4.144 0.000127 ***
fam.73a - fam.mel == 0  -0.03473    0.01321  -2.628 0.008964 **
fam.87 - fam.mel == 0   -0.10870    0.01321  -8.227 6.67e-14 ***
fam.9к - fam.mel == 0   -0.05137    0.01321  -3.888 0.000242 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Adjusted p values reported -- free method)
```

Мы видим, что при использовании пошаговой процедуры все *p*-значения уменьшились, в том числе и *p*-значение для сравнения с контролем семьи `fam.73a`, в результате чего значимые отличия теперь обнаружены во всех сравнениях.

6.7. Непараметрические аналоги

Выше мы рассматривали методы множественных сравнений для случая анализа данных, соответствующих нормальному распределению. Однако проблема множественных сравнений универсальна, она неизбежно возникает и при использовании непараметрических методов. Так, если критерий Крускала-Уоллиса выявил значимые отличия в совокупности групп, обычно возникает необходимость провести попарные сравнения.

Непараметрические методы множественных сравнений не так многочисленны, как параметрические, но они существуют для всех наиболее распространенных в практике анализа данных ситуаций.

Поправки к уровню значимости (или к *p*-значениям) при проведении попарных сравнений являются универсальным методом, их можно применять и в случае использования непараметрических критериев. Для удобства применения этих поправок в случае критерия Уилкоксона, который применяется для сравнения двух выборок, в R существует специальная функция `pairwise.wilcox.test()`, полностью аналогичная функции `pairwise.t.test()`, рассмотренной в разделе 15.1. Рассмотрим ее применение на примере попарных сравнений 8 семей пчел породы *A. m. carnica* по длине хоботка. Напомним, что этот признак не подчиняется нормальному распределению (см. раздел 3.3), а критерий Крускала-Уоллиса выявил значимость межсемейных отличий (см. раздел 5.5).

```
> pairwise.wilcox.test(carn$proboscis, carn$family,
+                          p.adjust.method = "bonferroni")

 Pairwise comparisons using Wilcoxon rank sum test

data:  carn$proboscis and carn$family
```

```
        fam.110a fam.24a fam.31 fam.41 fam.62a fam.73a fam.87
fam.24a 1.0000   -       -      -      -       -       -
fam.31  1.0000   1.0000  -      -      -       -       -
fam.41  0.1040   0.0082  0.0656 -      -       -       -
fam.62a 0.4343   0.0289  1.0000 1.0000 -       -       -
fam.73a 1.0000   0.0476  1.0000 0.5133 1.0000  -       -
fam.87  1.0000   0.2141  1.0000 0.1776 1.0000  1.0000  -
fam.9к  0.8488   0.1129  1.0000 1.0000 1.0000  1.0000  1.0000

P value adjustment method: bonferroni
```

Для спецификации типа поправки используется аргумент `p.adjust.method`, которому передается текстовая константа («`bonferroni`» – поправка Бонферрони, «`holm`» – метод Холма). В консоль выводится матрица *p*-значений с соответствующей поправкой, которые следует сравнивать с желаемым уровнем значимости.

Попарные сравнения с поправкой Бонферрони обнаружили значимые отличия только в 3 парах семей на 5 %-ном уровне значимости, что неудивительно с учетом консервативности этой процедуры. Проведем теперь такой же анализ с поправкой Холма:

```
> pairwise.wilcox.test(carn$proboscis, carn$family,
+                      p.adjust.method = "holm")

 Pairwise comparisons using Wilcoxon rank sum test

data:  carn$proboscis and carn$family

        fam.110a fam.24a fam.31 fam.41 fam.62a fam.73a fam.87
fam.24a 1.0000   -       -      -      -       -       -
fam.31  1.0000   1.0000  -      -      -       -       -
fam.41  0.0892   0.0082  0.0585 -      -       -       -
fam.62a 0.3102   0.0279  0.6080 1.0000 -       -       -
fam.73a 0.6080   0.0442  1.0000 0.3483 1.0000  -       -
fam.87  1.0000   0.1606  1.0000 0.1396 1.0000  1.0000  -
fam.9к  0.5457   0.0927  1.0000 1.0000 1.0000  1.0000  1.0000

P value adjustment method: holm
```

Несмотря на то, что *p*-значения уменьшились, метод Холма не выявил дополнительных пар со значимыми отличиями.

Критерий Уилкоксона основан на рангах. При его использовании для попарных сравнений ранги назначаются в каждой паре независимо от остальных сравнений. Другой подход заключается в том, чтобы назначать ранги для всей совокупности сравниваемых выборок (так делается при применении Крускала-

Уоллиса) и использовать их для проведения попарных сравнений. В этом случае ранги содержат больше информации, поэтому увеличивается мощность критерия.

Процедуры, основанные на общих рангах, в значительной степени аналогичны параметрическим критериям Тьюки и Даннетта, однако в них используется стандартное нормальное распределение. Непараметрические аналоги критериев Тьюки и Даннетта, реализованы в функции `kruskalmc()` пакета `pgirmess`. По умолчанию эта функция предполагает, что нам необходимо провести полный набор попарных сравнений.

```
> library(pgirmess)
> kruskalmc(proboscis ~ family, data = carn)
Multiple comparison test after Kruskal-Wallis
p.value: 0.05
Comparisons
                    obs.dif critical.dif difference
fam.110a-fam.24a 17.583333     55.99536      FALSE
fam.110a-fam.31   2.600000     55.99536      FALSE
fam.110a-fam.41  61.650000     55.99536       TRUE
fam.110a-fam.62a 37.450000     55.99536      FALSE
fam.110a-fam.73a 33.183333     55.99536      FALSE
fam.110a-fam.87  27.183333     55.99536      FALSE
fam.110a-fam.9к  31.383333     55.99536      FALSE
fam.24a-fam.31   20.183333     55.99536      FALSE
fam.24a-fam.41   79.233333     55.99536       TRUE
fam.24a-fam.62a  55.033333     55.99536      FALSE
fam.24a-fam.73a  50.766667     55.99536      FALSE
fam.24a-fam.87   44.766667     55.99536      FALSE
fam.24a-fam.9к   48.966667     55.99536      FALSE
fam.31-fam.41    59.050000     55.99536       TRUE
fam.31-fam.62a   34.850000     55.99536      FALSE
fam.31-fam.73a   30.583333     55.99536      FALSE
fam.31-fam.87    24.583333     55.99536      FALSE
fam.31-fam.9к    28.783333     55.99536      FALSE
fam.41-fam.62a   24.200000     55.99536      FALSE
fam.41-fam.73a   28.466667     55.99536      FALSE
fam.41-fam.87    34.466667     55.99536      FALSE
fam.41-fam.9к    30.266667     55.99536      FALSE
fam.62a-fam.73a   4.266667     55.99536      FALSE
fam.62a-fam.87   10.266667     55.99536      FALSE
fam.62a-fam.9к    6.066667     55.99536      FALSE
fam.73a-fam.87    6.000000     55.99536      FALSE
fam.73a-fam.9к    1.800000     55.99536      FALSE
fam.87-fam.9к     4.200000     55.99536      FALSE
```

В консоль выводится таблица, в которой для каждой сравниваемой пары указывается: значение критериальной статистики, представляющее собой разность средних рангов (`obs.dif`), критическое значение, вычисленное для необходимого уровня значимости (`critical.dif`), а также результат сравнения наблюдаемого и критического значений (`difference = TRUE` указывает на

значимость отличий). В случае анализа попарных сравнений длины хоботка в 8 семьях породы *A. m. carnica* статистически значимые отличия выявлены для 3 пар семей. Обратите внимание, что это не те пары семей, отличия между которыми выявлены с использованием критерия Уилкоксона (совпадает только одна пара из трех).

Наиболее современные непараметрические методы множественных сравнений учитывают взаимозависимость данных и используют многомерные распределения. Мы рассмотрим метод, предложенный Гао с соавторами (2008). Метод основан на использовании так называемых псевдо-рангов, которые отличаются от обычных рангов поправками, учитывающими неравенство объёмов выборки. Сама процедура является пошаговой. Метод Гао реализован в функции `gao_cs()` пакета `nparcomp`.

```
> gt <- gao_cs(proboscis ~ family, data = carn)


 #----Gao et al's (2008) modification of Campbell and Skillings (1985) (CS) stepwise multiple comparison procedure
 #---- This function uses joint ranks of the data. Attention: In the CS algorithm, the samples are jointly reranked!
 #----Reference: Gao, X. et al. (2008). Nonparametric Multiple Comparison Procedures for Unbalanced One-Way Factorial Designs. JSPI 138, 2574 - 2591.
```

Функция выдает несколько комментариев с описанием метода (чтобы избежать их вывода при каждом вызове функции нужно воспользоваться аргументом `silent = TRUE`) и возвращает список из трех фреймов: вспомогательная описательная статистика (`Info`), результаты одноэтапной процедуры (`Single.Analysis`) и результаты пошаговой процедуры (`CS.Analysis`). Нас интересует последний. В целях экономии места распечатаем ключевые показатели (описание пары, критериальная статистика, *p*-значение, итоговый результат сравнения):

```
> gt$CS.Analysis[c("Comp", "Statistic", "Adj.P", "Rejected")]
               Comp Effect Statistic  Adj.P Rejected
1    fam.41-fam.24a 0.3301    6.0994 0.0016     TRUE
2   fam.62a-fam.24a 0.2403    4.9547 0.0148     TRUE
3   fam.41-fam.110a 0.2688    5.0106 0.0137     TRUE
4   fam.73a-fam.24a 0.2294    4.7190 0.0177     TRUE
5  fam.62a-fam.110a 0.1660    3.4291 0.1649    FALSE
6     fam.41-fam.31 0.2557    4.3465 0.0362     TRUE
7    fam.9к-fam.24a 0.2109    4.4710 0.0204     TRUE
8  fam.73a-fam.110a 0.1523    3.1945 0.1733    FALSE
9    fam.62a-fam.31 0.1347    2.4047  0.442    FALSE
10    fam.41-fam.87 0.1680    2.9813 0.2313    FALSE
11   fam.87-fam.24a 0.1900    3.6751  0.056    FALSE
12  fam.9к-fam.110a 0.1324    2.8793 0.1869    FALSE
13   fam.73a-fam.31 0.1214    2.2218 0.4033    FALSE
```

```
14   fam.62a-fam.87 0.0172    0.3129 0.9961    FALSE
15    fam.41-fam.9к 0.1436    2.5564 0.2806    FALSE
16   fam.31-fam.24a 0.0702    1.2373 0.6582    FALSE
17  fam.87-fam.110a 0.1020    2.0753 0.3142    FALSE
18    fam.9к-fam.31 0.0937    1.7679 0.4293    FALSE
19   fam.73a-fam.87 0.0163    0.3005 0.9754    FALSE
20   fam.62a-fam.9к 0.0215    0.4002 0.9568    FALSE
21   fam.41-fam.73a 0.1607    3.0269 0.0916    FALSE
22 fam.110a-fam.24a 0.0922    1.7430 0.2229    FALSE
23  fam.31-fam.110a 0.0239    0.4461 0.7536    FALSE
24    fam.87-fam.31 0.0678    1.2734 0.3717    FALSE
25    fam.9к-fam.87 0.0222    0.4149 0.7703    FALSE
26   fam.73a-fam.9к 0.0311    0.5813 0.6826    FALSE
27  fam.62a-fam.73a 0.0156    0.2903 0.8381    FALSE
28   fam.41-fam.62a 0.1133    2.1565 0.1329    FALSE
```

Итак, значимые отличия выявлены в 6 парах. Дополнительно к 5 парам, выявленным с помощью критерия Уилкоксона и непараметрического аналога критерия Тьюки, обнаружена значимость отличий в длине хоботка между семьями `fam.24a` и `fam.9к`.

Рассмотрим теперь аналогичные непараметрические методы для сравнения с контрольной группой. Аналог критерия Даннетта (в случае неравного объема выборок его называют критерием Данна) реализован в функции `kruskalmc()` пакета `pgirmess`. Для спецификации вида анализа нужно воспользоваться аргументом `cont`, которому передается тип необходимого сравнения: «`one-tailed`» – односторонний, «`two-tailed`» – двусторонний. Рассмотрим применение этого метода для сравнения по длине хоботка 8 семей пчел породы *A. m. carnica* с условно контрольной выборкой пчел породы *A. m. mellifera*. Данные уже выбраны в отдельный фрейм `carn.mel` и фактор `family` преобразован таким образом, чтобы уровень `carn.mel` являлся базовым (это важно, поскольку базовый уровень выбирается в качестве контрольного, см. раздел 6.6). Перед проведением множественных сравнений необходимо убедиться в значимости межсемейных отличий:

```
> kruskal.test(proboscis ~ family, data = carn.mel)

Kruskal-Wallis rank sum test

data:  proboscis by family
Kruskal-Wallis chi-squared = 68.0559, df = 8, p-value = 1.197e-11
```

Отличия высоко значимы (`p-value = 1.197e-11`), можно переходить к множественным сравнениям с контрольной группой.

```
> kruskalmc(proboscis ~ family, data = carn.mel, cont = "two-tailed")
Multiple comparison test after Kruskal-Wallis, treatments vs control
(two-tailed)
p.value: 0.05
```

```
Comparisons
                   obs.dif critical.dif difference
fam.mel-fam.110a 106.20439     57.41832       TRUE
fam.mel-fam.24a  131.18772     57.41832       TRUE
fam.mel-fam.31    78.33772     57.41832       TRUE
fam.mel-fam.41    13.01228     57.41832      FALSE
fam.mel-fam.62a   49.58772     57.41832      FALSE
fam.mel-fam.73a   42.18772     57.41832      FALSE
fam.mel-fam.87    42.02105     57.41832      FALSE
fam.mel-fam.9к    66.02105     57.41832       TRUE
```

Структура вывода аналогична случаю полного перебора пар. Отличия от контроля выявлены для 4 семей из 8.

В заключение рассмотрим метод Гао и соавторов для сравнения с контрольной группой, реализованный в функции `gao()` пакета `nparcomp`.

```
> gd <- gao(proboscis ~ family, data = carn.mel)

 #----Xin Gao et al's (2008) Non-Parametric Multiple Test Procedure
 #----Type of Adjustment: Hochberg
 #----Level of significance = 0.05
 #----The procedure compares if the distribution functions F() are
equal. The FWER is strongly controlled
 #---- This function uses pseudo ranks of the data!
 #----Reference: Gao, X. et al. (2008). Nonparametric Multiple
Comparison Procedures for Unbalanced One-Way Factorial Designs. JSPI 138,
2574 - 2591.
```

Аналогично функции `gao_cs()` функция `gao()` выдает ряд сообщений с описанием метода (их вывод также можно отменить с помощью аргумента `silent = TRUE`) и возвращает список из двух фреймов с вспомогательными и основными результатами. Распечатаем ключевые результаты (описание пары, критериальная статистика, *p*-значение, итоговый результат сравнения):

```
> gd$Analysis[c("Comparison", "Statistic", "P.Hochberg", "Rejected")]
              Comparison Statistic P.Hochberg Rejected
1 F(fam.110a)-F(fam.mel)   -6.0926     0.0000     TRUE
2  F(fam.24a)-F(fam.mel)   -7.5988     0.0000     TRUE
3   F(fam.31)-F(fam.mel)   -4.0173     0.0023     TRUE
4   F(fam.41)-F(fam.mel)    0.0756     0.9403    FALSE
5  F(fam.62a)-F(fam.mel)   -2.4670     0.0393     TRUE
6  F(fam.73a)-F(fam.mel)   -2.5930     0.0393     TRUE
7   F(fam.87)-F(fam.mel)   -2.6294     0.0393     TRUE
8   F(fam.9к)-F(fam.mel)   -3.2085     0.0160     TRUE
```

На этот раз значимые отличия от «контроля» выявлены для 7 пар из 8.

Глава 7. Простая регрессия и корреляция

Основной целью регрессионного анализа является изучение формы зависимости между двумя величинами. В простейшем случае изучается линейная зависимость одной количественной переменной от другой. Уравнение регрессии позволяет прогнозировать значения одной переменной по значениям другой.

7.1. Основы регрессионного анализа

Набор стандартных морфометрических признаков пчел, с которым мы познакомились в предыдущих главах, включает четыре признака крыла (длина и ширина, кубитальный индекс, дискоидальное смещение), два признака груди (длина и ширина тергита) и длину хоботка. За исключением кубитального индекса и дискоидального смещения, которые описывают жилкование и отражают форму крыла, все признаки являются размерными характеристиками. Довольно логично предположить, что они так или иначе связаны между собой, ведь крупные пчелы должны иметь крупные крылья и крупную грудь. Менее очевидна связь размеров тела с длиной хоботка, но этот вопрос тоже может являться предметом анализа. Поставим конкретный вопрос: как связаны длина крыла и длина тергита? Или еще более конкретно: можно ли по длине крыла определить длину тергита? Ответы на эти вопросы дает уравнение регрессии.

Рассмотрим следующую регрессионную модель:

$$y_i = \alpha + \beta \cdot x_i + \varepsilon_i .$$

Переменная Y называется зависимой, а переменная X – независимой. y_i и x_i – конкретные значения зависимой и независимой переменных у каждого из наблюдаемых объектов, ε_i – случайная ошибка, подчиняющаяся нормальному распределению с нулевым средним и дисперсией σ^2, α и β – коэффициенты регрессии. Регрессионная модель состоит из двух частей. Детерминированная часть описывает линейную зависимость между переменными и, в нашем случае, соответствует уравнению прямой: $y = a + bx$. Стохастическая часть описывает характер случайных отклонений от этой прямой.

Первой задачей регрессионного анализа является оценка параметров регрессионной модели α, β и σ^2. Наиболее простым и эффективным способом оценки является *метод наименьших квадратов*, который минимизирует сумму квадратов отклонений от линии регрессии:

$$SS_{res} = \sum_i \left(y_i - (\alpha + \beta \cdot x_i)\right)^2 .$$

Оценка коэффициентов α и β (выборочные оценки мы будем обозначать латинскими буквами a и b) производится в соответствии со следующими выражениями:

$$b = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2},$$

$$a = \bar{y} - b\bar{x} .$$

Выборочная оценка остаточной дисперсии σ^2 производится на основе суммы квадратов SS_{res} и объема выборки n:

$$s^2 = \frac{SS_{res}}{n-2} .$$

Также как и выборочные оценки любых других генеральных параметров (средних, стандартных отклонений и т.п.), оценки параметров регрессии подвержены случайной изменчивости. Разные выборки из одной и той же генеральной совокупности будут давать отличающиеся значения оценок. Степень разброса оценок вокруг истинного значения параметра характеризует стандартная ошибка (см. раздел 4.1). Так же как можно рассчитать стандартную ошибку среднего по выборочным данным, можно рассчитать стандартные ошибки коэффициентов регрессии и использовать их для построения доверительных интервалов и проверки разного рода гипотез.

Чаще всего исследователя в первую очередь интересует вопрос о наличии связи между переменными. Соответствующая нуль-гипотеза имеет вид:

$$H_0: \beta = 0.$$

Если $\beta = 0$, регрессионная модель описывает горизонтальную прямую, то есть значения Y никак не зависят от X. Для проверки этой гипотезы можно использовать величину t, которая вычисляется как отношение оценки коэффициента b к ее стандартной ошибке s_b

$$t = \frac{b}{s_b}$$

и в условиях нуль-гипотезы имеет распределение Стьюдента с $n - 2$ степенями свободы. На основе этого распределения легко вычислить вероятность получения такого же или еще более экстремального значения t (т.е. p-значение) и принять решение относительно значимости коэффициента β. Аналогичную процедуру можно применить для проверки гипотезы о том, что линия регрессии проходит через начало координат ($H_0: \alpha = 0$), однако для большинства случаев это не самая осмысленная гипотеза (либо потому что нет причин предполагать

такое поведение модели, либо просто потому, что диапазон варьирования независимой переменной далек от нуля).

Возвращаясь к примеру с анализом зависимости длины тергита от длины крыла, нас в первую очередь будет интересовать оценка уравнения регрессии

$$\texttt{tergit.length} = \alpha + \beta \cdot \texttt{wing.length},$$

а также проверка гипотезы о равенстве нулю коэффициента β (то есть об отсутствии связи между длинами).

7.2. Построение и анализ линейной модели с помощью функции `lm()`

Для построения линейных (в том числе – регрессионных) моделей служит функция `lm()`, которой передается формула. Построим модель зависимости длины тергита от длины крыла пчел породы *A. m. mellifera*.

```
> load("apis.rda")
> mel <- subset(apis, subset = breed == "mellifera")
> reg.model <- lm(tergit.length ~ wing.length, data = mel)
```

Мы загрузили данные по пчелам, выбрали наблюдения по породе *A. m. mellifera* в отдельный фрейм `mel` и построили линейную модель `reg.model`. Структура использованной нами формулы аналогична использованным ранее: слева от тильды (~) стоит имя зависимой переменной (`tergit.length`), справа – имя независимой переменной (`wing.length`), сама тильда означает, что необходимо построить модель зависимости одной переменной от другой, аргумент `data = mel` указывает, в каком фрейме находятся переменные. Основное отличие от использованных нами ранее формул заключается в том, что переменная, стоящая справа от тильды, сама по себе является числовой, поэтому она не задает принадлежность наблюдений к выборкам (как в случае текстовых переменных или факторов), а используется для моделирования зависимой переменной.

Функция `lm()` возвращает объект класса «`lm`» – список из 12 элементов, в котором содержится большое количество информации, включая исходные данные. При выводе этого объекта в консоль, мы увидим отчет, который содержит минимум информации:

```
> reg.model
Call:
lm(formula = tergit.length ~ wing.length, data = mel)
Coefficients:
(Intercept)  wing.length
     1.0289       0.1526
```

Мы видим только структуру модели, т.е. повторение выражения, с помощью которого создана модель, а также полученные оценки коэффициентов. Свободный коэффициент *a* приводится первым: при выводе в консоль он обозначается термином `intercept` (т.е. пересечение с осью ординат), используемым в англоязычной литературе. Коэффициенты при независимых переменных обозначаются именами этих переменных, в нашем случае оценка коэффициента *b* находится под именем `wing.length`. Таким образом, мы получили искомое уравнение регрессии:

$$\texttt{tergit.length} = 1.0289 + 0.1526 \cdot \texttt{wing.length}.$$

Стандартная интерпретация свободного коэффициента *a* (он так называется, поскольку «свободен» от связи с независимыми переменными) заключается в том, что он определяет значение *Y* при нулевом значении *X*. В нашем случае эта интерпретация не имеет смысла, поскольку нулевая длина крыла невозможна (бескрылые пчелы, конечно же, существуют, но это слишком большие генетические отклонения, чтобы ожидать того, что наша простая регрессионная модель подошла бы для их описания). Коэффициент наклона *b* интерпретируется как изменение зависимой переменной в расчете на единицу независимой переменной. В нашем случае $b = 0.15$ означает, что при изменении длины крыла на 1 мм следует ожидать изменение длины тергита на 0.15 мм.

Для получения более подробной информации нужно воспользоваться специальными функциями. Рассмотрим результаты применения к модельному объекту функции `summary()`:

```
> summary(reg.model)

Call:
lm(formula = tergit.length ~ wing.length, data = mel)

Residuals:
      Min        1Q    Median        3Q       Max
-0.159492 -0.036180  0.001772  0.043407  0.171528

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.0289     0.3483   2.954  0.00382 **
wing.length   0.1526     0.0369   4.136 6.85e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.06035 on 112 degrees of freedom
Multiple R-squared:  0.1325, Adjusted R-squared:  0.1248
F-statistic: 17.11 on 1 and 112 DF,  p-value: 6.851e-05
```

Разберем представленные результаты по блокам выводимой информации.

```
Call:
lm(formula = tergit.length ~ wing.length, data = mel)
```

Как и при использовании многих других функций, сначала выдается описание данных, в данном случае оно эквивалентно выражению, которым создана линейная модель.

```
Residuals:
      Min        1Q    Median        3Q       Max
-0.159492 -0.036180  0.001772  0.043407  0.171528
```

Здесь приводится краткое описание отклонений от линейной регрессии, которые называются остатками (`Residuals`). В частности, приведен стандартный набор квартилей, которые дают представление о распределении остатков. Среднее значение остатков по определению равно нулю, поэтому близко к нулю и значение медианы. Остатки должны быть распределены симметрично вокруг нуля, поэтому близки абсолютные значения минимума и максимума, а также первого и третьего квартилей.

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.0289     0.3483   2.954  0.00382 **
wing.length   0.1526     0.0369   4.136 6.85e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Здесь мы видим таблицу, напоминающую таблицу дисперсионного анализа. Строки соответствуют компонентам модели (в данном случае – свободный член `Intercept` и длина крыла `wing.length`). В столбцах содержатся оценки коэффициентов (`Estimate`), их стандартные ошибки (`Std. Error`), значения статистики t (`t value`) и соответствующие p-значения (`Pr(>|t|)`). Коэффициенты, значимо отличающиеся от нуля, получают обозначения в виде точек и звездочек, расшифровка которых содержится под таблицей (`Signif. codes`). В нашем случае от нуля значимо отличны оба коэффициента.

```
Residual standard error: 0.06035 on 112 degrees of freedom
```

Это оценка остаточного стандартного отклонения s, которое является одним из параметров модели и описывает разброс остатков вокруг линии регрессии. Для него приведено число степеней свободы.

```
Multiple R-squared:  0.1325, Adjusted R-squared:  0.1248
```

Первый элемент (`Multiple R-squared`) – это коэффициент детерминации R^2, который в случае простой линейной регрессии является квадратом коэффициента корреляции Пирсона (см. раздел 7.6). Он имеет важную интерпретацию – это доля изменчивости зависимой переменной, которая

объясняется регрессионной моделью, то есть в определенном смысле коэффициент детерминации описывает качество модели. Коэффициент детерминации часто используется для сравнения моделей (чем он больше, тем модель лучше), но применительно к этой задаче он обладает существенным недостатком: его значение всегда увеличивается (не уменьшается) при добавлении в модель новых переменных, даже если эти переменные никакого отношения к объясняемой переменной не имеют. При селекции моделей применяется скорректированный коэффициент детерминации (`Adjusted R-squared`), который содержит поправку на число параметров и поэтому его нельзя в полной мере интерпретировать как долю объясненной дисперсии. Этот показатель всегда меньше единицы, но теоретически может быть и меньше нуля (при очень маленьком значении обычного коэффициента детерминации и большом количестве параметров).

```
F-statistic: 17.11 on 1 and 112 DF,  p-value: 6.851e-05
```

В последней строке содержится результат проверки нуль-гипотезы о равенстве нулю всех коэффициентов регрессии за исключением свободного коэффициента с помощью критериальной статистики *F*, которая подчиняется распределению Фишера. В нашем случае такой коэффициент один и результат применения *F*-критерия Фишера эквивалентен результату *t*-критерия Стьюдента (обратите внимание на точное совпадение *p*-значений).

7.3. Остатки и прогнозируемые значения

В объекте, создаваемом функцией `lm()`, хранится большое количество информации, в частности – прогнозируемые значения и остатки. Прогнозируемые значения – это значения зависимой переменной, вычисленные в соответствии с моделью в точках, соответствующих значениям независимой переменной в исходных данных. Они содержатся в элементе с именем `fitted.values`. Получить к ним доступ можно с помощью оператора `$`, а также с помощью специальной функции `fitted()`.

```
> model.fit <- reg.model$fitted.values
> model.fit <- fitted(reg.model)
```

Приведенные варианты эквивалентны.

Остатки представляют собой отклонения значений зависимой переменной от соответствующих прогнозов. Они содержатся в элементе с именем `residuals`, для доступа к ним можно также воспользоваться функцией `resid()`.

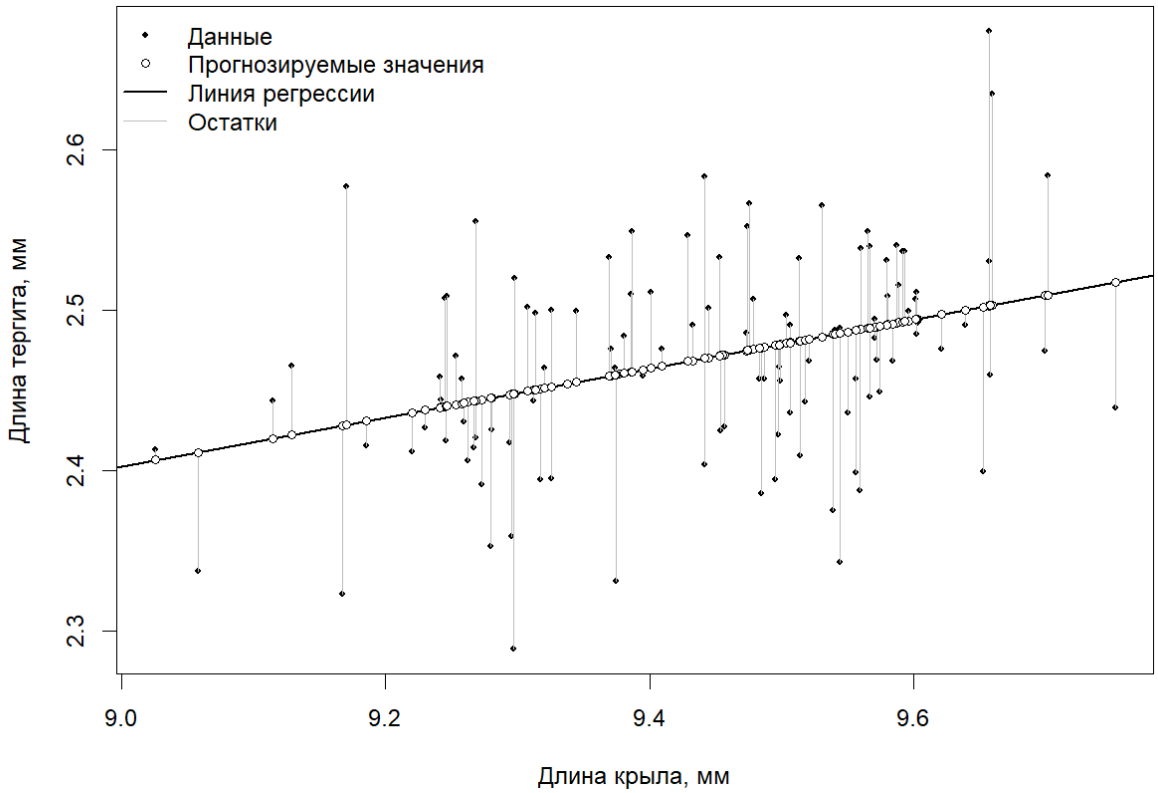

Рис. 21. Зависимость длины тергита от длины крыла с линией регрессии и остатками

```
> model.res <- reg.model$residuals
> model.res <- resid(reg.model)
```

Изобразим теперь все имеющие элементы на графике. Первым делом отображаются исходные данные. Такой график называют диаграммой рассеяния.

```
> plot(tergit.length ~ wing.length, data = mel,  pch = 20, cex = .8,
+         ylim = c(2.28, 2.7),
+         xlab = "Длина крыла, мм", ylab = "Длина тергита, мм")
```

Обратите внимание, что в данном случае мы воспользовались формулой. Функция `plot()` допускает такую форму вызова, если обе переменные являются числовыми и имеют одинаковую длину. Мы также расширили диапазон значений по ординате, чтобы оставить немного дополнительного места для легенды.

Линию регрессии проще всего добавить на график с помощью функции `abline()`, которой можно передать коэффициенты уравнения регрессии, однако удобнее передать линейную модель и функция сама извлечет из нее необходимую информацию.

```
> abline(reg.model, lwd = 2)
```

Прогнозируемые значения добавим с помощью функции `points()`, а остатки изобразим в виде отрезков с помощью функции `segments()`, которой передаются

вектора с координатами точек, которые необходимо изобразить в виде сегментов.

```
> points(mel$wing.length, model.fit, pch = 21, cex = .8, bg = "white")
> segments(mel$wing.length, mel$tergit.length,
+             mel$wing.length, model.fit, col = "grey")
```

Наконец, добавим легенду. Нам нужно использовать как точечные, так и линейные объекты, поэтому мы используем аргументы `pch` и `lwd` одновременно. Чтобы не отображалась линия для точечных объектов, для них нужно установить неопределенную толщину, то есть соответствующий элемент вектора, который передается аргументу `lwd`, должен иметь значение `NA`. Аналогичным образом следует поступить с типом символа для линейных объектов.

```
> legend("topleft", legend = c("Данные", "Прогнозируемые значения",
+                                 "Линия регрессии", "Остатки"),
+             col = c("black", "black", "black", "grey"),
+            pch = c(20, 21, NA, NA), pt.bg = c(NA, "white", NA, NA),
+             pt.cex = .8, lwd = c(NA, NA, 2, 1), bty = "n")
```

Для того чтобы не отображать рамку вокруг легенды, мы воспользовались аргументом `bty = "n"`. Итоговый график представлен на Рис. 21.

7.4. Прогнозы и доверительные интервалы

Коэффициенты регрессии, рассчитанные по выборочным данным, характеризуются неопределенностью, которая отражена в их стандартных ошибках. Графически эта неопределенность отображается в виде доверительных интервалов. В регрессионном анализе используются три вида доверительных интервалов: для коэффициентов регрессии, для линии регрессии и для прогнозируемых значений.

Доверительные интервалы для коэффициентов регрессии строятся с помощью функции `confint()`, которой передается линейная модель. По умолчанию строится 95 %-ный доверительный интервал для всех коэффициентов.

```
> confint(reg.model)
                 2.5 %    97.5 %
(Intercept) 0.33882495 1.7188835
wing.length 0.07952091 0.2257536
```

Функция `confint()` возвращает матрицу, строки которой соответствуют коэффициентам (они имеют соответствующие имена), а в столбцах располагаются нижняя и верхняя границы доверительного интервала. Если есть

необходимость получить интервальную оценку только какого-то конкретного коэффициента, его можно указать с помощью аргумента `parm`, которому можно передать имя параметра либо его номер (первым считается свободный член). Следующие два вызова эквивалентны:

```
> confint(reg.model, parm = 1)
                2.5 %   97.5 %
(Intercept) 0.3388249 1.718884
> confint(reg.model, parm = "(Intercept)")
                2.5 %   97.5 %
(Intercept) 0.3388249 1.718884
```

Если необходимо указать иную доверительную вероятность, то нужно воспользоваться аргументом `level`. Например, доверительный интервал для наклона линии регрессии можно рассчитать так:

```
> confint(reg.model, parm = "wing.length", level = 0.99)
                 0.5 %    99.5 %
wing.length 0.05593809 0.2493364
```

Доверительный интервал для линии регрессии отражает неопределённость оценки регрессионной модели: для заданного значения независимой переменной *X* истинная линия регрессии находится в пределах доверительного интервала с заданной вероятностью. Чем больше выборка, по которой строилась модель, тем точнее оценка параметров и тем у́же доверительный интервал для линии регрессии. Границы доверительного интервала для линии регрессии обладают отчетливой нелинейностью: чем дальше от центра, тем шире интервал.

Доверительный интервал для прогнозируемых значений отражает неопределенность регрессионной модели по отношению к новым данным: прогнозы с заданной вероятностью должны попадать в оцениваемый интервал. Ширина этого интервала гораздо меньше зависит от объема выборки, поскольку она определяется в первую очередь остаточной дисперсией.

Для расчета прогнозируемых значений и доверительных интервалов предназначена функция `predict()`. Этой функции необходимо передать модель, а также фрейм, в котором содержатся значения независимой переменной, для которых необходимо провести расчеты. Так, если мы хотим получить прогноз длины тергита для пчелы с длиной крыла, равной 9.5 мм, мы можем поступить следующим образом:

```
> predict(reg.model, newdata = data.frame(wing.length = 9.5))
       1
2.478908
```

Обратите внимание, что аргументу `newdata` обязательно должен быть передан фрейм, в котором есть переменная, которая была использована при

спецификации модели (в данном случае – wing.length). В приведенном примере мы создали такой фрейм прямо при вызове функции, этот фрейм содержит единственное значение, для которого мы хотели получить прогноз. Если аргумент newdata не использован, то прогнозы и интервалы будут рассчитаны для значений независимой переменной в исходных данных.

Для получения доверительных интервалов нужно воспользоваться аргументом interval, которому передаются либо значение «confidence» в случае доверительного интервала для линии регрессии (допускается также сокращенный вариант «c»), либо значение «prediction» в случае доверительного интервала для прогнозируемых значений (допускается также сокращенный вариант «p»).

Построим график с доверительными интервалами. Прежде всего необходимо создать фрейм со значениями длины крыла, для которых нам нужны доверительные интервалы. В целях визуализации можно просто взять диапазон значений длины крыла в выборке.

```
> pred.frame <- data.frame(wing.length = seq(min(mel$wing.length),
+                                max(mel$wing.length), length = 100))
```

Затем нужно рассчитать сами доверительные интервалы.

```
> conf.line <- predict(reg.model, int = "c", newdata = pred.frame)
> conf.data <- predict(reg.model, int = "p", newdata = pred.frame)
```

Функция predict() возвращает матрицу, в столбцах которой содержатся прогнозы, нижняя и верхняя границы доверительного интервала, а строки соответствуют значениям независимой переменной из фрейма newdata.

```
> plot(tergit.length ~ wing.length, data = mel,  pch = 20, cex = .8,
+      ylim = c(2.28, 2.7),
+      xlab = "Длина крыла, мм", ylab = "Длина тергита, мм")
> abline(reg.model, lwd = 2)
> lines(pred.frame$wing.length, conf.line[, 2],
+       lty = 2, lwd = 2, col = "darkgrey")
> lines(pred.frame$wing.length, conf.line[, 3],
+       lty = 2, lwd = 2, col = "darkgrey")
> lines(pred.frame$wing.length, conf.data[, 2],
+       lty = 5, lwd = 2, col = "darkgrey")
> lines(pred.frame$wing.length, conf.data[, 3],
+       lty = 5, lwd = 2, col = "darkgrey")
> legend("topleft", legend = c("Данные", "Линия регрессии",
+                   "Доверительный интервал для линии регрессии",
+                   "Доверительный интервал для новых данных"),
+        col = c("black", "black", "darkgrey", "darkgrey"),
+        pch = c(20, NA, NA, NA), lty = c(NA, 1, 2, 5),
+        pt.cex = .8, lwd = c(NA, 2, 2, 2), bty = "n")
```

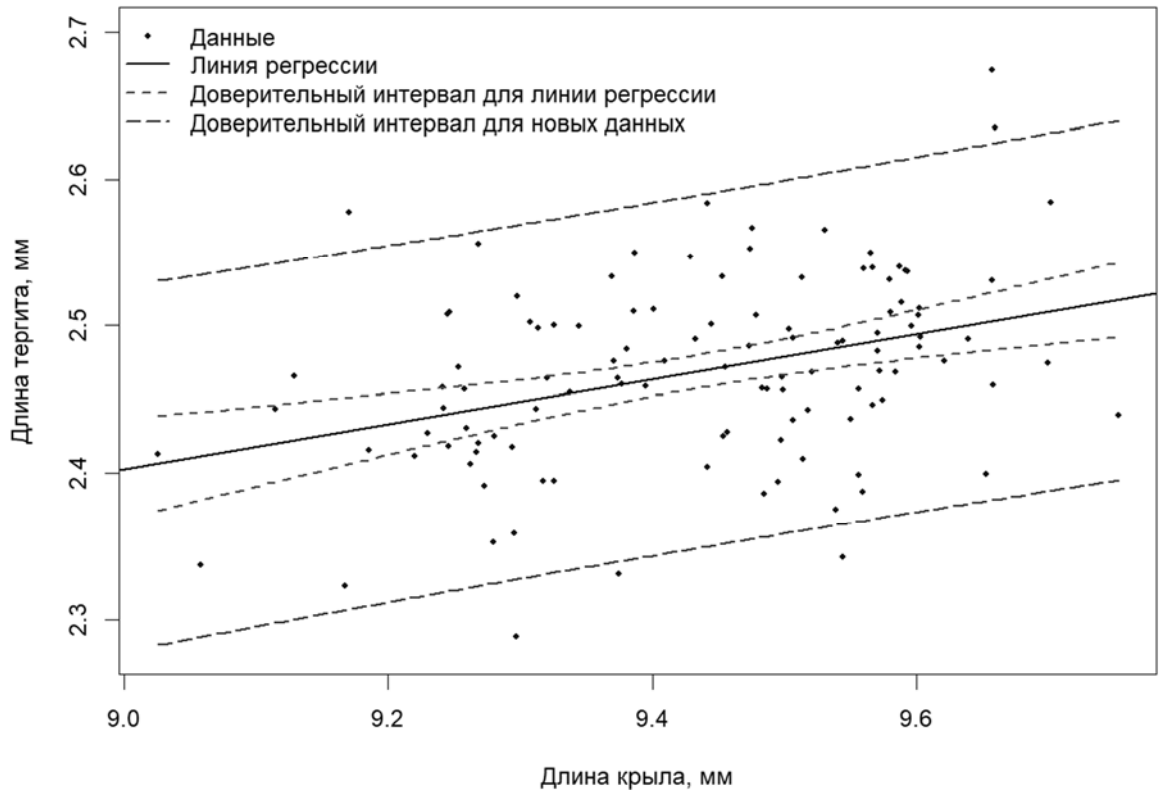

Рис. 22. Зависимость длины тергита от длины крыла с линией регрессии и доверительными интервалами

Итоговый график представлен на Рис. 22. Обратите внимание, что шесть точек лежат за пределами доверительного интервала для значений. Это нормально, поскольку в этом доверительном интервале должно содержаться 95 % значений (причем тех значений, которые будут получены в будущем, то есть не на тех данных, по которым строилась модель).

7.5. Предположения регрессионного анализа и их диагностика

Рассматриваемый в настоящей главе вид регрессионного анализа основывается на ряде предположений, касающихся характера распределения и типа взаимосвязи между переменными. Основные предположения следующие:

1) независимость наблюдений – данные в выборке не должны быть связаны между собой, т.е. должны представлять собой случайную выборку из некой генеральной совокупности;
2) линейность связи – зависимость между переменными должна иметь существенно линейный характер; если это условие не соблюдается, то прогнозы модели могут оказаться неадекватными вплоть до полной

бесполезности; в этом случае необходимо перейти от простой линейной регрессии к модели, описывающей в той или иной форме нелинейность связи;

3) нормальность распределения остатков – важнейшим компонентом теоретической регрессионной модели являются случайные ошибки ε_i, которые подчиняются нормальному распределению (см. раздел 16.1); в эмпирических данных случайным ошибкам соответствуют остатки, которые также должны подчиняться нормальному распределению;
4) гомогенность дисперсии (гомоскедастичность) – подразумевается, что дисперсия распределения остатков не зависит от значений независимой переменной (то есть разброс значений вокруг линии регрессии примерно одинаков как при низких значениях X, так и при высоких); отклонение от этого предположения существенно влияет на доверительные интервалы и результаты проверки гипотез.

Первое предположение относится к природе данных и никак не может быть проверено, исходя из самих численных значений переменных. Выполнение этого предположения должно гарантироваться структурой исследования.

Последние три предположения могут быть в той или иной степени проверены при анализе построенной модели. Для такой диагностики в R реализовано множество визуальных средств, помогающих получить представление о выполнении предположений. Эти графики можно получить с использованием базовой функции `plot()`, которой передается линейная модель. Таких диагностических графиков шесть, выбор конкретного осуществляется с помощью аргумента `which`, которому передается номер от 1 до 6. Мы разберем первые 4 графика.

```
> opar <- par(mfrow = c(2,2))
> plot(reg.model, which = 1, main = "(a)")
> plot(reg.model, which = 2, main = "(b)")
> plot(reg.model, which = 3, main = "(c)")
> plot(reg.model, which = 4, main = "(d)")
> par(opar)
```

Первый график (Рис. 23a) отображает остатки (Residuals) в зависимости от прогнозируемых значений (Fitted values). Этот график используется для диагностики линейности связи. В случае линейной зависимости остатки распределены вокруг линии регрессии без отклонений вверх или вниз. Для диагностики таких отклонений и служит рассматриваемый график. Если зависимость линейна, связь остатков и прогнозируемых значений отсутствует. В нашем случае остатки распределены вокруг нуля (серая горизонтальная линия), а сглаживающая кривая этой зависимости лежит практически горизонтально, что

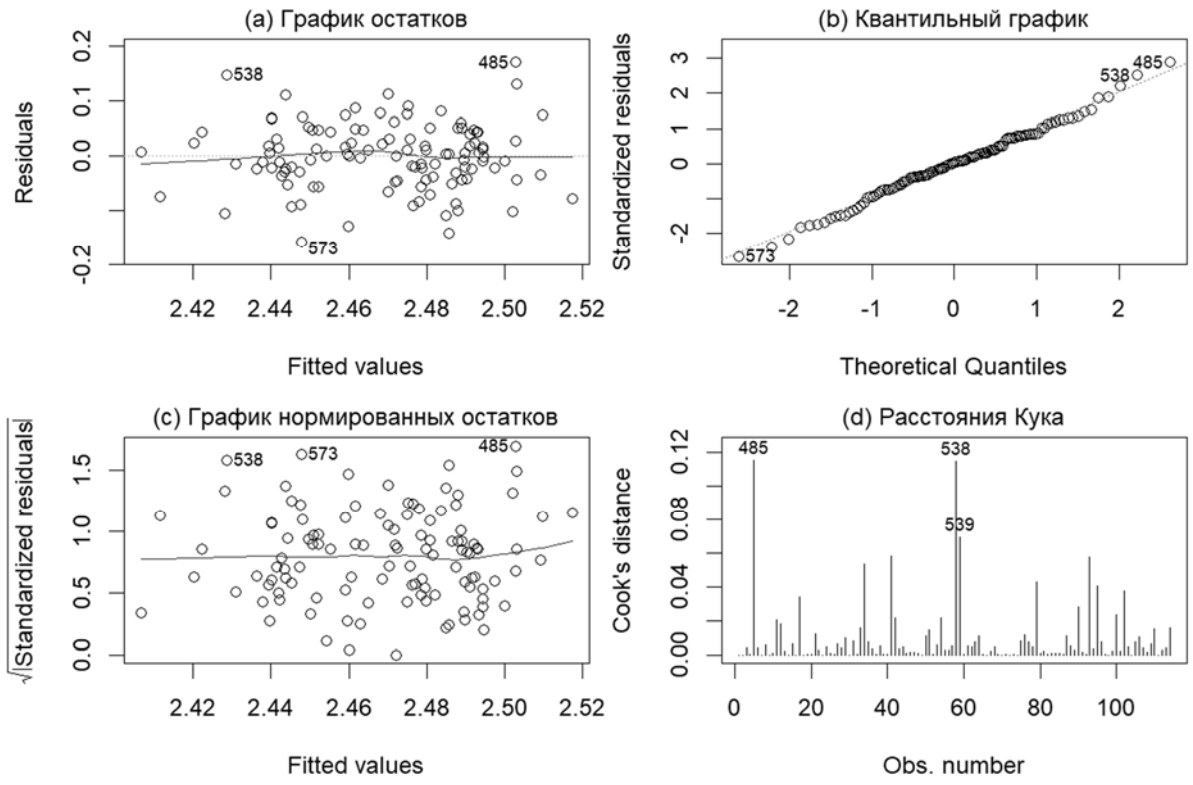

Рис. 23. Диагностические графики для регрессии
зависимости длины тергита от длины крыла

свидетельствует об отсутствии связи. Разумеется, такой простой визуальный анализ можно провести и с помощью обычной диаграммы рассеяния (Рис. 21 или Рис. 22), однако сглаживающая кривая может оказать существенную помощь. Кроме того, график зависимости остатков от прогнозируемых значений не имеет простых альтернатив в случае более сложных моделей, когда рассматривается несколько независимых переменных (см. главу 8).

При использовании функции `plot()` и линейной модели для создания диагностических графиков отсутствует возможность определения подписей самих графиков и осей, поэтому мы будем приводить код, воспроизводящий эти графики (сами полученные графики мало отличаются от Рис. 23, поэтому мы их приводить не будем). График зависимости остатков от прогнозируемых значений можно воспроизвести следующим образом:

```
> plot(fitted(reg.model), resid(reg.model))
> abline(h = 0, col = "grey", lty = 3)
> panel.smooth(fitted(reg.model), resid(reg.model))
```

Второй график (Рис. 23b) используется для диагностики отклонений от нормальности остатков и представляет собой обычный квантильный график, построенный для остатков модели. Чем точнее точки ложатся на прямую серую

линию, тем ближе распределение остатков к нормальности. В нашем случае существенных отклонений мы не наблюдаем.

Воспроизвести квантильный график для остатков можно следующим образом:

```
> qqnorm(resid(reg.model))
> qqline(resid(reg.model), lty = 3, col = "grey")
```

Для диагностики отклонений остатков регрессионной модели от нормальности можно также воспользоваться критериями согласия Шапиро-Уилка, Лиллиефорса и Андерсона-Дарлинга (см. раздел 3.3):

```
> shapiro.test(resid(reg.model))

Shapiro-Wilk normality test

data:  resid(reg.model)
W = 0.994, p-value = 0.9056
> library(nortest)
> lillie.test(resid(reg.model))

Lilliefors (Kolmogorov-Smirnov) normality test

data:  resid(reg.model)
D = 0.0527, p-value = 0.6115
> ad.test(resid(reg.model))

Anderson-Darling normality test

data:  resid(reg.model)
A = 0.2602, p-value = 0.7049
```

Ни один из критериев согласия значимых отклонений от нормальности не обнаружил (во всех трех случаях $p > 0.05$), что согласуется с результатами визуального анализа.

Третий график (Рис. 23c) используется для диагностики гомогенности дисперсии остатков. Он отображает зависимость квадратного корня из модуля нормированных остатков (√|Standardized residuals|) от прогнозируемых значений (Fitted values). В случае гомогенности дисперсии эта зависимость должна отсутствовать, что хорошо видно в нашем случае, поскольку сглаживающая кривая (красная линия) лежит практически горизонтально. Стандартизованные остатки, нормированные таким образом, чтобы их дисперсия была равна единице, можно получить с помощью функции `rstandard()`, которой передается линейная модель. Воспроизвести этот график можно следующим образом:

```
> plot(fitted(reg.model), sqrt(abs(rstandard(reg.model))))
> panel.smooth(fitted(reg.model), sqrt(abs(rstandard(reg.model))))
```

Помимо визуального анализа гомогенности дисперсии можно воспользоваться специальным статистическим критерием Бройша-Пагана (Breusch-Pagan test), который применяется для анализа гомоскедастичности в регрессионных моделях. Этот критерий реализован в функции `ncvTest()` пакета `car`, которой необходимо передать линейную модель.

```
> library(car)
> ncvTest(reg.model)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 0.05113717    Df = 1     p = 0.8210962
```

Критерий Бройша-Пагана проверяет нуль-гипотезу о гомоскедастичности модели, поэтому в нашем случае ($p = 0.82$) гетерогенность дисперсии остатков не выявлена, что согласуется с результатами визуального анализа.

Четвертый график (Рис. 23d) используется для выявления так называемых влиятельных наблюдений – это такие наблюдения, которые оказывают значительное влияние на параметры модели. При удалении таких наблюдений параметры модели меняются существенным образом. Часто это наиболее сильно отклоняющиеся наблюдения. В любом случае на такие наблюдения следует обратить внимание и рассмотреть вопрос об их исключении. Мерой влиятельности наблюдений является расстояние Кука (Cook's distance):

$$D_i = \frac{\sum_{j=1}^{n} \left(\hat{Y}_j - \hat{Y}_{j(i)}\right)^2}{k \cdot s^2},$$

где $\hat{Y}_j$ – прогноз полной регрессионной модели для наблюдения j, $\hat{Y}_{j(i)}$ – прогноз для наблюдения j в регрессионной модели, построенной после исключения наблюдения i, k – число параметров модели, s^2 – остаточная дисперсия. Таким образом, расстояние Кука измеряет эффект исключения данного наблюдения.

На графике расстояния Кука отображены в виде вертикальных линий, по горизонтали отложены номера наблюдений (Obs. number). Существуют разные мнения относительно того, какие наблюдения следует считать влиятельными. Сам автор меры расстояния считал, что влиятельными нужно считать наблюдения с $D_i > 1$. Многие другие специалисты считают такой критерий слишком либеральным и предлагают рассматривать в качестве влиятельных наблюдения с $D_i > 4/(n - k)$. Третьи полагают, что не следует устанавливать численную границу влиятельности наблюдений. Вместо этого нужно просто рассмотреть совокупность расстояний Кука на графике и обратить внимание на те наблюдения, которые характеризуются сильно отклоняющимся значением расстояния Кука по сравнению с остальными наблюдениями. В нашем случае

критерию $D_i > 1$ не отвечает ни одно наблюдение, критерию $D_i > 0.036$ отвечают 9 наблюдений, при этом сильно отклоняются от остальных два наблюдения (они подписаны на всех графиках как 485-е и 538-е; имена этих наблюдений соответствуют их номерам в исходном фрейме `apis`). Отметим, что при использовании линейной модели в функции `plot()`, три наиболее уклоняющиеся наблюдения автоматически подписываются на всех графиках.

Воспроизвести график расстояний Кука можно следующим образом. Для расчета самих расстояний служит функция `cooks.distance()`, которой передается линейная модель. График строится обычной функцией `plot()` с использованием гистограммо-подобного типа отображения (`type = "h"`) без указания переменной для абсциссы (напомним, что в этом случае в качестве абсциссы используются номера наблюдений).

```
> plot(cooks.distance(reg.model), type = "h")
```

Для визуализации границ влиятельности можно добавить их на график в виде горизонтальных линий:

```
> abline(h = 1, lty = 3, col = "grey")
> abline(h = 4/(length(resid(reg.model)) -
+                  length(reg.model$coefficients)),
+                  lty = 3, col = "grey")
```

В заключение нашего обзора методов диагностики регрессионных моделей отметим, что редкая модель не обладает теми или иными отклонениями от базовых предположений. Даже в нашем примере с анализом зависимости длины тергита от длины крыла пчел породы *A. m. mellifera* выявлено несколько влиятельных наблюдений. Подчеркнем, что невыполнение того или иного предположения не является приговором. Нужно помнить, на что конкретно влияют эти предположения. Метод наименьших квадратов, с помощью которого обычно подбираются линейные модели, в основе своей является не статистическим, а геометрическим, поэтому подобрать линейную модель можно для большинства наборов данных. Даже если модель отклоняется от требований нормальности или гомоскедастичности остатков, но соответствует линейности и обладает удовлетворяющей исследователя объяснительной силой (то есть высоким коэффициентом детерминации), использовать такую модель в целях прогнозирования значений зависимой переменной вполне корректно. Однако при анализе такой модели нельзя использовать статистические критерии и доверительные интервалы. Например, могут оказаться неверными выводы относительно нуль-гипотезы об отличии от нуля коэффициента наклона.

7.6. Корреляционный анализ

Регрессионный анализ изучает зависимость одной переменной от другой путем построения модели, пригодной для прогнозирования значений зависимой переменной по значениям независимой. На практике часто встречаются ситуации, когда нет необходимости в настолько подробном анализе. Напротив, исследователя интересует только сам факт и сила зависимости между двумя количественными переменными. В таких случаях применяется корреляционный анализ.

Корреляция – это статистическая взаимосвязь между двумя количественными переменными, когда изменения значений одной переменной связаны с систематическими изменениями другой переменной. Количественной мерой такой связи является коэффициент корреляции. Он варьирует от –1 до 1, крайние значения соответствуют строгой линейной связи, а нулевое значение – отсутствию связи. Коэффициент корреляции имеет отрицательный знак, когда взаимосвязь между переменными имеет обратный характер, то есть высокие значения одной переменной соответствуют низким значениям другой и наоборот. При положительном коэффициенте корреляции связь между переменными прямая, то есть переменные имеют тенденцию иметь большие или маленькие значения одновременно. Абсолютная величина коэффициента корреляции является мерой силы связи.

Наиболее распространенным является коэффициент корреляции Пирсона:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \cdot \sum_{i=1}^{n}(y_i - \bar{y})^2}}.$$

Можно показать, что | r | меньше единицы во всех случаях, за исключением строгой линейной связи между переменными, когда по значениям одной можно с абсолютной точностью определить значение второй.

Основной целью корреляционного анализа является оценка коэффициента корреляции и анализ его отличия от нуля, то есть выявления связи между переменными как таковой. Величина

$$t = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}}$$

подчиняется распределению Стьюдента с $n - 2$ степенями свободы. Этот факт и используется для проверки нуль-гипотезы об отсутствии связи H_0: $r = 0$.

Для расчета коэффициента корреляции Пирсона в R служит функция `cor()`, которой можно передать два вектора, матрицу либо фрейм. Рассчитаем

коэффициент корреляции между длинами крыла и тергита у пчел породы *A. m. mellifera*.

```
> cor(mel$wing.length, mel$tergit.length)
[1] 0.3640272
```

Коэффициент корреляции является симметричной мерой, то есть неважно, какая переменная является первой, а какая – второй. В этом легко убедиться, поменяв вектора местами при вызове функции `cor()`:

```
> cor(mel$tergit.length, mel$wing.length)
[1] 0.3640272
```

Мы получили то же самое значение коэффициента.

Для анализа значимости корреляции служит функция `cor.test()`:

```
> cor.test(mel$wing.length, mel$tergit.length)
Pearson's product-moment correlation

data:  mel$wing.length and mel$tergit.length
t = 4.1363, df = 112, p-value = 6.851e-05
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.1930362 0.5135592
sample estimates:
      cor
0.3640272
```

В консоль выводится название коэффициента корреляции (`Pearson's product-moment correlation`), описание данных (`data`), критериальная статистика (`t`), число степеней свободы (`df`), *p*-значение (`p-value`), формулировка альтернативной гипотезы (`alternative hypothesis`), а также интервальная (`95 percent confidence interval`) и точечная (`sample estimates`) оценки коэффициента. Ключевым показателем является *p*-значение: если оно меньше 0.05, нуль-гипотеза об отсутствии связи отвергается. Обратите внимание, что полученное *p*-значение идентично значению при проверке гипотезы о равенстве нулю коэффициента наклона в регрессионном анализе (см. раздел 7.2). Это не случайно, поскольку корреляционный и регрессионный анализы тесно связаны между собой.

При необходимости рассчитать доверительный интервал коэффициента корреляции для другой доверительной вероятности, можно воспользоваться аргументом `conf.level`, например, рассчитать 99 %-ный интервал можно так:

```
> cor.test(mel$wing.length, mel$tergit.length,
+                                          conf.level = 0.99)$conf.int
[1] 0.1361823 0.5552969
attr(,"conf.level")
[1] 0.99
```

Этот интервал шире 95 %-ного.

Коэффициент корреляции Пирсона является параметром двумерного нормального распределения, поэтому все виды статистического анализа, с ним связанные, основываются на предположении о нормальном распределении изучаемых переменных. Что же делать в случае отклонений от нормальности? Для таких случаев существуют непараметрические коэффициенты корреляции Спирмена и Кенделла, которые основаны на рангах.

Коэффициент корреляции Спирмена рассчитывается по формуле коэффициента корреляции Пирсона после замены исходных данных рангами. Его можно рассчитать с использованием функции `cor()` и аргумента `method = "spearman"` (по умолчанию `method = "pearson"`):

```
> cor(mel$wing.length, mel$tergit.length, method = "spearman")
[1] 0.3442769
```

Аналогичным образом, анализ отличия коэффициента корреляции Спирмена от нуля можно провести с использованием функции `cor.test()`:

```
> cor.test(mel$wing.length, mel$tergit.length, method = "spearman")

Spearman's rank correlation rho

data:  mel$wing.length and mel$tergit.length
S = 161901.3, p-value = 0.0001763
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.3442769

Warning message:
In cor.test.default(mel$wing.length, mel$tergit.length, method = "spearman"):
  Cannot compute exact p-value with ties
```

Структура вывода аналогична, за исключением того, что отсутствует интервальная оценка коэффициента, а в конце выводится предупреждение о том, что точное *p*-значение вычислить нельзя ввиду наличия совпадающих значений, вместо него используется аппроксимация. В данном случае $p < 0.05$, поэтому гипотезу о равенстве коэффициента корреляции Спирмена нулю можно отвергнуть.

Коэффициент корреляции Кенделла основан на расчете числа согласованных и несогласованных пар наблюдений. Согласованная пара – это такая, в которой разницы значений по двум переменным имеют одинаковый знак. В случае монотонной зависимости все пары будут либо согласованными

(если связь прямая) либо несогласованными (если связь обратная). В случае независимости переменных число согласованных пар будет примерно равно числу несогласованных пар. Коэффициент корреляции Кенделла рассчитывается как

$$\tau = 1 - \frac{4 \cdot R}{n \cdot (n-1)},$$

где R – число несогласованных пар. Расчет коэффициента корреляции Кенделла интегрирован в функцию `cor()`, в которой аргументу `method` нужно передать значение «`kendall`»:

```
> cor(mel$wing.length, mel$tergit.length, method = "kendall")
[1] 0.2287622
```

Численное значение коэффициента корреляции Кенделла меньше по сравнению с коэффициентами Пирсона и Спирмена. Анализ коэффициента корреляции Кенделла производится аналогично двум предыдущим с использованием функции `cor.test()`:

```
> cor.test(mel$wing.length, mel$tergit.length, method = "kendall")
Kendall's rank correlation tau
data:  mel$wing.length and mel$tergit.length
z = 3.6072, p-value = 0.0003095
alternative hypothesis: true tau is not equal to 0
sample estimates:
      tau
0.2287622
```

Результаты анализа коэффициента корреляции Кенделла свидетельствуют о значимости корреляции.

В заключение краткого обзора регрессионного и корреляционного анализов отметим, что факт выявления статистически значимой связи между двумя переменными ничего не говорит о природе этой связи. Часто у исследователя возникает соблазн объявить о выявлении причинно-следственной связи на основании регрессионного анализа. Однако регрессия и корреляция просто не предназначены для такого рода выводов. Типична ситуация, когда две изученные переменные определяются каким-то третьим фактором, в анализе не рассмотренном. В нашем примере с анализом зависимости длины тергита от длины крыла мы не можем утверждать, что длина тергита определяется длиной крыла, для такого вывода нет оснований. Мы можем прогнозировать значения длины тергита по длине крыла, но это не значит, что между ними есть причинно-следственная связь. Оба эти признака зависят от генетической основы, которая определяет общие размеры особей пчел. Однако в нашем исследовании количественные данные об этой генетической основе не рассматривались.

Глава 8. Анализ комплекса признаков

Во многих случаях исследователь имеет дело не с одной переменной, а сразу с комплексом признаков (переменных), которые измеряются у всех изучаемых объектов. Именно такая ситуация имеет место в примере с исследованием морфологической изменчивости пчел: у каждой особи измерялся комплекс из 7 признаков (6 количественных и один качественный). В предыдущей главе мы рассматривали статистические методы изучения зависимости между двумя признаками. Но всего у нас семь признаков. В данной главе мы рассмотрим методы анализа корреляции в комплексе признаков, а также методы изучения зависимости одной зависимой переменной от комплекса независимых переменных.

8.1. Анализ матрицы корреляций

При наличии комплекса признаков обычно первым вопросом является выявление тех из них, которые связаны между собой. Можно провести анализ корреляции в парах признаков с использованием методов, описанных в разделе 7.6, однако здесь мы сталкиваемся с уже знакомой нам проблемой множественных сравнений. Напомним, что при проведении попарных сравнений средних для совокупности выборок увеличивается вероятность ошибки первого рода. Это происходит вследствие того, что множество нуль-гипотез проверяется с использованием одного и того же набора данных (см. главу 6). Аналогичный рост вероятности ошибки первого рода происходит и при анализе попарных корреляций в комплексе признаков. К счастью, универсальные методы решения проблемы множественных сравнений, такие как поправки Бонферрони и Холма, подходят и для данного случая.

В качестве примера проанализируем корреляцию в комплексе из 5 количественных признаков пчел породы *A. m. mellifera*, которые подчиняются нормальному распределению, оставив пока в стороне длину хоботка, для которой выявлены отклонения.

```
> load("apis.rda")
> mel <- subset(apis, subset = breed == "mellifera")
```

Начнем мы с визуализации данных по комплексу признаков. Средством изображения зависимости двух переменных служит диаграмма рассеяния, на которой значения переменных откладываются по абсциссе и ординате, а сами наблюдения отображаются в виде облака точек. При работе с комплексом

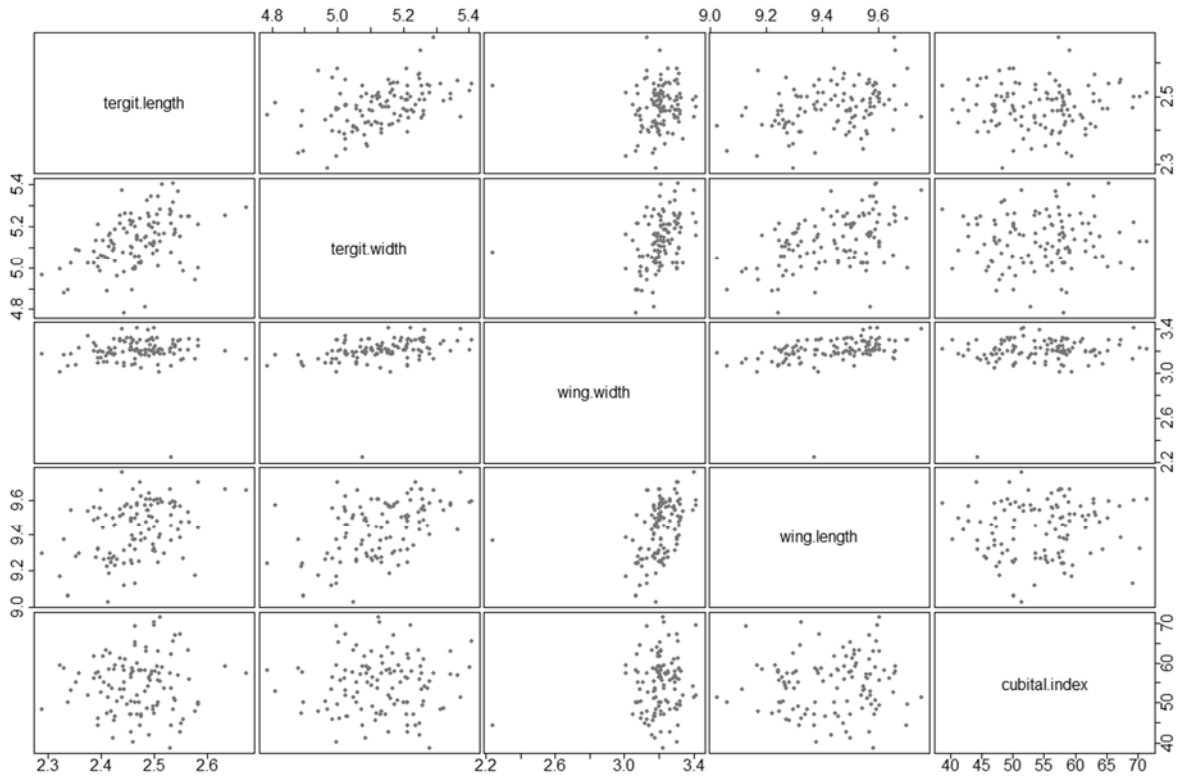

Рис. 24. Матрица диаграмм рассеяния для комплекса из 5 количественных морфологических признаков пчел

признаков таких диаграмм рассеяния будет множество. Для построения матрицы диаграмм рассеяния в R существует специальная функция `pairs()`, которой передается матрица либо фрейм (переменными считаются столбцы):

```
> pairs(mel[, 2:6], gap = 0.5, pch = 16, col = "darkgrey")
```

Мы воспользовались несколькими дополнительными аргументами, чтобы изменить тип и цвет символа (`pch = 16, col = "darkgrey"`) и уменьшить расстояние между графиками (`gap = 0.5`). Результат (Рис. 24) представляет собой матрицу графиков рассеяния размером 5 × 5, строки и столбцы соответствуют признакам, названия которых отображены на главной диагонали. Так, на пересечении первой строки и четвертого столбца находится уже знакомый нам график зависимости длины тергита от длины крыла. Матрица симметрична, поэтому на пересечении четвертой строки и первого столбца находится аналогичный график, однако здесь абсцисса и ордината поменялись местами: длина крыла изображена как функция длины тергита. Визуальное обследование позволяет получить представление о характере взаимосвязи между признаками. Так, хорошо прослеживается связь между длиной крыла и длинами тергита и крыла, а также шириной тергита (облако точек вытянуто и наклонено), но не с кубитальным индексом (облако точек лежит горизонтально).

Визуальный анализ также помогает выявить сильно уклоняющиеся наблюдения. Обратите внимание на диаграммы рассеяния для ширины крыла. Одна из точек лежит очень далеко от основного облака, что заставляет рассматривать ее в качестве уклоняющегося наблюдения.

```
> (out <- min(mel$wing.width))
[1] 2.2387
> which(mel$wing.width == out)
[1] 75
```

Значение ширины крыла 75-й особи лежит далеко за пределами варьирования этого признака в выборке.

```
> c(mean(mel$wing.width[-75]), sd(mel$wing.width[-75]),
+         min(mel$wing.width[-75]))
[1] 3.21225929 0.08241636 3.01040000
```

Скорее всего, в данном случае имела место банальная ошибка при записи результатов измерения: вместо 3.2387 оператор записал 2.2387. Тем не менее, поскольку полной уверенности в таком сценарии у нас нет, мы должны исключить это измерение из анализа, присвоив ему значение NA.

```
> mel$wing.width[75] <- NA
```

Вернемся теперь к вопросу об анализе корреляции в комплексе признаков. Функция cor(), использованная нами для расчета коэффициентов корреляции, может быть использована для расчета матрицы корреляций, если передать ей матрицу либо фрейм:

```
> cor(mel[, 2:6])
              tergit.length    tergit.width    wing.width    wing.length
cubital.index
tergit.length    1.00000000   0.43785256         NA  0.36402723    0.03243577
tergit.width     0.43785256   1.00000000         NA  0.46633676    0.02896152
wing.width               NA           NA          1          NA            NA
wing.length      0.36402723   0.46633676         NA  1.00000000    0.07908013
cubital.index    0.03243577   0.02896152         NA  0.07908013    1.00000000
```

Функция возвращает симметричную матрицу корреляций с именованными строками и столбцами. Эта матрица симметрична, на главной диагонали расположены единицы. Для всех значений корреляции с шириной крыла возвращено значение NA. Это связано с тем, что одно из значений по этому признаку отсутствует (мы сами только что исключили его из анализа). Обычно в таких ситуациях в стандартных функциях используется аргумент na.rm = TRUE, указывающий, что отсутствующие значения нужно проигнорировать. Однако при работе с комплексом признаков возможны разные варианты. Например, можно игнорировать объекты, в которых хотя бы по одному из признаков имеются отсутствующие значения (так называемые неполные наблюдения), и

тогда при расчете всех коэффициентов такие наблюдения (в нашем примере 75-е) будут полностью исключены из анализа. Другой вариант – игнорировать только пары, в которых есть отсутствующие значения: в нашем примере 75-е наблюдение будет исключено из расчета коэффициентов корреляции ширины крыла, но будет участвовать в расчете корреляции между другими признаками. Для указания варианта действий используется аргумент `use`, который может получать следующие значения: «`everything`» – если в паре переменных присутствует `NA`, результатом для этой пары также будет `NA` (этот вариант используется по умолчанию), «`all.obs`» – наличие хотя бы одного отсутствующего значения приведет к выводу сообщения об ошибке, «`complete.obs`» – используются только полные наблюдения, если таковые отсутствуют, выводится сообщение об ошибке, «`na.or.complete`» – аналогично предыдущему, но при отсутствии полных наблюдений возвращается `NA`, «`pairwise.complete.obs`» – корреляция в каждой паре переменных рассчитывается с использованием всех полных пар значений. На практике обычно используются третий и пятый варианты.

Пересчитаем матрицу корреляций с максимально полным использованием информации:

```
    > cor(mel[, 2:6], use = "pairwise.complete.obs")
                tergit.length tergit.width wing.width wing.length cubital.index
tergit.length  1.00000000   0.43785256 0.16776218  0.36402723    0.03243577
tergit.width    0.43785256   1.00000000 0.42965960  0.46633676    0.02896152
wing.width     0.16776218   0.42965960 1.00000000  0.51370391    0.06622357
wing.length    0.36402723   0.46633676 0.51370391  1.00000000    0.07908013
cubital.index  0.03243577   0.02896152 0.06622357  0.07908013    1.00000000
```

Для статистического анализа значимости коэффициента корреляции пары переменных используется функция `cor.test()`, однако для анализа корреляции комплекса признаков она не подходит. Конечно, можно провести анализ в парах переменных с использованием циклов и «ручным» применением той или иной поправки. Однако удобнее воспользоваться специально предназначенной для такого анализа функцией `corr.test()` из пакета `psych`. Обратите внимание на двойную букву `r` в названии этой функции.

```
    > (result <- corr.test(mel[,2:6]))
Call:corr.test(x = mel[, 2:6])
Correlation matrix
              tergit.length    tergit.width    wing.width    wing.length
cubital.index
tergit.length         1.00        0.44       0.17       0.36         0.03
tergit.width          0.44        1.00       0.43       0.47         0.03
wing.width            0.17        0.43       1.00       0.51         0.07
```

```
wing.length          0.36        0.47       0.51        1.00         0.08
cubital.index        0.03        0.03       0.07        0.08         1.00
Sample Size
              tergit.length    tergit.width    wing.width    wing.length
cubital.index
tergit.length          114         114         113         114          114
tergit.width           114         114         113         114          114
wing.width             113         113         113         113          113
wing.length            114         114         113         114          114
cubital.index          114         114         113         114          114
Probability values (Entries above the diagonal are adjusted for multiple
tests.)
              tergit.length    tergit.width    wing.width    wing.length
cubital.index
tergit.length         0.00        0.00       0.38        0.0            1
tergit.width          0.00        0.00       0.00        0.0            1
wing.width            0.08        0.00       0.00        0.0            1
wing.length           0.00        0.00       0.00        0.0            1
cubital.index         0.73        0.76       0.49        0.4            0

 To see confidence intervals of the correlations, print with the
short=FALSE option
```

В консоль выводятся три матрицы: матрица корреляций (`Correlation matrix`), матрица объемов выборки (`Sample Size`), использованных при расчете того или иного коэффициента (в нашем случае они варьируют из-за отсутствующего значения), а также матрица *p*-значений (`Probability values`), причем под главной диагональю отображены нескорректированные значения, а над главной диагональю – уже скорректированные. По умолчанию функция `corr.test()` использует поправку Холма (можно изменить на поправку Бонферрони с использованием аргумента `adjust = "bonferroni"`), для обработки отсутствующих значений используются все полные пары значений (можно изменить на любой другой метод с использованием аргумента `use`), анализ проводится для коэффициента корреляции Пирсона (можно изменить на коэффициенты Спирмена или Кенделла с использованием аргумента `method`).

В нашем примере значимо отличаются от нуля 5 коэффициентов корреляции из 10. Мы видим, например, что длина крыла и ширина тергита уверенно коррелирует со всеми остальными признаками за исключением кубитального индекса, который не коррелирует ни с чем.

Для визуализации результатов корреляционного анализа комплекса признаков удобно использовать функцию `corrplot()` из небольшого одноименного пакета. Эта функция графически отображает матрицу корреляций с использованием цветовой системы обозначений и обладает большим числом

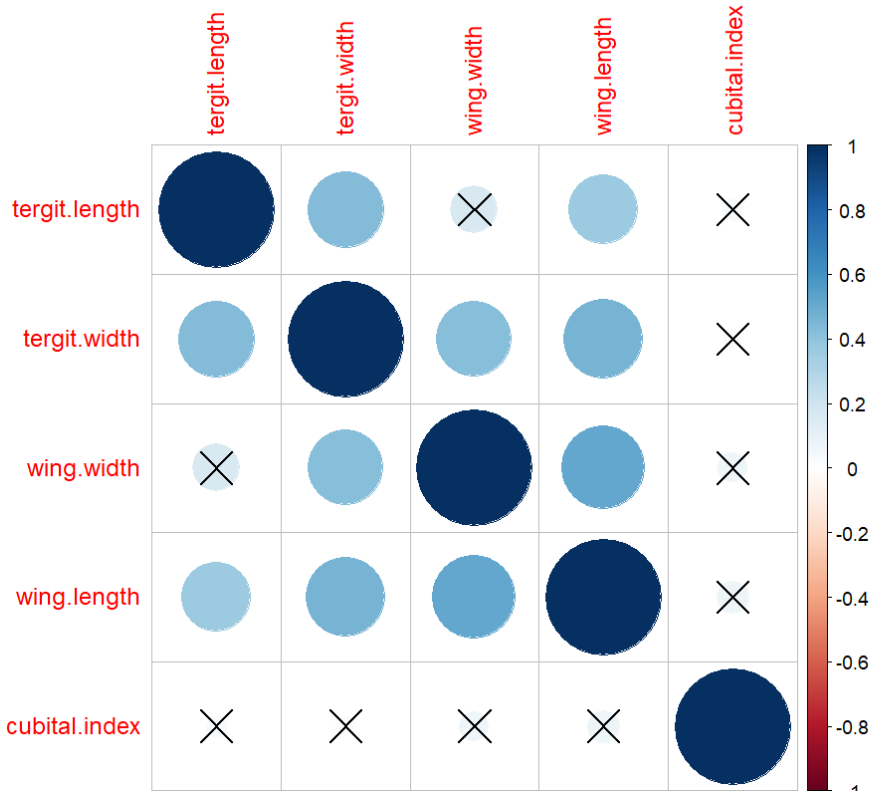

Рис. 25. Результаты анализа корреляции для комплекса из 5 количественных морфологических признаков пчел

параметров для настройки отображения. Мы рассмотрим один конкретный вариант с применением метода кругов (`method = "circle"`, используется по умолчанию) и зачеркивания незначимых коэффициентов (`insig = "pch"`, используется по умолчанию). Функции необходимо передать матрицу корреляций, а в случае необходимости визуализации значимости также и матрицу *p*-значений. И то, и другое можно получить из объекта, возвращаемого функцией `corr.test()`.

```
> r <- result$r
> p <- pmax(result$p, t(result$p))
```

Напомним, что матрица *p*-значений, возвращаемая функцией `corr.test()`, содержит скорректированные значения только над главной диагональю. Чтобы избежать путаницы, лучше использовать только скорректированные значения. Поэтому мы берем максимум из исходной и транспонированной матриц (скорректированное значение всегда больше исходного), для чего используем функцию для расчета параллельных максимумов `pmax()`.

```
> library(corrplot)
> corrplot(r, p.mat = p)
```

Полученный график представлен на Рис. 25. Коэффициенты отображены в виде закрашенных кругов. Размер круга пропорционален абсолютному значению

коэффициента: чем выше, тем больше. Цвет кодирует значение коэффициента: положительные коэффициенты отображаются в синей гамме, отрицательные – в красной, при этом, чем меньше абсолютное значение коэффициента, тем бледнее соответствующий круг. Шкала цветовой кодировки представлена справа от матрицы. Незначимые коэффициенты корреляции перечеркнуты.

8.2. Множественная регрессия: подбор линейной модели

При работе с комплексом признаков может возникнуть необходимость построения модели зависимости одной независимой переменной от множества зависимых. Для этой цели предназначен метод анализа, называемый множественной регрессией.

Общая модель множественной линейной регрессии имеет вид:

$$y = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + ... + \beta_k \cdot x_k + \varepsilon ,$$

где $x_1, x_2, \ldots, x_k$ – независимые переменные (часто называемые предикторами), k – их количество, $\beta_0, \beta_1, \ldots, \beta_k$ – коэффициенты регрессии, ε – случайный шум, подчиняющийся нормальному распределению с нулевым средним и дисперсией σ^2. Как и в случае анализа зависимости между двумя переменными, коэффициенты регрессии оцениваются методом наименьших квадратов, но теперь оценка происходит в многомерном пространстве с использованием алгоритмов линейной алгебры.

Интерпретация коэффициентов множественной регрессии аналогична простой регрессии: свободный коэффициент β_0 определяет значение зависимой переменной при равенстве нулю всех предикторов, а коэффициенты при предикторах определяют скорость изменения зависимой переменной в расчете на единицу данной независимой переменной при условии, что все остальные предикторы остаются неизменными.

В разделе 7.2 мы построили регрессионную модель зависимости длины тергита от длины крыла, которая объясняет 13 % изменчивости длины тергита. Попробуем улучшить эту модель с использованием множественной регрессии по всему комплексу имеющихся количественных признаков. Для построения модели множественной регрессии используется та же функция `lm()`, которая применяется следующим образом:

```
> reg.model <- lm(tergit.length ~ proboscis + tergit.width +
+               wing.length + wing.width + cubital.index, data = mel)
```

Обратите внимание, что теперь в формуле справа от тильды мы указали несколько переменных, которые будут считаться независимыми. Переменные «просуммированы», что означает аддитивность их эффекта.

Рассмотрим результаты анализа модели, которые можно получить с помощью функции summary():

```
> summary(reg.model)

Call:
lm(formula = tergit.length ~ proboscis + tergit.width + wing.length +
    wing.width + cubital.index, data = mel)

Residuals:
      Min        1Q    Median        3Q       Max
-0.134062 -0.032223  0.002236  0.035108  0.174722

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.6608373  0.3790407   1.743 0.084128 .
proboscis      0.0190432  0.0364975   0.522 0.602911
tergit.width   0.1945118  0.0518707   3.750 0.000287 ***
wing.length    0.1057438  0.0433223   2.441 0.016293 *
wing.width    -0.1001710  0.0792329  -1.264 0.208883
cubital.index  0.0002374  0.0007885   0.301 0.763937
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.05733 on 107 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.2453,     Adjusted R-squared:   0.21
F-statistic: 6.955 on 5 and 107 DF,  p-value: 1.164e-05
```

Структура вывода нам уже знакома: формула (Call), описание остатков (Residuals), таблица коэффициентов и их анализа (Coefficients), остаточное стандартное отклонение (Residual standard error), коэффициенты детерминации (Multiple R-squared), общий анализ модели (F-statistic). Мы видим, что модель в целом значима (p-value: 1.164e-05), то есть она обладает лучшей предсказательной силой по сравнению со случайным шумом. Данная модель объясняет 24.5 % изменчивости длины тергита, что значительно лучше по сравнению с простой регрессией. Анализ коэффициентов регрессии свидетельствует о значимости вклада ширины тергита и длины крыла, тогда как коэффициенты при длине хоботка, ширине крыла и кубитальном индексе не отличаются от нуля. Эти результаты неудивительны в свете результатов корреляционного анализа (см. предыдущий раздел), который выявил значимую

связь длины тергита только с шириной тергита и длиной крыла. Можно предположить, что рост предсказательной силы модели достигнут за счет добавления в нее именно ширины тергита.

Таким образом, включение в регрессионную модель трех переменных из пяти можно считать неоправданным: они не дают значимого вклада в предсказания модели. Разумно было бы оставить только значимые предикторы, чтобы минимизировать сложность модели. Здесь мы сталкиваемся с проблемой выбора оптимальной модели. Дело в том, что значимость вклада отдельных предикторов, оцениваемая по результатам сравнения с нулем коэффициента регрессии по критерию Стьюдента, зависит от присутствия в модели других предикторов. Обратите внимание, что результаты анализа значимости коэффициента при длине крыла существенно изменились. В отсутствие других предикторов (см. раздел 7.2): $t = 4.136$, $p = 6.85e\text{-}05$. В модели с пятью предикторами: $t = 2.441$, $p = 0.016293$.

Практически при любом изменении структуры модели результаты анализа значимости коэффициентов будут меняться. Обычно поступают следующим образом: по одной убирают из модели наименее значимые предикторы, на каждом этапе повторяя анализ. Именно тогда и встает проблема выбора оптимальной модели. Дело в том, что даже формально незначимые предикторы могут давать определенный вклад в объяснительную силу модели. Необходимо найти баланс между числом предикторов и адекватностью модели. Количественная мера такого баланса должна отражать как предсказательную силу, так и число параметров. Одной из таких мер является скорректированный коэффициент детерминации. Другой мерой является информационный критерий Акаике (Akaike information criterion, AIC), который учитывает и объяснительную силу модели, и ее экономность в смысле числа параметров. Наилучшей считается модель с наименьшим значением критерия Акаике.

Для расчета критерия Акаике для линейных моделей есть специальная функция `extractAIC()`, которой передается одна или несколько линейных моделей. Рассчитаем значения критерия Акаике для полной модели с пятью предикторами и модели зависимости длины тергита только от длины крыла:

```
> extractAIC(reg.model)
[1]    6.0000 -640.2659
> extractAIC(lm(tergit.length ~ wing.length, data = mel[-75, ]))
[1]    2.0000 -633.0825
```

Функция `extractAIC()` возвращает пару значений: число параметров модели и значение критерия Акаике. Видно, что модель с пятью предикторами имеет меньшее значение информационного критерия Акаике, поэтому она лучше

(в данном случае информационный критерий Акаике имеет отрицательное значение, поэтому наименьшим будет значение, большее по абсолютной величине). Обратите внимание, что если при расчете линейной модели встретятся отсутствующие значения NA, то неполные наблюдения будут исключены из анализа. Именно так функция lm() поступила с 75-м наблюдением при расчете полной модели reg.model. Но при расчете модели с двумя переменными (tergit.length ~ wing.length) значение NA не будет встречено, поскольку показатель ширины крыла не используется в формуле, и в модели будут задействованы уже все наблюдения. В результате модели будут основаны на разных наборах данных, что делает некорректным их прямое сравнение. Для разрешения этого затруднения, при расчете второй модели мы исключили неполное 75-е наблюдение вручную.

Попробуем теперь исключить из регрессионной модели кубитальный индекс (этот признак имеет наибольшее *p*-значение) и рассчитаем для новой модели критерий Акаике:

```
> extractAIC(lm(tergit.length ~ proboscis + tergit.width +
+                 wing.length + wing.width, data = mel[-75, ]))
[1]    5.0000 -642.1702
```

Значение критерия уменьшилось по сравнению с полной моделью, поэтому новая редуцированная модель лучше. Продолжим исключать незначимые переменные и избавимся от длины хоботка:

```
> extractAIC(lm(tergit.length ~ tergit.width + wing.length +
+                 wing.width, data = mel[-75, ]))
[1]    4.0000 -643.8734
```

Значение критерия снова уменьшилось. Исключим теперь ширину крыла:

```
> extractAIC(lm(tergit.length ~ tergit.width + wing.length,
+                 data = mel[-75, ]))
[1]    3.0000 -644.1947
```

Информационный критерий еще уменьшился. Таким образом, мы достигли минимума и подобрали оптимальную модель. В данном случае мы вручную применили так называемую обратную пошаговую процедуру поиска оптимальной модели, когда переменные последовательно исключаются из модели (при выборе порядка исключения переменных мы руководствовались результатами анализа значимости их коэффициентов в полной модели). Существует также прямая пошаговая процедура, когда предикторы последовательно включаются в модель в порядке роста их корреляции с зависимой переменной, а также комбинированная процедура, когда на каждом шаге рассматривается возможность как включения, так и исключения любой

переменной. Все три процедуры не гарантируют нахождение оптимальной модели, поэтому разработано множество альтернативных алгоритмов.

8.3. Пошаговые процедуры поиска оптимальной модели

Пошаговые процедуры реализованы в специальной функции `step()`, которой передается стартовая модель. Алгоритм поиска задается аргументом `direction`, который может принимать значения «`backward`» (обратная процедура, используется по умолчанию), «`forward`» (прямая процедура), либо «`both`» (комбинированная процедура). В ходе подбора оптимальной модели функция `step()` выдает в консоль много промежуточной информации, избежать вывода которой можно с использованием аргумента `trace = 0`. Попробуем применить обратную пошаговую процедуру к полной модели.

```
 > opt.model <- step(reg.model, trace = 0)
Error in step(reg.model, trace = 0) :
  number of rows in use has changed: remove missing values?
```

Функция выдала сообщение об ошибке (`number of rows in use has changed`: число использованных наблюдений изменилось) и предложила избавиться от отсутствующих значений (`remove missing values?`). Действительно, на одном из этапов отбора модели из нее была удалена ширина крыла, после чего 75-е наблюдение было включено в расчеты, изменился объем выборки и сравнение моделей стало некорректным. В таких случаях перед применением пошаговых процедур рекомендуется удалить все неполные наблюдения. Мы можем сделать это, перестроив нашу исходную модель с использованием отрицательного индекса при указании фрейма:

```
> reg.model <- lm(tergit.length ~ proboscis + tergit.width +
+        wing.length + wing.width + cubital.index, data = mel[-75, ])
```

Теперь затруднений, связанных с отсутствующими значениями, не возникнет и можно вернуться к пошаговой обратной процедуре. На сей раз мы не будем подавлять вывод промежуточных результатов.

```
> opt.model <- step(reg.model)
Start:  AIC=-640.27
tergit.length ~ proboscis + tergit.width + wing.length + wing.width +
    cubital.index

                Df Sum of Sq     RSS     AIC
- cubital.index  1  0.000298 0.35203 -642.17
- proboscis      1  0.000895 0.35263 -641.98
- wing.width     1  0.005254 0.35699 -640.59
```

```
<none>                       0.35174 -640.27
- wing.length    1  0.019585 0.37132 -636.14
- tergit.width   1  0.046225 0.39796 -628.31

Step:  AIC=-642.17
tergit.length ~ proboscis + tergit.width + wing.length + wing.width

               Df Sum of Sq     RSS     AIC
- proboscis     1  0.000926 0.35296 -643.87
- wing.width    1  0.005167 0.35720 -642.52
<none>                     0.35203 -642.17
- wing.length   1  0.019868 0.37190 -637.97
- tergit.width  1  0.046065 0.39810 -630.27

Step:  AIC=-643.87
tergit.length ~ tergit.width + wing.length + wing.width

               Df Sum of Sq     RSS     AIC
- wing.width    1  0.005283 0.35824 -644.19
<none>                     0.35296 -643.87
- wing.length   1  0.020786 0.37375 -639.41
- tergit.width  1  0.049048 0.40201 -631.17

Step:  AIC=-644.19
tergit.length ~ tergit.width + wing.length

               Df Sum of Sq     RSS     AIC
<none>                     0.35824 -644.19
- wing.length   1  0.015810 0.37405 -641.31
- tergit.width  1  0.044077 0.40232 -633.08
```

Функция начала работу со стартового значения AIC полной модели (`Start: AIC=-640.27`) и рассчитала для каждой из включенных в полную модель переменных значение AIC после исключения этой переменной. Результаты такого анализа представлены в таблице. Переменные упорядочены по возрастанию значения AIC. Здесь также присутствует строка `<none>`, которая соответствует текущей модели (то есть никакие переменные не удаляются). Таким образом, выше этой строки расположены кандидаты на удаление, первым из которых является `cubital.index`. Именно кубитальный индекс исключается на первом шаге.

Функция фиксирует новое базовое значение AIC (`Step:  AIC=-642.17`) и текущую модель (`tergit.length ~ proboscis + tergit.width + wing.length + wing.width`), после чего пересчитывает значения AIC после исключения из модели оставшихся переменных и снова отображает их в виде таблицы. На этом

этапе исключается длина хоботка, а на следующем – ширина крыла. Процедура продолжается до тех пор, пока исключение одной из переменных приводит к снижению значения AIC. На последнем шаге в таблице строка <none> находится сверху, потому что соответствующее значение AIC минимально. На этом функция прекращает расчеты и возвращает найденную модель.

В данном случае и порядок исключения переменных, и конечный результат совпали с нашим «ручным» отбором. Так будет происходить не всегда. Мы руководствовались *p*-значениями в полной модели и не проводили перерасчета на каждом шаге. Но результаты анализа зависят от присутствия других переменных, поэтому они могли измениться. Функция `step()` проводит анализ на каждом шаге, поэтому лучше воспользоваться предоставляемыми ею возможностями.

Рассмотрим теперь прямую пошаговую процедуру. Для ее запуска функции `step()` нужно передать «пустую» модель, в которую будут добавляться переменные, а также формулу, содержащую все потенциально пригодные предикторы. Пустую модель можно создать следующим образом:

```
> (empty.model <- lm(tergit.length ~ 1, data = mel[-75, ]))

Call:
lm(formula = tergit.length ~ 1, data = mel[-75, ])

Coefficients:
(Intercept)
      2.469
```

Единица в списке предикторов в формуле соответствует свободному члену. Его присутствие обычно предполагается по умолчанию, поэтому в формулах единица встречается нечасто. В некоторых случаях необходимо подобрать регрессионную модель, проходящую через начало координат (то есть со свободным членом, равным нулю), в таких ситуациях в формулу вводят минус единицу, указывающую на необходимость подавления свободного члена. В данном случае мы использовали единицу, чтобы создать модель без предикторов (если бы мы ничего не указали справа от тильды, произошла бы ошибка). Поскольку предикторов в этой модели нет, единственным коэффициентом является свободный член, который представляет собой среднее значение зависимой переменной:

```
> mean(mel$tergit.length[-75])
[1] 2.4686
```

Теперь мы можем запустить прямую пошаговую процедуру поиска оптимальной модели. Формула со всеми потенциальными предикторами передается аргументу scope.

```
> opt.model <- step(empty.model, direction = "forward",
+                   scope = tergit.length ~ proboscis + tergit.width +
+                           wing.length + wing.width + cubital.index)
Start:  AIC=-618.47
tergit.length ~ 1

                Df Sum of Sq     RSS     AIC
+ tergit.width   1  0.091997 0.37405 -641.31
+ wing.length    1  0.063730 0.40232 -633.08
+ wing.width     1  0.013117 0.45293 -619.69
+ proboscis      1  0.010799 0.45525 -619.12
<none>                       0.46605 -618.47
+ cubital.index  1  0.000987 0.46506 -616.71

Step:  AIC=-641.31
tergit.length ~ tergit.width

                Df Sum of Sq     RSS     AIC
+ wing.length    1 0.0158096 0.35824 -644.19
<none>                       0.37405 -641.31
+ proboscis      1 0.0018293 0.37222 -639.87
+ cubital.index  1 0.0005944 0.37346 -639.49
+ wing.width     1 0.0003059 0.37375 -639.41

Step:  AIC=-644.19
tergit.length ~ tergit.width + wing.length

                Df Sum of Sq     RSS     AIC
<none>                       0.35824 -644.19
+ wing.width     1 0.0052827 0.35296 -643.87
+ proboscis      1 0.0010414 0.35720 -642.52
+ cubital.index  1 0.0002375 0.35800 -642.27
```

Структура вывода аналогична обратной процедуре. На каждом шаге выводится значение AIC, текущая модель и таблица с анализом моделей после добавления переменных (обратите внимание на плюсики рядом с именами переменных в таблицах). На первом этапе в модель включается ширина тергита, затем длина крыла, на третьем этапе процедура завершается, поскольку добавление остальных переменных не ведет к снижению AIC.

Рассмотрим теперь комбинированную процедуру, вызов которой аналогичен вызову прямой процедуры, то есть также необходимо задать пустую стартовую модель и формулу с набором потенциальных предикторов.

```
> opt.model <- step(empty.model, direction = "both",
+                   scope = tergit.length ~ proboscis + tergit.width +
+                         wing.length + wing.width + cubital.index)
Start:  AIC=-618.47
tergit.length ~ 1

                Df Sum of Sq     RSS     AIC
+ tergit.width   1  0.091997 0.37405 -641.31
+ wing.length    1  0.063730 0.40232 -633.08
+ wing.width     1  0.013117 0.45293 -619.69
+ proboscis      1  0.010799 0.45525 -619.12
<none>                       0.46605 -618.47
+ cubital.index  1  0.000987 0.46506 -616.71

Step:  AIC=-641.31
tergit.length ~ tergit.width

                Df Sum of Sq     RSS     AIC
+ wing.length    1  0.015810 0.35824 -644.19
<none>                       0.37405 -641.31
+ proboscis      1  0.001829 0.37222 -639.87
+ cubital.index  1  0.000594 0.37346 -639.49
+ wing.width     1  0.000306 0.37375 -639.41
- tergit.width   1  0.091997 0.46605 -618.47

Step:  AIC=-644.19
tergit.length ~ tergit.width + wing.length

                Df Sum of Sq     RSS     AIC
<none>                       0.35824 -644.19
+ wing.width     1  0.005283 0.35296 -643.87
+ proboscis      1  0.001041 0.35720 -642.52
+ cubital.index  1  0.000237 0.35800 -642.27
- wing.length    1  0.015810 0.37405 -641.31
- tergit.width   1  0.044077 0.40232 -633.08
```

В ходе комбинированной процедуры на каждом этапе рассматривается не только возможность включения еще неиспользованных переменных (переменные с плюсами в таблицах), но также возможность исключения всех уже включенных в модель (переменные с минусами). В нашем примере и ход включения переменных, и итоговый результат прямой и комбинированной

процедур совпали, но в сложных случаях с большим числом потенциальных предикторов так будет происходить не всегда.

Мы рекомендуем применить все три варианта и в случае несовпадения выбрать результат с наименьшим AIC итоговой модели.

В заключение нам остается провести диагностику итоговой модели и убедиться в выполнении предположений, лежащих в основе регрессионного анализа: линейность связи, нормальность распределения остатков и гомогенность их дисперсии. Для диагностики отклонений можно использовать описанные в разделе 7.4 инструменты. Анализ линейности проведем графически с использованием графика зависимости остатков от прогнозируемых значений.

```
> plot(fitted(opt.model), resid(opt.model),
+      xlab = "Прогнозируемые значения", ylab = "Остатки")
> abline(h = 0, col = "grey", lty = 3)
> panel.smooth(fitted(opt.model), resid(opt.model))
```

Остатки распределены вокруг нуля без какой-либо выраженной зависимости (Рис. 26), что соответствует предположению линейности. В случае множественной регрессии этот график является практически единственным средством визуализации линейности связи, поскольку линия регрессии теперь находится не в двумерном, а в многомерном пространстве.

Анализ нормальности и гомоскедастичности остатков проведем с использованием критериев Шапиро-Уилка и Бройша-Пагана соответственно:

```
> shapiro.test(resid(opt.model))

 Shapiro-Wilk normality test

data:  resid(opt.model)
W = 0.9878, p-value = 0.4038
> library(car)
> ncvTest(opt.model)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 2.785743    Df = 1     p = 0.09510657
```

Итак, критерий согласия Шапиро-Уилка не выявил отклонений от нормальности, а критерий Бройша-Пагана не выявил гетероскедастичности. Теперь можно с полной уверенностью провести анализ итоговой модели.

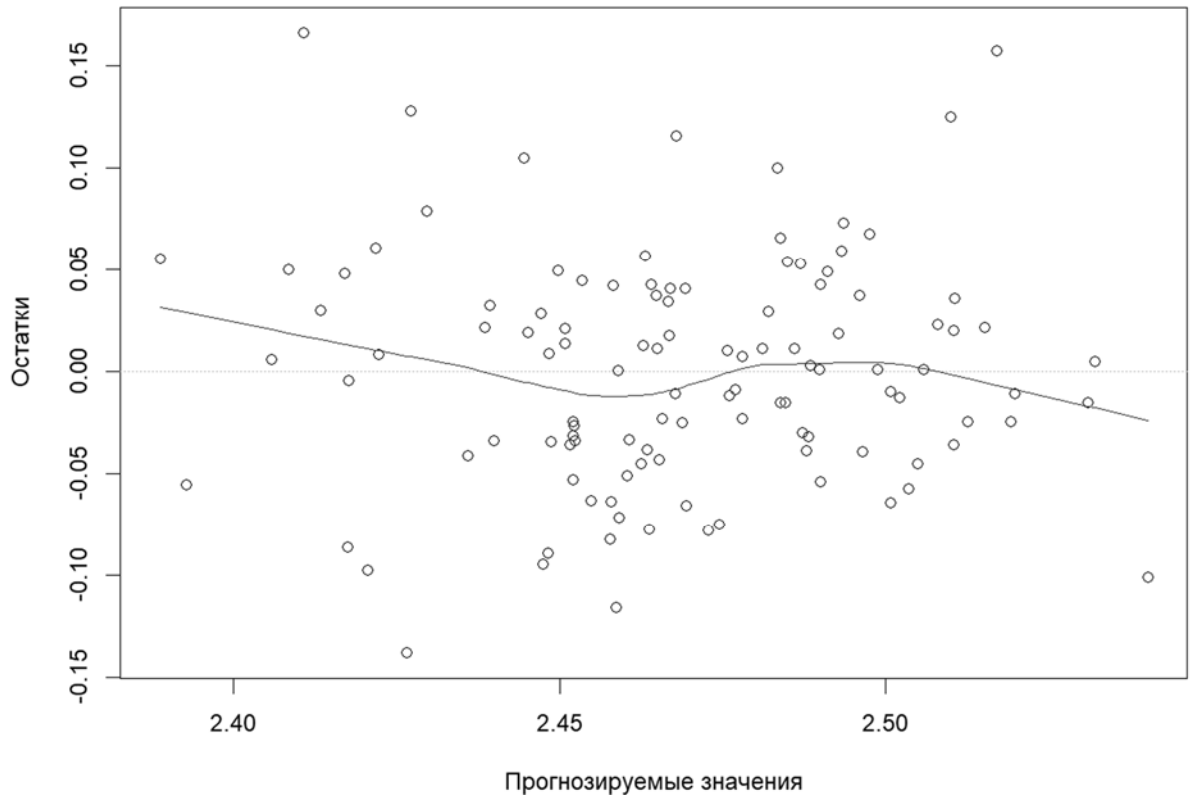

Рис. 26. Зависимость остатков от прогнозируемых значений в оптимальной множественной регрессионной модели

```
> summary(opt.model)
Call:
lm(formula = tergit.length ~ tergit.width + wing.length, data = mel[-75,])

Residuals:
      Min        1Q    Median        3Q       Max
-0.138208 -0.035955  0.000943  0.037402  0.166555

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.71517    0.33928   2.108 0.037309 *
tergit.width  0.18182    0.04942   3.679 0.000364 ***
wing.length   0.08694    0.03946   2.203 0.029660 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.05707 on 110 degrees of freedom
Multiple R-squared:  0.2313,     Adjusted R-squared:  0.2173
F-statistic: 16.55 on 2 and 110 DF,  p-value: 5.2e-07
```

Глава 9. Статистика нечисловых данных

В главах 3–8 мы рассматривали методы статистического анализа количественных переменных, основными выборочными параметрами которых являются среднее и дисперсия. Теперь же мы рассмотрим методы анализа данных нечисловой природы, то есть категориальных переменных, для которых основным выборочным параметром является доля.

Методы расчета долей и таблиц сопряженности рассмотрены в главе 2, основным инструментом для этого служит функция `table()`, рассчитывающая натуральные частоты категорий в выборке.

9.1. Сравнение доли с эталоном

Анализ долей определенных категорий основывается на биномиальном распределении вероятностей с объемом выборки N и вероятностью успеха p. В данном случае под испытанием в рамках схемы Бернулли подразумевается каждый элемент выборки, а под успехом – принадлежность элемента выборки к интересующей нас категории. Тогда вероятность наблюдения данной частоты m в выборке объема N определяется по формуле Бернулли, а совокупность этих вероятностей составляет биномиальное распределение (см. гл. 1). При больших объемах выборки дискретное биномиальное распределение хорошо аппроксимируется непрерывным нормальным распределением со средним $N \cdot p$ и дисперсией $N \cdot p \cdot (1 - p)$. Такая аппроксимация считается надежной, если ожидаемая частота успехов и неудач больше 5.

Рассмотрим задачу сравнения эмпирической доли с эталоном на конкретном примере. Согласно литературным данным в популяциях пчел породы *A. m. mellifera* доля организмов с положительным дискоидальным смещением составляет 65 %. Соответствует ли наша выборка пчел породы *A. m. mellifera* данному стандарту? Другими словами: отличается ли частота положительного дискоидального смещения в нашей выборке от 65 %?

Сформулируем нуль-гипотезу:

$$H_0: p = p_0,$$

где p – эмпирическая вероятность (относительная частота), p_0 – теоретическая вероятность (в нашем случае 0.65).

Загрузим данные и рассчитаем эмпирическую частоту.

```
> load("apis.rda")
> mel <- subset(apis, subset = breed == "mellifera")
```

```
> (f <- table(mel$discoidal.shift))
negative  neutral positive
      19       17       78
> (N <- sum(f))
[1] 114
> (p <- f[3]/N)
 positive
0.6842105
```

Итак, наша выборка имеет объем $N = 114$ и относительную частоту положительного дискоидального смещения $p = 0.684$, т.е. отклонение от теоретической частоты составляет $(0.68421 – 0.65) = 0.03421$. Насколько вероятно получить такое же или более сильное отклонение от теоретической вероятности в условиях нуль-гипотезы? Чтобы ответить на этот вопрос, нужно воспользоваться биномиальным распределением либо его аппроксимацией нормальным распределением. В нашем случае аппроксимация нормальным распределением надежна, поскольку натуральные частоты успеха $p_0 \cdot N = 0.65 \cdot 114 = 74.1$ и неудачи $(1 – p_0) \cdot N = 0.35 \cdot 114 = 39.9$ больше 5.

В условиях нуль-гипотезы натуральная частота m в выборке может быть аппроксимирована нормальным распределением со средним $\mu = Np_0 = 74.1$ и дисперсией $\sigma^2 = Np_0(1 – p_0) = 25.935$. Нормированная величина

$$z = \frac{m - Np_0}{\sqrt{Np_0(1 - p_0)}}$$

подчиняется стандартному нормальному распределению с нулевым средним и единичной дисперсией. Рассчитаем значение z для нашей выборки:

```
> (z <- (unname(f[3]) - N*0.65) / sqrt(N*0.65*(1-0.65)))
[1] 0.7658108
```

Рассчитаем теперь вероятность получения такого же или более сильного отклонения величины z от нуля в условиях нуль-гипотезы, для чего воспользуемся функцией стандартного нормального распределения (подробности см. в разделах 10.2 и 12.1):

```
> pnorm(-z) + 1 - pnorm(z)
[1] 0.4437889
```

Итак, вероятность получить такие же или еще более сильные отклонения от эталона, которые наблюдаются в нашей выборке (0.03421), составляет около 44.5 %. Другими словами – мы получили вполне типичные отклонения от теоретической частоты для выборки объемом 114 и у нас нет оснований отвергнуть нуль-гипотезу.

Использованная процедура часто называется z-критерием, поскольку основывается на величине z, подчиняющейся стандартному нормальному

распределению. Мы провели расчеты путем последовательного вычисления необходимых показателей. В R можно воспользоваться специальной функцией prop.test(), предназначенной для сравнения долей.

```
> prop.test(f[3], N, 0.65, correct = FALSE)

 1-sample proportions test without continuity correction

data:  f[3] out of N, null probability 0.65
X-squared = 0.5865, df = 1, p-value = 0.4438
alternative hypothesis: true p is not equal to 0.65
95 percent confidence interval:
 0.5940657 0.7623454
sample estimates:
        p
0.6842105
```

Первым аргументом мы передали натуральную частоту, вторым – объем выборки, а третьим – теоретическую вероятность, с которой мы хотим сравнить нашу выборку. Структура вывода типична для статистических критериев: название процедуры (1-sample proportions test without continuity correction – одновыборочный критерий для доли без поправки на непрерывность), описание данных (data), значение критериальной статистики (X-squared), число степеней свободы (df), *p*-значение (p-value), формулировка альтернативной гипотезы (alternative hypothesis), доверительный интервал для выборочной оценки доли (95 percent confidence interval) и точечная оценка доли (sample estimates). Обратите внимание, что в качестве критериальной статистики использована величина χ^2, тогда как мы использовали нормально распределенную величину *z*. Дело в том, что квадрат величины *z* подчиняется распределению χ^2 с одной степенью свободы, поэтому использование распределения χ^2 в данном случае эквивалентно использованию стандартного нормального распределения. При вызове функции prop.test() мы воспользовались аргументом correct = FALSE, чтобы отменить поправку Йейтса на непрерывность, которая используется по умолчанию. Использование этой поправки делает аппроксимацию дискретного биномиального распределения непрерывным нормальным более точной.

```
> prop.test(f[3], N, 0.65, correct = TRUE)

 1-sample proportions test with continuity correction

data:  f[3] out of N, null probability 0.65
X-squared = 0.4457, df = 1, p-value = 0.5044
alternative hypothesis: true p is not equal to 0.65
```

```
95 percent confidence interval:
 0.5895240 0.7662814
sample estimates:
        p
0.6842105
```

Обратите внимание, что *p*-значение в результате применения поправки Йейтса немного увеличилось. Наиболее точный результат можно получить с помощью функции `binom.test()`, которая не использует аппроксимаций и задействует дискретное биномиальное распределение. Вызов этой функции и структура вывода аналогичны функции `prop.test()`:

```
> binom.test(f[3], N, 0.65)

 Exact binomial test

data:  f[3] and N
number of successes = 78, number of trials = 114, p-value = 0.4924
alternative hypothesis: true probability of success is not equal to
0.65
95 percent confidence interval:
 0.5905101 0.7680674
sample estimates:
probability of success
             0.6842105
```

9.2. Сравнение выборочных долей

Следующая типичная задача – сравнение выборочных долей. Пусть мы имеем две выборки объемом N_1 и N_2 с наблюдаемыми частотами m_1 и m_2. Нас интересует вопрос об отличиях относительной частоты интересующего нас признака в двух выборках. Нуль-гипотеза формулируется следующим образом:

$$H_0: p_1 = p_2,$$

или эквивалентно:

$$H_0: p_1 - p_2 = 0.$$

Для построения критерия рассматривается распределение разницы выборочных долей в условиях нуль-гипотезы, которую можно аппроксимировать нормальным распределением с нулевым средним и дисперсией

$$\sigma^2 = \left(\frac{1}{N_1} + \frac{1}{N_2}\right) \cdot p_{pool} \cdot (1 - p_{pool}),$$

где p_{pool} – так называемая объединенная оценка истинной доли:

$$p_{pool} = \frac{m_1 + m_2}{N_1 + N_2}.$$

Нормированная величина

$$z = \frac{p_1 - p_2}{\sigma}$$

в условиях нуль-гипотезы имеет стандартное нормальное распределение, а ее квадрат – распределение χ^2 с одной степенью свободы. Любое из этих распределений может быть использовано для проверки гипотез.

В R процедура сравнения долей реализована в уже знакомой нам функции `prop.test()`, которой на сей раз в качестве первого и второго аргументов нужно передать не единичные значения, а вектора соответственно натуральных частот и объемов выборки.

Проведем сравнение долей пчел с положительным дискоидальным смещением в наших выборках пчел пород *A. m. mellifera* и *A. m. carnica*. Рассчитаем частоты и объемы выборки и сохраним их в виде векторов из двух элементов:

```
> carn <- subset(apis, subset = breed == "carnica")
> (m <- c(f[3], table(carn$discoidal.shift)[3]))
positive positive
      78      225
> (N <- c(N, dim(carn)[1]))
[1] 114 240
```

Теперь воспользуемся функцией `prop.test()` для сравнения выборочных долей:

```
> prop.test(m, N)

 2-sample test for equality of proportions with continuity correction

data:  m out of N
X-squared = 38.1828, df = 1, p-value = 6.442e-10
alternative hypothesis: two.sided

95 percent confidence interval:
 -0.3504155 -0.1561634
sample estimates:
   prop 1    prop 2
0.6842105 0.9375000
```

Структура вывода аналогична одновыборочному критерию, за исключением того, что доверительный интервал строится для разности долей. По умолчанию применяется поправка Йейтса на непрерывность. В нашем случае доли пчел с положительным дискоидальным смещением значимо отличаются

между двумя выборками: $p < 0.05$ и доверительный интервал для разности долей не покрывает ноль.

В случае сравнения выборочных долей аппроксимация нормальным распределением применима в случаях, когда для обеих выборок ожидаемые частоты успехов и неудач в условиях нуль-гипотезы превышают 5. Для проверки этого условия необходимо рассчитать ожидаемые частоты на основе объединенной оценки доли.

```
> (p.pool <- sum(m)/sum(N))
[1] 0.8559322
> c(N*p.pool, N*(1-p.pool))
[1]  97.57627 205.42373  16.42373  34.57627
```

Первые два элемента полученного вектора соответствуют частотам успехов, последние два – частотам неудач. Все эти частоты больше 5, поэтому аппроксимация правомерна. Мы вычислили ожидаемые частоты вручную, однако в типичном случае делать это нет необходимости, поскольку функция `prop.test()` сама проверяет условие применимости аппроксимации и в случае его невыполнения выдает соответствующее предупреждение:

```
Warning message:
In prop.test(m, N) : Chi-squared approximation may be incorrect
```

Если хотя бы одна из ожидаемых частот меньше 5, необходимо применять критерий, не использующий аппроксимацию – так называемый точный критерий Фишера. Он основан на переборе всех возможных вариантов таблиц сопряженности при данных объемах выборки, поэтому для его использования нужно сначала построить соответствующую таблицу сопряженности, которая обычно представляет собой матрицу размерностью 2 × 2. Строки соответствуют успехам и неудачам, а столбцы – выборкам. Частоты неудач легко рассчитать путем вычитания частот успеха из объемов выборки. Сформируем необходимую нам матрицу с помощью функции `rbind()`, которая объединяет векторы в матрицу построчно:

```
> (tab <- rbind(m, N-m))
  [,1] [,2]
m   78  225
    36   15
```

Полученную таблицу сопряженности нужно передать функции `fisher.test()`, в которой реализован точный критерий Фишера:

```
> fisher.test(tab)
 Fisher's Exact Test for Count Data

data:  tab
p-value = 1.538e-09
```

```
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.06989332 0.28939784
sample estimates:
odds ratio
 0.1453924
```

Структура вывода стандартна для статистических тестов. Обратите внимание, что основным показателем при расчете критерия Фишера служит так называемое отношение шансов (`odds ratio`). Оно представляет собой отношение

$$\frac{p_1}{1-p_1} \Big/ \frac{p_2}{1-p_2}.$$

Легко видеть, что в условиях нуль-гипотезы отношение шансов будет равно единице, именно так в случае точного критерия Фишера формулируется нуль-гипотеза. Доверительный интервал и точечная оценка также приведены для отношения шансов.

В нашем случае получено очень маленькое *p*-значение, что свидетельствует о малой вероятности получить наблюдаемое отношение шансов, что позволяет с уверенностью отвергнуть нуль-гипотезу и принять альтернативную гипотезу об отличиях в долях пчел с положительным дискоидальным смещением в выборках пчел пород *A. m. mellifera* и *A. m. carnica*. Это полностью соответствует результату, полученному с помощью *z*-критерия.

9.3. Критерий χ^2 для сравнения с теоретическим распределением

В биомедицине и экологии часто встречаются ситуации, когда имеется некое теоретическое распределение частот, с которым нужно сравнить наблюдаемое в выборке эмпирическое распределение. Типичным примером являются теоретически ожидаемые расщепления в скрещиваниях в классической генетике. Другая типичная ситуация – сравнение распределения частот категориального признака с неким эталонным, заранее известным распределением.

Задача сравнения эмпирического распределения с теоретическим встречается и применительно к количественным переменным. Мы уже встречались с ней, когда рассматривали способы выявления отклонений от нормальности с помощью критериев согласия (см. разд. 3.3). Рассмотренный в настоящем разделе критерий χ^2 также является критерием согласия и может быть применен для диагностики отклонений от нормальности, однако он имеет

меньшую мощность по сравнению с современными специализированными критериями.

Нуль-гипотезой любого критерия согласия является предположение о соответствии эмпирического распределения теоретическому. Альтернативная гипотеза соответственно состоит в отличии распределений. Критериальной статистикой служит величина

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i},$$

где O_i – наблюдаемые частоты (от англ. observed), E_i – ожидаемые частоты (от англ. expected), k – число категорий признака. Величина χ^2 в условиях нуль-гипотезы подчиняется одноименному распределению с $k - 1$ степенями свободы. Критерий χ^2 основан на аппроксимации дискретного распределения непрерывным, поэтому для его применения должно выполняться следующее условие: ожидаемые частоты должны быть больше 5.

Рассмотрим конкретный пример. В классическом опыте по дигибридному скрещиванию гороха в первом поколении были получены следующие гибриды: 108 высоких растений с розовыми цветками, 35 высоких растений с белыми цветками, 46 карликовых растений с розовыми цветками и 11 карликовых растений с белыми цветками. Соответствуют ли результаты опыта теоретическому расщеплению 9:3:3:1?

Каждый фенотип представляет собой отдельную категорию, при этом нам известны эмпирические натуральные частоты, а также теоретическое соотношение, которое можно рассматривать как натуральные частоты. Сохраним их в виде отдельных векторов.

```
> obs <- c(108, 35, 46, 11)
> exp <- c(9, 3, 3, 1)
```

В R критерий χ^2 реализован в функции `chisq.test()`. Первым аргументом (он имеет имя `x`, но его обычно можно опустить) нужно передать вектор эмпирических натуральных частот, дополнительно нужно использовать аргумент `p`, которому передаются теоретические вероятности. Чтобы перейти от теоретических частот к теоретическим вероятностям достаточно поделить частоты на их сумму, либо можно воспользоваться аргументом `rescale.p`, который нужно установить в значение `TRUE`.

```
> chisq.test(x = obs, p = exp/sum(exp))

 Chi-squared test for given probabilities

data:  obs
X-squared = 2.4533, df = 3, p-value = 0.4838
```

В консоль выводится название критерия (`Chi-squared test for given probabilities`), описание данных (`data`), значение статистики χ^2 (`X-squared`), число степеней свободы (`df`) и p-значение (`p-value`). В нашем случае p-значение велико (> 0.05), поэтому нуль-гипотеза о соответствии эмпирического распределения теоретическому не отклоняется.

Рассмотрим еще один пример. В разд. 9.1 мы сравнили долю пчел породы *A. m. mellifera* с положительным дискоидальным смещением в нашей выборке с породным стандартом, который составляет 65 %, и не обнаружили отличий. Однако имеются литературные данные не только о частоте положительного дискоидального смещения, но также и отрицательного и нейтрального, которые составляют соответственно 15 % и 20 %. Сравним эмпирическое распределение дискоидального смещения в нашей выборке с породным стандартом. Эмпирические частоты мы уже рассчитали в разд. 9.1, они сохранены в векторе `f`, поэтому нам остается просто применить функцию `chisq.test()`:

```
> chisq.test(f, p = c(15, 20, 65), rescale.p = TRUE)

 Chi-squared test for given probabilities

data:  f
X-squared = 1.8918, df = 2, p-value = 0.3883
```

Как и в случае анализа положительного дискоидального смещения, значимых отличий от породного стандарта не выявлено ($p > 0.05$).

9.4. Анализ зависимости между категориальными переменными

В разд. 7.6 мы рассмотрели основы корреляционного анализа, который позволяет выявить зависимость между двумя количественными переменными. Категориальные переменные также могут быть связанными между собой и для выявления такой связи также можно использовать критерий χ^2.

Обратимся к таблице сопряженности, описывающей распределение частот дискоидального смещения в выборках трех пород пчел.

```
> (tab <- table(apis$breed, apis$discoidal.shift))

           negative neutral positive
  carnica         2      13      225
  caucasica      87      64       89
  mellifera      19      17       78
```

Породную принадлежность пчел можно рассматривать в качестве категориального признака и поставить вопрос о связи между породой и дискоидальным смещением. Альтернативные формулировки: (1) зависит ли распределение частот дискоидального смещения от породы, (2) отличается ли соотношение пород в зависимости от дискоидального смещения, (3) связаны ли порода и дискоидальное смещение между собой?

Нуль-гипотеза для данного вида критерия χ^2 заключается в независимости категориальных переменных. Критериальная статистика вычисляется на основе наблюдаемых и ожидаемых частот по уже знакомой нам формуле

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i},$$

однако теперь k – это число ячеек таблицы сопряженности, то есть суммирование ведется по всем ячейкам двумерной таблицы. Ожидаемые частоты E_i вычисляются по следующей формуле:

$$E_i = \frac{[сумма_по_строке] \cdot [сумма_по_столбцу]}{[общая_сумма]}.$$

В случае необходимости суммы по строкам и столбцам (их иногда называют маргинальными суммами) можно добавить к таблице сопряженности с помощью функции `addmargins()`, которая возвращает исходную таблицу с добавленными суммами:

```
> addmargins(tab)

           negative neutral positive Sum
  carnica         2      13      225 240
  caucasica      87      64       89 240
  mellifera      19      17       78 114
  Sum           108      94      392 594
```

Чаще всего вручную рассчитывать суммы и ожидаемые частоты нет необходимости, поскольку эти процедуры уже реализованы в функции `chisq.test()`, которой нужно просто передать исходную таблицу сопряженности:

```
> chisq.test(tab)

 Pearson's Chi-squared test

data:  tab
X-squared = 175.7965, df = 4, p-value < 2.2e-16
```

В условиях нуль-гипотезы величина χ^2 подчиняется одноименному распределению с $(r-1)(c-1)$ степенями свободы, где r и c – число строк и

столбцов таблицы сопряженности соответственно. В нашем примере связь между породой и частотами дискоидального смещения статистически значима ($p < 0.05$).

В случае выявления значимой связи между категориальными переменными имеет смысл вычислить количественную меру такой связи, аналогичную коэффициенту корреляции между двумя количественными переменными. Такой мерой является коэффициент Крамера:

$$V = \sqrt{\frac{\chi^2 / N}{\min(r-1, c-1)}},$$

где N – объем выборки (эквивалентен сумме частот таблицы сопряженности). Коэффициент Крамера может принимать значения от 0 (отсутствие связи) до 1 (абсолютная связь).

Коэффициент Крамера не реализован в базовых пакетах среды R, но его легко рассчитать вручную, воспользовавшись результатом функции `chisq.test()`.

```
> (V <- sqrt(unname(chisq.test(tab)$statistic) / sum(tab)
+                                / min(nrow(tab)-1, ncol(tab)-1)))
[1] 0.3846776
```

Поскольку сумма квадратов отклонений наблюдаемых и ожидаемых частот имеет тенденцию возрастать с ростом числа ячеек таблицы сопряженности, коэффициент Крамера для таблиц большей размерности может оказаться выше только по этой причине (аналогичным образом ведет себя и величина χ^2). В связи с этим рекомендуется сравнивать между собой коэффициенты Крамера, только если они рассчитаны для таблиц сопряженности одинаковой размерности.

Глава 10. Элементы многомерного анализа

До настоящего момента мы рассматривали методы анализа одномерных данных, описывающих одну переменную. Даже тогда, когда мы применяли множественный регрессионный анализ, основной задачей оставалось объяснение поведения только одной зависимой переменной (отклика). На практике исследователь часто имеет дело с комплексом признаков, как в нашем примере с анализом морфологической изменчивости пчел. Помимо частных вопросов о влиянии тех или иных факторов на характер распределения отдельных признаков, исследователя может интересовать общая картина о влиянии факторов на поведение всего комплекса признаков. В этом случае анализируемая переменная отклика является векторной или многомерной, то есть состоит из нескольких компонентов, соответствующих отдельным признакам (переменным).

Для работы с такими данными используются методы многомерного статистического анализа. Часть из них предназначена для проверки традиционных статистических гипотез, часть решает совершенно другие задачи (например, снижение размерности или распознавание групп), в связи с чем нередко возникают ошибки в интерпретации результатов. Большинство методов многомерного статистического анализа основываются на работе с матричной алгеброй, для понимания которой необходимо знание соответствующего математического аппарата. Не вдаваясь в вычислительные тонкости, мы сосредоточим внимание на практических аспектах применения основных методов.

10.1. Многомерный дисперсионный анализ

Как несложно понять из названия, многомерный дисперсионный анализ является обобщением обычного одномерного дисперсионного анализа. Он предназначен для выявления отличий между группами по совокупности средних значений комплекса признаков. Нуль-гипотеза заключается в равенстве векторов средних значений во всех группах.

Сформулируем сразу конкретную задачу. Предположим, нас интересуют отличия между тремя породами пчел по комплексу количественных морфологических признаков (пока оставим в стороне длину хоботка, поскольку мы ранее выяснили, что этот признак не подчиняется нормальному распределению). Мы имеем три независимых выборки пчел, каждая из которых характеризуется вектором средних значений.

```
> load("apis.rda")
> m <- rbind(colMeans(apis[apis$breed == "carnica", 2:6]),
+       colMeans(apis[apis$breed == "caucasica", 2:6]),
+       colMeans(apis[apis$breed == "mellifera", 2:6]))
> rownames(m) <- c("carnica", "caucasica", "mellifera")
> t(m)
                carnica caucasica mellifera
tergit.length  2.389581  2.385632  2.469166
tergit.width   4.966272  4.873053  5.130930
wing.width     3.238970  3.163380  3.203719
wing.length    9.405860  9.350536  9.436175
cubital.index 36.861561 49.702671 54.508925
```

Векторы средних мы рассчитали с помощью функции `colMeans()`, которая вычисляет средние по столбцам матрицы или фрейма. Каждый столбец выведенной матрицы средних представляет собой вектор средних значений, описывающих выборку пчел определенной породы. Задача многомерного дисперсионного анализа заключается в проверке гипотезы о равенстве этих векторов. Другими словами, мы хотим выяснить, отличаются ли породы по комплексу морфологических признаков.

Для формирования модели многомерного дисперсионного анализа можно воспользоваться функциями `aov()`, `lm()` либо `manova()`, которые приведут к идентичным результатам. Отличия будут только в поведении вспомогательных методов (например, функции `summary()`). Основным аргументом, который нужно передать этим функциям, является формула, в которой слева от тильды будет находиться не вектор, а матрица, столбцы которой соответствуют изучаемым переменным.

```
> model <- aov(as.matrix(apis[, 2:6]) ~ apis$breed)
```

В данном случае мы использовали в формуле часть фрейма `apis` с интересующими нас переменными, при этом осуществив приведение фрейма к матричной структуре (функция `as.matrix()`). Аналогичного результата можно добиться с помощью функции `cbind()`, которая сформирует матрицу из отдельных векторов переменных. Следующая спецификация модели идентична предыдущей:

```
> model <- aov(cbind(tergit.length, tergit.width, wing.width,
+                    wing.length, cubital.index) ~ breed, data = apis)
```

Как и в случае одномерного дисперсионного анализа, основную таблицу результатов можно получить с помощью функции `anova()`:

```
> anova(model)
Analysis of Variance Table
```

```
             Df  Pillai approx F num Df den Df    Pr(>F)
(Intercept)   1 0.99974   457699      5    587 < 2.2e-16 ***
breed         2 0.87692       92     10   1176 < 2.2e-16 ***
Residuals   591
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Мы получили таблицу дисперсионного анализа. Первая строка таблицы (`Intercept`) описывает базовый уровень, его значимость свидетельствует только об отличии вектора глобальных средних от нулевого вектора. Основное внимание нужно обратить на вторую строку (`breed`), в которой и представлены результаты анализа влияния нашего фактора (породной принадлежности). В многомерном дисперсионном анализе несколько показателей используются в качестве критериальной статистики:

– след Пиллая (Pillai’s trace);

– лямбда Уилкса (Wlks’ lambda);

– след Хотеллинга-Лоули (Hotelling-Lawley’s trace);

– наибольший характеристический корень Роя (Roy’s largest root).

Распределение этих статистик в условиях нуль-гипотезы в общем случае неизвестно, однако они могут быть преобразованы в показатель, который аппроксимируется *F*-распределением Фишера. По умолчанию в R используется след Пиллая, этот показатель представлен во втором столбце таблицы (`Pillai`), далее следует результат его преобразования в величину *F* (`approx F`), числа степеней свободы числителя (`num Df`) и знаменателя (`den Df`) для *F*-распределения, а также соответствующее *p*-значение (`Pr(>F)`). В нашем случае межпородные отличия по комплексу морфологических признаков статистически значимы ($p < 0.05$).

Для того чтобы использовать другую критериальную статистику в функции `anova()` нужно использовать аргумент `test`, которому передается одно из следующий значений: «`Pillai`» (след Пиллая, используется по умолчанию), «`Wilks`» (лямбда Уилкса), «`Hotelling`» (след Хотеллинга-Лоули) либо «`Roy`» (наибольший характеристический корень Роя).

```
> anova(model, test = "Wilks")
Analysis of Variance Table

             Df    Wilks approx F num Df den Df    Pr(>F)
(Intercept)   1 0.000256   457699      5    587 < 2.2e-16 ***
breed         2 0.298076       98     10   1174 < 2.2e-16 ***
Residuals   591
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

При использовании лямбды Уилкса изменились значение *F*, числа степеней свободы, но общий вывод остался неизменным: влияние фактора высоко значимо.

Параметрический многомерный дисперсионный анализ основывается на предположении о том, что данные в группах подчиняются многомерному нормальному распределению, при этом матрицы ковариаций (аналог дисперсии для многомерного случая) в группах идентичны. Основная проблема заключается в том, что эти предположения крайне редко выполняются на практике.

Для проверки гипотезы о многомерном нормальном распределении данных используется многомерная версия критерия согласия Шапиро-Уилка, которая реализована в функции `mshapiro.test()` пакета `mvnormtest`. Этой функции передается матрица, строки которой соответствуют переменным, а столбцы – наблюдениям. Чтобы привести данные, хранящиеся во фрейме, к такому виду, нужно воспользоваться функцией транспонирования `t()`. Применим многомерный критерий Шапиро-Уилка к выборкам пчел:

```
> library(mvnormtest)
> mshapiro.test(t(apis[apis$breed == "carnica", 2:6]))
 Shapiro-Wilk normality test

data:  Z
W = 0.9236, p-value = 8.683e-10
> mshapiro.test(t(apis[apis$breed == "caucasica", 2:6]))
 Shapiro-Wilk normality test

data:  Z
W = 0.984, p-value = 0.008278
> mshapiro.test(t(apis[apis$breed == "mellifera", 2:6]))
 Shapiro-Wilk normality test

data:  Z
W = 0.6239, p-value = 1.173e-15
```

Во всех трех случаях выявлены значимые отклонения от многомерного нормального распределения.

Для проверки гипотезы о гомогенности матриц ковариаций используется так называемый *M*-критерий Бокса, который реализован в функции `boxM()` пакета `biotools`. Этой функции первым аргументом передается матрица данных (на сей раз без транспонирования), вторым – группирующий вектор.

```
> library(biotools)
> boxM(as.matrix(apis[, 2:6]), apis$breed)
 Box's M-test for Homogeneity of Covariance Matrices
```

```
data:  as.matrix(apis[, 2:6])
Chi-Sq (approx.) = 120.52, df = 30, p-value = 8.351e-13
```

Гетерогенность матриц ковариаций также статистически значима.

Таким образом, применение параметрической статистики к нашему набору данных необоснованно. Необходимо воспользоваться непараметрическим методом. Мы приведем реализацию наиболее простого подхода, который основан на применении перестановочной процедуры (permutation test). Этот метод является вычислительно интенсивным и требует применения рандомизации. Суть метода заключается в том, что распределение критериальной статистики в условиях нуль-гипотезы определяется не аналитически, а с помощью компьютерного моделирования. В частности, в случае однофакторного анализа, который мы сейчас рассматриваем, осуществляется большое число случайных перестановок наблюдений относительно уровней фактора. Для каждой перестановки проводится полный анализ и рассчитывается значение критериальной статистики. После осуществления перестановки данные оказываются распределены по группам случайным образом, что можно считать эквивалентом состояния в условиях нуль-гипотезы: группы являются случайными выборками из одной и той же совокупности. Осуществляя множество таких перестановок, можно восстановить распределение интересующей нас статистики. После этого наблюдаемое значение статистики для исходных данных сравнивается с восстановленным распределением и рассчитывается *p*-значение. Для этого находится доля значений в восстановленном распределении таких же или еще более экстремальных, чем наблюдаемое значение. Восстановление распределения путем рандомизации не основано на каких-либо предположениях о характере распределения исходных данных, поэтому этот метод в полной мере является непараметрическим.

Реализуем перестановочный критерий для однофакторного многомерного дисперсионного анализа в функции `manova.perm()`.

```
> manova.perm <- function(dat, gr, test = "Wilks", nperm = 1000)
+ {
+     perm <- rep(0, nperm+1)
+     for (ii in 1:nperm)
+         perm[ii] <- anova(aov(dat ~ sample(gr)), test = test)[2,2]
+     perm[nperm+1] <- obs <- anova(aov(dat ~ gr), test = test)[2,2]
+     if (test == "Wilks") p.value <- sum(perm <= obs) / (nperm+1)
+     else p.value <- sum(perm >= obs) / (nperm+1)
+     return(p.value)
+ }
```

Функция принимает матрицу переменных (`dat`), группирующий вектор (`gr`), название критериальной статистики (`test`, по умолчанию лямбда Уилкса) и число перестановок (`nperm`, по умолчанию 1000), а возвращает *p*-значение. Воспользуемся этой функцией для проведения анализа наших данных. Поскольку все манипуляции будут проведены большое число раз, вычисления могут занять длительное время (зависит от мощности компьютера).

```
> manova.perm(as.matrix(apis[, 2:6]), apis$breed)
[1] 0.000999001
> manova.perm(as.matrix(apis[, 2:6]), apis$breed, test = "Pillai")
[1] 0.000999001
```

В качестве примера мы провели анализ с применением лямбды Уилкса и следа Пиллая, оба раза получены значимые отличия. Таким образом, перестановочный критерий подтвердил результаты параметрического анализа. Так бывает не всегда. При отклонении от предположений параметрического анализа (нормальность распределения и гомогенность матриц ковариации) следует использовать непараметрический метод.

10.2. Анализ главных компонент

Основной проблемой многомерного дисперсионного анализа являются сложности при интерпретации результатов. В предыдущем разделе мы выяснили, что три породы пчел значимо различаются по имеющемуся набору стандартных морфологических признаков, однако мы не можем указать, в чем именно заключаются эти различия. В случае одномерного анализа можно выяснить, какой признак у какой породы имеет большее или меньшее значение, но как быть в случае одновременного анализа множества признаков?

Одним из подходов к решению обозначенной проблемы является снижение размерности данных. В разделе 17.1 мы выявили значимую корреляцию в 5 парах признаков из 10. Это означает, что наши морфологические данные в значительной степени избыточны для выявления каких-либо тенденций. Это довольно типичная ситуация при анализе многомерных данных. Избавиться от этой избыточности позволяет анализ главных компонент.

Главные компоненты – это линейные комбинации исходных переменных, то есть суммы значений всех переменных, помноженных на коэффициенты, называемые нагрузками (loadings). Для обозначения главных компонент традиционно используется аббревиатура PC (principal component). Главные компоненты рассчитываются по формуле:

$$PC_k = \sum_{j=1}^{m} a_{kj} \cdot x_j ,$$

где k – номер главной компоненты, x_j – значение j-й переменной, a_{kj} – нагрузка j-й переменной в k-й главной компоненте, суммирование ведется по m переменным.

Число главных компонент равно числу переменных и если мы работаем с набором из 5 морфологических признаков, то получим 5 главных компонент. Главные компоненты в совокупности содержат столько же информации, сколько содержится в исходных переменных, однако она целенаправленно перераспределена. Весовые коэффициенты a_{kj} подбираются таким образом, чтобы первая главная компонента объясняла максимум изменчивости, то есть была наиболее информативной. Вторая главная компонента содержит максимум информации, оставшейся в наборе данных после выделения первой главной компоненты и так далее. Последняя главная компонента наименее информативна. Мерой информативности главных компонент является дисперсия их значений: чем она больше, тем информативнее компонента. Ключевым свойством главных компонент является то, что они не коррелируют между собой, в отличие от исходных переменных. Именно за счет этого достигается концентрация информации.

После расчета главных компонент с ними можно работать как с обычными переменными и проводить любые виды анализа. При этом обычно для анализа оставляют несколько первых главных компонент, отбрасывая наименее информативные, в чем и заключается снижение размерности исходных данных.

Мы не будем разбирать математические особенности построения главных компонент, которые основаны на линейной алгебре и манипуляциях с матрицами, а рассмотрим конкретные функции для работы с главными компонентами в среде R.

Для проведения анализа и расчета весовых коэффициентов служат функции `prcomp()` и `princomp()`, которые отличаются методом расчета. Считается, что метод сингулярного разложения, используемый функцией `prcomp()`, дает более точные численные результаты, поэтому мы будем работать именно с этой функцией.

```
> apis.pca <- prcomp(apis[, 2:6])
> class(apis.pca)
[1] "prcomp"
```

```
> apis.pca
Standard deviations:
[1] 9.93236046 0.21432068 0.13353077 0.08481345 0.06115225
Rotation:
                         PC1          PC2          PC3          PC4
tergit.length  0.0014006681 -0.163232863  0.121858366 -0.084264356
tergit.width   0.0006970522 -0.643690238  0.725084331 -0.126958519
wing.width    -0.0027400949 -0.245306870 -0.034795944  0.967681239
wing.length   -0.0020331373 -0.706286346 -0.676895768 -0.200918071
cubital.index  0.9999929552 -0.001430831 -0.002147686  0.002449585
                        PC5
tergit.length -0.975398958
tergit.width   0.209276465
wing.width    -0.046896614
wing.length    0.050985408
cubital.index  0.001195501
```

Функции prcomp() передается матрица данных либо фрейм. Мы передали часть фрейма apis, содержащую 5 количественных переменных, которые мы анализировали в предыдущем разделе. Результатом является сложно устроенный список класса «prcomp». При его выводе в консоль распечатываются стандартные отклонения главных компонент (Standard deviations) и матрица нагрузок (Rotation). Дополнительную информацию можно получить с помощью функции summary():

```
> summary(apis.pca)
Importance of components:
                          PC1     PC2     PC3     PC4     PC5
Standard deviation     9.9324 0.21432 0.13353 0.08481 0.06115
Proportion of Variance 0.9992 0.00047 0.00018 0.00007 0.00004
Cumulative Proportion  0.9992 0.99971 0.99989 0.99996 1.00000
```

В консоль выводится таблица важности компонент. Первая строка отображает стандартные отклонения, вторая – долю объясненной дисперсии, третья – накопленную долю объясненной дисперсии, то есть сумму по всем компонентам от первой до данной (накопленная доля для последней компоненты всегда будет равна единице). Для нашего набора данных 99.9 % объясненной дисперсии приходится на первую главную компоненту, информативность остальных минимальна. Так произошло не потому, что у нас крайне сильно скоррелированные переменные и львиная доля информации может быть сконцентрирована в единственной компоненте. Главная причина в том, что анализ главных компонент крайне чувствителен к масштабу варьирования данных. Рассмотрим длину тергита и кубитальный индекс.

```
> summary(apis$tergit.length)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.107   2.355   2.400   2.403   2.450   2.674
> sd(apis$tergit.length)
[1] 0.07274029
```

Длина тергита варьирует от 2.107 до 2.674 и характеризуется стандартным отклонением 0.073.

```
> summary(apis$cubital.index)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  19.93   38.12   45.73   45.44   52.31   74.57
> sd(apis$cubital.index)
[1] 9.932291
```

Кубитальный индекс варьирует от 19.93 до 74.57 и характеризуется стандартным отклонением 9.93. Эти две переменные имеют разную размерность (миллиметры и проценты) и варьируют в совершенно разных масштабах. Если при анализе придавать численным значениям этих переменных одинаковое значение, то окажется, что вклад кубитального индекса будет определяющим. Если же преобразовать длину тергита из миллиметров в микроны, то ситуация изменится на противоположную.

Избавиться от такой неоднозначности и привести переменные к одному масштабу варьирования позволяет *z*-преобразование, т.е. стандартизация с использованием параметров нормального распределения: вычитание средней и деление разности на стандартное отклонение (см. разд. 3.1). Преобразованные переменные будут эквиваленты исходным, но при этом будут характеризоваться нулевым средним и единичным стандартным отклонением.

В R такое преобразование переменных к стандартному масштабу может быть осуществлено с помощью функции `scale()`, но в контексте анализа главных компонент проще воспользоваться аргументом `scale` функции `prcomp()` и установить его в значение `TRUE`. Приведенные ниже выражения приведут к одинаковому результату.

```
> apis.pca <- prcomp(scale(apis[,2:6]))
> apis.pca <- prcomp(apis[, 2:6], scale = TRUE)
```

Рассмотрим теперь таблицу важности полученных главных компонент:

```
> summary(apis.pca)
Importance of components:
                          PC1    PC2    PC3    PC4     PC5
Standard deviation     1.4249 1.1428 0.7940 0.7439 0.69262
Proportion of Variance 0.4061 0.2612 0.1261 0.1107 0.09594
Cumulative Proportion  0.4061 0.6673 0.7934 0.9041 1.00000
```

После масштабирования первая главная компонента объясняет 40.6 % дисперсии, вторая – 26.1 % и так далее. Доля последней пятой главной компоненты составляет около 9.5 %.

При анализе главных компонент всегда возникает вопрос о том, сколько компонент следует оставить для дальнейшего анализа, а сколько отбросить как малоинформативные. Универсального алгоритма отбора не существует, поэтому руководствоваться следует здравым смыслом и стоящими перед дальнейшим исследованием задачами. Одной из возможных стратегий является фиксация порогового значения объясненной дисперсии, ниже которого мы не можем позволить себе опуститься. Например, мы могли заранее решить, что нам необходим набор компонент, объясняющих не менее 75 % общей изменчивости в нашем наборе данных. Тогда мы оставили бы для дальнейшего анализа три компоненты, которые в сумме объясняют 79 % изменчивости (см. нижнюю строку таблицы важности компонент).

Другая распространенная стратегия – так называемый критерий каменной осыпи. Строится график объясненной дисперсии, который всегда убывает. Такой график часто и называют графиком каменной осыпи (screeplot). Этот график можно получить, просто передав объект класса «prcomp» функции plot():

```
> plot(apis.pca, xlab = "Главные компоненты",
+      main = "График каменной осыпи", col = "lightgrey")
```

Результат представлен на Рис. 27. Смысл метафоры каменной осыпи в том, что внизу осыпи камни нагромождены очень полого, но всегда есть резкое изменение наклона в месте перехода осыпи в склон горы. Если что-то похожее мы видим на нашем графике, то следует оставить те компоненты, которые объясняют значительно больше дисперсии, чем ближайшая следующая. В нашем случае это первые две главные компоненты.

Наконец, существует так называемый критерий Кайзера-Харриса, который в случае масштабированных данных сводится к тому, что необходимо оставить главные компоненты, дисперсия которых больше единицы, поскольку эти компоненты несут больше информации, чем в среднем каждая исходная переменная (напомним, что дисперсия масштабированных переменных составляет единицу). В нашем случае согласно критерию Кайзера-Харриса следует оставить первые две главные компоненты, в чем легко убедиться, если добавить на график каменной осыпи горизонтальную линию.

```
> abline(h = 1)
```

Таблица нагрузок главных компонент хранится в элементе rotation полученного объекта класса «prcomp», а сами значения главных компонент для

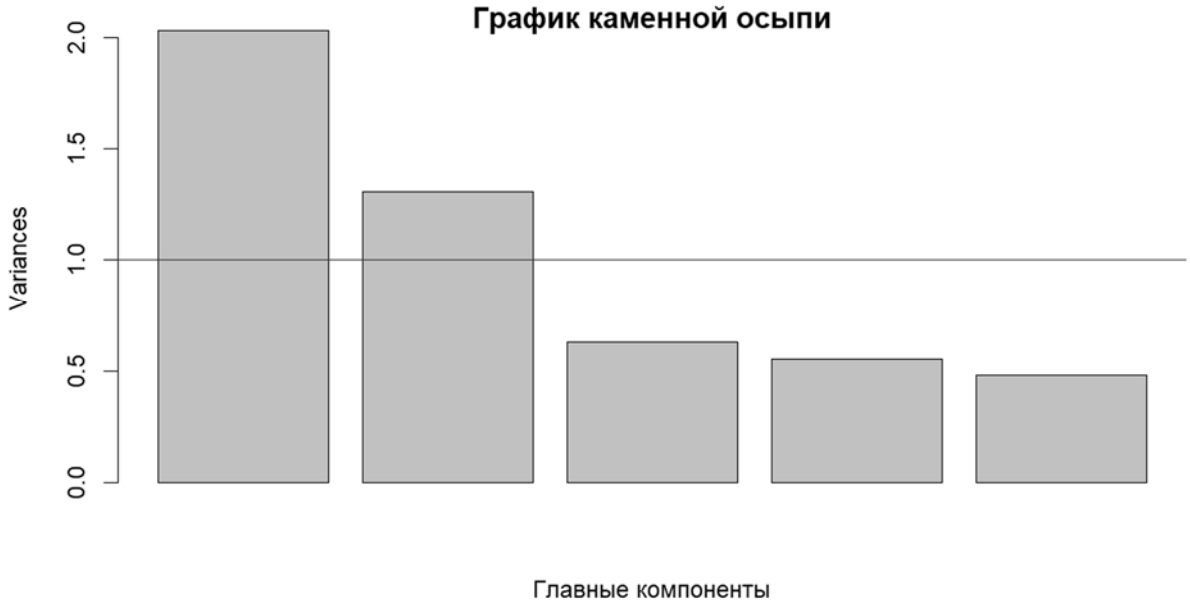

Рис. 27. Зависимость дисперсии главных компонент от номера

каждого из объектов (в нашем случае – отдельных пчел) содержатся в элементе x, который имеет размерность исходной матрицы данных.

В качестве примера дальнейшего использования главных компонент проведем анализ влияния породной принадлежности на первую главную компоненту. Анализ проведем с помощью непараметрического критерия Крускала-Уоллиса, чтобы не углубляться в детали проверки параметрических допущений, необходимых для проведения традиционного дисперсионного анализа.

```
> kruskal.test(apis.pca$x[,1], apis$breed)

 Kruskal-Wallis rank sum test

data:  apis.pca$x[, 1] and apis$breed
Kruskal-Wallis chi-squared = 140.0299, df = 2, p-value < 2.2e-16
```

Влияние породы на первую главную компоненту значимо.

Другое типичное использование главных компонент – визуализация межгрупповых отличий. Обычно для этих целей строят графики рассеяния в пространстве главных компонент (первая против второй, первая против третьей, вторая против третьей и так далее). В нашем примере достаточно построить один график, на котором мы отобразим по оси абсцисс первую главную компоненту, по оси ординат – вторую главную компоненту, а породную принадлежность обозначим цветом. Для цветовой кодировки пород мы воспользуемся своего рода трюком: преобразуем текстовые идентификаторы пород в фактор, а затем используем числовые значения фактора как вектор, кодирующий цвета.

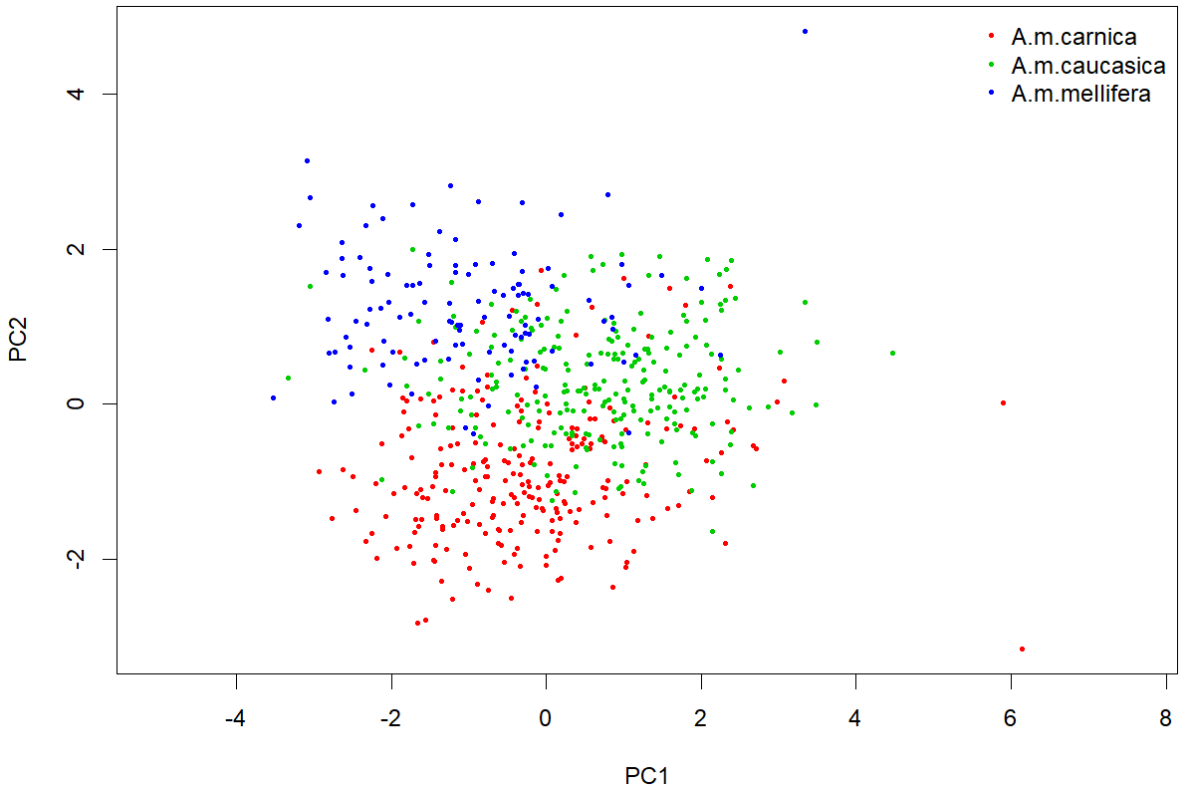

Рис. 28. График рассеяния пчел в пространстве двух первых главных компонент.

```
> apis$breed <- factor(apis$breed)
> plot(apis.pca$x[, 1], apis.pca$x[, 2], pch = 16, cex = 0.5,
+      col = as.numeric(apis$breed) + 1, asp = 1,
+      xlab = "PC1", ylab = "PC2")
> legend("topright", pch = 16, pt.cex = 0.5, col = 2:4, bty = "n",
+        legend = c("A.m.carnica", "A.m.caucasica", "A.m.mellifera"))
```

Результат представлен на Рис. 28. Мы получили облако точек, причем хорошо заметно, что оно состоит из трех пересекающихся подмножеств: пчелы породы *A. m. mellifera* группируются в верхней левой части общего облака, пчелы породы *A. m. carnica* группируются в нижней части облака, а пчелы породы *A. m. caucasica* – в правой части. Такая картина находится в полном соответствии с результатами проведенного многомерного дисперсионного анализа (см. предыдущий раздел).

В заключение раздела отметим, что анализ главных компонент не основан на каких-либо предположениях о характере распределения данных. В этом смысле он не является параметрическим методом. На самом деле анализ главных компонент не является статистическим методом в обычном смысле, а скорее геометрическим. В ходе анализа не проверяются гипотезы и не делается никаких выводов относительно генеральной совокупности, а просто одна система

координат (исходные переменные) преобразуется в другую (главные компоненты). Результаты анализа (собственно главные компоненты) могут использоваться как обычные переменные в статистическом анализе, но это уже следующий этап анализа.

10.3. Линейный дискриминантный анализ

В том случае, когда по совокупности переменных выявлены значимые отличия между группами (например, с помощью многомерного дисперсионного анализа), имеет смысл построить модель, описывающую структуру этих отличий. Совокупность методов, связанных с построением и использованием такой модели, называется дискриминантным анализом.

Дискриминантный анализ используется в двух целях. Во-первых, построенная модель позволяет делать прогноз групповой принадлежности новых объектов, то есть классифицировать их по группам. Во-вторых, анализ структуры самой модели (ее коэффициентов) может помочь понять, какие из переменных вносят большой вклад в различия между группами, а какие – малый. На основе такого анализа можно делать выводы об информативности переменных.

В определенном смысле дискриминантный анализ аналогичен множественной регрессии (см. раздел 8.2). Отличие заключается в том, что в регрессионном анализе зависимая переменная, подлежащая объяснению, является количественной, а в дискриминантном анализе – это качественная переменная, задающая группировку.

Технически дискриминантный анализ близок анализу главных компонент, в основе его модели также лежит совокупность линейных комбинаций исходных переменных, которые называются линейными дискриминантами (linear discriminant, LD) или каноническими переменными (canonical variate) и вычисляются по формуле:

$$LD_k = \sum_{j=1}^{m} c_{kj} \cdot (x_j - \bar{x}_j),$$

где k – номер линейного дискриминанта, x_j – значение j-й переменной, $\bar{x}_j$ – среднее значение j-й переменной (по всем группам), c_{kj} – вес j-й переменной в k-ом линейном дискриминанте, суммирование ведется по m переменным. Число дискриминантов определяется количеством групп: поскольку для различения двух групп требуется один дискриминант, то число дискриминантов меньше числа групп на единицу (но не может превышать число переменных). Первый

дискриминант подбирается таким образом, чтобы максимизировать отличия между группами, второй и последующие также максимизируют межгрупповую изменчивость, но подбираются ортогонально первому и всем предыдущим. Следствием ортогональности является отсутствие корреляции между дискриминантами. Аналогичным свойством обладают главные компоненты. Ключевое отличие между главными компонентами и линейными дискриминантами заключается в том, что первые подбираются так, чтобы максимизировать общую изменчивость, а последние – чтобы максимизировать межгрупповую изменчивость.

Для классификации новых объектов по группам служат специальные классификационные функции:

$$CF_g = d_{g0} + \sum_{j=1}^{m} d_{gj} \cdot x_j ,$$

где CF_g – классификационная функция для g-й группы, d_{gj} – вес j-й переменной, d_{g0} – свободный член. Обратите внимание, что классификационные функции не являются линейными комбинациями исходных переменных, поскольку содержат дополнительный свободный член. Кроме того, число классификационных функций на единицу больше числа дискриминантов, поскольку каждой группе соответствует своя функция. Для каждого объекта рассчитываются значения функций и объект классифицируется в ту группу, значение функции для которой оказалось наибольшим.

Дискриминантный анализ реализован в нескольких пакетах среды R, мы рассмотрим применение функции `lda()` пакета «`MASS`». В разделе 19.1 было установлено, что набор из 5 количественных морфологических признаков значимо различается между тремя породами медоносных пчел. Построим модель для классификации пчел на основе имеющихся данных.

Функции `lda()` можно передать матрицу данных и группирующую переменную или воспользоваться формулой. Ниже приведены два эквивалентных вызова:

```
> library("MASS")
> apis.lda <- lda(apis[,2:6], grouping = apis$breed)
> apis.lda <- lda(breed ~ tergit.length + tergit.width + wing.width +
+                     wing.length + cubital.index, data = apis)
```

Функция возвращает список из 10 элементов, относящийся к классу «`lda`».

```
> apis.lda
Call:
lda(breed ~ tergit.length + tergit.width + wing.width + wing.length +
    cubital.index, data = apis)
```

```
Prior probabilities of groups:
  carnica caucasica mellifera
0.4040404 0.4040404 0.1919192

Group means:
          tergit.length tergit.width wing.width wing.length
carnica        2.389581     4.966272   3.238970    9.405860
caucasica      2.385632     4.873053   3.163380    9.350536
mellifera      2.469166     5.130930   3.203719    9.436175
          cubital.index
carnica        36.86156
caucasica      49.70267
mellifera      54.50893

Coefficients of linear discriminants:
                     LD1         LD2
tergit.length  3.4701330  4.23626366
tergit.width   1.5830453  5.92357579
wing.width    -2.3189705  1.55872039
wing.length    0.4154899 -1.03463502
cubital.index  0.1349551 -0.03586981

Proportion of trace:
   LD1    LD2
0.7493 0.2507
```

При выводе в консоль печатается форма вызова функции (`Call`), априорные вероятности групп, которые по умолчанию пропорциональны объемам выборок (`Prior probabilities of groups`), матрица групповых средних (`Group means`), коэффициенты c_{kj} линейных дискриминантов (`Coefficients of linear discriminants`), а также доли дисперсии, объясненной дискриминантами (`Proportion of trace`). В нашем случае почти 75 % дисперсии приходится на долю первого дискриминанта и 25 % – на долю второго.

Для того чтобы рассчитать значения линейных дискриминантов для конкретной особи, необходимы глобальные средние. Рассчитаем в качестве примера значения дискриминантов для первой особи.

```
> apis.m <- colMeans(apis[, 2:6])
> w <- apis.lda$scaling
> sum((apis[1,2:6] - apis.m) * w[,1])
[1] -0.189467
> sum((apis[1,2:6] - apis.m) * w[,2])
[1] -2.022341
```

Умножение на весовые коэффициенты и суммирование можно осуществить с помощью перемножения матриц (оператор %*%), а для вычитания средних (то

есть для центрирования) можно воспользоваться функцией scale() с аргументом scale = FALSE, который предотвращает масштабирование (то есть деление каждой переменной на стандартное отклонение). Следующее выражение рассчитывает значения линейных дискриминантов для всех особей в нашем наборе данных:

```
> apis.ld.scores <- scale(apis[,2:6], scale = FALSE) %*% w
```

Чаще всего нет необходимости рассчитывать значения дискриминантов подобным образом, проще воспользоваться функцией predict(), которой передается объект класса «lda». Эта функция служит в первую очередь для классификации объектов, но она также возвращает значения дискриминантов в элементе x.

```
> apis.pr <- predict(apis.lda)
```

Результатом является список из следующих элементов: class – фактор с прогнозами относительно принадлежности объектов к группам, posterior – апостериорные вероятности принадлежности объектов к группам, x – значения линейных дискриминантов для каждого объекта. Мы провели апостериорную классификацию и рассчитали значения дискриминантов для исходного набора данных, по которому строилась модель. Чтобы классифицировать новые объекты, функции predict() нужно передать фрейм с новыми данными через аргумент newdata. Предположим, что у нас есть морфологические данные по пяти признакам трех пчел неизвестной породной принадлежности. Сформируем новый фрейм данных, структура которого максимально соответствует исходному (как минимум – в части названий задействованных в формуле переменных), и получим прогнозы для новых данных.

```
> apis.new <- matrix(c(2.31, 5.06, 3.3,  9.5, 40.1,
+                      2.41, 4.5,  3.15, 9.3, 47.5,
+                      2.6,  5.25, 3.31, 9.4, 58.4),
+                    ncol = 5, byrow = TRUE)
> colnames(apis.new) <- names(apis)[2:6]
> apis.new <- as.data.frame(apis.new)
> (apis.new.pr <- predict(apis.lda, newdata = apis.new))
$class
[1] carnica   caucasica mellifera
Levels: carnica caucasica mellifera

$posterior
       carnica   caucasica    mellifera
1 0.8558169818 0.136972195 0.0072108231
2 0.0526819022 0.946724187 0.0005939104
3 0.0005571318 0.007404209 0.9920386590
```

```
$x
         LD1        LD2
1 -1.0679255  0.4262509
2 -0.3440026 -2.7596428
3  2.6441304  2.2428799
```

Первая пчела с вероятностью 85.5 % отнесена к породе *A. m. carnica*, вторая с вероятностью 94.7 % – к породе *A. m. caucasica*, а третья с вероятностью 99.2 % – к породе *A. m. mellifera*.

Вернемся к анализу нашей дискриминантной модели. Основным показателем адекватности модели является точность прогноза. Для ее расчета строится так называемая таблица классификации, которая представляет собой двумерную таблицу сопряженности с частотами истинных и прогнозируемых значений.

```
> (class.tab <- table(apis$breed, apis.pr$class))

           carnica caucasica mellifera
  carnica       200        34         6
  caucasica      37       190        13
  mellifera       5        22        87
```

Строки полученной таблицы соответствуют истинным породам, столбцы – прогнозам модели. Главная диагональ содержит частоты верных прогнозов для каждой породы. Так, из 240 пчел породы *A. m. carnica* 200 были классифицированы верно, 34 были отнесены к породе *A. m. caucasica*, а оставшиеся 6 – к породе *A. m. mellifera*. Общая доля правильных прогнозов рассчитывается как отношение суммы элементов главной диагонали к общей сумме таблицы:

```
> sum(diag(class.tab))/sum(class.tab)
[1] 0.8030303
```

Итак, доля правильных прогнозов составляет более 80 %. Однако не нужно забывать о том, что рассчитанные прогнозы получены для тех же данных, по которым строилась модель. Вероятность правильного распознавания для новых данных может оказаться существенно ниже. Для того, чтобы улучшить оценку правильного распознавания, существует процедура перекрестной проверки (cross-validation). Она заключается в том, что из полного набора данных последовательно отбрасывается по одному наблюдению. По оставшимся данным строится дискриминантная модель и делается прогноз для отброшенного наблюдения. Такие прогнозы гораздо лучше подходят для проверки адекватности модели, поскольку они делаются для наблюдений, не участвовавших в построении модели.

Перекрестная проверка реализована в базовой функции `lda()`. Чтобы ее провести, нужно воспользоваться аргументом `CV = TRUE`. В этом случае функция вернет объект с вектором классификации `class` и матрицей апостериорных вероятностей принадлежности к группам `posterior`.

```
> apis.lda.cv <- lda(breed ~ tergit.length + tergit.width + wing.width
+              + wing.length + cubital.index, data = apis, CV = TRUE)
```

Полученный объект можно использовать для построения новой таблицы классификации:

```
> (class.tab.cv <- table(apis$breed, apis.lda.cv$class))

           carnica caucasica mellifera
  carnica      198        36         6
  caucasica     38       189        13
  mellifera      6        24        84
```

Мы видим, что число правильных прогнозов уменьшилось незначительно.

```
> sum(diag(class.tab.cv))/sum(class.tab.cv)
[1] 0.7929293
```

Доля правильных прогнозов снизилась до 79.9 %, что свидетельствует о высокой адекватности модели. Оценить модель можно также визуально, построив график рассеяния наблюдений в пространстве линейных дискриминантов, чаще всего – двух первых.

```
> plot(apis.ld.scores[, 1], apis.ld.scores[, 2], pch = 1, cex = 0.7,
+      col = as.numeric(apis$breed) + 1, asp = 1,
+      xlab = "LD1", ylab = "LD2")
> legend("bottomright", pch = 1, pt.cex = 0.7, col = 2:4, bty = "n",
+        legend = c("A.m.carnica", "A.m.caucasica", "A.m.mellifera"))
```

Сравните полученный график с Рис. 28, на котором представлено рассеяние этих же данных в пространстве главных компонент. Обратите внимание на то, что первый дискриминант расположен таким образом, чтобы максимизировать межгрупповую изменчивость, то есть расстояние между центрами групповых кластеров. Добавим эти центры (они называются центроидами) на наш график. Координаты центроидов можно рассчитать путем усреднения значений дискриминантов, а можно воспользоваться групповыми средними, результат будет идентичный.

```
> breed.center <- (apis.lda$means -
+                  matrix(rep(apis.m, 3), nrow = 3, byrow = T)) %*% w
> points(breed.center, pch = 21, cex = 2, bg = "white",
+        col = c("darkred", "darkgreen", "darkblue"))
> points(breed.center, pch = 4, cex = 1.3,
+        col = c("darkred", "darkgreen", "darkblue"))
```

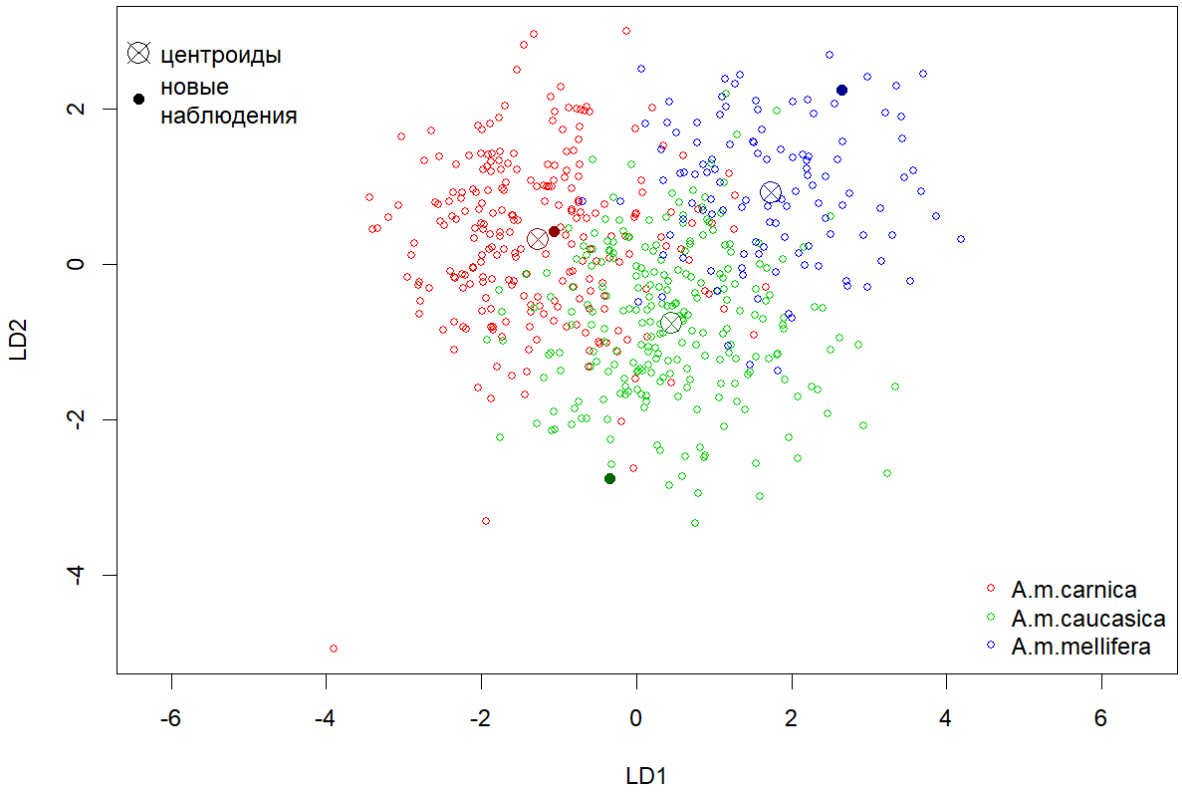

Рис. 29. График рассеяния пчел в пространстве линейных дискриминантов.

Центроид породы *A. m. carnica* занимает левое положение, центроид породы *A. m. caucasica* находится в центре, а центроид *A. m. mellifera* – справа. Добавим теперь на график новые наблюдения, которые мы классифицировали с помощью нашей модели. Итоговый график представлен на Рис. 29.

```
> points(apis.new.pr$x, col = c("darkred", "darkgreen", "darkblue"),
+                        pch = 16)
> legend("topleft", legend = c("центроиды", "новые\nнаблюдения"),
+              pch = c(13, 16), pt.cex = c(2, 1), bty = "n")
```

Вернемся к анализу апостериорных вероятностей породной принадлежности новых наблюдений.

```
> apis.new.pr$posterior
       carnica   caucasica    mellifera
1 0.8558169818 0.136972195 0.0072108231
2 0.0526819022 0.946724187 0.0005939104
3 0.0005571318 0.007404209 0.9920386590
```

Первое наблюдение располагается очень близко к центроиду породы *A. m. carnica*, поэтому с вероятностью 85 % оно относится к этой породе. Тем не менее, остается небольшая вероятность (13 %) принадлежности этой пчелы к породе *A. m. caucasica*. Вторая и третья же пчелы хотя и расположены далеко от центроидов пород, к которым они в итоге отнесены, но они отклоняются в

направлении, противоположном от центроидов других пород, поэтому вероятности принадлежности их к другим породам крайне невелики.

До настоящего момента мы занимались использованием нашей модели для прогнозирования породной принадлежности и выясняли ее общую адекватность. Обратимся теперь к анализу вклада отдельных исходных переменных в дискриминацию между породами. Рассмотрим весовые коэффициенты линейных дискриминантов.

```
> apis.lda$scaling
                     LD1         LD2
tergit.length  3.4701330  4.23626366
tergit.width   1.5830453  5.92357579
wing.width    -2.3189705  1.55872039
wing.length    0.4154899 -1.03463502
cubital.index  0.1349551 -0.03586981
```

Обращают на себя внимание высокие значения весов первых трех переменных и относительно небольшие – двух последних. Такая ситуация не отражает реального вклада переменных в дискриминацию, поскольку в ходе анализа переменные центрируются, но не масштабируются (обратите внимание на то, что при расчете значений дискриминантов из значений переменных вычитается общее среднее). Чтобы оценить относительный вклад отдельных переменных в линейные дискриминанты необходимо провести анализ на данных, подвергнутых *z*-преобразованию.

```
> apis.lda.scaled <- lda(scale(apis[,2:6]), gr = apis$breed)
```

Результаты анализа в плане апостериорной классификации и даже значений линейных дискриминантов для каждого наблюдения не изменятся. Однако весовые коэффициенты переменных теперь можно сравнивать между собой, поскольку переменные масштабированы к единичной дисперсии.

```
> apis.lda.scaled$scaling
                      LD1        LD2
tergit.length  0.25241846  0.3081470
tergit.width   0.26834310  1.0041094
wing.width    -0.23501185  0.1579657
wing.length    0.07408199 -0.1844758
cubital.index  1.34041277 -0.3562694
```

Первый дискриминант в значительной степени определяется кубитальным индексом, вес которого максимален (1.34) и существенно превышает веса остальных переменных. Наибольший вклад во второй дискриминант дает ширина тергита (вес 1.004). Таким образом, эти две переменные можно считать наиболее информативными для распознавания породной принадлежности пчел.

В заключение раздела рассмотрим вопрос о предположениях, лежащих в основе линейного дискриминантного анализа. Аналогично анализу главных компонент, метод построения линейных дискриминантов является чисто геометрическим и не требует выполнения каких-либо предположений о характере распределения данных. Однако методы построения классификационных функций и оценки апостериорных вероятностей принадлежности объектов к группам основываются на предположениях о многомерном нормальном распределении данных и о гомогенности матриц ковариации в группах. В этом смысле линейный дискриминантный анализ является параметрическим методом, который может приводить к ошибочным выводам (например, может быть построена неоптимальная классификационная функция) в случае невыполнения предпосылок. Тем не менее, даже в случае отклонений от предположений нормальности и гомогенности применение дискриминантного анализа для построения модели для классификации данных представляется корректным, если результаты не используются для проверки гипотез и выводов относительно генеральной совокупности.

В нашем примере анализа породной принадлежности пчел данные отклоняются как от многомерного нормального распределения, так и от предположения о гомогенности матриц ковариации (см. раздел 10.1). Однако мы построили дискриминантную модель, которая показала довольно высокую степень адекватности (79 % правильно классифицированных объектов по результатам перекрестной проверки). Дальнейшее использование этой модели в практических целях вполне оправданно. Однако при анализе вклада отдельных переменных в линейные дискриминанты мы ограничились качественным анализом (на уровне сравнения абсолютных величин весовых коэффициентов) и воздержались от применения статистических критериев. Такие методы существуют, но все они основаны на использовании многомерного дисперсионного анализа и их использование было бы некорректным.

Список рекомендуемой литературы

1. Волкова П.А., Шипунов А.Б. Статистическая обработка данных в учебно-исследовательских работах / под ред. Ю.А. Багринцева, Б.М. Каплан, Е.Н. Сиднева. М.: Экспресс, 2008. 60 с.
2. Буховец А.Г., Москалев П.В., Богатова В.П., Бирючинская Т.Я. Статистический анализ данных в системе R. Учебное пособие / под ред. А.Г. Буховца. Воронеж: ВГАУ, 2010. 124 с.
3. Зарядов И.С. Введение в статистический пакет R: типы переменных, структуры данных, чтение и запись информации, графика. М.: Изд-во РУДНБ, 2010. 207 с.
4. Зарядов И.С. Статистический пакет R: теория вероятностей и математическая статистика. М.: Изд-во РУДНБ, 2010. 141 с.
5. Кабаков Р.И. R в действии: Анализ и визуализация данных в программе R. М.: ДМК Пресс, 2014. 580 с.
6. Мастицкий С.Э., Шитиков В.К. Статистический анализ и визуализация данных с помощью R. М.: ДМК Пресс, 2015. 496 с.
7. Савельев А.А., Мухарамова С.С., Пилюгин А.Г. Использование языка R для статистической обработки данных. Учебно-методическое пособие. Казань: Казанский университет, 2007. 28 с.
8. Шипунов А.Б., Балдин Е.М., Волкова П.А., Коробейников А.И., Назарова С.А., Петров С.В., Суфиянов В.Г. Наглядная статистика. Используем R! М.: ДМК Пресс, 2012. 298 с.
9. Шитиков В.К., Розенберг Г.С. Рандомизация и бутстреп: статистический анализ в биологии и экологии с использованием R. Тольятти: Кассандра, 2014. 314 с.
10. Adler J. R in a Nutshell: A Desktop Quick Reference. 2nd ed. O'Reilly Media, 2012. 724 pp.
11. Dalgaard P. Introductory Statistics with R. 2nd ed. Springer, 2008. 364 pp.
12. Logan M. Biostatistical Design and Analysis Using R: A Practical Guide. Wiley-Blackwell, 2010. 576 pp.
13. Venables W.N., Ripley B.D. Modern Applied Statistics with S. 4th ed. Springer, 2002. 505 pp.
14. Zuur A.F., Ieno E.N., Meesters E. A Beginner's Guide to R. Springer, 2009. 220 pp.

Программное обеспечение и Интернет-ресурсы

1. R Project: The R Project for Statistical Computing / R Foundation for Statistical Computing. URL: http://cran.r-project.org/.
2. CRAN: The Comprehensive R Archive Network / R Foundation for Statistical Computing. URL: http://cran.r-project.org/.
3. R Studio web-site. URL: http://www.rstudio.com/.
4. R-bloggers: R news and tutorials contributed by (750) R bloggers. URL: http://www.r-bloggers.com/.
5. Quick-R: accessing the power of R / Kabacoff R. URL: http://www.statmethods.net/.
6. R: Анализ и визуализация данных / Мастицкий С. URL: http://r-analytics.blogspot.ru/.

Василий Николаевич **Якимов**

Основы анализа биомедицинских и экологических данных в среде R

Часть 2

Учебное пособие

Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский Нижегородский государственный университет
им. Н.И. Лобачевского»
603950, Нижний Новгород, пр. Гагарина, 23.
Подписано в печать . Формат 60×84 1/16.
Бумага офсетная. Печать офсетная. Гарнитура Таймс.
Усл. печ. л. 12,5. Уч.-изд. л.
Заказ № . Тираж 100 экз.

Отпечатано в типографии Нижегородского государственного
университета им. Н.И. Лобачевского
603600, г. Нижний Новгород, ул. Большая Покровская, 37