

Разведочный анализ данных

Exploratory Data Analysis (EDA)

Храмов Д.А.

10.02.2020

Содержание

- ▶ Типы данных
- ▶ Описательная статистика: меры центра и разброса
- ▶ Разведочный анализ

Типы данных: статистические шкалы

- ▶ **Номинативные (категориальные, качественные):**
цвета светофора, пол — арифметические операции невозможны
- ▶ **Ранговые:** финишный протокол гонки — возможно сравнение по величине
- ▶ **Количественные**
 - ▶ непрерывные: рост, вес
 - ▶ дискретные: число потомков, тестовые баллы

Типы данных в R

- ▶ Количественные данные
 - ▶ числовые (numeric)
 - ▶ символьные (character)
 - ▶ логические (logical)
- ▶ Категориальные данные
 - ▶ факторы (factor)

```
> 174
[1] 174
> class(174)
[1] "numeric"
> class(174L)
[1] "integer"
> class("174L")
[1] "character"
> class('174L')
[1] "character"
> class(TRUE)
[1] "logical"
> class(T)
[1] "logical"
```

Что означает [1]?

Количественные переменные. Векторы

```
height <- c(174, 162, 188, 192, 165, 168, 174)
```

```
class(height)      # класс переменной
```

```
## [1] "numeric"
```

```
str(height)        # структура переменной
```

```
##  num [1:7] 174 162 188 192 165 168 174
```

```
is.vector(height)  # проверка: это вектор?
```

```
## [1] TRUE
```

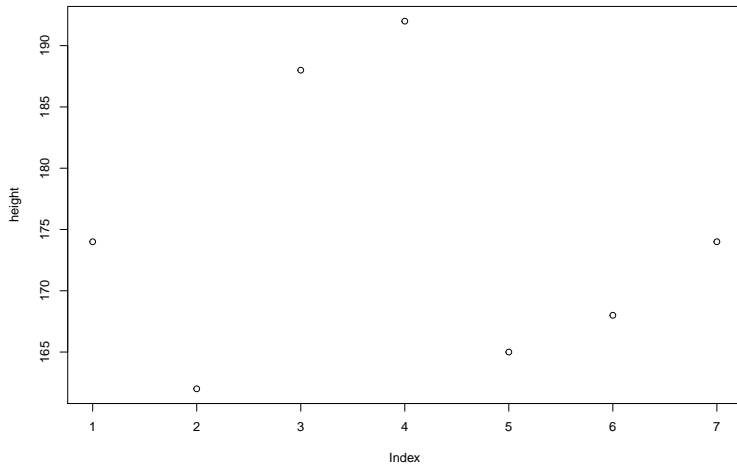
Векторы в R – числовые, символьные и логические – играют роль элементарных типов данных, из которых строятся все остальные типы. Скаляры представляют собой векторы единичной длины.

Особенности синтаксиса R

1. В именах переменных можно использовать точку '.'. Часто ее используют вместо '_'. Обращение к методам объекта или элементам данных осуществляется через '\$'.
2. Присваивание обозначается стрелкой <-. Можно использовать обычное равенство =.

График

```
plot(height)
```



Доступ к элементам

```
height[1]           # 1-й элемент
```

```
## [1] 174
```

```
length(height)      # длина вектора
```

```
## [1] 7
```

```
height[2:5]         # элементы со 2-го по 5-й
```

```
## [1] 162 188 192 165
```

```
height[-1]          # все элементы, кроме 1-го
```

```
## [1] 162 188 192 165 168 174
```

```
height[length(height)] # последний элемент
```

```
## [1] 174
```


Номинативные данные

```
sex <- c("m", "f", "m", "m", "f", "f", "m")
```

```
str(sex)
```

```
## chr [1:7] "m" "f" "m" "m" "f" "f" "m"
```

```
is.character(sex) # проверка: это символьные данные?
```

```
## [1] TRUE
```

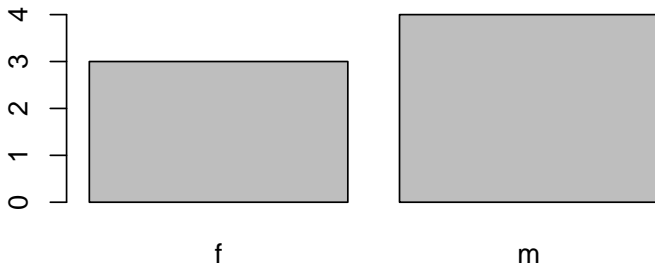
```
# plot(sex)           # выдает сообщение об ошибке
```

Создаем фактор

```
sex.f <- factor(sex)  
str(sex.f)
```

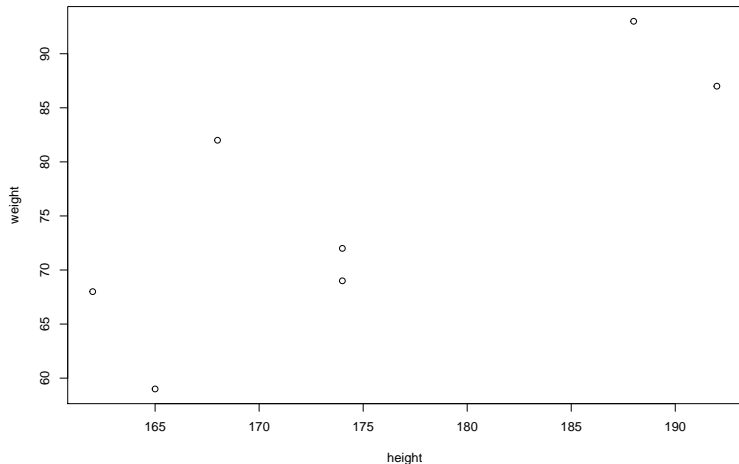
```
## Factor w/ 2 levels "f","m": 2 1 2 2 1 1 2
```

```
plot(sex.f)
```



Группируем данные

```
weight <- c(69, 68, 93, 87, 59, 82, 72)  
plot(height, weight)
```

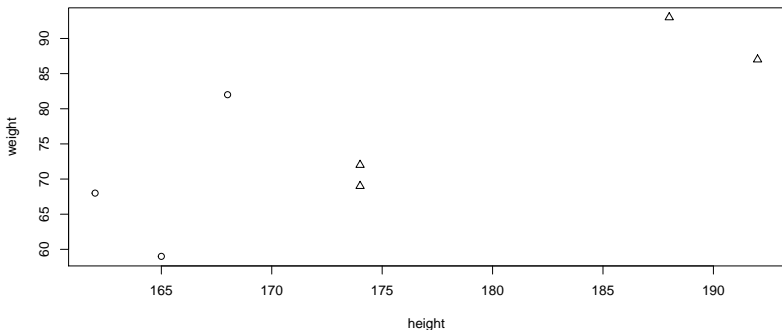


Добавляем тип маркера (pch)

```
as.numeric(sex.f) # тип маркера должен быть числом
```

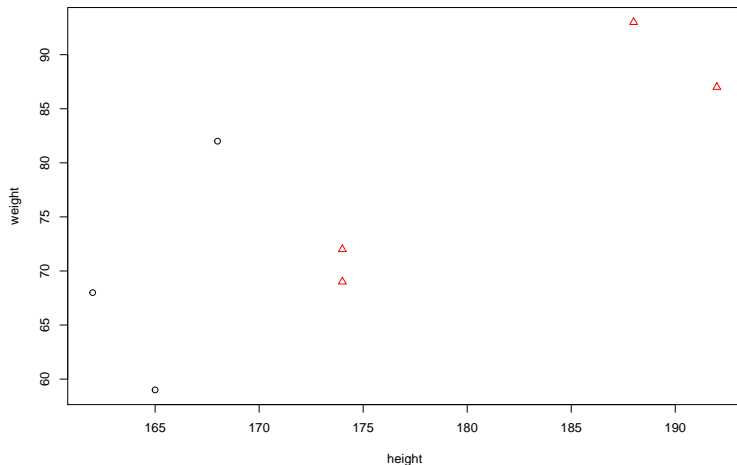
```
## [1] 2 1 2 2 1 1 2
```

```
plot(height, weight, pch=as.numeric(sex.f))
```



Добавляем цвет (col)

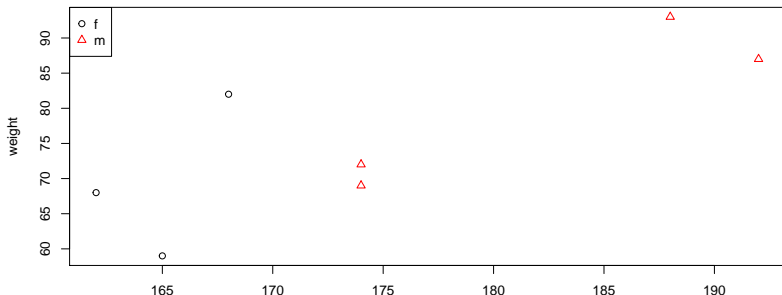
```
plot(height, weight, pch=as.numeric(sex.f),  
      col=as.numeric(sex.f))
```



.. и легенду

```
levels(sex.f) # уровни фактора  
nlevels(sex.f) # число уровней
```

```
plot(height, weight, pch=as.numeric(sex.f),  
      col=as.numeric(sex.f))  
legend("topleft", pch=1:nlevels(sex.f),  
      col=1:nlevels(sex.f), legend=levels(sex.f))
```



Контейнеры разнородных элементов: списки

```
l <- list(1,"a", c(TRUE, F))
```

```
str(l)
```

```
## List of 3  
## $ : num 1  
## $ : chr "a"  
## $ : logi [1:2] TRUE FALSE
```

```
str(l[1])    # l[1]    - доступ к 1-му подписку
```

```
## List of 1  
## $ : num 1
```

```
str(l[[1]])  # l[[1]] - доступ к 1-му элементу
```

```
## num 1
```

Списки: имена элементов

```
l <- list(num=1, ch="a", log=c(TRUE, F))  
str(l)
```

```
## List of 3  
## $ num: num 1  
## $ ch : chr "a"  
## $ log: logi [1:2] TRUE FALSE
```

```
l$log
```

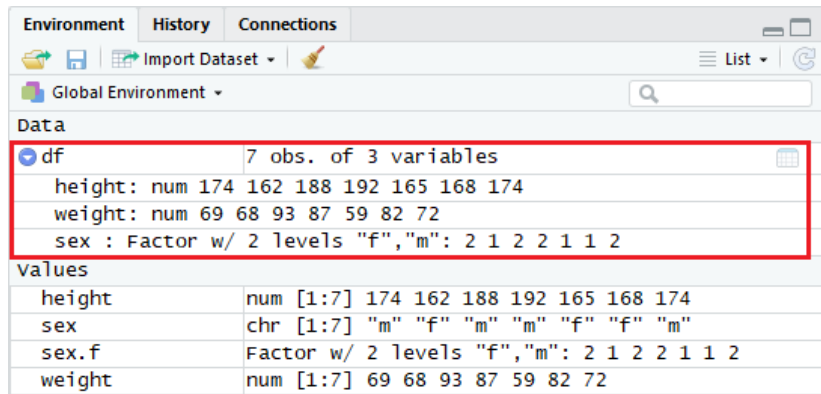
```
## [1] TRUE FALSE
```

```
length(l$log)
```

```
## [1] 2
```


Таблица (data frame)

```
df <- data.frame(height, weight, sex)
```



The screenshot shows the RStudio Environment pane with the 'df' data frame selected. The data frame has 7 observations and 3 variables: height (numeric), weight (numeric), and sex (factor with levels 'f' and 'm'). The data is displayed in a table format with a red border around the main data rows.

Environment	History	Connections
Global Environment		
Data		
df	7 obs. of 3 variables	
height:	num 174 162 188 192 165 168 174	
weight:	num 69 68 93 87 59 82 72	
sex :	Factor w/ 2 levels "f","m": 2 1 2 2 1 1 2	
values		
height	num [1:7] 174 162 188 192 165 168 174	
sex	chr [1:7] "m" "f" "m" "m" "f" "f" "m"	
sex.f	Factor w/ 2 levels "f","m": 2 1 2 2 1 1 2	
weight	num [1:7] 69 68 93 87 59 82 72	

Таблица df в окне Environment RStudio

Таблицы: просмотр содержимого


```
View(df)
```

	height	weight	sex
1	174	69	m
2	162	68	f
3	188	93	m
4	192	87	m
5	165	59	f
6	168	82	f
7	174	72	m

Просмотр содержимого таблицы df

Операции в окне Environment

Environment History Connections

Import Dataset  Очистка памяти List

Global Environment

Data

df 7 obs. of 3 variables

height: num 174 162 188 192 165 168 174

weight: num 69 68 93 87 59 82 72

sex : Factor w/ 2 levels "f","m": 2 1 2 2 1 1 2


values

height num [1:7] 174 162 188 192 165 168 174

sex chr [1:7] "m" "f" "m" "m" "f" "f" "m"

sex.f Factor w/ 2 levels "f","m": 2 1 2 2 1 1 2

weight num [1:7] 69 68 93 87 59 82 72

Просмотр содержимого 

Таблицы: доступ к элементам

```
df[1,1]      # элемент 1-й строки и 1-го столбца
```

```
## [1] 174
```

```
df$height[1] # 1-й элемент столбца с именем height
```

```
## [1] 174
```

```
df[,1]       # 1-й столбец
```

```
## [1] 174 162 188 192 165 168 174
```

```
df[1,c("height","sex")] # 1-я строка столбцов height и sex
```

```
##   height sex
```

```
## 1    174  m
```

Таблицы: терминология

Таблица — основной способ представления данных. Строки таблицы содержат наблюдения, столбцы — признаки.

- ▶ строка = наблюдение = объект
- ▶ колонка = столбец = признак (feature) = переменная

Генеральная совокупность и выборка

Генеральная совокупность (population) — множество всех объектов, относительно которых мы хотим сделать выводы в рамках нашего исследования.

На какое множество объектов вы хотели бы обобщить результаты ваших исследований? — это и есть генеральная совокупность.

Некоторые элементы, случайным образом взятые из генеральной совокупности называются, **выборкой** (sample).

Выборка должна быть **репрезентативной**, то есть служить моделью (уменьшенной копией) генеральной совокупности.

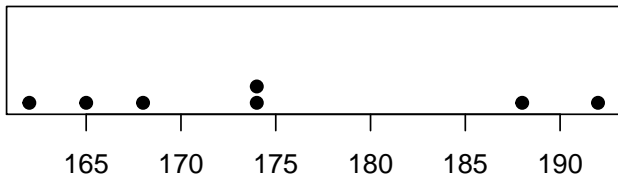
Меры центральной тенденции: среднее

```
height <- c(174, 162, 188, 192, 165, 168, 174)
```

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

```
mean(height) # sum(height)/length(height)
```

```
## [1] 174.7143
```



Меры центральной тенденции: медиана

```
height <- c(174, 162, 188, 192, 165, 168, 174)
median(height)
```

```
## [1] 174
```

```
order(height)
```

```
## [1] 2 5 6 1 7 3 4
```

```
height[order(height)]
```

```
## [1] 162 165 168 174 174 188 192
```


Меры центральной тенденции: влияние выбросов

```
height <- c(height, 225)  
mean(height)
```

```
## [1] 181
```

```
median(height)
```

```
## [1] 174
```

```
mean(height, trim=1/8)
```

```
## [1] 176.8333
```

Если в выборке присутствуют выбросы, выбираем в качестве меры среднего медиану или усеченное (урезанное) среднее (`mean()` с опцией `trim`).

Меры разброса: дисперсия, стандартное отклонение и квантили

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

```
var(height)
```

```
## [1] 427.1429
```

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

```
sd(height)
```

```
## [1] 20.66743
```

Меры разброса: квантили

```
## [1] 162 165 168 174 174 188 192 225
```

```
max(height) - min(height) # размах выборки
```

```
## [1] 63
```

```
summary(height) # квартили
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    162.0   167.2   174.0   181.0   189.0   225.0
```

```
IQR(height) # межквартильный размах
```

```
## [1] 21.75
```

Меры центра характеризуют точность, а меры разброса — кучность.

Немного терминологии

“Статистика” имеет три значения:

1. Наука, изучающая общие вопросы сбора, измерения и анализа массовых статистических (количественных или качественных) данных.
2. Сами статистические данные.
3. Числовая функция от выборки, не зависящая от параметров распределения: среднее, дисперсия и т. д.

Говоря о статистиках мы имеем в виду третье из указанных значений.

Среднее и дисперсия как и другие статистики относятся к генеральной совокупности.

Если статистика относится к выборке, говорят о выборочном среднем, выборочной дисперсии и т. п.

Но: на практике мы всегда (почти) имеем дело с выборками. Поэтому слово “выборочный” мы будем опускать, говоря просто о среднем и дисперсии.

Разведочный анализ



Швейцарская банкнота

Набор данных Swiss Bank Notes

Данные из книги: *Flury, Riedwyl. Multivariate statistics. A practical approach, Chapman & Hall, 1988.*

- ▶ X1: длина банкноты
- ▶ X2: высота банкноты, измеренная слева
- ▶ X3: высота справа
- ▶ X4: кайма нижняя
- ▶ X5: кайма верхняя
- ▶ X6: диагональ центральной картинки

Нужно определить, подлинная банкнота или фальшивая.

Фрагмент данных

Length	H_l	H_r	dist_l	dist_up	Diag
214,8	131,0	131,1	9,0	9,7	141,0
214,6	129,7	129,7	8,1	9,5	141,7
214,8	129,7	129,7	8,7	9,6	142,2
214,8	129,7	129,6	7,5	10,4	142,0
215,0	129,6	129,7	10,4	7,7	141,8
215,7	130,8	130,5	9,0	10,1	141,4
215,5	129,5	129,7	7,9	9,6	141,6
214,5	129,6	129,2	7,2	10,7	141,7

Загрузка данных

```
getwd() # где находится рабочий каталог?
```

```
## [1] "D:/интеллектуальный_анализ_данных_2020/lectures/02"
```

```
# setwd('path/to/data') # установить рабочий каталог  
swiss.0 <- read.table("data/Swiss Bank Notes.dat",  
                      header=T, sep=" ", dec=".",")  
head(swiss.0, n=3) # выведем "голову" таблицы
```

```
##   Length  H_l  H_r dist_l dist_up Diag  
## 1  214.8 131.0 131.1    9.0    9.7 141.0  
## 2  214.6 129.7 129.7    8.1    9.5 141.7  
## 3  214.8 129.7 129.7    8.7    9.6 142.2
```


Знакомство с данными. Проверка на пропуски

```
dim(swiss.0)      # размеры таблицы swiss.0
```

```
## [1] 200    6
```

```
summary(swiss.0)
```

##	Length	H_l	H_r	dist_l
##	Min. :213.8	Min. :129.0	Min. :129.0	Min. : 7.200
##	1st Qu.:214.6	1st Qu.:129.9	1st Qu.:129.7	1st Qu.: 8.200
##	Median :214.9	Median :130.2	Median :130.0	Median : 9.100
##	Mean :214.9	Mean :130.1	Mean :130.0	Mean : 9.418
##	3rd Qu.:215.1	3rd Qu.:130.4	3rd Qu.:130.2	3rd Qu.:10.600
##	Max. :216.3	Max. :131.0	Max. :131.1	Max. :12.700
##	dist_up	Diag		
##	Min. : 7.70	Min. :137.8		
##	1st Qu.:10.10	1st Qu.:139.5		
##	Median :10.60	Median :140.4		
##	Mean :10.65	Mean :140.5		
##	3rd Qu.:11.20	3rd Qu.:141.5		
##	Max. :12.30	Max. :142.4		

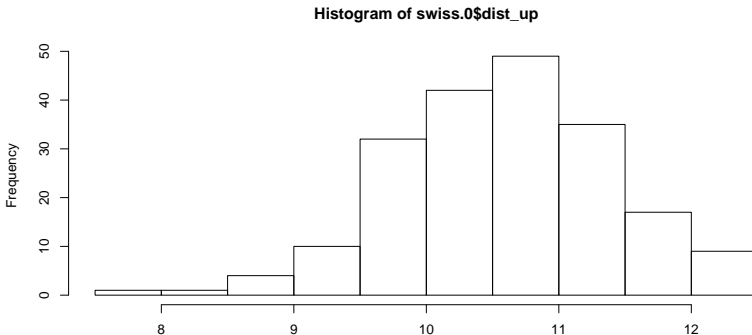
summary позволяет увидеть возможные ошибки в данных.

Гистограммы

Гистограмма — “оценка” плотности распределения случайной величины, построенная по выборке. Пусть n_i число элементов выборки, попавших в i -й интервал Δ_i .

$$h_i = n_i$$

```
hist(swiss.0$dist_up)
```



Построение гистограммы

1. Множество значений, которое может принимать элемент выборки, разбивается на несколько интервалов (bins), чаще всего — одинаковых, но не обязательно.
2. Эти интервалы откладываются на горизонтальной оси.
3. Над каждым интервалом строится прямоугольник, высота которого пропорциональна числу элементов выборки, попавших в соответствующий интервал.

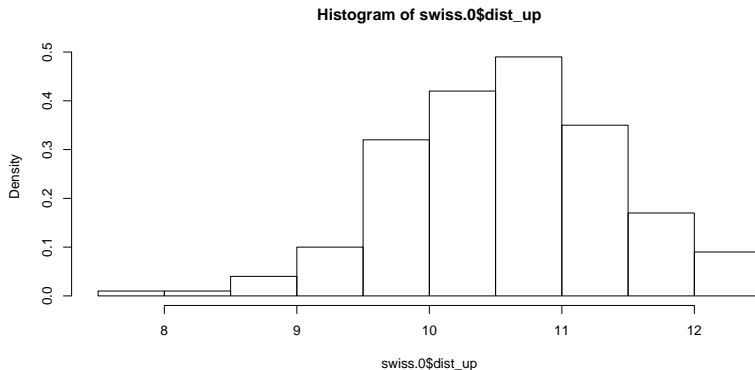
Если используются интервалы разной длины, то высота прямоугольника выбирается таким образом, чтобы его площадь была пропорциональна числу элементов выборки, которые попали в этот интервал.

Нормализация гистограммы

Нормализация позволяет сравнить разные выборки и оценить долю от общего числа данных, попавших в определенный интервал

$$h_i = \frac{n_i}{n\Delta_i}$$

```
hist(swiss.0$dist_up, freq = F)
```

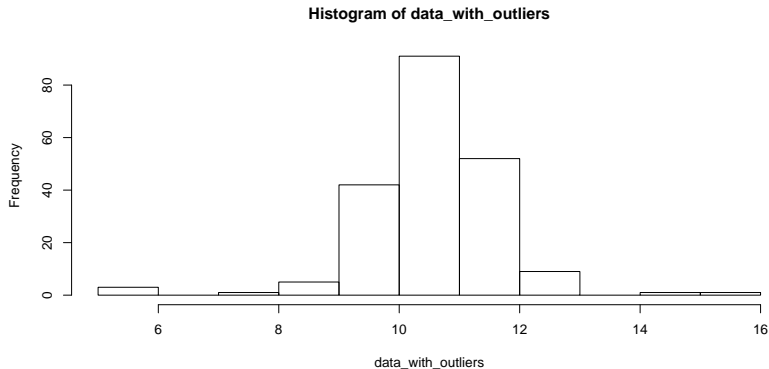


Гистограмма позволяет увидеть выбросы

— аномально большие или аномально малые наблюдения

```
outliers <- c(5.9, 5.1, 15.1, 14.9, 5.2)
data_with_outliers <- c(swiss.0$dist_up, outliers)
```

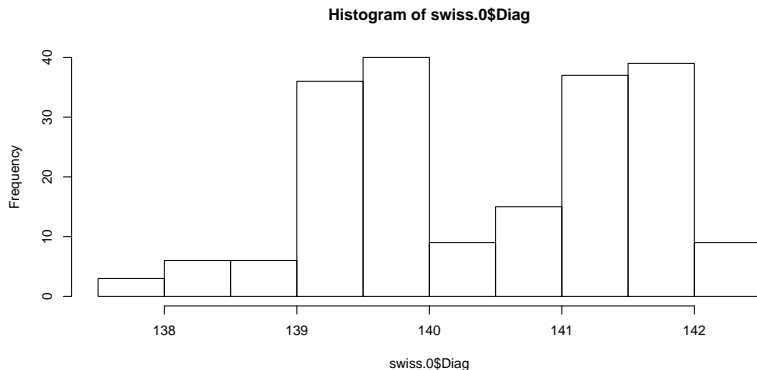
```
hist(data_with_outliers)
```



Гистограмма порождает гипотезы

Гипотеза: длина ≤ 140 — один вид банкнот; длина > 141 — другой вид.

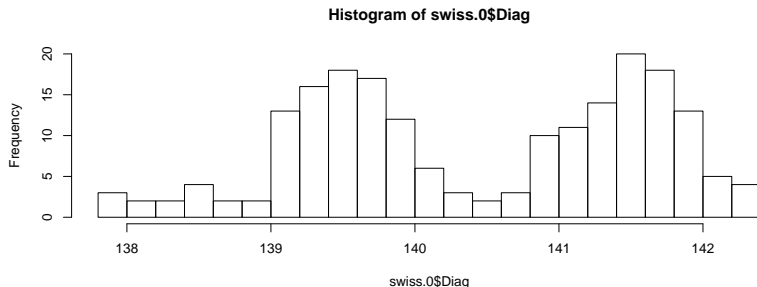
```
hist(swiss.0$Diag)
```



Гистограмма: открытые вопросы

- ▶ Сколько должно быть интервалов? Histogram. Number of bins and width
- ▶ Должны ли интервалы быть равными?

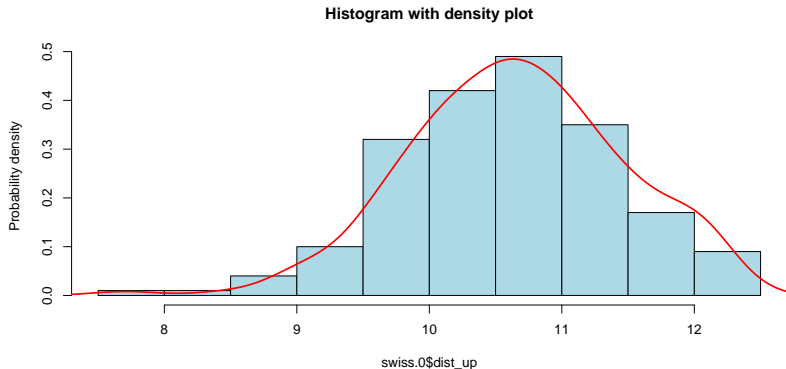
```
hist(swiss.0$Diag, breaks=18)
```



Почему интервалов не 18? Из справки: "... the number is a suggestion only; the breakpoints will be set to pretty values".

Гистограмма и оценка плотности распределения

```
hist(swiss.0$dist_up, freq = F, col = "lightblue",  
     ylab = "Probability density",  
     main = "Histogram with density plot")  
lines(density(swiss.0$dist_up), col = "red", lwd = 2)
```



Раскроем карты

Из 200 банкнот первые 100 — подлинные, остальные фальшивые.

```
> swiss.0[swiss.0$Diag <= 140,]
```

	Length	H_l	H_r	dist_l	dist_up	Diag
70	214.9	130.2	130.2	8.0	11.2	139.6
101	214.4	130.1	130.3	9.7	11.7	139.8
102	214.9	130.5	130.2	11.0	11.5	139.5
105	214.7	130.2	130.3	11.8	10.9	139.7
106	215.0	130.2	130.2	10.6	10.7	139.9
...						
198	214.8	130.3	130.4	10.6	11.1	140.0
199	214.7	130.7	130.8	11.2	11.2	139.4
200	214.3	129.9	129.9	10.2	11.5	139.6

```
> nrow(swiss.0[swiss.0$Diag <= 140,])
```

```
[1] 91
```

Матрица диаграмм рассеивания

```
pairs(swiss.0) # или: plot(swiss.0)
```

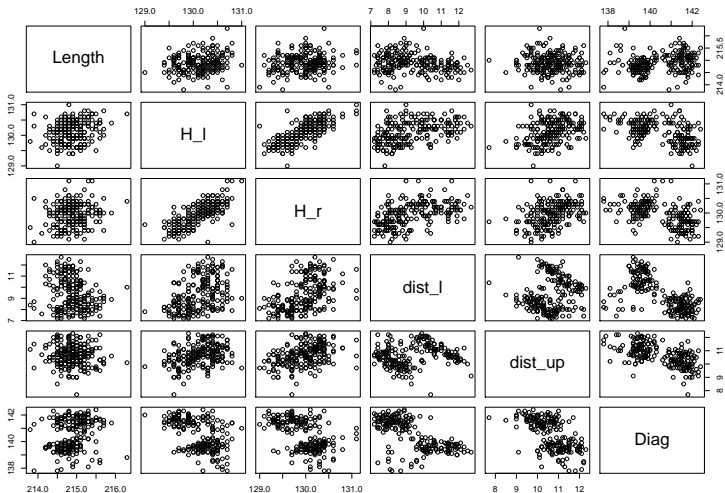
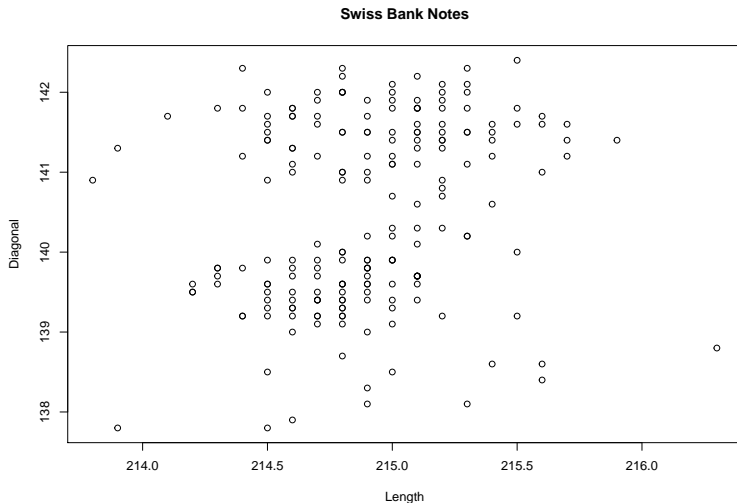


Диаграмма рассеивания (scatter plot)

```
plot(swiss.0$Length, swiss.0$Diag, xlab="Length",  
ylab="Diagonal", main="Swiss Bank Notes")
```



Добавим столбец — индикатор типа банкнот

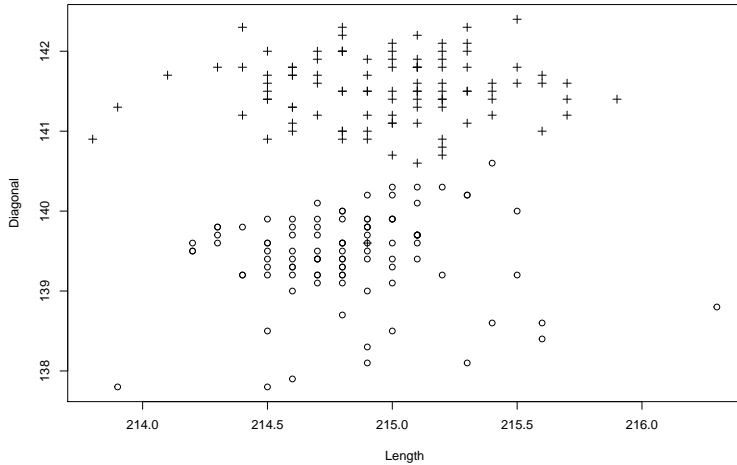
Добавим в таблицу столбец-индикатор `origin`, сообщающий о подлинности банкноты. Код '1' означает, что банкнота подлинная, код '0' — банкнота фальшивая.

```
origin <- c(rep(1, 100), rep(0, 100))  
# Объединяем таблицу и вектор в новую таблицу  
swiss.1 <- data.frame(swiss.0, origin)
```

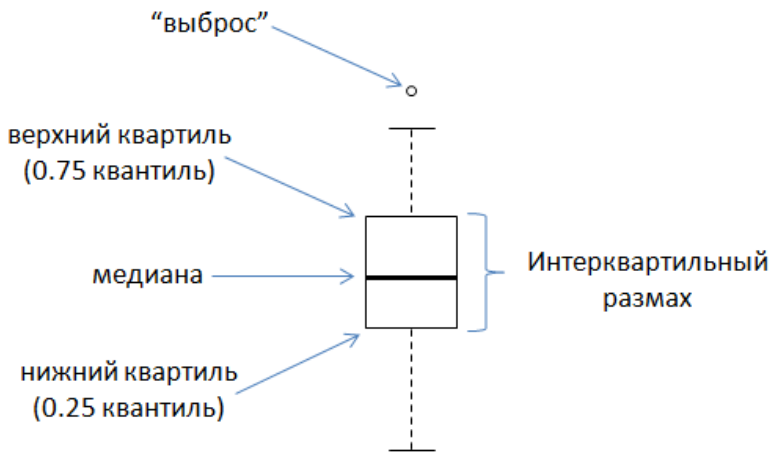
Диаграмма рассеивания: использование маркеров точек

```
# Рисуем правильно смасштабированные оси координат,  
# но данные не выводим (type="n")  
plot(swiss.1$Length, swiss.1$Diag, type="n",  
xlab="Length", ylab="Diagonal",  
main="Swiss Bank Notes")  
# Условия выбора подлинных и фальшивых банкнот  
l <- swiss.1$origin == 1  
o <- swiss.1$origin == 0  
# Добавляем точки, соответствующие подлинным банкнотам  
points(swiss.1$Length[l], swiss.1$Diag[l], pch=3)  
# Добавляем точки, соответствующие фальшивым банкнотам  
points(swiss.1$Length[o], swiss.1$Diag[o], pch=1)
```

Swiss Bank Notes



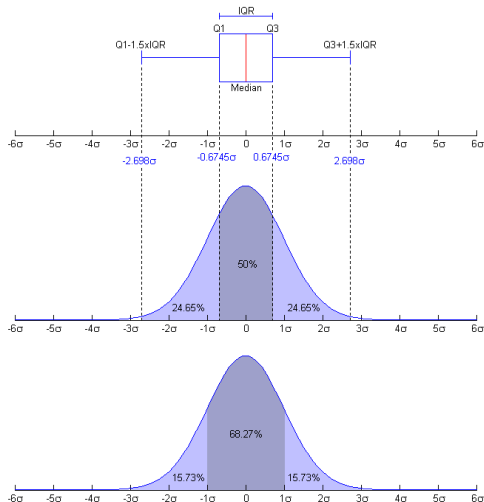
Ящик с усами, он же боксплот (box-whisker plot, boxplot)



$$X_1 = Q_1 - 1.5(Q_3 - Q_1) = Q_1 - 1.5IQR, \quad X_2 = Q_3 + 1.5IQR$$

X_1 — нижняя граница уса, X_2 — верхняя граница уса, Q_1 — первый квартиль, Q_3 — третий квартиль

Квартили нормального распределения



Источник:

https://commons.wikimedia.org/wiki/File:Boxplot_vs_PDF.png

Боксплот: замечания

Длину интервала $1.5 * IQR$ можно изменить при помощи аргумента `range` функции `boxplot()`.

Наблюдения, находящиеся за пределами “усов”, потенциально могут быть выбросами. Следует внимательно относиться к такого рода нестандартным наблюдениям - они вполне могут оказаться “нормальными” для исследуемой совокупности, и поэтому не должны удаляться из анализа без дополнительного расследования причин их появления.

Выбросы (outliers) находятся в пределах от $> 1.5 * IQR$ до $< 3 * IQR$. Они отображаются кружками.

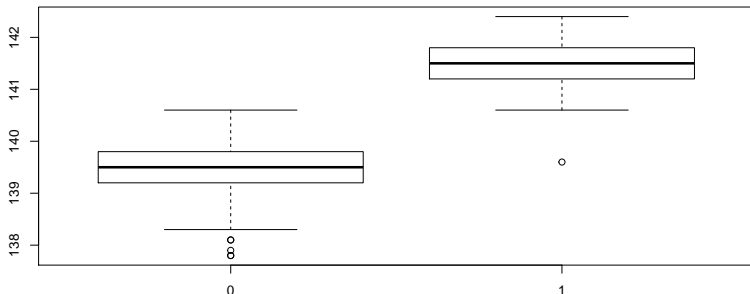
Экстремальные выбросы (extrems) $> 3 * IQR$. Обозначаются звездочками.

Строим боксплот: plot

```
# При использовании plot координата X должна быть фактором  
is.factor(swiss.1$origin)
```

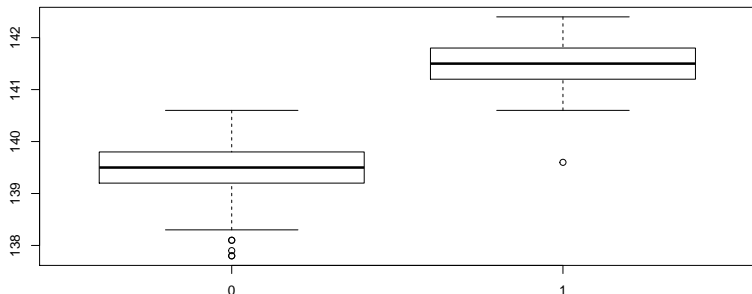
```
## [1] FALSE
```

```
swiss.1$origin <- as.factor(swiss.1$origin)  
plot(swiss.1$origin, swiss.1$Diag)
```



Строим боксплот: `boxplot`

```
boxplot(Diag ~ origin, swiss.1)
```

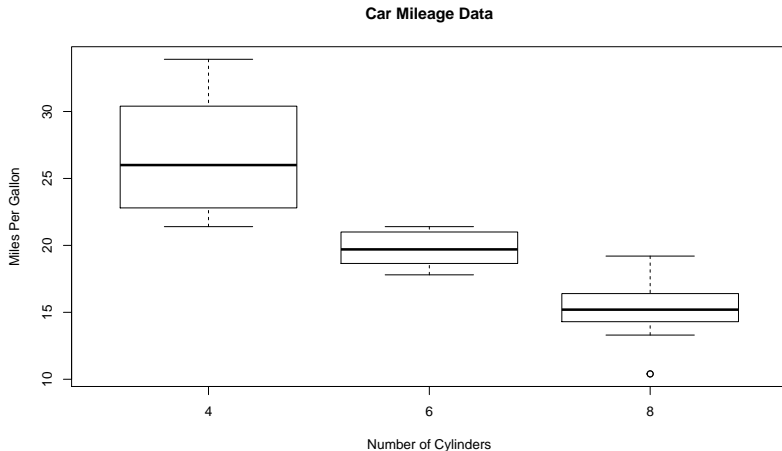


- ▶ $y \sim x_1 + x_2 + \dots$ — формула. Запись означает “ y зависит от x_1, x_2 ” взятых аддитивно.
- ▶ `origin` не нужно делать фактором.

Зачем нужен боксплот?

Ящик с усами — упрощенная версия гистограммы. Он хорош для сравнения нескольких выборок.

```
boxplot(mpg ~ cyl, data=mtcars, main="Car Mileage Data",  
xlab="Number of Cylinders", ylab="Miles Per Gallon")
```



Резюме по разведочному анализу

Разведочный анализ данных — анализ основных свойств данных, нахождение в них общих закономерностей, распределений и аномалий, построение начальных моделей с использованием инструментов визуализации.

Цель разведочного анализа: максимальное “проникновение” в данные.

Вопросы, которые нужно выяснить:

1. Какой тип у данных, каким способом они представлены?
2. Однородны ли данные? В каких единицах измерены показатели?
3. Можно ли предположить нормальное распределение данных?
4. Нужна ли очистка данных: есть ли пропущенные данные, выбросы, опечатки?

Инструменты

- ▶ `read.table()`
- ▶ `summary()`
- ▶ `hist()`
- ▶ `boxplot()`

Если при попытке загрузки данных в R возникает ошибка, возможно это результат неправильного оформления и/или ввода данных.

Пропуски в данных

```
dat <- read.table("data/Albuquerque_Home_Prices_data.txt",  
                  header = T)  
summary(dat)
```

##	PRICE	SQFT	AGE	FEATS
##	Min. : 540	Min. : 837	Min. : -9999	Min. : 0.00
##	1st Qu.: 780	1st Qu.: 1280	1st Qu.: -9999	1st Qu.: 3.00
##	Median : 960	Median : 1549	Median : 4	Median : 4.00
##	Mean : 1063	Mean : 1654	Mean : -4179	Mean : 3.53
##	3rd Qu.: 1200	3rd Qu.: 1894	3rd Qu.: 15	3rd Qu.: 4.00
##	Max. : 2150	Max. : 3750	Max. : 53	Max. : 8.00
##	NE	CUST	COR	TAX
##	Min. : 0.0000	Min. : 0.0000	Min. : 0.000	Min. : -9999.0
##	1st Qu.: 0.0000	1st Qu.: 0.0000	1st Qu.: 0.000	1st Qu.: 553.0
##	Median : 1.0000	Median : 0.0000	Median : 0.000	Median : 701.0
##	Mean : 0.6667	Mean : 0.2308	Mean : 0.188	Mean : -128.9
##	3rd Qu.: 1.0000	3rd Qu.: 0.0000	3rd Qu.: 0.000	3rd Qu.: 899.0
##	Max. : 1.0000	Max. : 1.0000	Max. : 1.000	Max. : 1765.0

- ▶ В AGE и TAX есть пропуски.
- ▶ -9999 — обозначает пропуск в данных.

Исправляем пропуски: na.strings

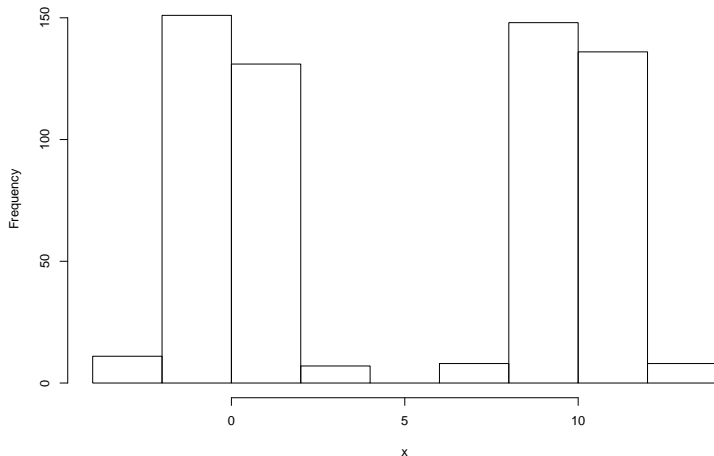
```
dat <- read.table("data/Albuquerque_Home_Prices_data.txt",  
                  header = T, na.strings = "-9999")  
summary(dat)
```

```
##          PRICE          SQFT          AGE          FEATS  
## Min.      : 540    Min.      : 837    Min.      : 1.00    Min.      :0.00  
## 1st Qu.: 780    1st Qu.:1280    1st Qu.: 5.75    1st Qu.:3.00  
## Median : 960    Median :1549    Median :13.00    Median :4.00  
## Mean   :1063    Mean   :1654    Mean   :14.97    Mean   :3.53  
## 3rd Qu.:1200    3rd Qu.:1894    3rd Qu.:19.25    3rd Qu.:4.00  
## Max.    :2150    Max.    :3750    Max.    :53.00    Max.    :8.00  
##                                     NA's      :49  
##          NE          CUST          COR          TAX  
## Min.      :0.0000    Min.      :0.0000    Min.      :0.000    Min.      : 223.0  
## 1st Qu.:0.0000    1st Qu.:0.0000    1st Qu.:0.000    1st Qu.: 600.0  
## Median :1.0000    Median :0.0000    Median :0.000    Median : 731.0  
## Mean   :0.6667    Mean   :0.2308    Mean   :0.188    Mean   : 793.5  
## 3rd Qu.:1.0000    3rd Qu.:0.0000    3rd Qu.:0.000    3rd Qu.: 919.0  
## Max.    :1.0000    Max.    :1.0000    Max.    :1.000    Max.    :1765.0  
##                                     NA's      :10
```

Что делать с пропусками дальше — зависит от задачи.

Бимодальное распределение

Медиана не чувствительна к выбросам. Но есть ситуации, когда отказывает и медиана.



Среднее и медиану легко подсчитать, но что это даст?

```
summary(x)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -2.970  -0.123   5.067   4.931   9.970  13.117
```

Здесь нужна **мода** — число, которое встречается среди наблюдений наиболее часто. В нашем случае таких числа два (0 и 10), поэтому распределение называется *бимодальным* (дву-модальным)

Бимодальное распределение — повод продолжать исследование, чтобы найти причину, которая делит наблюдения на два класса.

Выделение воды на снимке космического радара

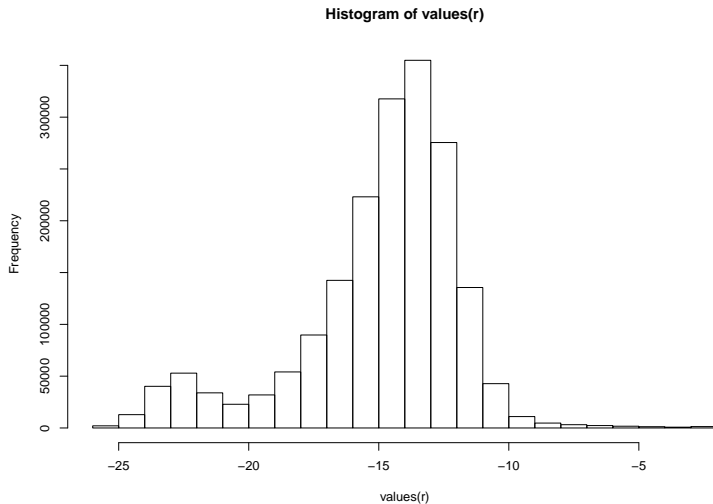


р. Миссисипи, шт. Луизиана — снимок спутника Sentinel-1, поляризация VH

```
library(raster)
```

```
## Loading required package: sp
```

```
r <- raster("data/louisiana_SAR.tif")  
hist(values(r))
```



- ▶ Нижнее распределение с пиком -22.5 дБ — водные объекты.
- ▶ Верхнее распределение с пиком -13.5 дБ — суша.

Что такое “типичный город”?

Анализируем численность населения городов России по данным переписи 1959 года. Названия городов даны современные. Население задано в тысячах человек.

Нужно узнать численность населения, проживающего в типичном городе и попутно определить, что же такое “типичный город”.

Данные находятся в файле `town_1959.csv`

Задаем рабочую папку и начинаем анализ с импорта данных в R.

```
town.1959 <- read.table("data/town_1959.csv", header=T,  
sep=";", encoding = 'UTF-8')
```

Проверим себя

Посмотрим на данные.

Зачем смотреть, если все вроде бы правильно?

Но: если бы мы пропустили любой из параметров `header=T` или `sep=","`, то результат импорта был бы неправильным.

```
head(town.1959, n=5)
```

##	номер	город	население
## 1	1	Москва	5046
## 2	2	Санкт-Петербург	3003
## 3	3	Нижний_Новгород	941
## 4	4	Новосибирск	885
## 5	5	Самара	806

Столбец с номером нам не нужен, уберем его

```
town.1959$`номер` <- NULL
```

Посмотрим описательные статистики

Гипотеза: типичный город задается средним арифметическим по выборке.

```
summary(town.1959[,2])
```

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	0.10	10.70	19.25	52.92	37.98	5046.00

Наблюдение 1. Среднее арифметическое больше 3 квантили!
Уточним.

```
sum(town.1959[,2] < 52.93)/nrow(town.1959) * 100
```

```
## [1] 82.37052
```

Наблюдение 2. Если в качестве населения типичного города России взять среднее арифметическое, то 82% городов России имеет население меньше, чем население типичного города. Что вызывает дискомфорт. Такое наблюдение не воспринимается как типичное.

Выборка содержит выбросы?

Сколько всего наблюдений?

```
nrow(town.1959)
```

```
## [1] 1004
```

Если принять, что Москва и Санкт-Петербург — выбросы, и исключить их из выборки, получим следующее

```
summary(town.1959[-c(1,2),2])
```

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	0.10	10.70	19.15	45.00	37.55	941.00

Посмотрим, на сколько процентов изменилось среднее арифметическое

```
(52.93 - 45.00) / 52.93 * 100
```

```
## [1] 14.98205
```

Какую долю городов мы удалили из выборки?

Наблюдение 3. После отбрасывания 0.2% наблюдений среднее арифметическое уменьшилось на 15%. При этом медиана уменьшилась на 100 человек.

Вывод

Если выборка содержит выбросы, т. е. аномально большие или аномально маленькие наблюдения, то вычисление среднего арифметического становится ненадежным методом определения типичного значения.

Медиана лучше, потому что она устойчива к выбросам.

Некоторые полезные команды

Вычисление среднего

```
mean(town.1959[,2])
```

```
## [1] 52.9252
```

Вычисление медианы

```
median(town.1959[,2])
```

```
## [1] 19.25
```

Вычисление усеченного среднего, $p=0.95$ (trim отсекает с каждой стороны распределения по .025)

```
mean(town.1959[,2], trim = .025)
```

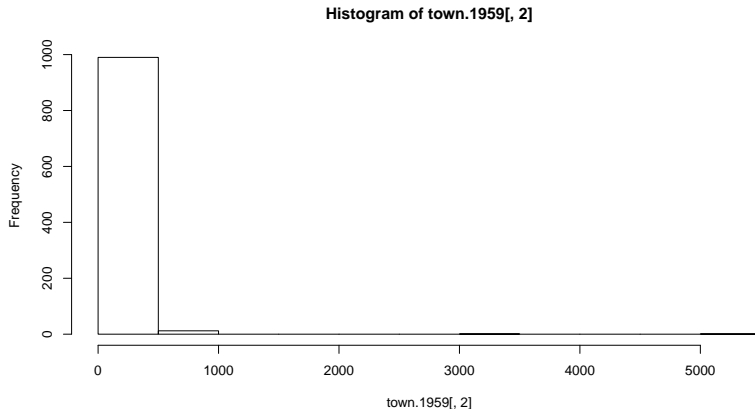
```
## [1] 34.35765
```

Но! Усеченное среднее плохо воспринимается заказчиком.

Сколько выбросов и каково распределение данных?

На гистограмме видны только выбросы

```
hist(town.1959[,2])
```



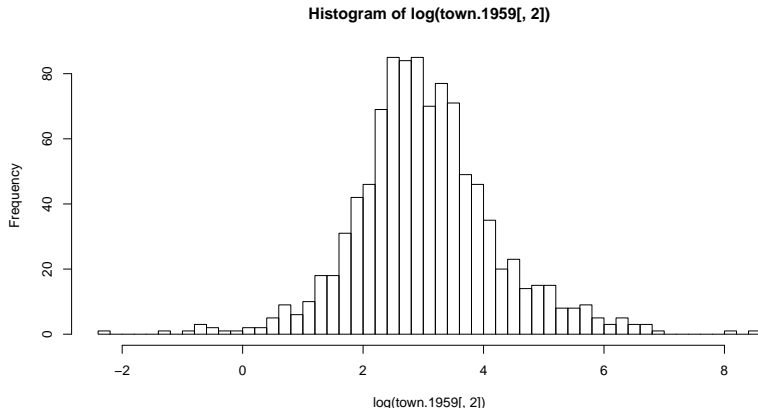
Но сколько их?

Логорифмируем данные

Внешне распределение похоже на лог-нормальное. Поэтому логорифмируем данные.

Теперь на гистограмме видно, что у нас 3 выброса

```
hist(log(town.1959[,2]), breaks=44)
```



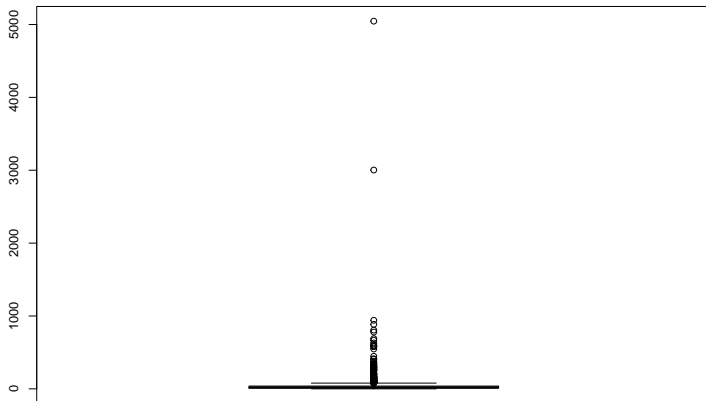
Итак...

когда мы описываем типичное значение, стоит серьезно подумать и...

- ▶ если распределение данных колоколообразное, то лучше использовать среднее арифметическое.
- ▶ если распределение несимметричное, то используем медиану. Допустимо также использовать усеченное среднее.

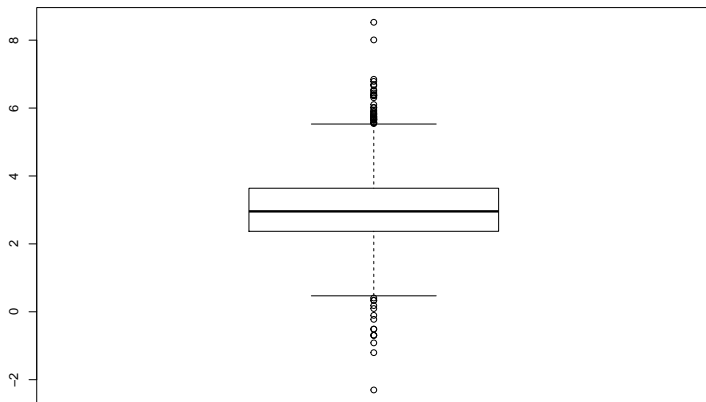
Удаление выбросов: проверка

```
# Выделим численность населения в отдельную переменную  
x <- town.1959[,2]  
# Посмотрим, много ли кандидатов на выбросы  
boxplot(x)
```



Нормализация распределения

```
x <- log(town.1959[,2])  
boxplot(x)
```



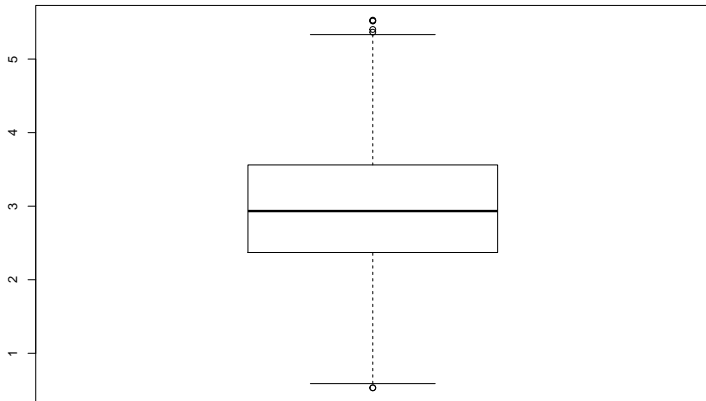
Удалим выбросы

Выбросами считаются объекты, выходящие за границы “усов”

```
# Вычислим 1-ю и 3-ю квантиль  
qnt <- quantile(x, probs=c(.25, .75))  
# Выбросы выходят за границы 1.5*IQR  
H <- 1.5 * IQR(x)  
  
y <- x  
y[x < (qnt[1] - H)] <- NA  
y[x > (qnt[2] + H)] <- NA
```

Выборка после удаления выбросов

```
boxplot(y)
```



Что это были за города?

```
head(town.1959[is.na(y),c(1,2)])
```

##	город	население
## 1	Москва	5046
## 2	Санкт-Петербург	3003
## 3	Нижний_Новгород	941
## 4	Новосибирск	885
## 5	Самара	806
## 6	Екатеринбург	779

```
tail(town.1959[is.na(y),c(1,2)])
```

##	город	население
## 999	Нерюнгри	0.5
## 1000	Усть-Илимск	0.5
## 1001	Ясный	0.5
## 1002	Мегион	0.4
## 1003	Надым	0.3
## 1004	Вуктыл	0.1

Замечания по выбросам

```
# Сколько всего выбросов?  
sum(is.na(y))
```

```
## [1] 51
```

Лечение может быть опаснее болезни: вместе с “выбросами” могут быть удалены ценные данные.

Самостоятельно удалите из выборки экстремальные выбросы (экстремумы)

```
H <- 3 * IQR(x)
```

и проверьте, какие города будут удалены в этом случае.

Удалять выбросы можно без нормализации распределения.

Типичные города в 1959 г.

```
tmd <- median(town.1959[,2])  
cnd <- (town.1959[,2] >= 0.99*tmd) &  
      (town.1959[,2] <= 1.01*tmd)  
town.1959[cnd,]
```

##	город	население
## 497	Беслан	19.4
## 498	Дятьково	19.4
## 499	Липки	19.4
## 500	Сорочинск	19.4
## 501	Шумиха	19.4
## 502	Поворино	19.3
## 503	Нытва	19.2
## 504	Богданович	19.1
## 505	Фрязино	19.1