Анализ данных с R (II).

© А. Б. Шипунов*, А. И. Коробейников‡, Е. М. Балдин**

*dactylorhiza@gmail.com
‡asl@math.spbu.ru
**E.M.Baldin@inp.nsk.su

Эмблема **R** взята с официального сайта проекта `http://developer.r-project.org/Logo/`

Оглавление

Глава 5

Работа с двумя переменными

Повествование об **R** возобновляется на новом уровне. С января (LXF100/101) по апрель (LXF104) в журнале была опубликована серия из четырёх статей представляющих этот замечательный язык обработки данных. То был предварительный обзор возможностей, а сейчас начинается серьёзная работа.

5.1. Проверка гипотез однородности

Две разные выборки называются однородными, если они одинаково распределены. Проверка «гипотез однородности» в математической статистике занимает особое место. Этот факт связан с тем, что для практических приложений, как правило, характерны задачи сравнения двух (и более) групп наблюдений. Сравнивать выборки можно совершенно разными способами: исследователя может интересовать различие в средних, медианах, дисперсиях при разнообразных предположениях относительно самих наблюдений.

5.1.1. Параметрические критерии проверки однородности выборок

Величины, имеющие нормальное распределение, в реальных экспериментах возникают совершенно естественным образом, При измерении любой характеристики всегда имеется ошибка измерения. Если предполагать, что ошибка прибора имеет нормальное распределение, то среднее отвечает за систематическую ошибку, а дисперсия — за величину случайной ошибки. Поэтому, критерии представленные в данном разделе предполагают, что выборка имеет нормальное распределение. Если это заранее не известно, то этот факт нужно проверять отдельно

(см. раздел 5.2). В противном случае все выводы, полученные на основе этих критериев, будут ошибочными.

Двухвыборочный критерий Стьюдента равенства средних

Двухвыборочный t-критерий используется для проверки гипотезы о равенстве средних в двух независимых выборках, имеющих нормальное распределение. В своей классической постановке критерий проводится в предположении равенства дисперсий в двух выборках. В **R** реализована модификация критерия, позволяющая избавиться от этого предположения.

Воспользуемся классическим набором данных, который использовался в оригинальной работе Стьюдента (псевдоним Уильяма Сили Госсета). В упомянутой работе производилось сравнение влияния двух различных снотворных на увеличение продолжительности сна (рис. 5.1). В **R** этот массив данных доступен под названием `sleep` в пакете **stats**. В столбце `extra` содержится среднее приращение продолжительности сна после начала приёма лекарства (по отношению к контрольной группе), а в столбце `group` — код лекарства (первое или второе).

```
> plot(extra ~ group, data = sleep)
```

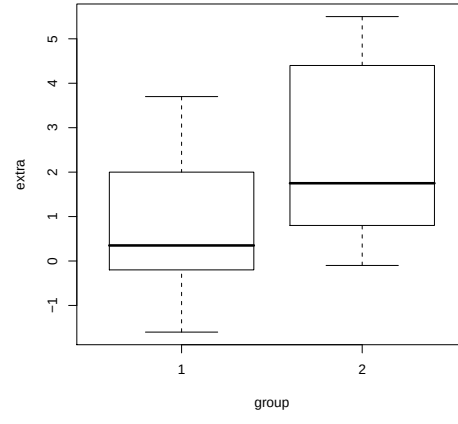

Рис. 5.1. Среднее приращение продолжительности сна после начала приёма разных лекарств в двух группах по отношению к контрольной.

Влияние лекарства на каждого человека индивидуально, но среднее увеличение продолжительности сна можно считать вполне логичным показателем «силы» лекарства. Основываясь на этом предположении, попробуем проверить при

помощи t-критерия, значимо ли различие в средних для этих двух выборок (соответствующих двум разным лекарствам):

```
> with(sleep, t.test(extra[group == 1],
+             extra[group == 2], var.equal = FALSE))

        Welch Two Sample t-test

data:  extra[group == 1] and extra[group == 2]
t = -1.8608, df = 17.776, p-value = 0.0794
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.3654832  0.2054832
sample estimates:
mean of x mean of y
     0.75      2.33
```

Параметр `var.equal` позволяет выбрать желаемый вариант критерия: оригинальный t-критерий Стьюдента в предположении одинаковых дисперсий (`TRUE`) или же t-критерий в модификации Уэлча (Welch), свободный от этого предположения (`FALSE`).

Хотя формально гипотеза о равенстве средних не отвергается на стандартных уровнях значимости, мы видим, что возвращаемое p-значение (0.0794) достаточно маленькое, поэтому к данном результату стоит относится критично. Это означает, что возможно, стоит попробовать другие методы для проверки гипотезы, увеличить количество наблюдений, ещё раз убедиться в нормальности распределений и т. д.

Можно ли использовать t-критерий, если необходимо сравнить среднее в *зависимых* выборках (например, при сравнении какого-либо жизненного показателя у пациента до и после лечения)? Ответ на этот вопрос: «Да, можно, но не обычный, а модифицированный специальным образом под такую процедуру». В литературе такая модификация называется *парным t-критерием*. Ничего специального для использования парного t-критерия в **R** делать не надо. Достаточно выставить опцию `paired` в `TRUE`:

```
> with(sleep, t.test(extra[group == 1],
+             extra[group == 2], paired = TRUE))

        Paired t-test

data:  extra[group == 1] and extra[group == 2]
t = -4.0621, df = 9, p-value = 0.002833
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
```

```
 -2.4598858 -0.7001142
sample estimates:
mean of the differences
                  -1.58
```

Здесь видно, что парный t-критерий отвергает гипотезу о равенстве средних с достаточно большой надёжностью. Следует отметить, что использованные в этом примере выборки не были «парными», поэтому, строго говоря, применять парный t-критерий нельзя, и все выводы носят сугубо иллюстративный характер.

Двухвыборочный критерий Фишера равенства дисперсий

Естественной характеристикой «размаха» распределения при нормальной модели является дисперсия. Предположим, что потребовалось проверить гипотезу об отсутствии различий в дисперсиях двух выборок. При этом не хочется делать абсолютно никаких допущений относительно значений средних в этих выборках. Такая задача может возникнуть, например, при сравнении точности двух приборов. Напомним, что при наличии ошибок измерений мерой систематической ошибки прибора является среднее, а случайной — дисперсия. Систематическую ошибку иногда можно уменьшить за счёт точной калибровки прибора. Случайную же ошибку убрать почти никогда не представляется возможным. В связи с этим задача проверки равенства дисперсий (например, при сравнении эталонного прибора и проверяемого) становится достаточно актуальной.

Решением такой задачи служит F-критерий Фишера (Fisher). В **R** он реализован в функции var.test():

```
> x <- rnorm(50, mean = 0, sd = 2);
> y <- rnorm(30, mean = 1, sd = 1);
> var.test(x, y)

        F test to compare two variances

data:  x and y
F = 3.8414, num df = 49, denom df = 29, p-value = 0.0002308
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 1.930003 7.227256
sample estimates:
ratio of variances
          3.841391
```

В этом примере участвуют две выборки с разным количеством наблюдений (50 и 30 для x и y, соответственно), разными средними (0 и 1) и дисперсиями ($2^2 = 4$ и $1^2 = 1$). Гипотеза о равенстве дисперсий безусловно отвергается (p-значение

мало). Кроме самого значения тестовой статистики, p-значения и величин степеней свободы функция выводит оценку отношения дисперсий и доверительный интервал от него. Значение оценки 3.55 не очень сильно отличается от истинного значения отношения дисперсий 4.

На самом деле, критерий проверяет несколько более общую гипотезу. Идёт проверка того, что отношение дисперсий двух выборок имеет какое-то наперёд заданное значение. Проверка гипотезы о равенстве дисперсий является частным случаем такой гипотезы.

Предполагаемое значение отношения дисперсий можно задать с помощью опции `ratio`:

```
> x <- rnorm(50, mean = 0, sd = 2);
> y <- rnorm(30, mean = 1, sd = 1);
> var.test(x, y, ratio = 4)

        F test to compare two variances

data:  x and y
F = 1.1097, num df = 49, denom df = 29, p-value = 0.7778
alternative hypothesis: true ratio of variances is not equal to 4
95 percent confidence interval:
 2.230136 8.351157
sample estimates:
ratio of variances
           4.43876
```

Здесь видно, что при задании истинного значения отношения дисперсий гипотеза не отвергается.

5.1.2. Непараметрические критерии проверки однородности выборок

Критерии, приведённые в предыдущем разделе работают только в предположении нормальности распределения данных. Что делать, если заранее известно, что выборки имеют другое распределение, или по каким-либо причинам проверить нормальность не получается? В таких случаях используются так называемые *непараметрические* критерии, т. е. критерии свободные от предположения какой-либо параметрической модели данных. Естественно, ввиду того, что эти критерии оперируют гораздо меньшим «объёмом информации», то они не смогут заметить те тонкие различия, которые были бы обнаружены при использовании параметрических критериев.

Критерий Вилкоксона

Критерий Вилкоксона (Wilcoxon), двухвыборочный вариант которого ещё известен под именем критерия Манна-Уитни, является непараметрическим аналогом t-критерия.

Стандартный набор данных `airquality` содержит информацию о величине озона в воздухе города Нью-Йорка с мая по сентябрь 1973 года. Проверим, например, гипотезу о том, что распределения уровня озона в мае и в августе было одинаковым:

```
> wilcox.test(Ozone ~ Month, data = airquality,
+                       subset = Month %in% c(5, 8))

        Wilcoxon rank sum test with continuity correction

data:  Ozone by Month
W = 127.5, p-value = 0.0001208
alternative hypothesis: true location shift is not equal to 0
```

Критерий отвергает гипотезу о согласии распределений уровня озона в мае и в августе с достаточно большой надёжностью. В принципе это достаточно легко вытекает из «общих соображений», так как уровень озона в воздухе сильно зависит от солнечной активности, температуры и ветра. И если с солнечной активностью всё в порядке:

```
> wilcox.test(Solar.R ~ Month, data = airquality,
+                       subset = Month %in% c(5, 8))

        Wilcoxon rank sum test with continuity correction

data:  Solar.R by Month
W = 422.5, p-value = 0.4588
alternative hypothesis: true location shift is not equal to 0
```

то распределения ветра и температуры, напротив, сильно различаются:

```
> wilcox.test(Temp ~ Month, data = airquality,
+                       subset = Month %in% c(5, 8))

        Wilcoxon rank sum test with continuity correction

data:  Temp by Month
W = 27, p-value = 1.747e-10
alternative hypothesis: true location shift is not equal to 0
```

```
> wilcox.test(Wind ~ Month, data = airquality,
+                         subset = Month %in% c(5, 8))

        Wilcoxon rank sum test with continuity correction

data:  Wind by Month
W = 687.5, p-value = 0.003574
alternative hypothesis: true location shift is not equal to 0
```

Эти различия легко видны, скажем, на изображении ящиков с усами (рис. 5.2):

```
> boxplot(Temp ~ Month, data = airquality,
+                       subset = Month %in% c(5, 8))
```

но в сложных случаях только использование критериев позволяет получать объективные результаты.

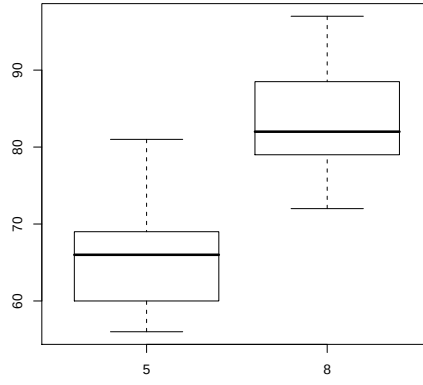

Рис. 5.2. Распределение температуры в Нью-Йорке в мае и в августе 1973 года.

По умолчанию критерий проверяет гипотезу о том, что распределения двух выборок отличаются лишь постоянным и известным сдвигом (который, в свою очередь, по умолчанию равен нулю). Задать его можно при помощи параметра `mu`. Например:

```
> x <- rnorm(50, mean = 0);
> y <- rnorm(50, mean = 2);
> wilcox.test(x, y);
```

```
        Wilcoxon rank sum test with continuity correction

data:  x and y
W = 230, p-value = 2.091e-12
alternative hypothesis: true location shift is not equal to 0

> wilcox.test(x, y, mu = -2);

        Wilcoxon rank sum test with continuity correction

data:  x and y
W = 1335, p-value = 0.5602
alternative hypothesis: true location shift is not equal to -2
```

Критерий Манна-Уитни так же, как и t-критерий, тоже бывает парным. Для использования парной модификации необходимо выставить опцию `paired` в значение `TRUE`. Задание опции `conf.int` позволяет получить оценку различия в сдвиге между двумя выборками[1] и доверительный интервал для него:

```
> x <- rnorm(50, mean = 0);
> y <- rnorm(50, mean = 2);
> wilcox.test(x, y, conf.int = TRUE);

        Wilcoxon rank sum test with continuity correction

data:  x and y
W = 227, p-value = 1.803e-12
alternative hypothesis: true location shift is not equal to 0
95 percent confidence interval:
 -2.418617 -1.500210
sample estimates:
difference in location
             -1.941988

> wilcox.test(x, y, mu = -2);

        Wilcoxon rank sum test with continuity correction

data:  x and y
W = 1292, p-value = 0.7748
alternative hypothesis: true location shift is not equal to -2
```

[1] Естественно, при условии, что кроме сдвига распределения двух выборок ничем не отличаются

Гипотеза о полном равенстве распределений была отвергнута, а гипотеза о том, что одно распределение отличает от другого просто сдвигом не отвергнута, что и требовалось показать.

Непараметрические критерии сравнения масштаба

Дисперсия является естественным параметром масштаба для выборки из нормальной совокупности. Этот факт позволяет заменить гипотезу о совпадении масштабов на гипотезу о совпадении дисперсий в случае нормального распределения. При отказе от нормальной модели дисперсия уже не является характеристикой масштаба и поэтому надо честно проверять гипотезу именно о совпадении масштабов распределений двух выборок.

Формально предполагается, что одна из выборок имеет распределение с плотностью $f(x-a)$, другая — $sf\left(s\left(x-a\right)\right)$. Здесь функция плотности f и параметр сдвига a считаются неизвестными. Мы заинтересованы в проверке совпадения масштабов у двух выборок, то есть проверке того, что $s=1$.

В стандартном пакете **stats** реализованы два классических непараметрических критерия, позволяющих проверить равенство масштабов: критерий Ансари-Брэдли (Ansari-Bradley) и критерий Муда (Mood). Начнём с первого.

```
> ansari.test(runif(50), rucunif(50, max = 2))

        Ansari-Bradley test

data:  runif(50) and runif(50, max = 2)
AB = 1404, p-value = 0.07526
alternative hypothesis: true ratio of scales is not equal to 1
```

Распределение выборок в данном случае отличается только масштабом. Критерий отвергает гипотезу о совпадении дисперсий на стандартных уровнях значимости, как это и должно быть. Всё аналогично для критерия Муда:

```
> mood.test(runif(50), runif(50, max = 2))

        Mood two-sample test of scale

data:  runif(50) and runif(50, max = 2)
Z = -2.5685, p-value = 0.01021
alternative hypothesis: two.sided
```

5.2. Проверка гипотез нормальности распределения

Большая часть инструментов статистического вывода работает в предположении о том, что выборка получена из нормальной совокупности. За примерами далеко ходить не надо: t-критерий, критерий Фишера, построение доверительных интервалов для линейной регрессии и проверка гипотезы о линейной независимости двух выборок.

Как уже отмечалось ранее, нормальное распределение естественным образом возникает практически везде, где речь идёт об измерении с ошибками. Более того, в силу центральной предельной теоремы, распределение многих выборочных величин (например, выборочного среднего) при достаточно больших объёмах выборки хорошо аппроксимируется нормальным распределением вне зависимости от того, какое распределение было у выборки исходно.

В связи с этим становится понятным, почему проверке распределения на нормальность стоит уделить особое внимание. В дальнейшем речь пойдёт о так называемых *критериях согласия* (goodness-of-fit tests). Проверяться будет не просто факт согласия с нормальным распределением с определёнными фиксированными значениями параметров, а несколько более общий факт принадлежности распределения к семейству нормальных распределений со всевозможными значениями параметров.

Основные классические критерии проверки на нормальность собраны в пакете **nortest**. Пакет можно установить с CRAN при помощи вызова функции install.packages():

```
> install.packages(pkgs=c("nortest"))
```

При этом Tcl/Tk-форма предлагает выбрать зеркало CRAN откуда скачать пакет. Подключить установленный пакет можно при помощи функции library():

```
> library(nortest)
```

Может возникнуть вопрос: «А зачем столько много разных критериев для проверки одного факта? Нельзя ли выбрать наилучший и всегда его использовать?». Ответ на этот вопрос не утешителен: «В общем случае, к сожалению, нельзя». Попробуем это объяснить.

Напомним, что ошибкой первого рода статистического критерия называется факт принятия альтернативной гипотезы, в ситуации когда верна гипотеза по умолчанию. Например, пусть статистический критерий используется для разграничения доступа к какому-нибудь ресурсу. Тогда отказ в доступе для авторизованного пользователя и будет ошибкой первого рода для такого критерия. Ясно, что возможна «симметричная» ошибочная ситуация, заключающаяся в предоставление доступа к ресурсу не авторизованному пользователю. Такая ошибка называется *ошибкой второго рода*: принятие критерием гипотезы по умолчанию в ситуации, когда она не верна (то есть имеет место альтернативная гипотеза).

Как правило, чувствительность выбранного критерия к ошибкам первого рода мы настраиваем самостоятельно (как раз выбором тех самых «стандартных уровней значимости», с которым и сравниваем p-значение, выданное критерием). С ошибкой второго рода всё гораздо хуже: её вероятность сильно зависит от выбранной альтернативной гипотезы и является неотъемлемой характеристикой самого критерия. В редких случаях удаётся построить критерий, который является наилучшим (это так называемые *равномерно наиболее мощные критерии*), и, к сожалению, это невозможно для нашей задачи. В нашем случае нулевая гипотеза формулируется просто: *выборка имеет нормальное распределение с некоторыми неизвестными параметрами*, а альтернативная гипотеза — это полное её отрицание. Альтернативная гипотеза гораздо «богаче» нулевой, туда входят все распределения, отличные от нормального. Для того и понадобилась целая батарея критериев: какие-то работают лучше против одного семейства альтернатив, другие — против другого. Использование всего набора позволяет быть хоть как-то уверенным в том, что распределение, не являющееся нормальным будет «распознано» хотя бы одним из критериев.

Критерий Лиллифорса

Критерий Лиллифорса (Lilliefors) является вариантом известного классического критерия Колмогорова-Смирнова, специально модифицированного для проверки нормальности. Эта модификация существенна. Для проверки гипотезы нормальности *нельзя* использовать классический непараметрический критерий Колмогорова-Смирнова, реализованный в функции `ks.test()`. Критерий Лиллифорса реализован в функции `lillie.test()`:

```
> lillie.test(rnorm(100, mean = 6, sd = 4));

        Lilliefors (Kolmogorov-Smirnov) normality test

data:  rnorm(100, mean = 6, sd = 4)
D = 0.0463, p-value = 0.8621

> lillie.test(runif(100, min = 2, max = 4));

        Lilliefors (Kolmogorov-Smirnov) normality test

data:  runif(100, min = 2, max = 4)
D = 0.0732, p-value = 0.2089
```

Критерии Крамера-фон Мизеса и Андерсона-Дарлинга

Первый критерий известен в русскоязычной литературе под именем критерия ω^2 или критерия Смирнова. Эти критерии менее известны, но обычно работают

гораздо лучше, нежели критерий Лиллифорса. Они реализованы в функциях cvm.test() и ad.test() соответственно:

```
> cvm.test(rnorm(50, mean = 6, sd = 4));

        Cramer-von Mises normality test

data:  rnorm(50, mean = 6, sd = 4)
W = 0.0321, p-value = 0.8123

> ad.test(runif(50, min = 2, max = 4));

        Anderson-Darling normality test

data:  runif(50, min = 2, max = 4)
A = 1.5753, p-value = 0.0004118
```

Критерий Шапиро-Франсиа

Этот критерий работает достаточно хорошо в большинстве не очень «сложных» случаев. Получить p-значение можно посредством функции sf.test():

```
> sf.test(rexp(50, rate = 2));

        Shapiro-Francia normality test

data:  rexp(50, rate = 2)
W = 0.7803, p-value = 2.033e-06
```

Критерий хи-квадрат Пирсона

В отличие от задач проверки пропорций, критерий хи-квадрат обычно очень плохо работает в задачах проверки распределения на нормальность. Вероятность ошибки второго рода очень велика для достаточно широкого класса альтернативных распределений. В связи с этим, использовать его не рекомендуется.

Тем не менее реализация его предоставлена функцией pearson.test(). У этой функции есть булевская опция `adjusted`, которая позволяет внести поправки в p-значение из-за наличия двух неизвестных параметров. Рекомендуемая последовательность действий такая: получить два p-значения, одно, соответствующее `adjusted=TRUE`, второе — `adjusted=FALSE`. Истинное p-значение обычно находится между. Кроме того, полезно поварьировать объем выборки и посмотреть, насколько сильно меняется p-значение. Если влияние объёма выборки сильное, то от использования критерия стоит отказаться во избежание ошибок.

```
> pearson.test(rnorm(50, mean = 6, sd = 4));

        Pearson chi-square normality test

data:  rnorm(50, mean = 6, sd = 4)
P = 5.2, p-value = 0.6356

> pearson.test(runif(50, min = -1, max = 1));

        Pearson chi-square normality test

data:  runif(50, min = -1, max = 1)
P = 7.6, p-value = 0.3692
```

► Теперь мы можем не только отличать нормальное распределения от «не нормального», но и сравнивать разные распределения. Это одна из самых первых ступенек на пути понимания сути данных, которые волей не волей приходится собирать для познания природы абсолютно любых явлений.

5.3. Взаимосвязь случайных величин

Одной из основных задач статистики является изучение зависимостей между данными. Под словом «зависимость» следует понимать зависимость в самом широком её смысле. Не однозначную функциональную зависимость, когда имея только один набор данным становится возможным полностью определить другой набор, а гораздо более общий случай, когда по одному набору данных можно получить хоть какую-нибудь информацию о втором.

5.3.1. Корреляция

Начнём всё же с функциональной зависимости, как наиболее просто формализуемой. Классическим инструментом для измерения *линейной зависимости* между двумя наборами данных является коэффициент корреляции. Коэффициент корреляции — это числовая величина, находящаяся в интервале от -1 до $+1$. Чем она больше по модулю (т. е. ближе к $+1$ или -1), тем выше линейная связь между наборами данных. Знак коэффициента корреляции показывает, в одном ли направлении изменяются наборы данных. Если один из наборов возрастает, а второй убывает, то коэффициент корреляции отрицателен, а если оба набора одновременно возрастают или убывают, то коэффициент корреляции положителен. Значение коэффициента корреляции по модулю равное 1 соответствует точной линейной зависимости между двумя наборами данных. Линейная зависимость является самой любимой у экспериментаторов всех мастей.

► Обратите внимание, что значение коэффициента корреляции близкое к нулю *не означает* независимости наборов данных. Коэффициент корреляции — это мера линейной зависимости, поэтому этот факт означает лишь отсутствие линейной зависимости, но не исключает любой другой. Отсутствие линейной зависимости равносильно независимости только для нормально распределённых выборок, факт нормальности, естественно, надо проверять отдельно.

Для вычисления коэффициента корреляции в R реализована функция cor:

```
> cor(5:15, 7:17)
[1] 1
> cor(5:15, c(7:16, 23))
[1] 0.9375093
```

В самом её простейшем случае ей передаются два аргумента (векторы одинаковой длины). Кроме того, возможен вызов функции с одним аргументом, в качестве которого может выступать матрица или набор данных (data frame). В этом случае функция cor вычисляет так называемую *корреляционную матрицу*, составленную из коэффициентов корреляций между столбцами матрицы или набора данных, взятых попарно:

```
>  cor(longley)

             GNP.deflator       GNP Unemployed Armed.Forces
GNP.deflator    1.0000000 0.9915892  0.6206334    0.4647442
GNP             0.9915892 1.0000000  0.6042609    0.4464368
Unemployed      0.6206334 0.6042609  1.0000000   -0.1774206
Armed.Forces    0.4647442 0.4464368 -0.1774206    1.0000000
Population      0.9791634 0.9910901  0.6865515    0.3644163
Year            0.9911492 0.9952735  0.6682566    0.4172451
Employed        0.9708985 0.9835516  0.5024981    0.4573074
               Population      Year  Employed
GNP.deflator    0.9791634 0.9911492 0.9708985
GNP             0.9910901 0.9952735 0.9835516
Unemployed      0.6865515 0.6682566 0.5024981
Armed.Forces    0.3644163 0.4172451 0.4573074
Population      1.0000000 0.9939528 0.9603906
Year            0.9939528 1.0000000 0.9713295
Employed        0.9603906 0.9713295 1.0000000
```

Если все данные присутствуют, то всё просто, но что делать, когда есть пропущенные наблюдения? Есть несколько способов, как вычислить корреляционную матрицу в этом случае. Для этого в команде cor есть опция `use`. По-умолчанию она равна `all.obs`, что при наличии хотя бы одного пропущенного наблюдения приводит к ошибки исполнения cor. Если опцию `use` приравнять значению `complete.obs`, то из данных до вычисления корреляционной матрицы удаляются

все наблюдения, в которых есть хотя бы один пропуск. Может оказаться так, что пропуски раскиданы по исходному набору данных достаточно хаотично и их много, так что после построчного удаления от матрицы фактически ничего не остаётся. В таком случае поможет попарное удаление пропусков, то есть удаляются строчки с пропусками не из всей матрицы сразу, а только лишь из двух столбцов непосредственно перед вычислением коэффициента корреляции. Для этого опцию `use` следует приравнять значению `pairwise.complete.obs`.

► В последнем случае следует принимать во внимание то, что коэффициенты корреляции вычисляются по *разному* количеству наблюдений и сравнивать их друг с другом может быть опасно.

```
> cor(swiss)
                 Fertility Agriculture Examination
Fertility        1.0000000  0.35307918  -0.6458827
Agriculture      0.3530792  1.00000000  -0.6865422
Examination     -0.6458827 -0.68654221   1.0000000
Education       -0.6637889 -0.63952252   0.6984153
Catholic         0.4636847  0.40109505  -0.5727418
Infant.Mortality 0.4165560 -0.06085861  -0.1140216
                   Education   Catholic Infant.Mortality
Fertility        -0.66378886  0.4636847  0.41655603
Agriculture      -0.63952252  0.4010951 -0.06085861
Examination       0.69841530 -0.5727418 -0.11402160
Education         1.00000000 -0.1538589 -0.09932185
Catholic         -0.15385892  1.0000000  0.17549591
Infant.Mortality -0.09932185  0.1754959  1.00000000

> # Создаём копию данных.
> swissNA <- swiss
> # Удаляем некоторые данные.
> swissNA[1,2] <- swissNA[7,3] <- swissNA[25,5] <- NA
> cor(swissNA)
Ошибка в cor(swissNA) : пропущенные наблюдения в cov/cor

> cor(swissNA, use = "complete")
                 Fertility Agriculture Examination
Fertility        1.0000000  0.37821953  -0.6548306
Agriculture      0.3782195  1.00000000  -0.7127078
Examination     -0.6548306 -0.71270778   1.0000000
Education       -0.6742158 -0.64337782   0.6977691
Catholic         0.4772298  0.40148365  -0.6079436
Infant.Mortality 0.3878150 -0.07168223  -0.1071005
                   Education   Catholic Infant.Mortality
Fertility        -0.67421581  0.4772298  0.38781500
```

```
Agriculture       -0.64337782  0.4014837 -0.07168223
Examination        0.69776906 -0.6079436 -0.10710047
Education          1.00000000 -0.1701445 -0.08343279
Catholic          -0.17014449  1.0000000  0.17221594
Infant.Mortality  -0.08343279  0.1722159  1.00000000

> cor(swissNA, use = "pairwise")

                  Fertility Agriculture Examination
Fertility         1.0000000  0.39202893  -0.6531492
Agriculture       0.3920289  1.00000000  -0.7150561
Examination      -0.6531492 -0.71505612   1.0000000
Education        -0.6637889 -0.65221506   0.6992115
Catholic          0.4723129  0.41520069  -0.6003402
Infant.Mortality  0.4165560 -0.03648427  -0.1143355
                    Education   Catholic Infant.Mortality
Fertility         -0.66378886  0.4723129  0.41655603
Agriculture       -0.65221506  0.4152007 -0.03648427
Examination        0.69921153 -0.6003402 -0.11433546
Education          1.00000000 -0.1791334 -0.09932185
Catholic          -0.17913339  1.0000000  0.18503786
Infant.Mortality  -0.09932185  0.1850379  1.00000000
```

Есть ещё два момента, на которые стоит обратить внимание. Первый — это ранговый коэффициент корреляции Спирмена (Spearman) ρ. Коэффициент ρ отражает меру монотонной зависимости и является более робастным (то есть он менее подвержен влиянию случайных «выбросов» в данных). Он полезен в случае, когда набор данных не получен выборкой из двумерного нормального распределения. Для подсчёта ρ достаточно приравнять опцию `method` значению `spearman`:

```
> x <- rexp(50);
> cor(x, log(x), method="spearman")
[1] 1
```

Можно сравнить, насколько сильно отличаются обычный коэффициента корреляции от коэффициента корреляции Спирмена:

```
> clP <- cor(longley)
> clS <- cor(longley, method = "spearman")
> i <- lower.tri(clP)
> cor(cbind(P = clP[i], S = clS[i]))
         P        S
P 1.000000 0.980239
S 0.980239 1.000000
```

Второй момент — это проверка гипотезы о значимости коэффициента корреляции. Это равносильно проверке гипотезы о равенстве нулю коэффициента корреляции. Если гипотеза отвергается, то влияние одного набора данных на другой считается *значимым*. Для проверки гипотезы используется функция cor.test:

```
> x <- rnorm(50)
> y <- rnorm(50, mean = 2, sd = 1);
> # Тестируем независимые данные.
> cor.test(x,y)

        Pearson's product-moment correlation

data:  x and y
t = 0.2496, df = 48, p-value = 0.804
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.2447931  0.3112364
sample estimates:
       cor
0.03600814

> # Тестируем линейно зависимые данные.
> cor.test(x, 2*x);

        Pearson's product-moment correlation

data:  x and 2 * x
t = Inf, df = 48, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 1 1
sample estimates:
cor
  1
```

Видно, что в первом случае гипотеза о равенстве нулю коэффициента корреляции не отвергается, что соответствует исходным данным. Во втором случае вызов был осуществлён с заведомо линейно-зависимыми аргументами и критерий отвергает гипотезу о равенстве нулю коэффициента корреляции с большим уровнем надёжности. Кроме непосредственно p-значения функция выводит оценку коэффициента корреляции и доверительный интервал для него. Выставить доверительный уровень для него можно с помощью опции `conf.level`. Также при помощи опции `method` можно выбирать, относительно какого коэффициента корреляции (простого или рангового) проводить проверку гипотезы значимости.

Следует признать, что смотреть на матрицу, полную чисел, не очень удобно. В **R** есть несколько способов, с помощью которых можно визуализировать корреляционную матрицу.

Первый способ — это использовать функцию symnum, которая выведет матрицу по-прежнему в текстовом виде, но все числа будут заменены на буквы, в зависимости от того, какому диапазону принадлежало значение:

```
> symnum(cor(longley))
             GNP. GNP U A P Y E
GNP.deflator 1
GNP          B    1
Unemployed   ,    ,   1
Armed.Forces .    .     1
Population   B    B   , . 1
Year         B    B   , . B 1
Employed     B    B   . . B B 1
attr(,"legend")
[1] 0 ‘ ’ 0.3 ‘.’ 0.6 ‘,’ 0.8 ‘+’ 0.9 ‘*’ 0.95 ‘B’ 1
```

Эта функция имеет большое количество разнообразных настроек, но по умолчанию они все выставлены в значения, оптимальные для отображения корреляционных матриц.

Второй способ — это графическое представление корреляционных коэффициентов. Идея проста: нужно разбить область от -1 до $+1$ на отдельные диапазоны, назначить каждому свой цвет, а затем всё это отобразить. Для этого следует воспользоваться функциями image и axis (рис. 5.3):

```
> C <- cor(longley)
> image(1:ncol(C), 1:nrow(C), C, col = rainbow(12),
+       axes = FALSE, xlab = "", ylab = "")
> # Подписи к осям.
>  axis(1, at = 1:ncol(C), labels=colnames(C))
>  axis(2, at = 1:nrow(C), labels=rownames(C), las = 2)
```

Ещё один интересный способ представления корреляционной матрицы предоставляется пакетом **ellipse**. В этом случае значения коэффициентов корреляции рисуются в виде эллипсов, отражающих форму плотности двумерного нормального распределения с данным значением корреляции между компонентами. Чем ближе значение коэффициента корреляции к $+1$ или -1 — тем более вытянутым становится эллипс. Наклон эллипса отражает знак. Для получения изображения необходимо вызвать функцию plotcorr (рис. 5.4):

```
> # Устанавливаем библиотеку.
> install.packages(pkgs=c("ellipse"))
> # Загружаем.
```

Рис. 5.3. Графическое представление корреляционной матрицы.

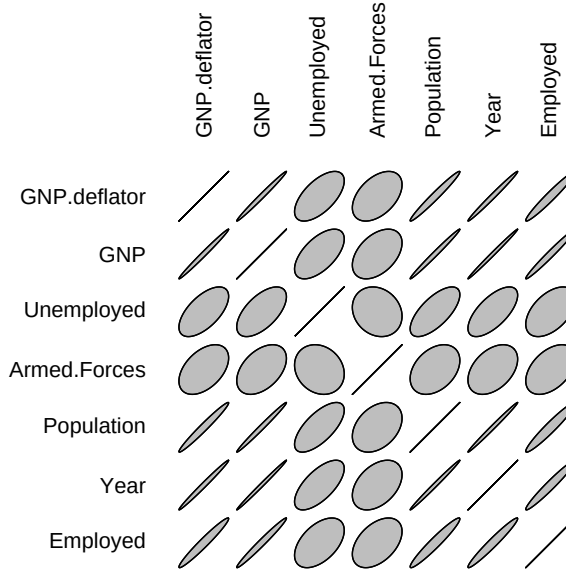

Рис. 5.4. Результат работы команды plotcorr из пакета **ellipse**.

```
> library(ellipse)
> # Используем.
> plotcorr(cor(longley))
```

5.3.2. Таблицы сопряжённости

Таблицы сопряжённости (contigency tables) — это удобный способ изображения категориальных переменных и исследования зависимостей между ними. Таблица сопряжённости представляет собой таблицу, ячейки которой индексируются градациями участвующих факторов, а числовое значение ячейки — количество наблюдений с данными градациями факторов. Построить таблицу сопряжённости можно с помощью функции table. В качестве аргументов ей нужно передать факторы, на основе которых будет строиться таблица сопряжённости:

```
> #  Таблица сопряжённости для выборки, имеющей
> # распределение Пуассона (n = 100, λ = 5).
> table(rpois(100,5))

 0  2  3  4  5  6  7  8  9 10 11
 1  7 18 17 22 13 13  4  1  1  3

> with(airquality, table(cut(Temp, quantile(Temp)), Month))

         Month
           5  6  7  8  9
  (56,72] 24  3  0  1 10
  (72,79]  5 15  2  9 10
  (79,85]  1  7 19  7  5
  (85,97]  0  5 10 14  5
```

R использует «честное» представление для трёх- и более мерных таблиц сопряжённости, то есть каждый фактор получает по своему измерению. Однако, это не очень удобно при выводе подобных таблиц на печать или сравнении с таблицами в литературе. Традиционно для этого используются «плоские» таблицы сопряжённости, когда все факторы, кроме одного, объединяются в один «многомерный» фактор и именно градации такого фактора используются при построении таблицы сопряжённости. Построить плоскую таблицу сопряжённости можно с помощью функцией ftable:

```
> ftable(Titanic, row.vars = 1:3)

                    Survived  No Yes
Class Sex    Age
1st   Male   Child            0   5
```

```
              Adult          118  57
      Female Child            0   1
              Adult           4 140
2nd   Male   Child            0  11
              Adult          154  14
      Female Child            0  13
              Adult           13  80
3rd   Male   Child           35  13
              Adult          387  75
      Female Child           17  14
              Adult           89  76
Crew  Male   Child            0   0
              Adult          670 192
      Female Child            0   0
              Adult           3  20
```

Опция `row.vars` позволяет указать номера переменных в наборе данных, которые следует объединить в один единый фактор, градации которого и будут индексировать строки таблицы сопряжённости. Опция `col.vars` проделывает то же самое, но для столбцов таблицы.

Функцию table можно использовать и для других целей. Самое простое — это подсчёт частот. Например, можно считать пропуски:

```
> d <- factor(rep(c("A","B","C"), 10), levels=c("A","B","C","D","E"))
> is.na(d) <- 3:4
> table(factor(d, exclude = NULL))

   A    B    C <NA>
   9   10    9    2
```

Функция mosaicplot позволяет получить графическое изображение таблицы сопряжённости[2] (рис. 5.5):

```
> mosaicplot(Titanic, main = "Survival␣on␣the␣Titanic", color = TRUE)
```

При помощи функции chisq.test[3] можно проверить гипотезу о независимости двух факторов. Например, проверим гипотезу о независимыми цвет глаз и волос:

```
> x <- margin.table(HairEyeColor, c(1, 2))
> chisq.test(x)
```

[2]Если стандартной функции plot передать в качестве аргумента таблицу сопряжённости, то вызывается именно эта функция

[3]Того же эффекта можно добиться если функции summary передать таблицу сопряжённости в качестве аргумента.

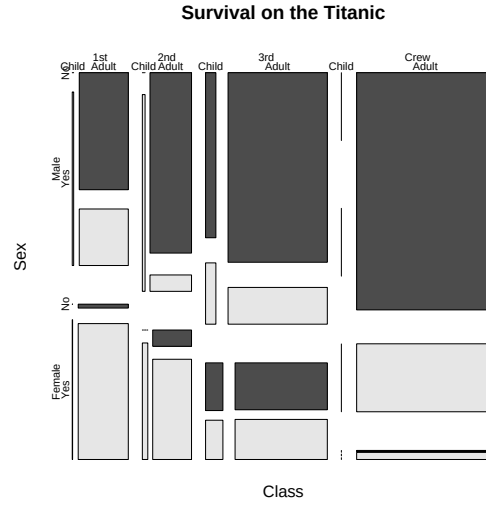

Рис. 5.5. Графическое представление таблицы сопряжённости.

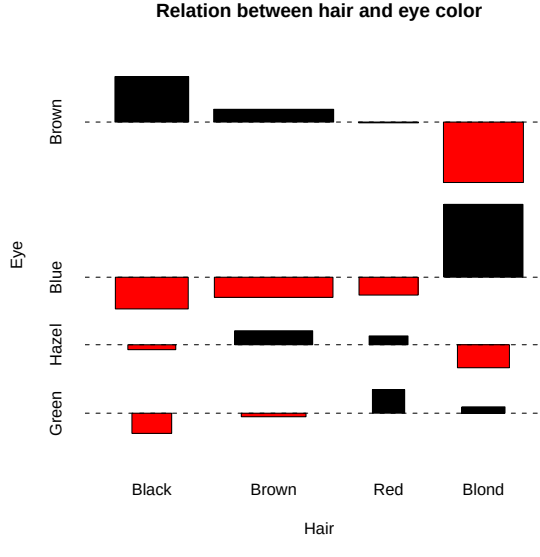

Рис. 5.6. Сравнение цвета глаз и волос с помощью функции assocplot.

```
        Pearson's Chi-squared test

data:  x
X-squared = 138.2898, df = 9, p-value < 2.2e-16
```

Набор данных `HairEyeColor` — это многомерная таблица сопряжённости. Здесь для суммирования частот по всем кроме двух «измерениям» использовалась функция margin.table. Таким образом в результате была получена двумерная таблица сопряжённости.

Чтобы зависимости между градациями двух факторов изобразить графически можно воспользоваться функцией assocplot. На рис. 5.6 показаны отклонения ожидаемых (при предположении независимости факторов) частот от наблюдаемых величин. Высота прямоугольника показывает абсолютную величину этого отклонения, а положение — знак отклонения. Ширина прямоугольника отображает собственно, саму ожидаемую величину значения в ячейка, но она не так информативна для предварительного анализа.

```
> x <- margin.table(HairEyeColor, c(1, 2))
> assocplot(x, main = "Relation␣between␣hair␣and␣eye␣color")
```

Здесь отчётливо видно, что, для людей со светлыми волосами характерен голубой цвет глаз и совсем не характерен карий цвет, а для обладателей чёрных волос ситуация в точности обратная.

► Поиск зависимостей между различными наборами данных — это любимое занятий человека разумного. Одно из самых больших интеллектуальных удовольствий — это обнаружение новой, ранее неизвестной, зависимости. Приятнее всего, если эта зависимость линейная.