МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

**Нижегородский государственный университет
им. Н.И. Лобачевского**

# Численные методы математической статистики в пакете R

Учебно-методическое пособие

Рекомендовано методической комиссией
Института ИТММ для студентов ННГУ,
обучающихся по направлениям подготовки
010302 «Прикладная математика и информатика»,
020302 «Фундаментальная информатика
и информационные технологии» и
090304 «Программная инженерия»

Нижний Новгород
2017

УДК 519.25
ББК В17

Численные методы математической статистики в пакете R. Составители: Зорин А.В., Кудрявцев Е.В., Рачинская М.А.: Учебно-методическое пособие. — Нижний Новгород: Нижегородский госуниверситет, 2017. — 37 с.

Рецензент: к.ф.-м.н., доцент кафедры МОСТ **К.А. Баркалов**

Настоящее пособие содержит базовые сведения о среде статистических вычислений R. Описаны основные типы данных и операции над ними. Приведены примеры, иллюстрирующие статистическую обработку данных как введенных пользователем, так и сгенерированных с помощью средств пакета R. Подробно описаны некоторые методы точечного оценивания неизвестных параметров распределения и их использование в пакете R.

Пособие предназначено для студентов бакалавриата и магистратуры, обучающихся по направлениям подготовки «Прикладная математика и информатика», «Фундаментальная информатика и информационные технологии» и «Программная инженерия», и может быть использовано при чтении специального курса «Прикладная математическая статистика».

УДК 519.25
ББК В17

# Содержание

## Введение

Специализированные программные комплексы для статистического анализа данных играют важную роль во многих областях науки и бизнеса. Такие программы должны обладать несколькими специфическими особенностями: уметь эффективно хранить данные разных типов и обеспечивать легкий доступ к ним, эффективно выполнять трудоемкие вычислительные задачи, осуществлять преобразование данных из разных форматов, представлять графически данные и результаты анализа, иметь встроенный язык для создания расширений и добавления новой функциональности в различных прикладных областях, обладать богатым набором методов решения основных статистических задач. В настоящее время существует несколько конкурирующих между собой коммерческих продуктов, таких как SPSS, STATISTICA, и т.д., поддерживающих весь спектр возможностей. Отметим также наличие статистических функций и дополнительных надстроек у программ электронных таблиц Microsoft Excel и OpenOffice Calc. Наконец, практически все современные программы для проведения математических расчетов (Mathematica, Maple, Matlab, Mathcad, Scilab, Octave, и т. д.) общего назначения имеют функции для выполнения основных процедур прикладного статистического анализа.

В научном сообществе пользуется популярностью пакет статистического анализа данных R [1]. Это свободно распространяемые на условиях лицензии GNU язык и среда вкупе с большим набором библиотек, доступные для всех основных платформ — UNIX, Linux, Windows и MacOS. Постоянно выходят новые версии и расширения. Важным показателем популярности продукта является выход книг с описанием применения R для тех или иных видов анализа [2, 3, 4]. Знакомство с этим пакетом позволит получить базовые навыки постановки статистической проблемы и решения ее «до числа», продемонстрирует основные методы проведения статистического анализа данных и интерпретации численных значений разнообразных статистик и оценок, возникающих в повседневной практике.

Минимальная сессия в R представлена ниже. На ее примере мы объясним некоторые принятые шрифтовые и типографский соглашения.

```
R version 2.11.1 (2010-05-31)
Copyright (C) 2010 The R Foundation for Statistical
Computing ISBN 3-900051-07-0

R -- это свободное ПО, и оно поставляется безо всяких
гарантий. Вы вольны распространять его при соблюдении
некоторых условий. Введите 'license()' для получения
более подробной информации.

```

```
10 R -- это проект, в котором сотрудничает множество
11 разработчиков. Введите 'contributors()' для получения
12 дополнительной информации и 'citation()' для ознакомления
13 с правилами упоминания R и его пакетов в публикациях.
14
15 Введите 'demo()' для запуска демонстрационных программ,
16 'help()' -- для получения справки, 'help.start()' --
17 для доступа к справке через браузер. Введите 'q()',
18 чтобы выйти из R.
19
20 > q()
21 Save workspace image? [y/n/c]: n
```

Шрифтом и рамкой выделяется то, что печатает пользователь и выдает программа. Числа слева от строк не являются частью диалога и присутствуют там для удобства ссылок. Строки 1–18 содержат стандартный начальный заголовок программы. Приглашение пользователя отмечается знаком `>` (строка 20). В данном сеансе была выполнена команда `q()`, завершающая работу с программой. В строке 21 предлагалось сохранить на диск образ рабочего пространства программы для загрузки его при следующем запуске. Наряду с интерфейсом командной строки существуют несколько графических интерфейсов пользователя: Rcommander, Rkward, и др.

## 1. Элементы входного языка пакета R

В R все объекты имеют два обязательных атрибута: тип данных и длина. В зависимости от значений этих и других атрибутов объекты в R делятся на вектора (`vector`), факторы (`factor`), матрицы (`matrix`), массивы (`array`), таблицы (`data.frame`), списки (`list`) и пр. Данные в R не хранятся скалярно. Основные типы данных в R — это `logical` — логический, `numeric` — числовой (`integer` и `double`), `character` — символьный, `complex` — комплексный.

Результат вычисления выражения, введенного в качестве команды, выводится на экран. Присваивание значений переменным осуществляется операторами `<-`, `->` или `=`. Результат последней команды автоматически сохраняется в переменной `.Last.value`.

```
1 > (5/4)^3+2*(pi+1)^2
2 [1] 36.2587
3 > x <- 65.4/4+exp(10); x
4 [1] 22042.82
```

```
5 > is.numeric(x); is.vector(x)
6 [1] TRUE
7 [1] TRUE
```

Команды, перечисленные через ';', как в строках 3 и 5, выполняются последовательно, и результаты выводится на экран. Символы `[1]` означают, что вывод осуществляется с первого элемента массива. Из строк 5–7 видим, что результат простейших арифметических операций есть числовой вектор.

Справку по имеющимся функциям в R можно получить с помощью команды `help(имя_функции)` или `?имя_функции`. При запросе справки по оператору его название должно быть заключено в кавычки. Бывает также полезной функция `apropos(what, ...)`, которая возвращает название объектов (функций, переменных), содержащих в своем имени строку `what`, заключенную в кавычки. Ради экономии места некоторые допустимые аргументы функций или их значения по умолчанию будем далее опускать, заменяя многоточием, если это не мешает пониманию.

### 1.1. Работа с векторами в пакете R

Следующие функции в пакете R используются для создания вектора длины `length` и инициализации его значениями по умолчанию:

- `logical(length)`, `numeric(length)` и т. п.;
- `vector(mode = "logical", length)`.

Например, для логического типа значением по умолчанию является `FALSE`, для символьного — `""`. Заполнить элементы вектора данными можно, например, с помощью функции `scan(...)`, организующей ввод данных из файла или консоли. Кроме того, существуют различные способы создания вектора с заранее заданными элементами:

- `c(...)` — объединяет аргументы в вектор;
- `seq(from, to, by, length.out, ...)` — генерирует последовательность из `length.out` чисел от `from` до `to` с шагом `by`;
- `rep(x, times)` — создает вектор из `times` копий элемента `x`.

Например, вектор `(-4, -3, -2, -1, 0, 1)` можно получить различными способами:

```
> c(-4:1)
[1] -4 -3 -2 -1  0  1
> c(-4, -3, -2, -1, 0, 1)
[1] -4 -3 -2 -1  0  1
> seq(-4, 1, 1)
[1] -4 -3 -2 -1  0  1
> seq(-4, 1, len = 6)
[1] -4 -3 -2 -1  0  1
```

При работе со статистическими данными полезны бывают следующие функции (описание функции опущено, если оно очевидно из названия): `length(...)`, `sort(...)`, `max(...)`, `min(...)`, `range(...)`, `sum(...)`, `mean(...)` — возвращает среднее арифметическое элементов вектора, `prod(...)` — возвращает произведение элементов вектора, `rev(...)` — переставляет элементы вектора в обратном порядке, `rank(...)` — присваивает элементу вектора его позицию в ряду всех элементов, упорядоченных по возрастанию, `cumsum(...)` — возвращает вектор накопленных сумм и т. п.

Отметим также функцию `which(...)`, возвращающую индексы элементов `TRUE` логического вектора. Например, ожидается, что в выборке из 1000 значений стандартной нормальной случайной величины примерно половина будет неотрицательной.

```
> length(which(rnorm(1000) >= 0))
[1] 526
```

Знакомство с функцией `rnorm(...)` произойдет позднее в разделе 2.1.

## 1.2. Работа с матрицами и многомерными массивами в R

Матрица в R создается функцией `matrix(data, nrow, ncol, ...)`. Данная функция заполняет по строкам матрицу размером `nrow` × `ncol` элементами вектора `data`. Если `data` содержит меньшее количество элементов, то данные повторяются, начиная с первого элемента.

Доступ к элементам матрицы проиллюстрируем на примере:

```
> matrix(runif(9), 3, 3) -> m1; m1
          [,1]      [,2]       [,3]
[1,] 0.6098199 0.6233537 0.92173507
[2,] 0.8389689 0.2792989 0.01528674
[3,] 0.9091849 0.7372735 0.91355633
```

```
> m1[1,2]; m1[4]
[1] 0.6233537
[1] 0.6233537
> m1[,2]; m1[c(1,3), c(2,3)]
[1] 0.6233537 0.2792989 0.7372735
          [,1]      [,2]
[1,] 0.6233537 0.9217351
[2,] 0.7372735 0.9135563
```

В строке 1 создается матрица $3 \times 3$ и заполняется случайными числами из интервала $(0,\ 1)$. Более подробно функция `runif(...)` разбирается в разделе 2.1. Строка 6 демонстрирует возможные обращения к одному и тому же элементу матрицы. В строке 9 иллюстрируется выделение столбца (строки) матрицы или подматрицы.

Отметим, что поскольку в языке R нет скалярных типов данных, то арифметические операции и элементарные математические функции применяются поэлементно. В примере ниже в строке 13 видим результат поэлементного перемножения матриц одной размерности. Ошибка в строке 11, в свою очередь, указывает на несовпадение размерностей матриц-множителей. В строке 16 выполняется сложение матрицы и вектора. При этом вектор приводится к матрице соответствующей размерности.

```
> m2 <- matrix(-1:1, 3, 2); m3 <- matrix(c(1,0), 2, 2)
> m2; m3
     [,1] [,2]
[1,]   -1   -1
[2,]    0    0
[3,]    1    1
     [,1] [,2]
[1,]    1    1
[2,]    0    0
> m2*m3
Error in m2 * m3 : non-conformable arrays
> m3*m1[c(1,3), c(2,2)]
[,1]      [,2]
[1,] 0.6233537 0.6233537
[2,] 0.0000000 0.0000000
> m3+c(1,3,3,5)
     [,1] [,2]
[1,]    2    4
[2,]    3    5
```

Для выполнения матричных операций в R существуют следующие функции и операторы: `%*%` — матричное умножение; `t(...)` — транспонирование матрицы; `diag(...)` — выделение главной диагонали матрицы или создание единичной матрицы; `colSums(...)`, `rowSums(...)`, `colMeans(...)`, `rowMeans(...)` — подсчет сумм или средних арифметических элементов по столбцам и строкам соотвественно. Функция `solve(a, b, ...)` решает матричное уравнение вида `a %*% x = b`, при чем `b` может быть как вектором, так и матрицей. Вызов функции `solve(a)` позволяет найти матрицу, обратную к `a`.

Создание многомерного массива из вектора данных `data` осуществляется вызовом функции `array(data, dim, ...)`, при этом в качестве аргумента `dim` передается вектор размеров массива в каждой размерности, например `dim = c(2,3,2)`. Набор операций для работы с массивами аналогичен работе с матрицами.

### 1.3. Создание функций в R

В рамках проекта R написано большое число пакетов со специализированными функциями для прикладного статистического анализа. Однако часто бывает полезно создать собственные пользовательские функции.

В языке R имеются все основные логические операторы (`>`, `<`, `>=`, `<=`, `==`, `!=`) и операторы управления:

- `if (cond) expr`, `if (cond) cons.expr else alt.expr` — выполняет команды `expr`, `cons.expr` или `alt.expr` в зависимости от логического значения условия `cond`; если `cond` имеет длину, большую единицы, то проверяется только первый его элемент;
- `ifelse(test, yes, no)` — выполняет `yes` или `no` в зависимости от логического значения условия `test`; в отличии от оператора `if` может работать с переменными любой длины;
- `for(var in seq) expr` или `while(cond) expr` — выполняют `expr`, пока `var` находится в рамках последовательности `seq` или выполнено условие `cond` соответственно;
- `repeat expr` — запускает бесконечный цикл выполнения `expr`, выйти из которого можно с помощью оператора `break`;
- `switch(expr, ...)` — выполняет команду из списка `...` с номером, являющимся результатом вычисления `expr`.

Каждое условие или выражение в этом списке может быть заменено блоком выражений вида `{expr1; expr2; ...}`. В этом случае они выполняются последовательно и возвращается результат выполнения последней команды.

Функция в R создается следующим образом: `function( arglist ) expr`. Формальным аргументам из списка `arglist` может присваиваться значение по умолчанию в виде `arg = value`. Телом функции `expr` является команда или блок команд. Тело может содержать в конце команду `return(value)` для явного указания возвращаемого значения. В случае, если эта команда отсутствует, функция возвращает результат выполнения последней команды из тела. Приведем пример функции, реализующей алгоритм Евклида для поиска наибольшего общего делителя двух чисел.

```
> Euclid <- function(x, y)
+ {while ((x != 0) && (y != 0)) if (x>y) x <- x%%y else y<-y%%x;
+ return (x+y)}
> Euclid(25367, 6375)
[1] 1
```

Отметим, что тип данных для аргументов не указывается, поэтому в качестве аргумента может также выступать любая другая функция.

# 2. Генерация случайных чисел и метод Монте–Карло в пакете R

## 2.1. Вероятностные распределения в R

В пакете R присутствует набор функций для работы с некоторыми типовыми дискретными и непрерывными одномерными распределениями вероятностей. С каждым распределением связаны четыре функции. Имена функций состоят из двух частей: однобуквенной приставки и имени распределения. Приставка `r` означает генерирование выборки значений, обязательный аргумент — объем выборки. Приставка `d` означает вычисление плотности распределения для непрерывных распределений или вероятности значения для дискретных распределений. Обязательный аргумент — точка или массив точек, в которых вычисляется плотность. Приставка `p` соответствует вычислению функции распределения. Обязательный аргумент — точка или массив точек, в которых вычисляется функция распределения. Наконец, приставка `q` означает вычисление квантиля распределения. Обязательный аргумент — вероятность или массив вероятностей, для которых вычисляются квантили. Таблица 1 содержит перечень распределений с указанием необязательных аргументов, соответствующих параметрам распределений. За более полной информацией о параметрах следует обратиться к справке по соответствующей функции. Например, выборку из 100 значений из нормального распределения со средним 1 и среднеквадратичными отклонением 0,3 можно получить

командой `rnorm(100, 1, .3)`. Следующий пример рисует графики плотностей $\chi^2$-распределения с разным числом степеней свободы (рис. 1).

```
1 > xvals <- seq(0,10,0.1)
2 > plot(xvals, dchisq(xvals, 1),"l",
3 + ylab="chi-square density", xlab="x")
4 > lines(xvals, dchisq(xvals, 4),lty=3)
5 > lines(xvals, dchisq(xvals, 2),lty=2)
```

Функция `seq` в строке 1 создает вектор значений независимой переменной от 0 до 10 с шагом 0,1. Функция `plot` создает новый график, `lines` добавляет на существующий график кривую, заданную вектором абсцисс и вектором ординат точек.

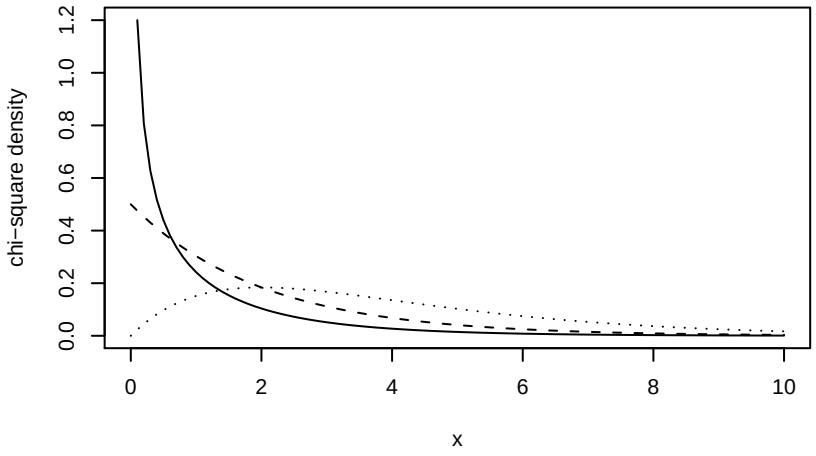


Рис. 1: Плотность $\chi^2$-распределения с одной степенью свободы (сплошная линия), с двумя степенями свободы (пунктирная линия), с четырьмя степенями свободы (точечная линия)

### 2.2. Генерирование величин с распределением пользователя

Для моделирования выборки из распределений, не представленных в пакете R, необходимо использовать общие методы моделирования случайных величин с заданным законом распределения.

Рассмотрим сначала моделирование дискретной случайной величины с конечным числом значений $a_1$, $a_2$, ..., $a_K$. Пусть $p_j$ — вероятность значения $a_j$, $j = 1, 2, \ldots, K$. Разобьём отрезок $[0,\ 1]$ на полуинтервалы $\Delta_1 = [0,\ p_1)$, $\Delta_2 = [p_1,\ p_1 + p_2)$,

Таблица 1: Типовые распределения в R

| Имя | Плотность (вероятность) | Имя в R | Параметры |
|---|---|---|---|
| Биномиальное | $C_n^k p^k (1-p)^{n-k}$ $0 \leqslant p \leqslant 1,\ k = 0,\ 1,\ \ldots,\ n$ | `binom` | `size` $= n$, `prob` $= p$ |
| Пуассоновское | $\frac{\lambda^k}{k!} e^{-\lambda}$ $\lambda > 0,\ k = 0,\ 1,\ \ldots$ | `pois` | `lambda` $= \lambda$ |
| Геометрическое | $p(1-p)^k$ $0 < p \leqslant 1,\ k = 0,\ 1,\ \ldots$ | `geom` | `prob` $= p$ |
| Отрицательно-биномиальное | $\frac{\Gamma(k+n)}{\Gamma(n)k!} p^n (1-p)^k$ $0 < p \leqslant 1,\ n = 1,\ 2,\ \ldots,$ $k = 0,\ 1,\ \ldots$ | `nbinom` | `size` $= n$, `prob` $= p$ |
| Гипергеометрическое | $\frac{C_m^k C_n^{r-k}}{C_{m+n}^r}$ $m,\ n = 1,\ 2,\ \ldots,$ $r = 1,\ 2,\ \ldots,\ m+n,$ $k = 0,\ 1,\ \ldots,\ m$ | `hyper` | `m` $= m$, `n` $= n$, `r` $= r$ |
| Равномерное | $(b-a)^{-1},\ a < x < b$ $a < b$ | `unif` | `min` $= a$, `max` $= b$ |
| Экспоненциальное | $\lambda e^{-\lambda x},\ x \geqslant 0$ $\lambda > 0$ | `exp` | `lambda` $= \lambda$ |
| Нормальное | $\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-a)^2}{2\sigma^2}}$ $\sigma > 0$ | `norm` | `mean` $= a$, `sd` $= \sigma$ |
| Гамма | $\frac{x^{a-1}}{s^a \Gamma(a)},\ x \geqslant 0,$ $s > 0,\ a \geqslant 0$ | `gamma` | `shape` $= a$, `scale` $= s$ |
| Коши | $s\left(\pi(s^2 + (x-l))\right)^{-1}$ | `cauchy` | `location` $= l$, `scale` $= s$ |
| Стьюдента $t$ | $\frac{\Gamma\left(\frac{n+1}{2}\right)}{\sqrt{n\pi}\Gamma\left(\frac{n}{2}\right)} \left(1 + \frac{x^2}{n}\right)^{-\frac{n+1}{2}}$ $n > 0$ | `t` | `df` $= n$ |
| F | $\frac{\Gamma\left(\frac{1}{2}(n_1+n_2)\right)}{\Gamma\left(\frac{1}{2}n_1\right)\Gamma\left(\frac{1}{2}n_2\right)} \left(\frac{n_1}{n_2}\right)^{\frac{1}{2}n_1} x^{\frac{1}{2}n_1 - 1} \times$ $\times \left(1 + \frac{n_1}{n_2} x\right)^{-\frac{1}{2}(n_1+n_2)}$ $n_1 > 0,\ n_2 > 0$ | `f` | `df1` $= n_1$, `df2` $= n_2$ |

$\ldots$, $\Delta_K = [p_1+p_2+\ldots+p_{K-1},\ 1]$. Пусть $u$ — значение случайной величины с равномерным распределением на $(0,\ 1)$. Найдём интервал $\Delta_j$, которому принадлежит число $u$. Тогда в качестве значения моделируемой случайной величины выберем $a_j$. Для реализации этой схемы в R предназначена функция `findInterval()` в сочетании с функцией `cumsum`. Функция `cumsum` вычисляет накопленные суммы для своего аргумента $\mathtt{u} = (u_1,\ u_2,\ \ldots,\ u_n)$:

$$\mathtt{cumsum}(\mathtt{u}) = (u_1,\ u_1 + u_2,\ \ldots,\ u_1 + u_2 + \ldots + u_n).$$

Для числа $a$ по заданному $\mathtt{v} = (v_1,\ v_2,\ \ldots,\ v_n)$, $v_1 < v_2 < \ldots < v_k$ функция `findInterval(a,v)` возвращает индекс $j$ такой, что $v_j \leqslant a < v_{j+1}$. Если первый аргумент является вектором, то эта процедура применяется к каждому его элементу и возвращается вектор из индексов соответственно. В следующем примере генерируется выборка значений дискретной случайной величины $X$ с распределением: $\mathbf{P}(X = -1) = 0{,}15$, $\mathbf{P}(X = 0) = 0{,}20$, $\mathbf{P}(X = 5) = 0{,}65$.

```
> a <- c(-1, 0, 5)
> pr <- c(.15, .2, .65)
> x <- a[ findInterval( runif(20), cumsum(c(0,pr)) ) ]
> x
 [1]  0  5  5  5 -1  0  0  0  0  5  5  0  0  5  0
```

На этом примере также хорошо видно особенное использования квадратных скобок “`[`”, “`]`” . Именно, для вектора индексов (возможно, повторяющихся), возвращается новый вектор соответствующих элементов (возможно, повторяющихся): $\mathtt{a[c(i,\ j,\ k)]} = (a_i,\ a_j,\ a_k)$.

Второй способ решения задачи — использование функции `sample`. Ее аргументами являются: `x` — вектор различных значений случайной величины $X$, `size` — объем выборки, `prob` — вектор вероятностей, соответствующих значениям. В следующем примере генерируется выборка значений дискретной случайной величины $X$ с распределением: $\mathbf{P}(X = -1) = 0{,}15$, $\mathbf{P}(X = 0) = 0{,}20$, $\mathbf{P}(X = 5) = 0{,}65$.

```
> a <- c(-1, 0, 5)
> pr <- c(.15, .2, .65)
> sample(a, 20, replace=TRUE, prob = pr)
 [1]  5  0 -1  5  0  5  0 -1 -1  5  5  5  5  5  5  0  0  5 -1  5
> sample(a, 20, replace=TRUE, prob = pr)
 [1]  5  5  5  5  5  5  5  0 -1  0  5  5  5  5  0  0  5  5  5  5
```

Параметр `replace=TRUE` указывает на повторную выборку. В данном примере два вызова функции `sample` генерировали две выборки объема 20. Для более

полного знакомства с другими возможностями функции читателю рекомендуется обратиться к справке (команда `?sample`).

Пусть непрерывная случайная величина задана своей плотностью распределения $f(x)$. Функция распределения $F(x)$ будет непрерывной. Предположим дополнительно, что $F(x)$ монотонно возрастает на некотором интервале $(x_{\min},\ x_{\max})$, $-\infty \le x_{\min} < x_{\max} \le \infty$ и постоянна вне его. Рассмотрим функцию $G(x)\colon (0,\ 1) \to$ $\to (x_{\min},\ x_{\max})$, обратную к $F(x)$ для $x \in (x_{\min},\ x_{\max})$. Пусть $U$ имеет равномерное распределение на $(0,\ 1)$. Покажем, что случайная величина $G(U)$ имеет функцию распределения $F(x)$. Действительно, пусть $x \in (x_{\min},\ x_{\max})$, тогда

$$\mathbf{P}(G(U) < x) = \mathbf{P}(U < F(x)) = F(x).$$

Рассмотрим пример. Пусть требуется получить значение случайной величины со смещённым показательным распределением, заданным функцией распределения

$$F(x) = \begin{cases} 0, & \text{если } x < \theta, \\ 1 - e^{-(x-\theta)}, & \text{если } x \ge \theta. \end{cases}$$

Найдём обратную функцию. Решая уравнение $y = 1 - e^{-(x-\theta)}$, получаем: $G(x) =$ $= \theta - \ln(1-x)$. Заметим, что если $U$ имеет равномерное распределение на $(0,\ 1)$, то такое же распределение имеет величина $(1-U)$. Окончательно имеем: $\eta = \theta - \ln U$. Напишем функцию, которая будет генерировать выборку значений случайной величины со смещенным показательным распределением с параметром смещения `shift` $= \theta$ объема `size` $= n$:

```
> rshiftedexp <- function(size, shift) {
+ shift-log(runif(size))
+ }
> rshiftedexp(20, 1.2)
 [1] 1.478200 1.351191 1.243315 2.019118 3.148568 2.113016
 [7] 3.829635 2.029789 1.755985 2.969904 2.176174 1.686405
[13] 3.809035 1.838412 1.927286 1.241551 1.904208 1.377962
[19] 3.936998 1.522257
```

### 2.3. Приближенное вычисление интегралов методом Монте–Карло

К методам Монте–Карло относится группа численных методов, основанных на моделировании вспомогательных случайных величин. При этом приближенное решение исходной задачи представляется как функция от них. К примеру,

рассмотрим модельную задачу о приближенном вычислении интеграла

$$I = \int\limits_G h(\mathbf{x})d\mathbf{x}$$

по некоторой ограниченной $m$–мерной области $G \subset R^m$. Здесь и далее жирным шрифтом будем обозначать векторные величины. Заметим, что любой подобный интеграл можно считать интегралом вида

$$I = \int\limits_G p(\mathbf{x})f(\mathbf{x})d\mathbf{x}, \tag{1}$$

где $f(\mathbf{x})$ — некоторая плотность вероятностей $m$–мерного вектора $\boldsymbol{X}$ и справедливо условие нормировки $\int_G f(\mathbf{x})d\mathbf{x} = 1$. Действительно, рассмотрим случайный вектор, равномерно распределенный в области $G$ с площадью $S_G$. Тогда плотность отлична от нуля только в точках $\mathbf{x} \in G$ и равна $f(\mathbf{x}) = \frac{1}{S_G}$. Положим теперь $p(\mathbf{x}) = S_G \cdot h(\mathbf{x})$ и получим интеграл вида (1).

При довольно общих условиях интеграл $I$ можно считать математическим ожиданием функции $p(\mathbf{x})$ от случайного вектора с плотностью $f(\mathbf{x})$, т. е. $I = \mathbf{M}p(\boldsymbol{X})$. Рассмотрим $N$ независимых случайных векторов $\boldsymbol{X}_1$, $\boldsymbol{X}_2$, …, $\boldsymbol{X}_N$, имеющих плотность $f(\mathbf{x})$. Тогда согласно закону больших чисел имеется сходимость по вероятности

$$\widetilde{I} = \frac{1}{N}\sum_{i=1}^{N} p(\boldsymbol{X}_i) \xrightarrow[N\to\infty]{p} I.$$

Таким образом, для приближенного вычисления интеграла $I$ можно использовать формулу $I \approx \sum\limits_{i=1}^{N} p(\mathbf{x}_i)$, где $\mathbf{x}_i$ есть реализация вектора $\boldsymbol{X}_i$.

Продемонстрируем представленный метод на примере приближенного вычисления интеграла, точное значение которого известно:

$$I = \int\limits_0^1 x^3 dx = 0{,}25.$$

1. Пусть $f(x) = 1$ при $x \in [0,\ 1]$ и $f(x) = 0$ в остальных точках, т. е. рассматривается равномерно распределенная на $[0,\ 1]$ случайная величина $U$.

```
> p1 <- function(x) {x^3}
> mean(p1(runif(10000)))
[1] 0.2532294
```

2. Пусть теперь $f(x)$ пропорциональна $x^2$ на отрезке $[0,\ 1]$. Из условия нормировки для плотности находим, что $f(x) = 3x^2$, а интегральная функция распределения в этом случае есть $F(x) = x^3$ при $x \in [0,\ 1]$. Методом обратных функций находим, что величина $X$ с плотностью $f(x)$ моделируется равенством $X = \sqrt[3]{U}$. В этом случае $p(x) = \frac{x}{3}$ и численное вычисление интеграла даст следующий результат:

```
> p2 <- function(x) {x/3}
> mean(p2(runif(10000)^{1/3}))
[1] 0.2492707
```

3. Рассмотрим независимые равномерно распределенные на $[0,\ 1]$ случайные величины $U_1$, $U_2$ и $U_3$. Найдем распределение функции $\max\{U_1,\ U_2,\ U_3\}$:

$$F(x) = \mathbf{P}(\max\{U_1,\ U_2,\ U_3\} < x) = \mathbf{P}(U_1 < x,\ U_2 < x,\ U_3 < x) =$$
$$= \prod_{i=1}^{3} \mathbf{P}(U_i < x) = x^3.$$

Получаем еще один вариант моделирования случайной величины с плотностью $f(x) = 3x^2$.

```
> g <- function(n) {pmax(runif(n), runif(n), runif(n))}
> mean(p2(g(10000)))
[1] 0.249169
```

Здесь функция `pmax()` находит максимальное значение среди элементов с одинаковыми индексами.

Если теперь для оценки значения интеграла (1) применять представленный выше метод Монте–Карло, то возникает вопрос: какого объема наблюдений $n$ достаточно, чтобы при заданной точности $\varepsilon$ и надежности $\gamma$ гарантировать выполнение неравенства

$$\mathbf{P}(|I - \widetilde{I}| < \varepsilon) \geqslant \gamma?$$

Основываясь на центральной предельной теореме, можно вывести неравенство $n \geqslant (\Phi^{-1}(\frac{1+\gamma}{2})\frac{\sigma}{\varepsilon})^2$, где $\Phi(x)$ есть функция распределения стандартной нормальной случайной величины, а среднее квадратическое отклонение $\sigma = \sqrt{\mathbf{D}(p(\boldsymbol{X}))}$ можно оценить по выборке, если оно неизвестно. Например, для достижения с вероятностью 0,95 точности 0,001 при оценке интеграла $\int_0^1 x^3 dx$ первым способом получим необходимое число наблюдений

```
> gamma <- 0.95; epsilon <- 0.001; sigma <- sd(p1(runif(10000)))
> ceiling((qnorm((1+gamma)/2)/epsilon*sigma)^2)
[1] 303286
```

в то время как для второго и третьего способа получим

```
> sigma <- sd(p2(runif(10000)^{1/3}))
> ceiling((qnorm((1+gamma)/2)/epsilon*sigma)^2)
[1] 16024
```

Здесь функция `ceiling()` осуществляет округление числа вверх.

### 2.4. Зашумленные распределения

Во многих статистических исследованиях используется т. н. $t$–статистика, построенная по выборке объема $n$ из распределения с математическим ожиданием $M$ следующим образом:

$$t = \frac{\overline{x} - M}{s}\sqrt{n-1} = \frac{\overline{x} - M}{s_0}\sqrt{n}, \tag{2}$$

где $\overline{x}$ — выборочное среднее, $s^2$ — выборочная дисперсия, $s_0^2$ — несмещенная оценка дисперсии. Известно, что если выборка получена из нормального распределения, то $t$–статистика имеет распределение Стьюдента с $(n-1)$ степенью свободы. На практике часто встречается случай, когда в собранных выборочных данных присутствует некий «шум», примесь. Рассмотрим следующий пример.

```
1 > n = 30; N = 10000; alpha = 0.05
2 > x <- matrix(data = ifelse(runif(n*N) < alpha,
3 + rnorm(n*N, 3, 0.05), rnorm(n*N)), nrow = N)
4 > t.test.res <- numeric(N)
5 > for (i in 1:N) t.test.res[i] <- t.test(x[i,])$statistic
6 > hist(t.test.res, pr = T, col = "lightgrey")
7 > curve(dt(x, n-1), add = T)
```

Матрица `x` содержит `N` строк, в каждой из которых располагается выборка из `n` наблюдений. Данные представляют собой смесь двух нормальных распределений: с малой вероятностью `alpha` выбирается «шумовое» распределение с положительным средним и небольшой дисперсией, с вероятностью (1 - `alpha`) выбирается стандартное нормальное распределение, отвечающее за исходные данные задачи. Небольшая доля зашумленных наблюдений может быть не замечена исследователем. Для каждой выборки априорно считается, что все наблюдения порождены одним и тем же нормальным распределением с неизвестной дисперсией. Выдвигается гипотеза о том, что математическое ожидание равно нулю. Для проверки гипотезы используется критерий Стьюдента, основанный на статистике (2). В строке 5 происходит вычисление $t$–статистики для каждой из `N` выборок.

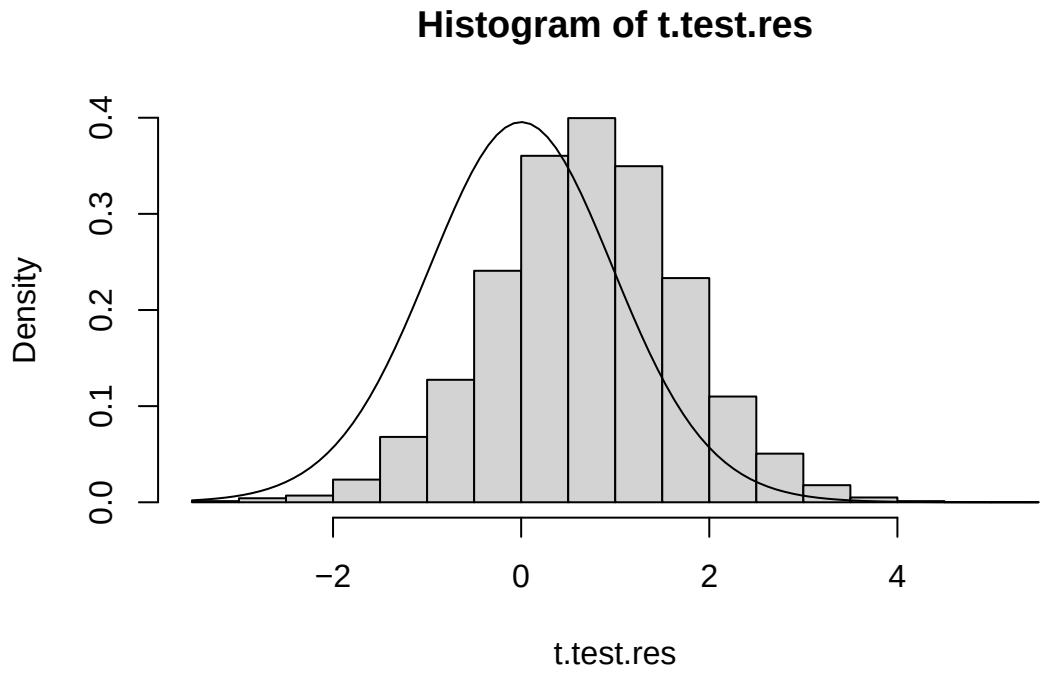


Рис. 2: Гистограмма значений $t$–статистики для зашумленных данных

В строке 6 строится гистограмма для полученных значений $t$–статистики. Результат представлен на рис. 2. График плотности распределения Стьюдента с $(n-1)$ степенями свободы выводится в строке 7. По рис. 2 видно, что гистограма имеет смещение от плотности. Следовательно, наблюдения будут чаще оказываться больше правой границы доверительного интервала и верная выдвинутая гипотеза будет чаще ошибочно отвергаться. Действительно, из-за 5% шума в данном случае при уровне значимости 0,1 в 17,5% выборках верная гипотеза отвергается.

```
1 > sum(ifelse(apply(x, 1, function(x) t.test(x)$p.value)
2 + <= rep(0.1, N), 1, 0))/N
3 [1] 0.175
```

Здесь `apply(x, margin, fun, ...)` применяет функцию `fun` к данным `x` по строкам или столбцам в зависимости от фактического значения аргумента `margin`. В строке 1 функция `t.test(x)$p.value` вычисляет $p$–значение критерия по строкам (`margin` $= 1$) данных `x`, т. е. для каждой из `N` выборок. Далее при помощи функции `ifelse(...)` происходит сравнение каждого из полученных $p$–значений с уровнем значимости 0,1. Количество выборок, для которых гипотеза отвергается, т. е. когда $p$–значение не превышает уровня значимости, подсчитывается функцией `sum(...)`. Доля таких выборок среди общего количества равна 0,175.

# 3. Статистические данные

## 3.1. Представление данных в R

Исходные данные для статистического анализа являются как правило результатом измерения нескольких переменных или характеристик у нескольких объектов. Переменные делятся на числовые и нечисловые (качественные). В свою очередь, числовые переменные бывают целочисленные или вещественные (дробные). Пусть обследуются сотрудники некоторого предприятия. Тогда примером вещественной переменной является рост в метрах, примером целочисленной переменной — число детей, а примером качественной — пол или цвет волос. Принципиальным отличием качественных переменных от количественных является невозможность упорядочить естественным образом значения качественной переменной. Еще одним важным типом переменной является календарная дата. Их можно считать особым типом числовых переменных. Пусть изучаются переменные $X$, $Y$, ..., $Z$, а результаты обследования $i$-го объекта представлены вектором $(x_i,\ y_i,\ \ldots,\ z_i)$, $i = 1, 2, \ldots, n$. Тогда выборка может быть сведена в таблицу

| $i$ | $X$ | $Y$ | ... | $Z$ |
|---|---|---|---|---|
| 1 | $x_1$ | $y_2$ | ... | $z_1$ |
| 2 | $x_2$ | $y_2$ | ... | $z_2$ |
| ⋮ | ⋮ | ⋮ | ⋱ | ⋮ |
| $n$ | $x_n$ | $y_n$ | ... | $z_n$ |

В терминологии R такая таблица называется *кадром данных* (data frame). Строки и столбы в этой таблице неравноправны. Каждый столбец представляет одну переменную. Каждая строка — одно наблюдение или один *случай*. Часто вместо номеров случаи могут иметь символические имена.

Для создания кадра данных используется функция `data.frame`. На примере этой команды рассмотрим некоторые особенности синтаксиса R. Полное описание возможностей этой функции (как и других) содержится во встроенной справке. Для доступа к странице справки необходимо напечатать `help(data.frame)` или `?data.frame`. Кадр данных с переменными разных типов создается и отображается в следующем примере:

```
> d.f <- data.frame(x=c(1,2), y=c(.2,.3),  z=c('Y','N'))
> d.f
  x   y z
1 1 0.2 Y
2 2 0.3 N
```

Каждое равенство здесь задает отдельную переменную. Функция `c()` осуществляет конкатенацию своих аргументов. Знак равенства (=) связывает имя переменной с ее значением внутри вызванной функции. Значением переменной `x` является целочисленный вектор, значением переменной `y` — вектор вещественных чисел, а значением переменной `z` — вектор из строк. Заметим, что нечисловые строковые значения категориальной переменной `z` были превращены в значения типа *фактор* (factor), то есть каждому уникальному значению был присвоен числовой код, который используется для внутреннего представления этой переменной. Имена переменных и функций в строке 1 типичны для R: они состоят из латинских и русских букв (и цифр), разделенных для удобства чтения точкой. Таким образом, точка в имени переменной не несет никакой синтаксической нагрузки для R. Оператор `<-` устанавливает вновь созданный объект типа `data.frame` в качестве значения переменной `d.f`. Следующая команду устанавливает новые имена для случаев (строк).

```
1 > row.names(d.f) <- c('A', 'B')
2 > d.f
3   x   y z
4 A 1 0.2 Y
5 B 2 0.3 N
```

Визуальное редактирование кадра данных `d.f` осуществляется с помощью функции `edit`. Перед пользователем открывается окно с электронной таблицей, позволяющей добавлять и удалять данные и переменные. По завершении редактирования таблицы функция `edit` возвращает *новый* кадр данных. Поэтому результат нужно сохранить в переменной, чтобы не потерять следы редактирования. Например, добавим третье значение, 3, переменной `x` и выведем результат:

```
1 > new.d.f <- edit(d.f); new.d.f
2      x   y    z
3 A    1 0.2    Y
4 B    2 0.3    N
5 row3 3  NA <NA>
```

В результате в кадре данных появились *пропущенные значения* `NA` (строка 5). Наличие пропущенных значений является отличительной чертой реальных статистических данных. Реальные программы для статистического анализа должны уметь представлять внутри себя пропущенные данные и должны предусматривать особое поведение в случаях, когда некоторые значения наблюдениях пропущены.

В стандартную установку R входит большое количество наборов данных. Например, набор данных `trees` содержит данные об охвате ствола (girth), высоте

(height) и объеме (volume) ствола для 31 сваленных черных вишневых деревьев (см. `?trees`).

```
> trees
   Girth Height Volume
1    8.3     70   10.3
2    8.6     65   10.3
3    8.8     63   10.2
...
```

Приведены только несколько первых строк.

Удобный способ ввода данных — импорт данных из файла. Рассмотрим импорт из текстовых файлов. В типичном текстовом файле данные могут располагаться следующим образом: первая строка содержит имена переменных, отделенные пробелами или табуляцией; в следующих строках располагаются разделенные пробелами или табуляцией значения переменных. В целом такой файл выглядит содержащим несколько колонок данных. Первая колонка может содержать имена случаев, а не данные. Функция `read.table` имеет следующие важные в первую очередь аргументы: имя файла — строка с именем файла, `header` — логическое значение `TRUE` или `FALSE` показывает наличие имен переменных в первой строке, `dec` — строка, указывающая разделитель целой и дробной части в десятичной дроби (например, "," или "."), `row.names` — либо число – номер колонки с именами случаев, либо вектор с именами случаев. Отметим возможность чтения данных из буфера обмена. Для этого надо указать имя файла `"clipboard"`.

Для доступа к переменным в кадре данных используется оператор `$`. Например, значения переменной `Girth` в кадре данных `trees` можно получить так:

```
> trees$Girth
 [1]  8.3  8.6  8.8 10.5 10.7 10.8 11.0 11.0 11.1 11.2
[11] 11.3 11.4 11.4 11.7 12.0 12.9 12.9 13.3 13.7 13.8
[21] 14.0 14.2 14.5 16.0 16.3 17.3 17.5 17.9 18.0 18.0
[31] 20.6
```

Числа в квадратных скобках в начале строк 2–5 указывают индексы элементов вектора `trees$Girth`, а которых начинается строка. Средой R поддерживается список пакетов и кадров данных, в которых производиться поиск переменной по ее имени. Для упрощения доступа к переменным в кадре данных можно добавить к этому списку функцией `attach`. Удаление из этого списка выполняется функцией `detach`. Например,

```
> attach(trees)
> Girth
```

```
 [1]  8.3  8.6  8.8 10.5 10.7 10.8 11.0 11.0 11.1 11.2
[11] 11.3 11.4 11.4 11.7 12.0 12.9 12.9 13.3 13.7 13.8
[21] 14.0 14.2 14.5 16.0 16.3 17.3 17.5 17.9 18.0 18.0
[31] 20.6
> detach(trees)
> Girth
Ошибка: объект 'Girth' не найден
```

Имя переменной в строке 8 приводит к сообщению об ошибке в строке 9, так как кадр данных `trees` уже выключен их списка поиска.

### 3.2. Выборочные числовые характеристики

Выборочной числовой характеристикой случайной величины $X$ называется числовая характеристика, соответствующая выборочной функции распределения $\hat{F}_n(x)$. Рассмотрим основные выборочные числовые характеристики, встречающиеся на практике. К характеристикам положения случайной величины относятся математическое ожидание, медиана, квантили. К характеристикам разброса — дисперсия, размах, среднеквадратичное отклонение.

**Выборочное среднее**

$$\bar{x} = \frac{x_1 + x_2 + \ldots + x_n}{n}$$

является при фиксированных $x_1$, $x_2$, …, $x_n$ математическим ожиданием, соответствующем функции распределения $\hat{F}_n(x)$. Действительно, предположим, что все выборочные значения различны. Тогда $\hat{F}_n(x)$ задает равномерное распределение на множестве $\{x_1, x_2, \ldots, x_n\}$. Легко видеть, что если среди выборочных значений есть совпадающие, что формула для $\bar{x}$ переходит в формулу для математического ожидания соответствующего (неравномерного) дискретного распределения. Напомним, что выборочное среднее $\bar{x}$ является несмещенной оценкой математического ожидания $\mathbf{M}X$ в предположении, что $\mathbf{M}X$ существует. Для оценки выборочного среднего в R используется функция `mean`:

```
1 > mean(trees$Girth)
2 [1] 13.24839
3 > mean(trees)
4    Girth   Height   Volume
5 13.24839 76.00000 30.17097
```

В строке 1 аргументом функции `mean` является одна переменная с вектором выборочных значений, а в строке 3 — кадр данных. В последнем случае вычисляется

выборочное среднее для каждой числовой переменной из указанного кадра данных. Поведение функции `mean` при наличии пропущенных значений управляется аргументом `na.rm`. Значение `TRUE` предписывает сначала удалить пропущенные случаи, а по оставшимся провести вычисления; значение `FALSE` приведет к результату `NA`, «значение недоступно»:

```
> mean(c(1, NA, 2))
[1] NA
> mean(c(1, NA, 2),na.rm=TRUE)
[1] 1.5
```

**Выборочная медиана** $\hat{\mu}$ для выборки нечетного объема $n = 2m + 1$ определяется как $(m + 1)$-е по величине выборочное значение, а для выборки четного объема $n = 2m$ — как среднее арифметическое из $m$-го и $(m + 1)$-го по величине выборочных значений. Из определения медианы следует, что одинаковое число выборочных значений оказываются больше и меньше медианы. В R вычисляется с помощью функции `median`.

**Выборочные квантили.** Напомним, что квантиль уровня $p$ определяется как точная верхняя грань $z(p)$ чисел $z$, для которых $F(z) \leqslant p$. Для непрерывной функции распределения $F$ имеем $F(z(p)) = p$. Если медиана $\mu$ существует, то она есть квантиль уровня $\frac{1}{2}$. Аналогично можно определить выборочный квантиль $\hat{z}(p)$ на основе выборочной функции распределения $\hat{F}_n$. Вычисление выборочных квантилей в R реализовано в функции `quantile`. Аргументами являются выборка значений случайной величины, вектор вероятностей $p$, для которых требуется найти соответствующие квантили, наконец, тип процедуры оценки квантиля — целое число от 1 до 9. Пусть $j = [np]$ — наибольшее целое, не превосходящее $np$. Выборочный квантиль $\hat{z}_1(p)$ типа 1 соответствует определению, данному для квантиля выше:

$$\hat{z}_1(p) = \begin{cases} x_{(j)}, & \text{если } np \text{ — целое} \\ x_{(j+1)}, & \text{если } np \text{ — не целое} \end{cases}$$

Так определенная величина есть разрывная функция от $p$, $0 < p < 1$. По умолчанию в R используется тип 7:

$$\hat{z}_7(p) = x_{([h])} + (h - [h])(x_{([h]+1)} - x_{[h]}), \qquad h = (n - 1)p + 1.$$

Это определение есть линейная интерполяция между двумя соседними порядковыми статистиками. В результате оценка $z_7(p)$ есть непрерывная функция от $p$, что весьма желательно при оценивании квантилей непрерывных функций распределения.

**Выборочная дисперсия**

$$S^2 = \frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \ldots + (x_n - \bar{x})^2}{n}$$

является смещенной оценкой для $\mathbf{D}X$. Несмещенной оценкой дисперсии является величина

$$S_0^2 = \frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \ldots + (x_n - \bar{x})^2}{n-1}.$$

В пакете R реализовано вычисление величины $S_0^2$ с помощью функции `var`.

```
> var(trees$Girth)
[1] 9.847914
```

Если аргументом функции `var` является кадр данных с несколькими переменными, то будет вычислена соответствующая ковариационная матрица. Дисперсии переменных будут размещены на главной диагонали:

```
> var(trees)
            Girth   Height    Volume
Girth   9.847914 10.38333  49.88812
Height 10.383333 40.60000  62.66000
Volume 49.888118 62.66000 270.20280
```

Например, несмещенная оценка дисперсии переменной `Height` равна 40,6.

**Выборочное среднеквадратичное отклонение** $S_0$ вычисляется как квадратный корень из несмещенной выборочной дисперсии. Для вычисления используется функция `sd`.

```
> sd(trees)
    Girth    Height    Volume
 3.138139  6.371813 16.437846
```

**Выборочный размах** определяется как разность между наибольшим и наименьшим из выборочных значений. Функция `range` возвращает пару из наименьшего и наибольшего значений в выборке. Для вычисления размаха теперь достаточно вызвать функцию `diff`, которая для вектора $(x_1,\ x_2,\ \ldots,\ x_n)$ строит вектор $(x_2 - x_1,\ x_3 - x_2,\ \ldots,\ x_n - x_{n-1})$.

```
> diff(range(trees$Volume))
[1] 66.8
```

**Выборочный междуквартильный размах.** Первым и третьим квартилями называются квантили уровня $1/4$ и $3/4$ соответственно. Междуквартильный размах определяется как величина $IQR = z(3/4) - z(1/4)$.

```
> IQR(trees$Girth)
[1] 4.2
```

Существует возможность простого наглядного представления данных о положении и разбросе выборочных значений случайной величины. Функция `boxplot` создает график типа «ящик с усами» (англ. box-and-wiskers). Такой график состоит из вертикального прямоугольника, нижняя и верхняя стороны которого находятся соответственно на высоте первого и третьего квартиля, горизонтальной черты внутри треугольника на высоте медианы, вертикальных отрезков–«усов», отходящих вверх и вниз от прямоугольника. Усы могут либо доставать до наибольшего и наименьшего из наблюдаемых значений, либо отходить на расстояние, пропорциональное междуквартильному размаху и тогда все не попадающие между «усами» наблюдения отмечаются отдельно (рис. 4).

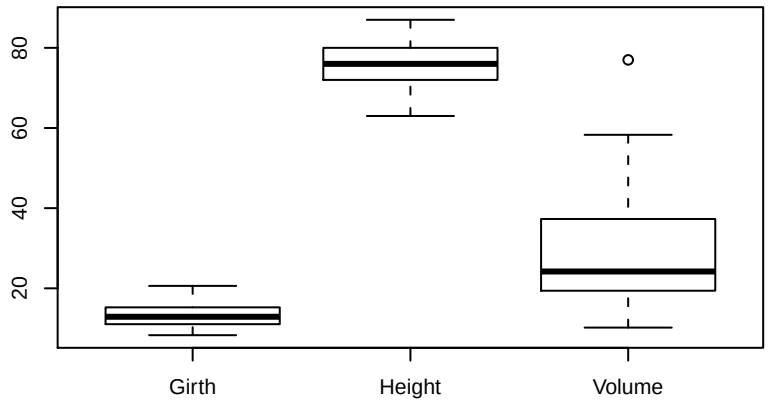


Рис. 3: График «ящик с усами» для переменных `Girth`, `Height`, `Volume`

# 4. Методы точечного оценивания параметров

## 4.1. Метод максимального правдоподобия

Пусть случайная величина $X$ дискретна, $p(a;\ \theta) = \mathbf{P}\{X = a\}$, $\theta$ — скалярный или векторный параметр, значение которого требуется определить по выборке $x_1$,

$x_2, \ldots, x_n$ конечного объема $n$. Совместное распределение выборочных значений имеет вид

$$L(x_1,\ x_2,\ \ldots,\ x_n;\ \theta) = p(x_1;\ \theta)p(x_1;\ \theta)\cdots p(x_n;\ \theta).$$

При фиксированных значениях $x_1, x_2, \ldots, x_n$ величина

$$L = L(x_1,\ x_2,\ \ldots,\ x_n;\ \theta)$$

как функция аргумента $\theta$ определяет правдоподобие данной выборки при значении параметра $\theta$ и называется *функцией правдоподобия*. Для непрерывной случайной величины с плотностью распределения $p(u;\ \theta)$ функция правдоподобия определяется как

$$L(x_1,\ x_2,\ \ldots,\ x_n;\ \theta) = p(x_1;\ \theta)p(x_1;\ \theta)\cdots p(x_n;\ \theta)$$

и задает совместную плотность распределения выборочных значений. Оценкой максимального правдоподобия называется статистика

$$\hat{\theta} = \hat{\theta}(x_1,\ x_2,\ \ldots,\ x_n),$$

доставляющая максимум функции правдоподобия. Для типовых распределений найдены явные формулы для оценок максимального правдоподобия их параметров.

Для нахождения оценок максимального правдоподобия в R можно воспользоваться функцией `fitdistr` из библиотеки `MASS`. Эта функция предназначена для нахождения оценки максимального правдоподобия как для типовых распределений, так и для не типовых распределений пользователя. К типовым распределениям относятся нормальное распределение, распределение хи-квадрат, гамма-распределение, экспоненциальное распределение, геометрическое распределение, отрицательно-биномиальное распределение, распределение Пуассона, биномиальное распределение и некоторые другие. Следующий пример показывает, как оценить параметры нормального распределения для переменной `Height` из кадра данных `trees`.

```
> fitdistr(trees$Height, "normal")
      mean          sd
  76.0000000    6.2681993
 ( 1.1258018) ( 0.7960621)
> norm.params <- fitdistr(trees$Height, "normal")$estimate;
> xvals <- seq(60,90,.5);
> hist(trees$Height,probability=TRUE,col="grey",main="");
> lines(xvals, dnorm(xvals, mean=norm.params[1],
+ sd=norm.params[2]));
```

Строка 3 содержит вычисленные оценки параметров нормального распределения: $a = 76$, $\sigma = 6{,}26$. В строке 4 в скобках указаны среднеквадратичные отклонения найденных оценок, так называемые *стандартные ошибки*. Для зрительной проверки соответствия подобранной плотности реальным данным в строках 6–9 строятся на одном графике гистограмма и график плотности нормального распределения. График строится в результате выполнения двух команд: `hist` и

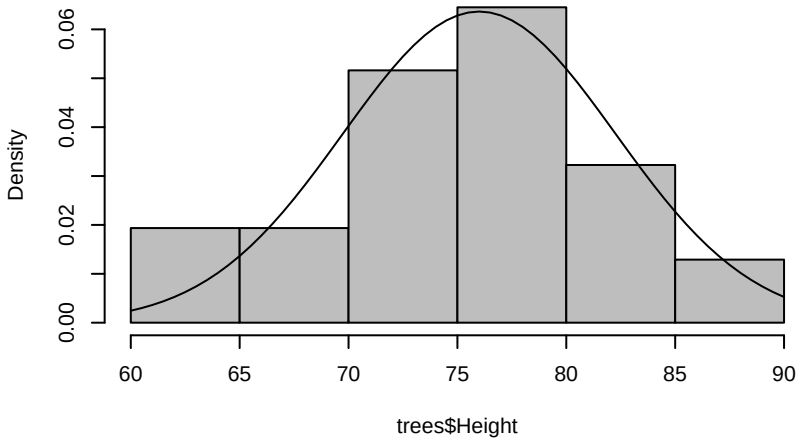


Рис. 4: Подогнанная плотность нормального распределения

`lines`. Вторая команда рисует линию по точкам, координаты которых указаны в аргументах. Абсциссы и ординаты точек передаются в виде двух одномерных массивов. Ординаты точек вычисляются функцией `dnorm`, которая возвращает значения плотности нормального распределения с параметрами `mean` и `sd` во всех точках массива — первого аргумента.

В случае распределения пользователя функция `fitdistr` осуществляет численную максимизацию функции правдоподобия. Для этого используются некоторые общие методы численной оптимизации (например, метод Недлера–Мида или метод Бройдена–Флетчера–Гольдфарба–Шанно), которые имеют ограничения на класс минимизируемых функций. Как правило, функции должны быть достаточное число раз дифференцируемы. В качестве аргументов функции `fitdistr` необходимо передать функцию $p(x;\ \theta)$ и список с начальным значением параметра $\theta$. В демонстрационных целях рассмотрим задачу оценивания параметров смеси распределений. Пусть экспериментальное частотное распределение дискретной случайной величины $X$ имеет вид из таблицы 2[1].

[1]Емельянов Г.В., Скитович В.П. Задачник по теории вероятностей и математической статистике. Л.: Изд-во Ленинградского университета. 1967. 330 с.

Таблица 2: Экспериментальные данные

| $k$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $n_k$ | 28 | 47 | 81 | 67 | 53 | 24 | 13 | 8 | 3 | 2 | 1 |

Предполагаемое распределение для величины $X$ задано формулой

$$\mathbf{P}(X = k) = p(k;\ \lambda_1,\ \lambda_2) = \frac{1}{2} \cdot \frac{\lambda_1^k}{k!} e^{-\lambda_1} + \frac{1}{2} \cdot \frac{\lambda_2^k}{k!} e^{-\lambda_2}, \qquad k = 0,\ 1,\ \ldots; \qquad (3)$$

$\lambda_1$ и $\lambda_2$ — некоторые положительные постоянные. К такому распределению можно придти, если считать, что параметр $\lambda$ пуассоновского распределения является случайным и принимает с равными вероятностями значения $\lambda_1$, $\lambda_2$. Найдем оценки максимального правдоподобия для параметров $\lambda_1$, $\lambda_2$.

```
1  > vals <- 0:10
2  > counts <- c(28,47,81,67,53,24,13,8,3,2,1)
3  > x <- rep(vals,counts)
4  > my.p <- function(x, lambda1, lambda2)
5  + .5*(dpois(x,lambda1)+dpois(x,lambda2))
6  > my.p(0:4,.4,3.2)
7  [1] 0.3555411 0.1992835 0.1311640 0.1148830 0.0894039
8  > fitdistr(x, my.p, list(lambda1=1, lambda2=2))
9      lambda1      lambda2
10   2.1491470    3.5283241
11  (0.1924748) (0.2149279)
```

В строке 1 создается вектор значений — целых чисел от 0 до 10. Строки 1–3 создают фиктивную выборку, как это было описано выше. В строках 4 и 5 определяется функция, вычисляющая вероятности значений в векторе `x` при значениях параметров `lambda1` $= \lambda_1$ и `lambda2` $= \lambda_2$. В строке 6 вычисляются вероятности значений 0, 1, 2, 3, 4 при $\lambda_1 = 0{,}4$ и $\lambda_2 = 3{,}2$. Оценка параметров производится в строках 8–11 с начальными значениями $\lambda_1 = 1$, $\lambda_2 = 2$. В строке 11 в скобках приведены оценки среднеквадратичных отклонений найденных оценок параметров. Интересно сравнить результат оценки параметров с исходными данными. На рис. 5 сплошная линия с кружками изображает полигон частот, пунктирная линия — полигон распределения $\{p(k;\ \lambda_1,\ \lambda_2);\ k \geqslant 0\}$. Интересно сравнить с подгонкой пуассоновского распределения.

```
1  > fitdistr(x,"poisson")
2      lambda
```

```
  2.8379205
 (0.0931593)
> plot(0:10, table(x)/length(x),type="b")
> lines(0:10,my.p(0:10,2.15,3.53),lty=2,col="blue")
> lines(0:10,dpois(0:10,2.84),lty=3,col="red")
```

На рис. 5 пуассоновскому распределению соответствует сплошная ломанная. Вид-

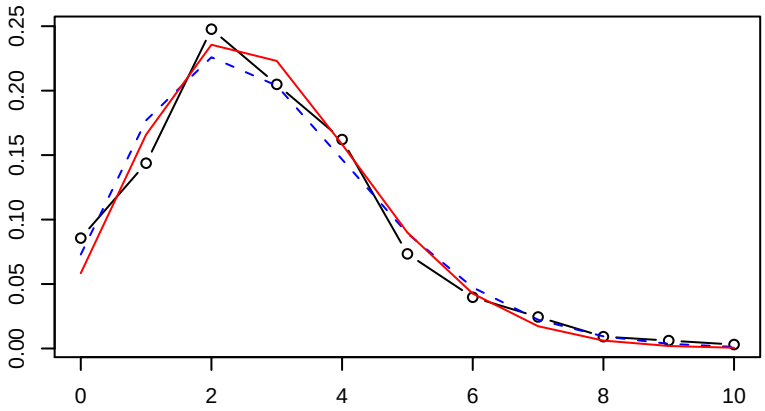


Рис. 5: Полигон частот и подгонка различных распределений

но, что пуассоновское распределение лучше приближает выборочное в окрестности точки максимума, а в области больших значений случайной величины точнее первое из рассматриваемых теоретических распределений.

В случае смещенного экспоненциального распределения с плотностью

$$f(x;\ \theta) = e^{-(x-\theta)}$$

(стр.14) функция правдоподобия имеет вид

$$L(x_1,\ \ldots,\ x_n;\ \theta) = \begin{cases} e^{n(\theta-\bar{x})}, & \theta \leqslant x_{(1)}, \\ 0, & \theta > x_{(1)}. \end{cases}$$

Эта функция разрывна при $\theta = x_{(1)}$, где и достигает наибольшего значения.

### 4.2. Метод аналогий

Метод аналогий заключается в подборе таких значений параметров гипотетического распределения, при котором некоторый набор выборочных числовых

характеристик совпадает с теоретическими. Поскольку можно рассматривать различные наборы числовых характеристик, оценки по методу аналогий определяются не единственным образом.

В качестве примера рассмотрим сдвинутое экспоненциальное распределение. Легко найти, что математическое ожидание имеет вид $1+\theta$, медиана равна $\theta+\ln 2$; кроме того, физический смысл параметра $\theta$ — наименьшее возможное значение случайной величины. Поэтому естественно рассмотреть три оценки:

$$\hat{\theta}_1 = \bar{x} - 1, \qquad \hat{\theta}_2 = \hat{\mu} - \ln 2, \qquad \hat{\theta}_3 = \min\{x_1,\ x_2,\ \ldots,\ x_n\} = x_{(1)}.$$

В следующем примере генерируются 100 выборок объема 50 для $\theta = 1$ и по каждой из этих выборок находятся оценка параметра $\theta$ каждым из трех способов. Затем строится график «ящик с усами», чтобы сравнить качество этих оценок.

```
> x <- data.frame(mean=rep(NA,100), median=rep(NA,100),
+ min=rep(NA,100))
> for (i in 1:100) {
+  samp <- 1-log(runif(50));
+  x$mean[i] <- mean(samp)-1;
+  x$median[i] <- median(samp)-log(2);
+  x$min[i] <- min(samp);
+ };
> x.b <- boxplot(x);
> mn <- mean(x);
> sd <- sd(x);
> xi <- .3+seq(x.b$n);
> points(xi, mn, pch=18, col="blue");
> arrows(xi, mn - sd, xi, mn + sd,
+  code = 3, col = "blue", angle = 75, length = .1)
> abline(1,0,lty=2, lwd=1.5,col="red");
```

График приведен на рис. 6. Значения оценки $\hat{\theta}_1$ сохранены в переменной `x$mean`, значения оценки $\hat{\theta}_2$ — в переменной `x$median`, а значения оценки $\hat{\theta}_3$ записаны в переменной `x$min`. Точно значение параметра $\theta$ отмечено горизонтальной пунктирной линией. Кроме «ящиков с усами» а графике представлены выборочные средние и среднеквадратичные отклонения в виде стрелок с жирной точкой посередине. Видно, что оценки $\hat{\theta}_1$ и $\hat{\theta}_2$ обладают сходными качествами — отклонением от среднего и разбросом значений. Распределение $\hat{\theta}_1$ более симметрично, поскольку выборочное среднее аппроксимируется нормальным распределением на основании центральной предельное теоремы. Оценка $\hat{\theta}$ имеет распределение с малым разбросом. В курсе теории вероятностей и математической статистики доказывается, что оценка $\hat{\theta}_1$ является несмещенной оценкой, $\mathbf{M}\hat{\theta}_1 = \theta$, а $\hat{\theta}_3$ — смещенной,

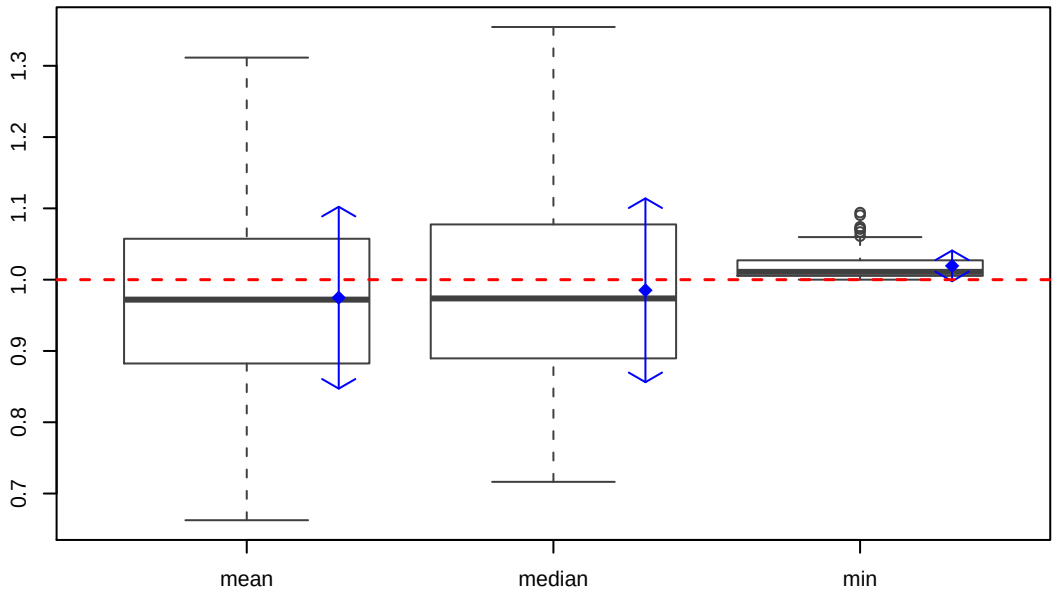


Рис. 6: Выборочное распределение трех оценок параметра сдвига показательного распределения

$\mathbf{M}\hat{\theta}_3 > \theta$. С другой стороны, $\hat{\theta}_3$ имеет меньшую дисперсию по сравнению с $\hat{\theta}_1$, она *более эффективна*.

Для оценки параметров по методу аналогий прежде всего требуется найти аналитическое выражение числовых характеристик через параметры распределения. Эта задача целиком возлагается на исследователя-человека.

### 4.3. Байесовское оценивание

Суть байесовского подхода состоит в том, что неизвестный параметр распределения $\theta$ рассматривается как *случайная величина*. Для непрерывной случайной величины $\theta$ известна ее плотность распределения $q(t)$. В дискретном случае $q(t)$ задает распределение вероятностей. Функция $q(t)$ называется априорной, т. е. данной до эксперимента. Байесовский подход предполагает, что неизвестный параметр $\theta$ выбирается случайным образом из распределения $q(t)$ один раз до получения выборки значений случайной величины $X$.

После проведения эксперимента мы получаем дополнительную информацию в виде апостериорного распределения параметра $\theta$. В качестве байесовской оценки выбирают некоторые характеристики полученного апостериорного распределения. Часто при байесовском подходе выбирают значение $\hat{\theta}^B$ оценки, минимизи-

рующее среднеквадратическое отклонение $\mathbf{M}(\hat{\theta} - \theta)^2$.

В случае дискретного распределения параметра $\theta$ и дискретного распределения случайной величины $X$ получаем

$$q(u) = \mathbf{P}(\theta = u), \quad u \in \{u_1,\ u_1,\ u_3,\ \ldots\},$$
$$f(x;\ u) = \mathbf{P}(X = x|\theta = u), \quad x \in \{a_1,\ a_2,\ a_3,\ \ldots\}.$$

Тогда функция правдоподобия будет равна $L(x_1,\ \ldots,\ x_n;\ u) = \prod_{i=1}^{n} f(x_i;\ u)$. Вероятность получить значения $x_1,\ \ldots,\ x_n$ выборки случайных величин $X_1,\ \ldots,\ X_n$ при условии $\theta = u$ вычисляется по следующей формуле:

$$\mathbf{P}(X_1 = x_1,\ \ldots,\ X_n = x_n|\theta = u) = q(u)L(x_1,\ \ldots,\ x_n;\ u).$$

Чтобы получить апостериорное распределение параметра $\theta$, воспользуемся формулой Байеса

$$\mathbf{P}(\theta = u|X_1 = x_1,\ \ldots,\ X_n = x_n) = \frac{\mathbf{P}(X_1 = x_1,\ \ldots,\ X_n = x_n|\theta = u)}{\sum\limits_i q(u_i)L(x_1,\ \ldots,\ x_n;\ u_i)}.$$

Будем искать байесовскую оценку $\hat{\theta}^B(x_1,\ \ldots,\ x_n)$, которая минимизирует по $\hat{\theta}$ функционал

$$\mathbf{M}(\hat{\theta}(X_1,\ \ldots,\ X_n) - \theta)^2.$$

Можно доказать, что при данном критерии байесовской оценкой дискретного параметра $\theta$ для дискретной случайной величины $X$ является

$$\hat{\theta}^B(x_1,\ \ldots,\ x_n) = \mathbf{M}(\theta|X_1 = x_1,\ \ldots,\ X_n = x_n) = \\ = \sum_i u_i \mathbf{P}(\theta = u_i|X_1 = x_1,\ \ldots,\ X_n = x_n).$$

Пусть теперь векторный параметр $\theta = (Y_1,\ \ldots,\ Y_m) \in \mathbf{R}^m$ и случайный вектор $X = (X_1,\ \ldots,\ X_n) \in \mathbf{R}^n$ имеют непрерывное распределение. Тогда при $x = (x_1,\ \ldots,\ x_n)$ и $y = (y_1,\ \ldots,\ y_m)$ имеем совместную интегральную функцию распределения $F(x,\ y) = \mathbf{P}(X < x,\ \theta < y)$ и совместную плотность распределения

$$f(x,\ y) = \frac{\partial^{n+m}}{\partial x_1 \ldots \partial x_n \partial y_1 \ldots \partial y_m} F(x,\ y).$$

Для определения апостериорной плотности распределения параметра $\theta$ вы-

полним следующие выкладки

$$\mathbf{P}(X < x|\theta = y) = \lim_{\Delta \to +0} \mathbf{P}(X < x|y \leqslant \theta \leqslant y + \Delta),$$
$$f_X(x|\theta = y) = \frac{\partial^n}{\partial x_1 \dots \partial x_n} \mathbf{P}(X < x|\theta = y) =$$
$$= \frac{f(x,\ y)}{f_\theta(y)} = \frac{f(x,\ y)}{\int\limits_{\mathbf{R}^n} f(x,\ y)dx}.$$

В результате получаем формулу Байеса для плотности случайной величины $X$

$$f_X(x|\theta = y) = \frac{f_X(x) f_\theta(y|X = x)}{\int\limits_{\mathbf{R}^n} f_X(x) f_\theta(y|X = x)dx}.$$

Условная плотность величины $\theta$ при условии $X = x$ равна

$$f_\theta(y|X = x) = \frac{f_\theta(y) f_X(x|\theta = y)}{\int\limits_{\mathbf{R}^m} f_\theta(y) f_X(x|\theta = y)dy}. \tag{4}$$

Эта плотность определяет так называемое апостериорное распределение параметра $\theta$. А равенство (4) называется формулой Байеса для апостериорного распределения.

Тогда байесовской оценкой, минимизирующей среднеквадратическое отклонение для непрерывного параметра $\theta$, является функция

$$\hat{\theta}^B(x_1,\ \dots,\ x_n) = \mathbf{M}(\theta|X = x) = \int\limits_{\mathbf{R}^m} y f_\theta(y|X = x)dy.$$

Рассмотрим следующий пример. В схеме Бернулли производится $n$ экспериментов с вероятностью успеха $\mathbf{P}(X = 1) = p,\ 0 \leqslant p \leqslant 1$. Пусть параметр $p$ неизвестен и имеет априорное распределение $\mathbf{P}(p = p_i) = q_i,\ i = 1, 2, \dots, r$. Если в результате серии из $n$ опытов число успехов равно $k$, то функция правдоподобия имеет следующий вид

$$L(x_1,\ \dots,\ x_n;\ p_i) = p_i^k (1 - p_i)^{n-k}.$$

Тогда апостериорные вероятности при числе успехов $k = x_1 + \dots + x_n$ будут равны

$$\mathbf{P}(p = p_i|X_1 = x_1,\ \dots,\ X_n = x_n) = \frac{q_i p_i^k (1 - p_i)^{n-k}}{\sum\limits_{j=1}^{r} q_j p_j^k (1 - p_j)^{n-k}}.$$

Таким образом, байесовской оценкой параметра $p$ является

$$\hat{p}^B(x_1,\ \ldots,\ x_n) = \sum_{i=1}^{r} p_i \frac{q_i p_i^k (1-p_i)^{n-k}}{\sum\limits_{j=1}^{r} q_j p_j^k (1-p_j)^{n-k}}.$$

Продемонстрируем процедуру байесовского оценивания с помощью пакета R. Функции, необходимые для вычисления байесовских оценок, содержатся в пакете LearnBayes, который подключается с помощью команды `library("LearnBayes")`. В данном примере дано априорное распределение вероятности успеха $p$ в единичном опыте

$$\mathbf{P}(p = 0{,}1) = \mathbf{P}(p = 0{,}4) = \mathbf{P}(p = 0{,}6) = \mathbf{P}(p = 0{,}8) = 0{,}25.$$

Далее мы генерируем выборку `X` из 20 значений при некоторой вероятности успеха $p$. Количество успехов `k` в выборке получилось равным 10. Функция `pdisc` строит апостериорное распределение параметра $p$. Последняя команда `sum(p*post)` вычисляет байесовскую оценку $\hat{p}^B = 0{,}5025679$ параметра $p$ для дискретных случайных величин.

```
> p<-c(.1,.4,.6,.8)
> q<-c(.25,.25,.25,.25)
> X
[1] 0 1 1 0 0 0 1 0 0 0 0 0 1 1 1 0 1 1 1 1
> k<-sum(X)
> k
[1] 10
> data<-c(k, 20-k)
> post<-pdisc(p,q,data)
> round(cbind(p, q, post),4)
p    q   post
[1,] 0.1 0.25 0.0000
[2,] 0.4 0.25 0.4957
[3,] 0.6 0.25 0.4957
[4,] 0.8 0.25 0.0086
> sum(p*post)
[1] 0.5025679
```

В примере апостериорные вероятности для двух значений $p = 0{,}4$ и $p = 0{,}6$ параметра $p$ оказались равны. Апостериорная вероятность значения $p = 0{,}1$ оказалась очень мала. Байесовская оценка не совпадает ни с одним из возможных значений параметра $p$. Выборка `X` была получена при значении параметра $p = 0{,}4$. Таким

образом, байесовская оценка оказалась довольно близка к истинному значению параметра при малом объеме выборки.

```
> X<-sample(c(1,0),20,replace=TRUE, c(.4,.6))
```

Рассмотрим пример, в котором случайная величина и параметр распределения этой случайной величины являются непрерывными случайными величинами. Пусть случайная величина $X$ имеет экспоненциальное распределение с математическим ожиданием $\lambda > 0$ и плотностью

$$f_X(x|\lambda) = \begin{cases} \lambda^{-1}e^{-x/\lambda}, & x \geqslant 0, \\ 0, & x < 0. \end{cases}$$

И пусть параметр $\lambda$ имеет непрерывное распределение на отрезке $[a,\ b]$, $b > a > 0$, с плотностью

$$q(u) = \begin{cases} k/u, & u \in [a,\ b], \\ 0, & u \notin [a,\ b], \end{cases}$$

где константа $k = 1/(\ln(b) - \ln(a))$ находится из условия нормировки плотности $\int\limits_{-\infty}^{\infty} q(u)du = 1$. Тогда плотность апостериорного распределения параметра $\lambda$ при повторной выборке объема $n$ и $s = x_1 + \ldots + x_n$ вычисляется следующим образом

$$q(\lambda|X_1 = x_1,\ \ldots,\ X_n = x_n) = \frac{\dfrac{k}{\lambda}\prod\limits_{i=1}^{n}\left(\lambda^{-1}e^{-x_i/\lambda}\right)}{\int\limits_a^b \dfrac{k}{\lambda}\prod\limits_{i=1}^{n}\left(\lambda^{-1}e^{-x_i/\lambda}\right)d\lambda} = \frac{\lambda^{-n-1}\exp(-s/\lambda)}{\int\limits_a^b \lambda^{-n-1}\exp(-s/\lambda)d\lambda}.$$

Интеграл в знаменателе, уменьшая $a$ и увеличивая $b$, можно сделать сколь угодно близким к $s^{-n}\Gamma(n)$. Таким образом, плотность апостериорного распределения можно с любой заданной точностью аппроксимировать

$$q(\lambda|X_1 = x_1,\ \ldots,\ X_n = x_n) = \frac{\lambda^{-n-1}\exp(-s/\lambda)}{s^{-n}\Gamma(n)}.$$

Используя формулу для байесовской оценки непрерывного параметра $\lambda$, получаем

$$\hat{\lambda}^B(x_1,\ \ldots,\ x_n) = \int\limits_0^\infty \lambda q(\lambda|x_1,\ \ldots,\ x_n)d\lambda = \int\limits_0^\infty \frac{\lambda^{-n}\exp(-s/\lambda)}{s^{-n}\Gamma(n)}d\lambda =$$

$$= \frac{s^{-(n-1)}\Gamma(n-1)}{s^{-n}\Gamma(n)} = \frac{s}{n-1}.$$

Параметр $\lambda$ является математическим ожиданием, поэтому его несмещенной оценкой является выборочное среднее.

Рассмотрим следующую задачу [5]. Во время тестирования были измерены времена (в часах) непрерывной работы пяти лампочек, которые оказались равны 751, 594, 1213, 1126 и 819. Предполагается, что время непрерывной работы лампочки имеет экспоненциальное распределение с неизвестным средним $\lambda$. Необходимо с помощью пакета R получить 1000 реализаций случайной величины $\theta$, имеющей апостериорное распределение случайной величины $1/\lambda$. Заметим, что случайная величина $\theta$ имеет гамма-распределение с параметрами `shape = n` и `scale = 1/s` (или `rate = s`). Следующая последовательность команд иллюстрирует решение задачи.

```
> data<-c(751, 594, 1213, 1126, 819)
> data
[1]  751  594 1213 1126  819
> n=5; s=sum(data)
> rgamma(1000, shape=n, scale=1/s)
[1] 0.0016505574 0.0010949532 0.0015292299 0.0018146785 0.00...
[6] 0.0004224466 0.0012462338 0.0015636684 0.0010781766 0.00...
[11] 0.0004354318 0.0012353216 0.0011187441 0.0004500492 0.00...
[16] 0.0006452703 0.0010386560 0.0006597630 0.0008603517 0.00...
...
```

В следующем примере генерируется 1000 выборок по 30 значений экспоненциальной случайной величины со средним $\lambda$. В каждой выборке сначала выбирается случайный параметр $\lambda$, распределенный в интервале $(a,\ b)$ с плотностью, пропорциональной $1/\lambda$. Для каждой выборки вычисляется выборочное среднее и байесовская оценка. В пятой строке происходит нормировка оценок, а команда в шестой строке строит «ящик с усами» для полученной нормировки. Из рисунка 7 видно, что обе оценки дают близкие результаты.

```
> a=0.0001; b=10000
> lambda<-a*((b/a)^runif(1000))
> for (i in 1:1000) {data<-rexp(30,1/lambda[i]);
+ m[i]<-mean(data); s[i]<-sum(data)/29}
> mnorm<-m/lambda; snorm<-s/lambda;
> boxplot(mnorm,snorm);
> mean(mnorm); mean(snorm)
[1] 1.004463
[1] 1.0391
```

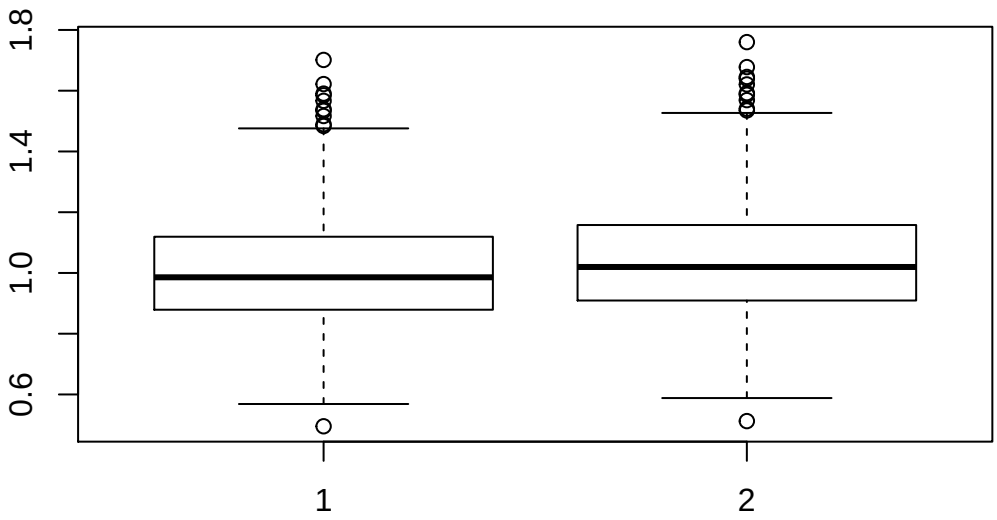


Рис. 7: Нормированное значение выборочного среднего и байесовской оценки для параметра $\lambda$ экспоненциального распределения

## Литература

1. The R project for satistical computing // http://www.r-project.org

2. Everitt B., Hothorn T. A handbook of statistical analysis using R. 2nd ed. Chapman and HALL/CRC. 2010.

3. Duller C. Einfürung in die nichtparametrische Statistik mit SAS und R. Physica-Verlag Heidelberg. 2008.

4. Shumway R.H., Stoffer D.S. Time series analysis and its applications with R examples. 3rd ed. Springer, 2011.

5. Albert J. Bayesian Computation with R. 2rd ed. Springer, 2009.

# ЧИСЛЕННЫЕ МЕТОДЫ МАТЕМАТИЧЕСКОЙ СТАТИСТИКИ В ПАКЕТЕ R

Составители:
Андрей Владимирович **Зорин**
Евгений Владимирович **Кудрявцев**
Мария Анатольевна **Рачинская**

**Учебно-методическое пособие**

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский
Нижегородский государственный университет
им. Н. И. Лобачевского».
603950, Нижний Новгород, пр. Гагарина, 23