



Институт технологий управления

Программные средства анализа данных

01.03.05 Статистика

Профиль «Анализ данных в бизнесе и экономике»

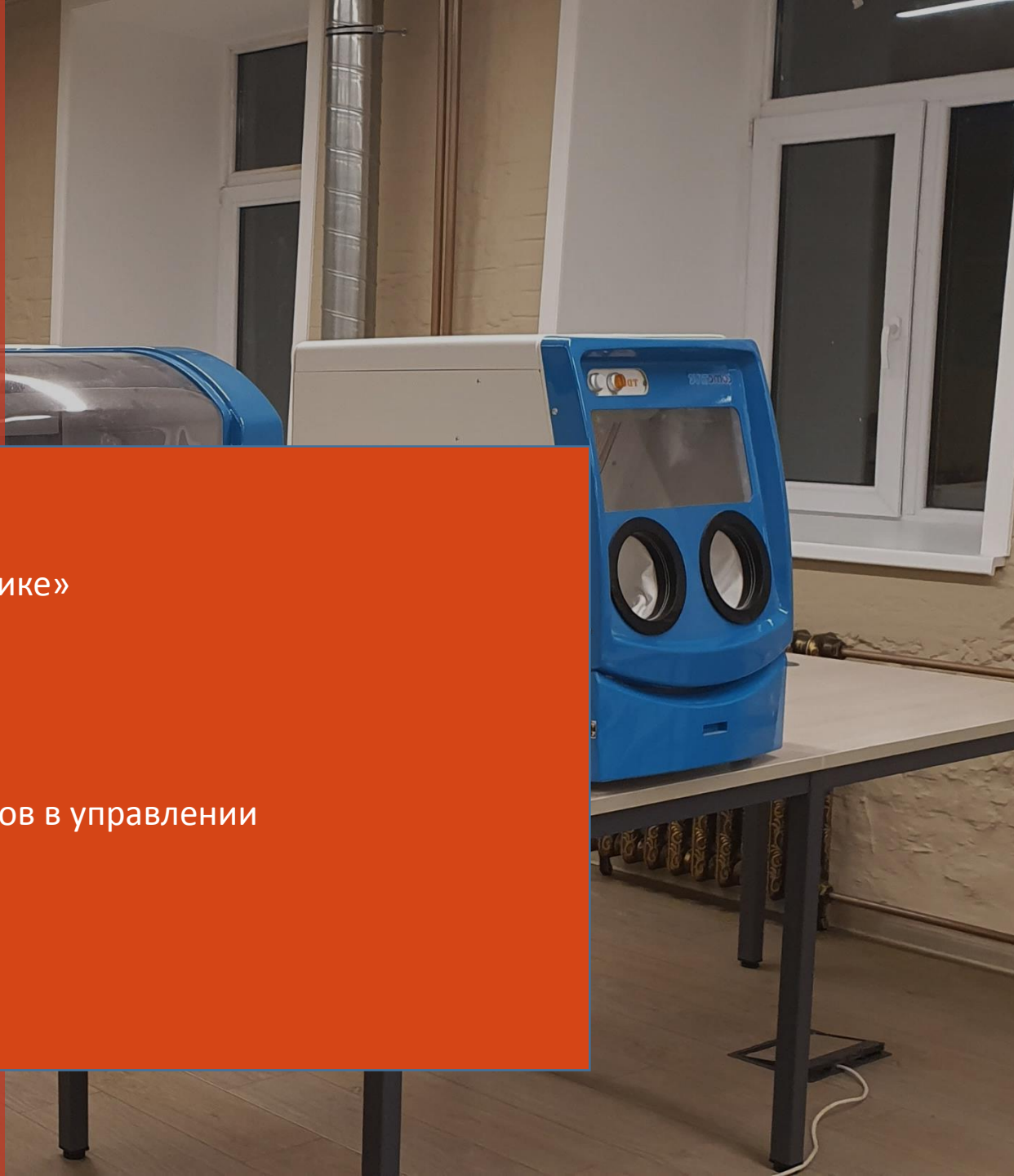
Квалификация «бакалавр»

Бурцева Татьяна Александровна

Профессор кафедры статистики и математических методов в управлении

Burceva\_t@mirea.ru

Москва, 2022



## Тема 2. Разведывательный анализ в R

### План лекции

1. Алгоритмы разведочного анализа в R.
2. Команды статистического анализа.

## 1. Алгоритмы разведочного анализа (Exploratory Data Analysis) в R

- Процедуры разведочного анализа данных предполагают **расчёт основных описательных статистических характеристик** и различные способы визуализации имеющихся данных.
- Основной целью процедур разведочного анализа является получение предварительного описания имеющихся данных, для оценивания **однородности** имеющейся информации, определения наличия **выбросов** и **пропусков** в данных, определения **законов распределения показателей**, **возможных линейных и нелинейных взаимосвязей между рассматриваемыми признаками**.

## Описательные статистики – среднее

$$x_1 \leq \dots \leq x_m$$

### **Выборочное среднее**

$$\text{mean}(X) = \frac{x_1 + \dots + x_m}{m}$$

### **Усечённое среднее**

$$\frac{x_k + \dots + x_{m-k+1}}{m - 2k + 2}$$

+ **весовые схемы**

+ **сглаживание**

### **Медиана**

$$\text{median}(X) = q_{0.5}(X) = \frac{x_{\lfloor m/2 \rfloor} + x_{\lceil m/2 \rceil}}{2}$$

### **Мода (частое значение)**

$$\text{mode}(X) = \arg \max_x | \{i \in \{1, 2, \dots, m\} \mid x = x_i\} |$$

### **mid-range (mid-extreme)**

$$\text{mid-range}(X) = \frac{x_1 + x_m}{2}$$

**тоже одно из решений  
оптимизационных задач...**

### **midhinge**

$$\text{midhinge}(X) = \frac{q_{0.25} + q_{0.75}}{2}$$

## Что такое среднее?

средний, типичный, среднестатистический...

**Естественная формализация – среднее арифметическое**

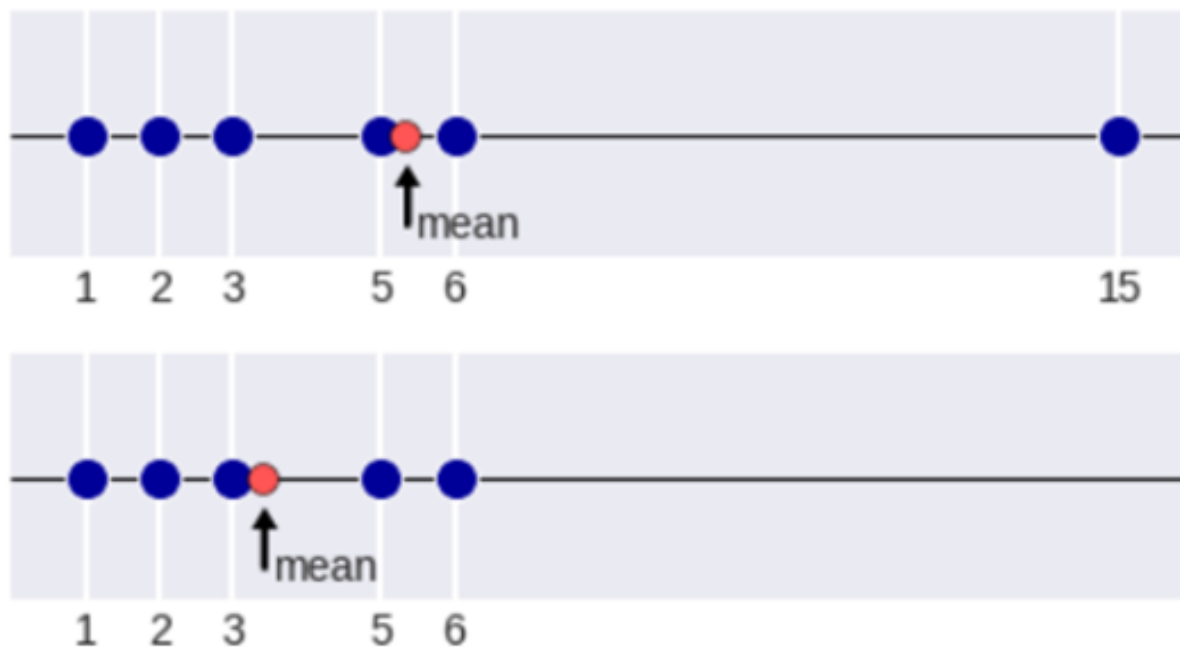
$$\text{mean}(X) = \frac{x_1 + \dots + x_m}{m}$$

**Какие плюсы и минусы?**

## Среднее арифметическое

Большой плюс – среднее можно вычислять в  $\mathbb{R}^n$

### Проблема выбросов



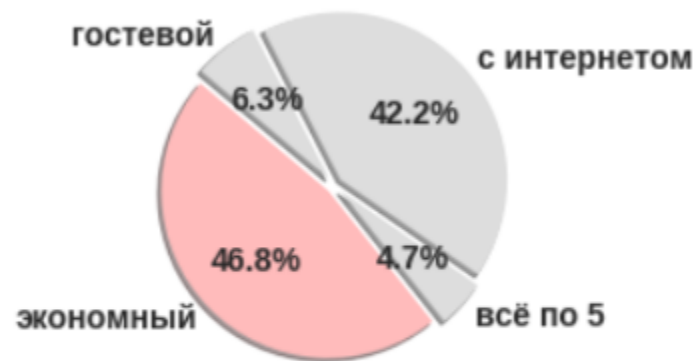
## Среднее как решение оптимизационной задачи

Если суммарные затраты

$$\sum_{i=1}^m |x_i - a| \rightarrow \min$$

то решение – медиана

Что такое среднее для номинальных признаков?



**Мода – самое популярное значение**  
– самое вероятное значение

## Описание

Выборочной средней  $\bar{x}$  называют среднее арифметическое значение признака выборочной совокупности. Если все значения  $x_1, x_2, \dots, x_n$  признака выборки объема  $n$  различны, то

$$\bar{x} = (x_1 + x_2 + \dots + x_n) / n$$

## Описание функции

***mean(x, ...)***

## Параметры

***x*** Вектор, матрица или data.frame.

## Пример

```
> x<-c(3.6,7.8,9.6,5.7,8.9)
> mean(x)
7.12 (значение среднего)
```



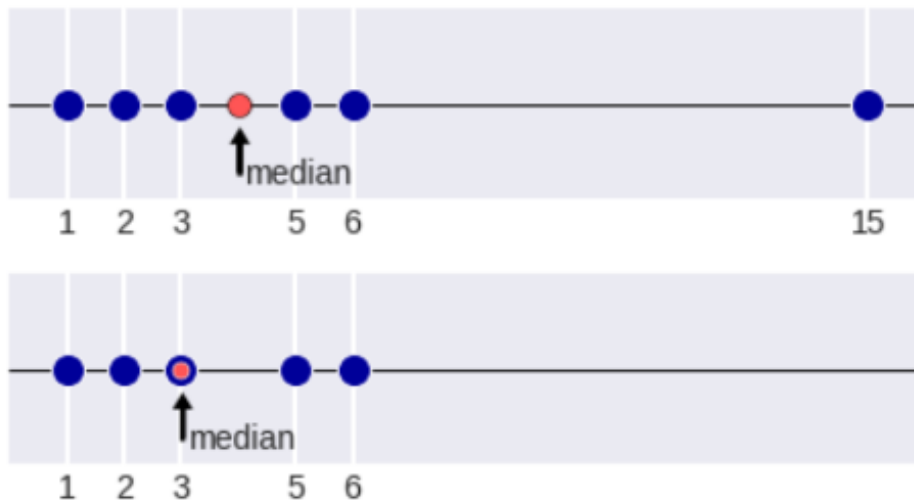
## Что такое среднее?

**Решение проблемы – медиана, для  $x_1 \leq x_2 \leq \dots \leq x_m$  :**

$$\text{median}(X) = \frac{x_{\lfloor m/2 \rfloor} + x_{\lceil m/2 \rceil}}{2}$$

**1) устойчива к выбросам**

**2) является (можно сделать!) точкой выборки**



Важное свойство медианы – сумма абсолютных отклонений значений признака от медианы меньше, чем от любой другой величины:

$\sum_{i=1}^n |x_i - Me| = \min$  Как известно, свойство средней состоит в следующем:

$$\sum_{i=1}^n (x_i - \bar{x}) = 0$$

## Описание

Медианой  $m_e$  называют варианту, которая делит вариационный ряд (упорядоченный по возрастанию) на две части, равные по числу вариант.

Если число вариант нечетно, т.е.  $n = 2r + 1$ , то  $m_e = x_{r+1}$ ;

при четном  $n = 2r$ , то  $m_e = (x_r + x_{r+1})/2$

Модой  $M_0$  называют варианту, которая имеет наибольшую частоту. В R для этого можно использовать построение таблицы вариант

## Описание функции

***median(x, na.rm=FALSE)***

## Параметры

*x* Вектор, матрица или data.frame

*na.rm* Удалить отсутствующие данные?

## Пример

```
> x<-c(3.6,7.8,9.6,5.7,8.9,9.6,5.7,8.9,9.6)
```

```
> median(x, na.rm=FALSE)
```

```
8.9
```

## Описательные статистики – разброс значений

### Среднее линейное (абсолютное) отклонение Mean Absolute Deviation

$$\frac{1}{m} \sum_{i=1}^m |x_i - \text{mid}(X)|$$

$\text{mid}(X)$  – любая формализация среднего

### Среднеквадратическое отклонение Mean Squared Error (MSE) / Mean Squared Deviation (MSD)

$$\sqrt{\frac{1}{m} \sum_{i=1}^m (x_i - \text{mid}(X))^2}$$

## Описательные статистики – абсолютные вариации

**Чаще: стандартное отклонение**

$$\text{std}(X) = \sqrt{\frac{\sum_{i=1}^m (x_i - \text{mean}(X))^2}{m-1}}$$

**Размах**

$$\text{range}(X) = x_m - x_1$$

**Дисперсия (рассеяние, разброс)**

$$\text{var}(X) = \text{std}^2(X)$$

**Median Absolute Deviation (MAD)**

$$\text{MAD}(X) = \text{median}(\{| \text{median}(X) - x_i | \}_{i=1}^m)$$

**Среднее квартильное расстояние**

**Интерквартильный размах**

$$q_{0.75}(X) - q_{0.25}(X)$$

**тоже обобщается на n-мерный случай**

## Описание функции

`var(x, y = NULL, na.rm = FALSE)`

`sd(x, na.rm = FALSE)`

## Параметры

- `x` Вектор, матрица или data.frame
- `y` NULL(по умолчанию) или вектор, матрица или data.frame такой же размерности, что и `x`
- `na.rm` Удалить данные, значения которых отсутствуют

## Пример

```
> x<-c(3.6,7.8,9.6,5.7,8.9)
```

```
> y<-c(2.7,8.9,6.5,8.9,6.5)
```

```
> var(x, y, na.rm = FALSE)
```

```
> sd(x, na.rm = FALSE)
```

```
[1] 2.9 (значение дисперсии)
```

```
[1] 2.459065 (значение СКО)
```

## Описательные статистики – характерные элементы

**Минимум**

$$x_1$$

**Максимум**

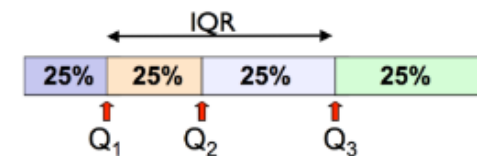
$$x_m$$

**Квантиль** – значение, которое с.в. не превышает с заданной вероятностью

$$X = \{x_1, \dots, x_m\}$$

**Квартили**

$$q_{0.75}(X), q_{0.5}(X), q_{0.25}(X)$$



**Децили**

$$q_{0.1}(X), q_{0.2}(X), \dots, q_{0.8}(X), q_{0.9}(X)$$

**Процентили**

$$q_{1\%}(X), q_{2\%}(X), \dots, q_{98\%}(X), q_{99\%}(X)$$

## Описательные статистики – относительные вариации

абсолютная вариация / среднее

**Коэффициент вариации**  
**Coefficient of variation**

$$\frac{\text{std}(X)}{\text{mean}(X)}$$

$$V_{\sigma} = \frac{\sigma}{\bar{x}} \quad 100$$

**Индекс дисперсии**  
**Index of dispersion**

$$\frac{\text{std}^2(X)}{\text{mean}(X)}$$

**Относительный размах вариации (коэффициент осцилляции)**

$$\frac{\text{range}(X)}{\text{mean}(X)}$$

## Описательные статистики – другое

### Стандартная ошибка среднего

$$\frac{\text{std}(X)}{\sqrt{m}}$$



## Описательные статистики – стандартизованные моменты Standardized moments

$$\frac{\mathbf{E}[(X - \mathbf{E}X)^k]}{\mathbf{D}[X]^{k/2}}$$

$$k = 1$$

$$\mathbf{0}$$

$$k = 2$$

$$\mathbf{1}$$

**Асимметрия – skewness**

$$k = 3 \quad \frac{\mathbf{E}[(X - \mathbf{E}X)^3]}{\mathbf{D}[X]^{3/2}}$$

**Экссесса (островершинность) – kurtosis**

$$k = 4 \quad \frac{\mathbf{E}[(X - \mathbf{E}X)^4]}{\mathbf{D}[X]^2} - 3$$

**инвариантны относительно изменения масштаба**

## Выбор статистических характеристик в зависимости от типа данных

	Количественные	Порядковые	Номинальные
Среднее	+		
Медиана	+	+	
Мода	+	+	+

Источник: Сигел Э.Ф. Практическая бизнес-статистика: пер. с англ.-М.: Вильямс, 2008.

# Основные характеристики, рассчитываемые как «Описательные статистики»

Статистика	Формула, комментарий
Среднее (арифметическое)	$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$
Стандартная ошибка среднего	$SE = S / \sqrt{n}$
Медиана	Значение признака, приходящееся на середину ранжированной (упорядоченной) последовательности
Мода	наиболее часто встречающееся значение признака
Размах	$R = x_{\max} - x_{\min}$
Межквартильный размах	разница между верхним квартильным значением и нижним
Выборочная дисперсия (несмещенная оценка)	$S^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}$
Среднеквадратическое отклонение	квадратный корень из дисперсии
Коэффициент асимметрии (skewness) - Ас	характеризует степень асимметричности, скошенности распределения данных. Правосторонняя асимметрия: Ас>0; левосторонняя асимметрия: Ас<0; симметричность распределения Ас=0.
Коэффициент эксцесса (kurtosis) - Ек	служит мерой крутости (островершинности/плосковершинности) графика вариационного ряда в сравнении с кривой нормального распределения. Ек<0 - плосковершинность; Ек>0 – островершинность.

# Расчёт описательных статистик: мин, max, Me, среднее, квартили (SUMMARY)

1	регионы	2014	2015	2016	2017	2018	2019
2	Белгородская область	764,455	754,042	756,835	757,9	752,6	754,1
3	Брянская область	557,766	547,669	540,643	530,2	523	508,6
4	Владимирская область	669,711	664,413	647,434	640,6	628,2	635,8
5	Воронежская область	1117,774	1092,535	1094,752	1102,1	1110,2	1106,4
6	Ивановская область	455,875	451,493	447,059	456,3	444,9	443,3
7	Калужская область	518,995	508,039	508,877	504,8	503	498,4
8	Костромская область	307,561	299,406	293,153	290,8	282,2	276,8
9	Курская область	529,53	520,324	520,554	519,6	510,8	505,5
10	Липецкая область	599,317	565,151	565,45	565,8	566,1	565,1
11	Московская область	3405,262	3366,883	3376,991	3450,2	3385,7	3437,1
12	Орловская область	340,605	335,905	330,187	321,1	314,5	298,7
13	Рязанская область	513,557	504,808	505,504	511	498,3	494,6
14	Смоленская область	470,036	460,832	443,85	445,9	432,5	411,4
15	Тамбовская область	502,179	499,777	492,131	482,4	466	454,1
16	Тверская область	640,287	630,146	608,474	610	605	593,5
17	Тульская область	749,993	742,625	731,46	719,9	715,1	705,4
18	Ярославская область	633,407	635,935	626,55	621,1	622,2	607,4
19	г. Москва	8613,866	8598,014	8692,036	8730	8838,2	8875,1
20	Республика Карелия	290,066	283,992	283,631	274,8	269,8	266,3
21	Республика Коми	446,783	437,72	421,763	409,9	408,9	401

```

7 #загрузка годовых данных по численности занятого населения РФ с 2014 по 2019 гг по регионам РФ
8 library(readxl)
9 types = c("text", rep("numeric", 6))
10 t1 <- as.data.frame(read_excel("C:/Users//компьютер/Documents/employed.xlsx", 1,
11                               col_types = types))
12 #смотрим структуру переменной
13 str(t1)
14 # найдем описательные статистики для каждого года
15
16 summary.data.frame(t1)

```

> summary.data.frame(t1)

регионы	2014	2015	2016	2017
Length:80	Min. : 33.26	Min. : 33.05	Min. : 31.92	Min. : 33.7
Class :character	1st Qu.: 364.95	1st Qu.: 359.13	1st Qu.: 356.26	1st Qu.: 356.6
Mode :character	Median : 559.96	Median : 556.22	Median : 546.70	Median : 535.4
	Mean : 901.01	Mean : 892.50	Mean : 888.36	Mean : 885.3
	3rd Qu.:1125.99	3rd Qu.:1106.75	3rd Qu.:1103.06	3rd Qu.:1092.7
	Max. :8613.87	Max. :8598.01	Max. :8692.04	Max. :8730.0
2018	2019			
Min. : 33.3	Min. : 33.5			
1st Qu.: 354.4	1st Qu.: 349.5			
Median : 526.6	Median : 515.5			
Mean : 881.5	Mean : 875.2			
3rd Qu.:1091.2	3rd Qu.:1088.0			
Max. :8838.2	Max. :8875.1			

## Что бывает в данных

	дата	пол	образование	сумма	платёжная строка	число просрочек	?????	x_m	неясность
0	12/01/2017	1	высшее	5000.0	0000	0	0	0.00000	
1	13/01/2017	1	высшее	2500.0	0000	1	1	1.00000	дубликаты
2	13/01/2017	1	высшее	2500.0	001000	1	1	1.00000	
3	13/01/2017	0		13675.0	111	3	3	0.00000	
4	25/01/2017	0		NaN	0	0	0	0.00000	
5		1	начальное	NaN	00	0	0	0.00000	
6	02/02/2017	1	среднее	1000.0		0	0	0.00000	
7	01/01/0001 13/01/2017		среднее	0.0		-7	-7	-0.00001	

ошибка

нечисловой признак

пропуски

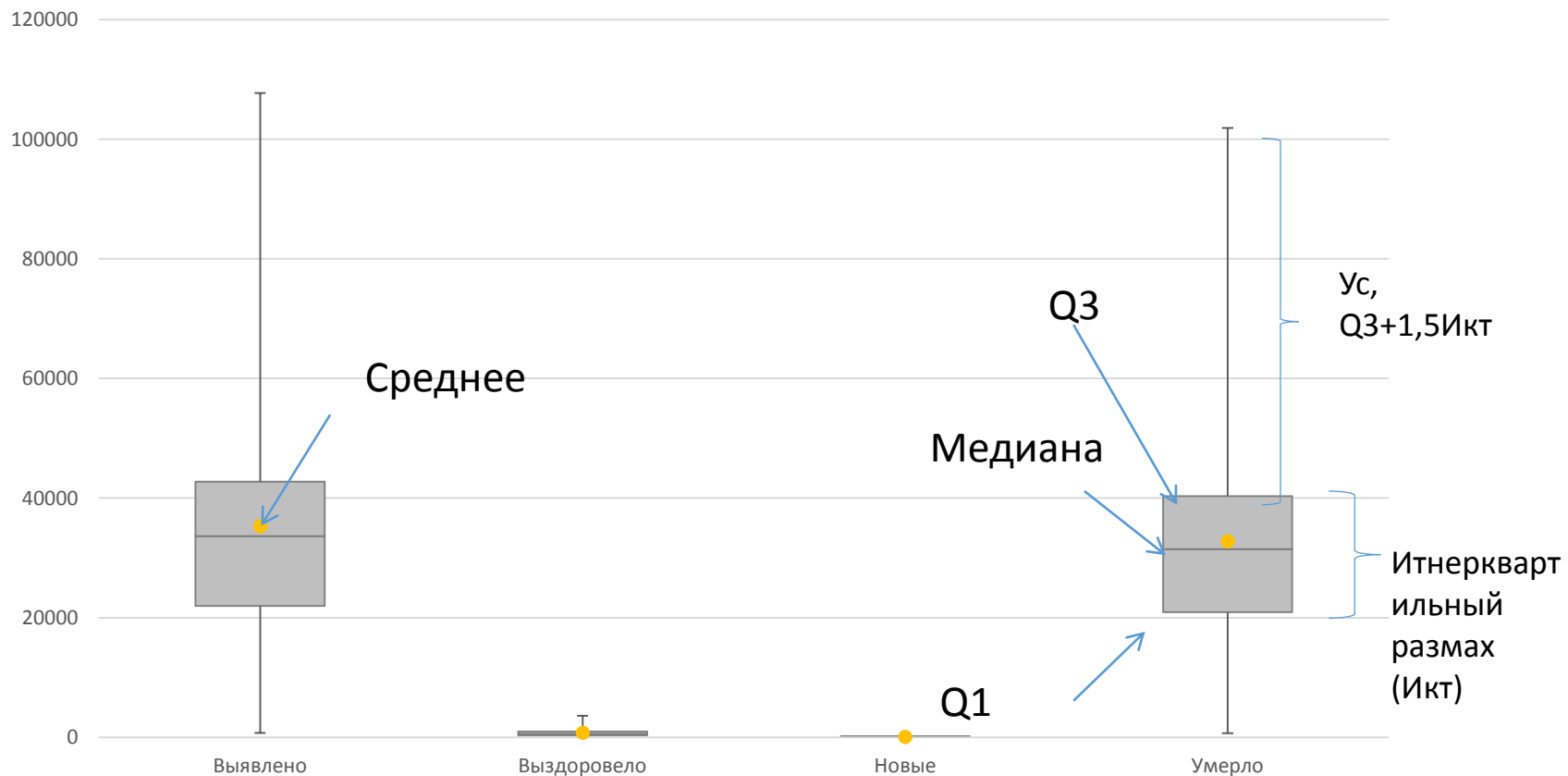
дубликаты

выброс

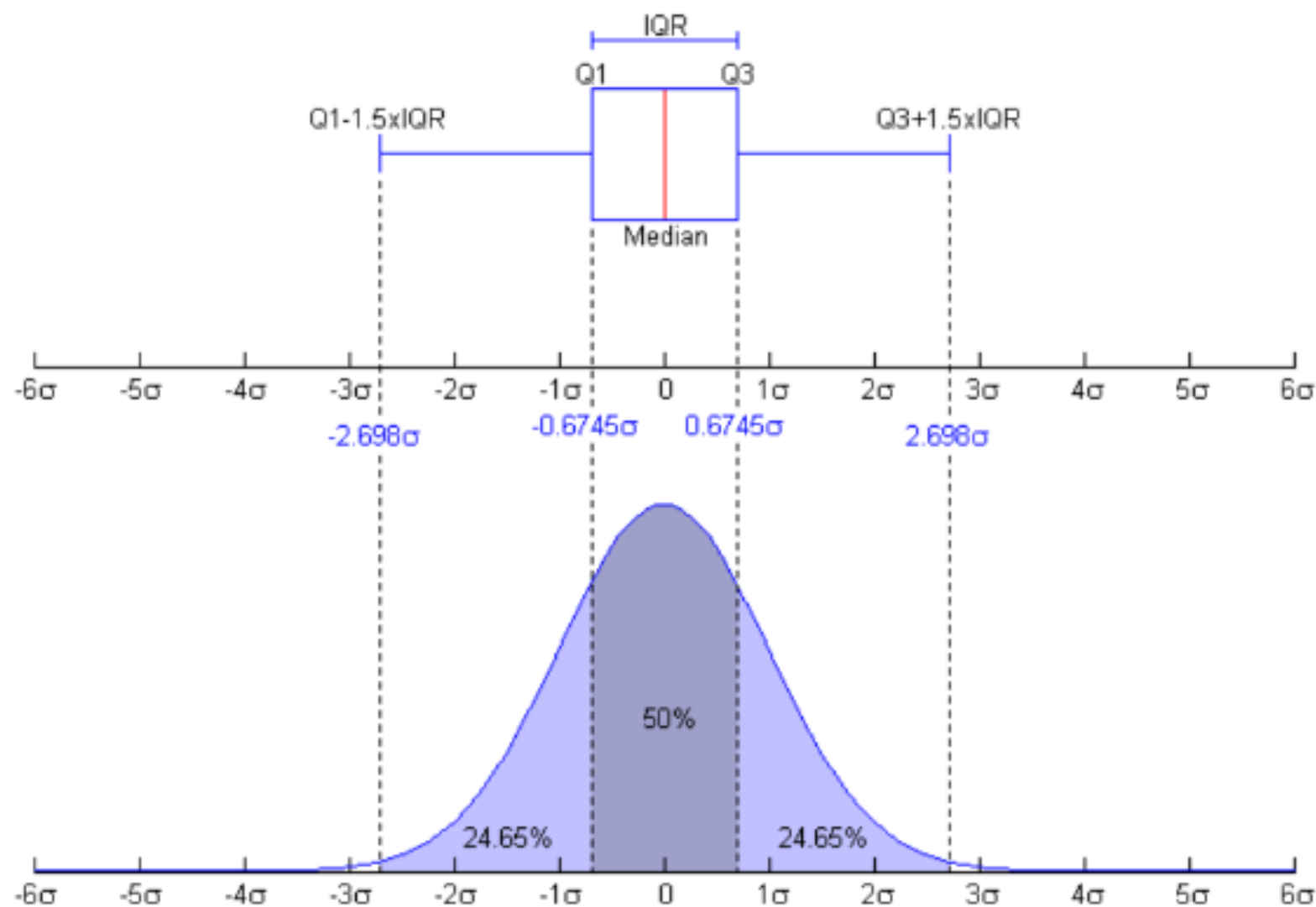
некорректность

# Визуализация результатов анализа (диаграмма размахов, box-plot, гистограмма)

Видное место в разведочном анализе данных занимают графические методы и процедуры. Рассмотрим, предложенную Тьюки *ящичную диаграмму* (boxplot)

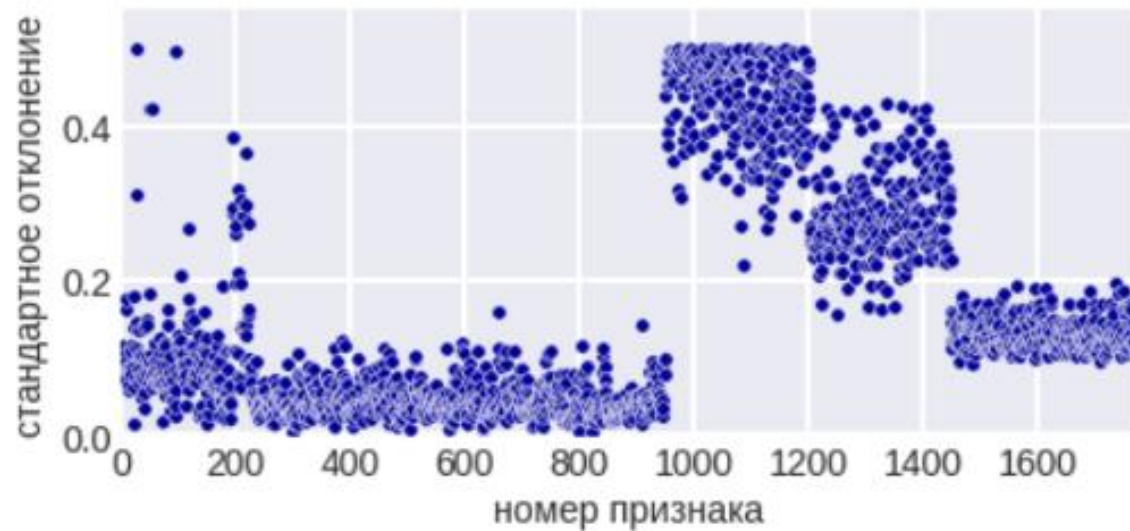
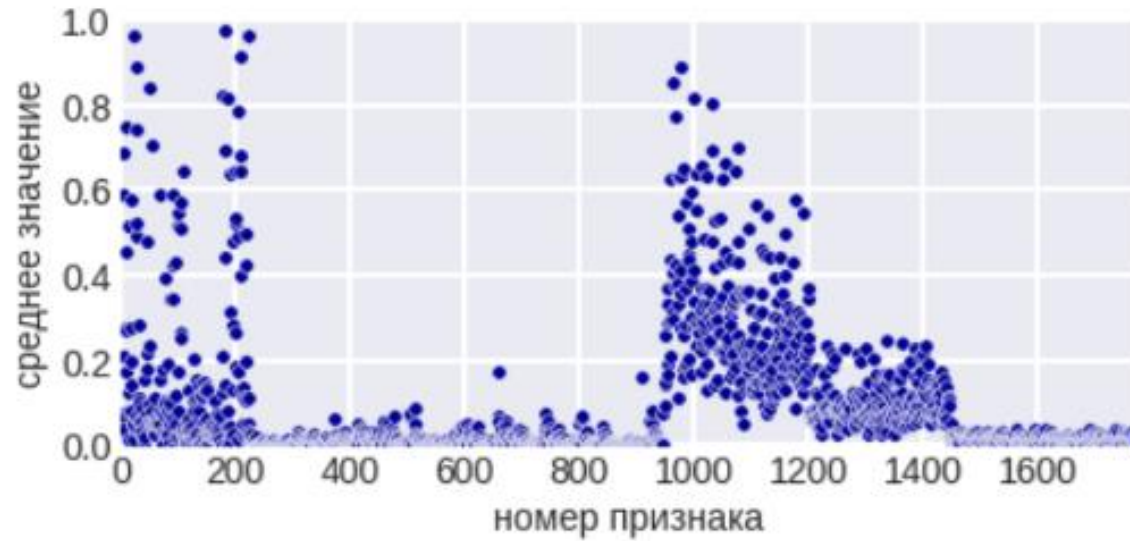


# Описательные статистики – характерные элементы





# Визуализация описательных статистик: задача Biological Response



**Чётко видны группы**



`plot()` – это функция общего назначения, которая строит диаграммы в R

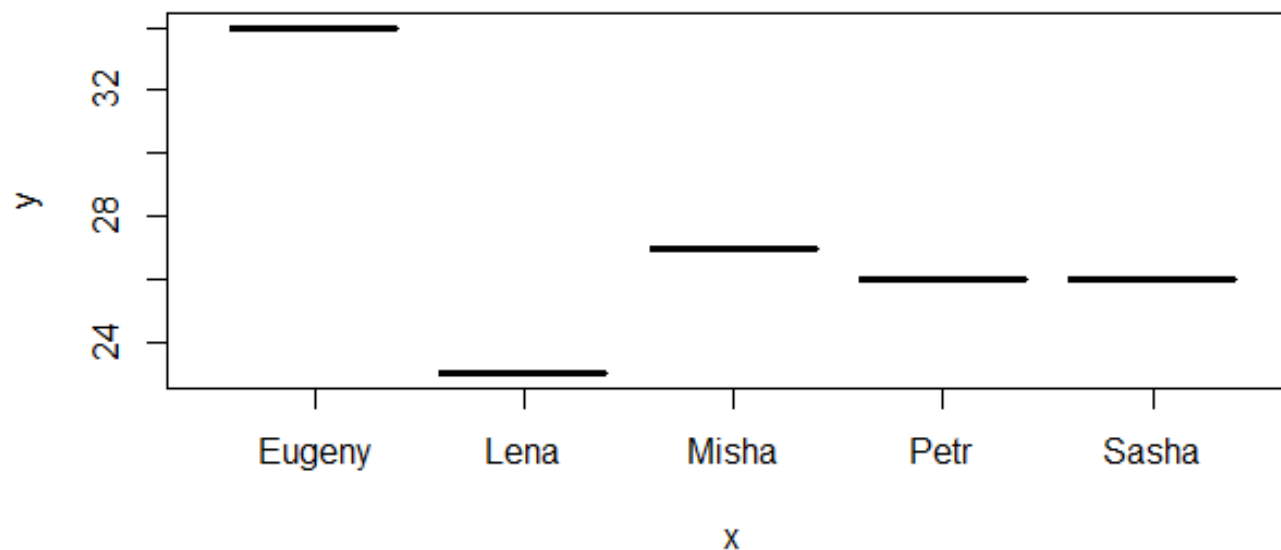
```
name <- c("Petr", "Eugeny", "Lena", "Misha", "Sasha")
```

```
age <- c(26, 34, 23, 27, 26)
```

```
student <- c(F, F, T, T, T)
```

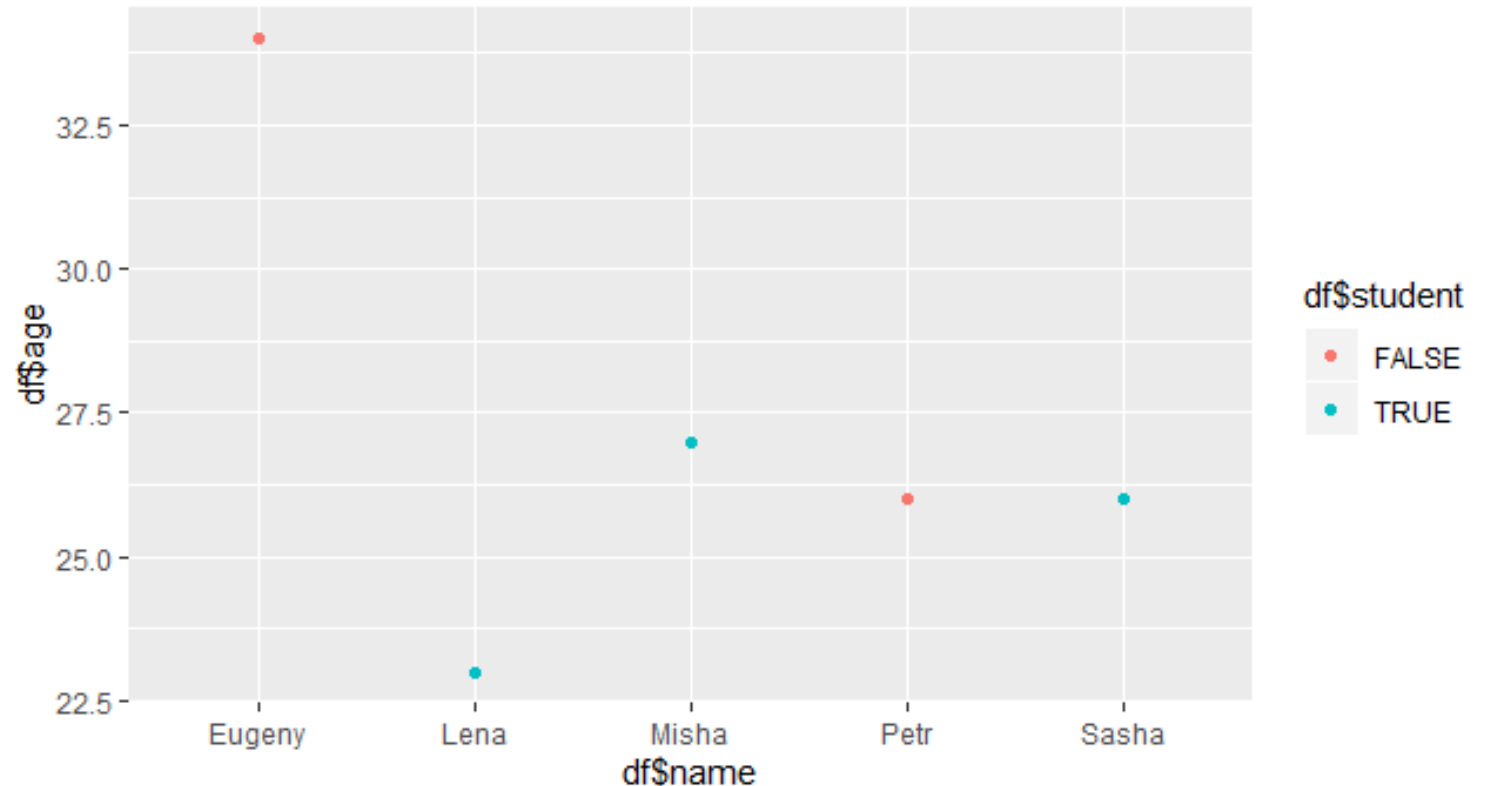
```
df <- data.frame(name, age, student)
```

```
plot(df$name, df$age)
```



ggplot() – это функция, которая строит диаграммы в R и позволяет учитывать условие (фактор)

```
library(ggplot2)  
ggplot(df, aes(x = df$name, y = df$age,  
               col = df$student,  
               size = 3))+ geom_point()
```



# Задание графических параметров в диаграммах R

## Функция colors() выводит на экран список всех доступных цветов

### Параметры для указания типов символов и линий

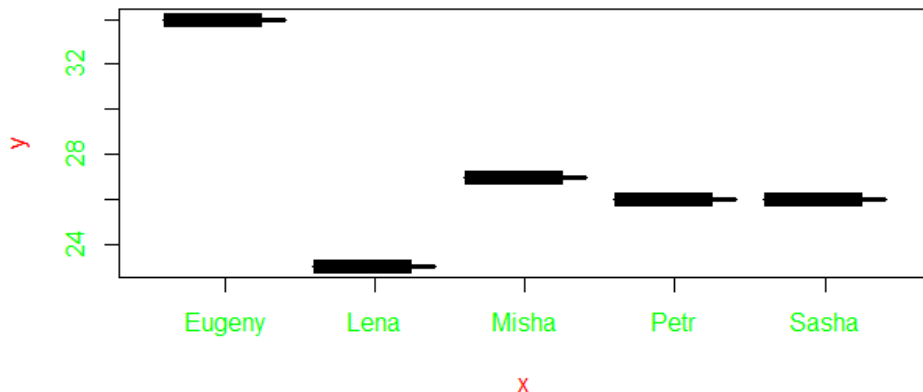
Параметр	Описание
<code>pch</code>	Определяет тип символа
<code>cex</code>	Определяет размер символа. <code>cex</code> – это число, обозначающее, как символы должны быть масштабированы по отношению к размеру по умолчанию. 1 = размер по умолчанию, 1.5 – на 50% крупнее, 0.5 – на 50% мельче и т. д.
<code>lty</code>	Определяет тип линии
<code>lwd</code>	Определяет толщину линии по сравнению с толщиной линии по умолчанию (1). Например, <code>lwd=2</code> делает линию в два раза толще, чем по умолчанию

Типы символов: <code>pch=</code>	
□ 0	◇ 5 ⊕ 10 ■ 15 ● 20 ▽ 25
○ 1 ▽ 6 ✖ 11 ● 16 ○ 21	
△ 2 ▣ 7 ▤ 12 ▲ 17 □ 22	
+ 3 * 8 ✖ 13 ♦ 18 ◇ 23	
× 4 ♦ 9 ▣ 14 ● 19 △ 24	

Типы линий: <code>lty=</code>	
6	-----
5	- - - - -
4	· · · · ·
3	· · · · ·
2	- - - - -
1	_____

### Прошлый пример

```
plot(df$name, df$age, lty=5, lwd=3, pch=15, cex=3,
col.lab="red", col.axis="green")
```



### Параметры для назначения цвета

Параметр	Описание
<code>col</code>	Цвет элементов на графике. Для некоторых функций (таких как <code>lines</code> и <code>pie</code> ) можно указывать вектор из значений, которые используются по очереди. Например, если <code>col=c("red", "blue")</code> и изображены три линии, первая будет красной, вторая – синей и третья – красной
<code>col.axis</code>	Цвет значений осей
<code>col.lab</code>	Цвет подписей осей
<code>col.main</code>	Цвет заголовков
<code>col.sub</code>	Цвет подзаголовков
<code>fg</code>	Цвет графика
<code>bg</code>	Цвет фона

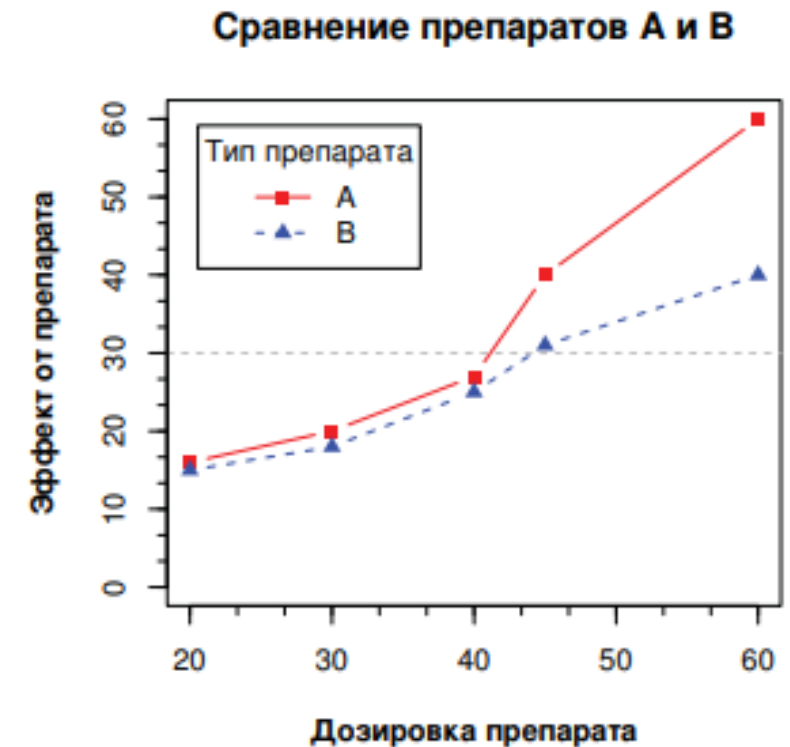
```
dose <- c(20, 30, 40, 45, 60)
drugA <- c(16, 20, 27, 40, 60)
drugB <- c(15, 18, 25, 31, 40)
opar <- par(no.readonly=TRUE)
par(lwd=2, cex=1.5, font.lab=2)
plot(dose, drugA, type="b",
      pch=15, lty=1, col="red", ylim=c(0, 60),
      main="Сравнение препаратов А и В",
      xlab="Дозировка препарата", ylab="Эффект от препарата")
lines(dose, drugB, type="b",
      pch=17, lty=2, col="blue")
abline(h=c(30), lwd=1.5, lty=2, col="gray")
library(Hmisc)
minor.tick(nx=3, ny=3, tick.ratio=0.5)
legend("topleft", inset=.05, title="Тип препарата", c("А", "В"),
      lty=c(1, 2), pch=c(15, 17), col=c("red", "blue"))
par(opar)
```

Увеличиваем ширину  
линии, размер  
символов и подписей

Создаем график

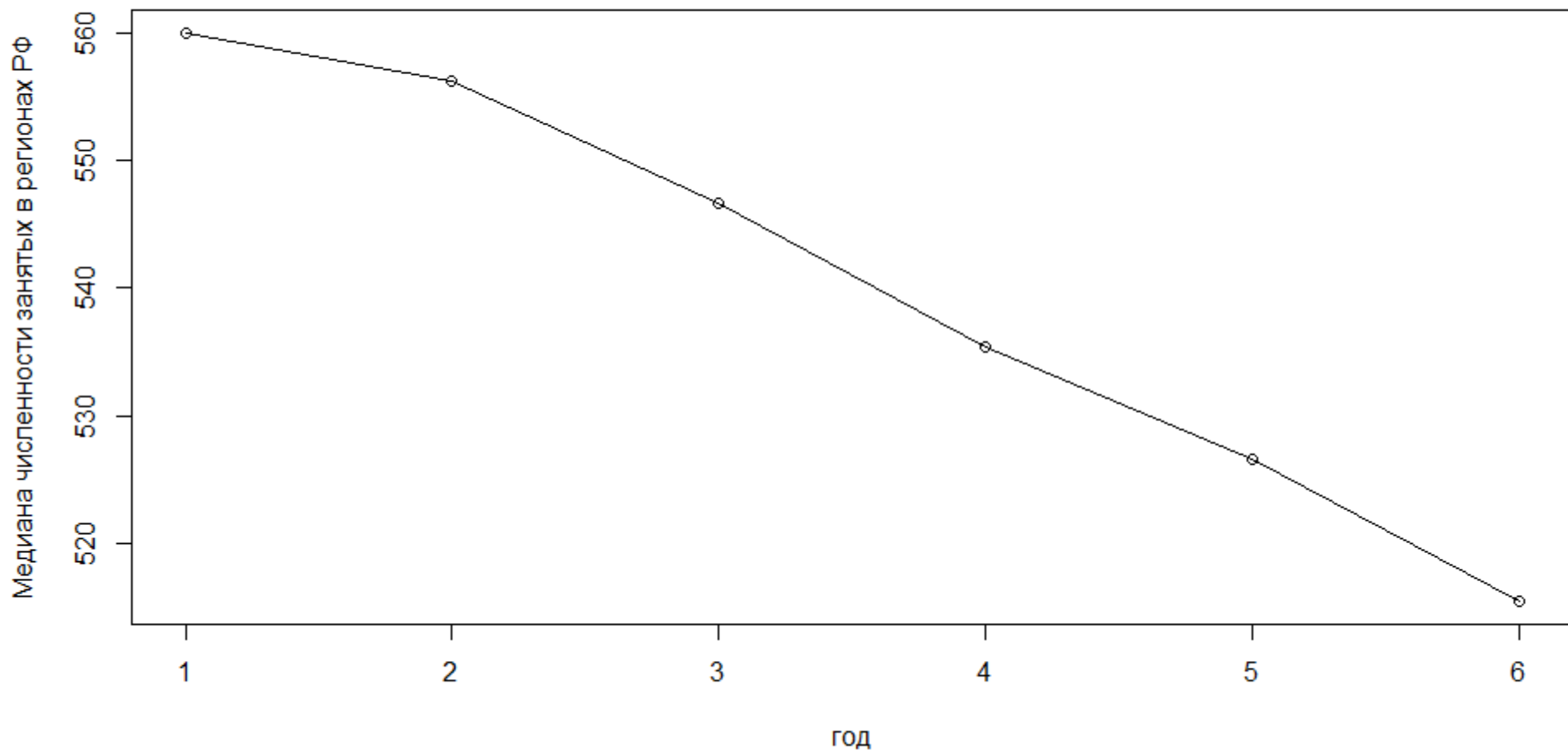
Добавляем  
промежуточные  
деления на осях

Добавляем  
условные  
обозначения



## Визуализация Ме по годам

Динамика Ме занятого населения в регионах РФ с 2014-2019гг



```
summary.data.frame(t1)
```

```
me=c(median(t1$`2014`), median(t1$`2015`), median(t1$`2016`), median(t1$`2017`), median(t1$`2018`), median(t1$`2019`))  
plot(me, xlab="год", ylab="Медиана численности занятых в регионах РФ", main = "Динамика Ме занятого населения в регионах РФ с 2014-2019гг", type = "b")
```

```

#загрузка годовых данных по численности занятого
населения РФ с 2014 по 2019 гг по регионам РФ
library(readxl)
types = c("text", rep("numeric", 2))
t1 <-
as.data.frame(read_excel("C:/Users/компьютер/Documents/e
mployed2.xlsx", 1,
                        col_types = types))
me=c(median(t1$занятые[t$год=='2014']),
median(t1$занятые[t$год=='2015']),
median(t1$занятые[t$год=='2016']),
median(t1$занятые[t$год=='2017']),
median(t1$занятые[t$год=='2018']),
median(t1$занятые[t$год=='2019']))
year=c("2014", "2015", "2016", "2017", "2018", "2019")
dfme=data.frame(me, year)
plot( dfme$year, dfme$me, xlab="год", ylab="Медиана в
регионах РФ", main = "Занятое население, тыс. чел.", type =
"l")

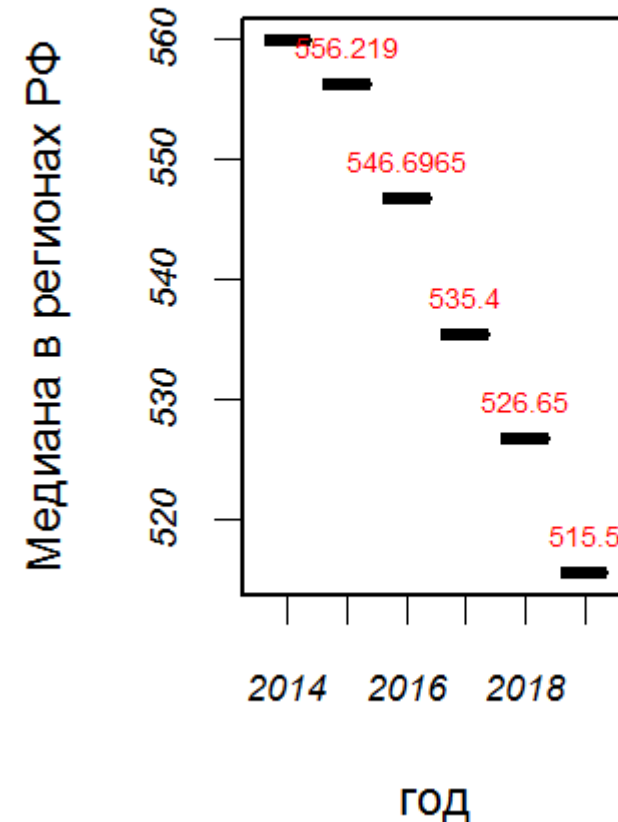
```

```

text(dfme$year, dfme$me,
dfme$me,
cex=0.6, pos=3, col="red")

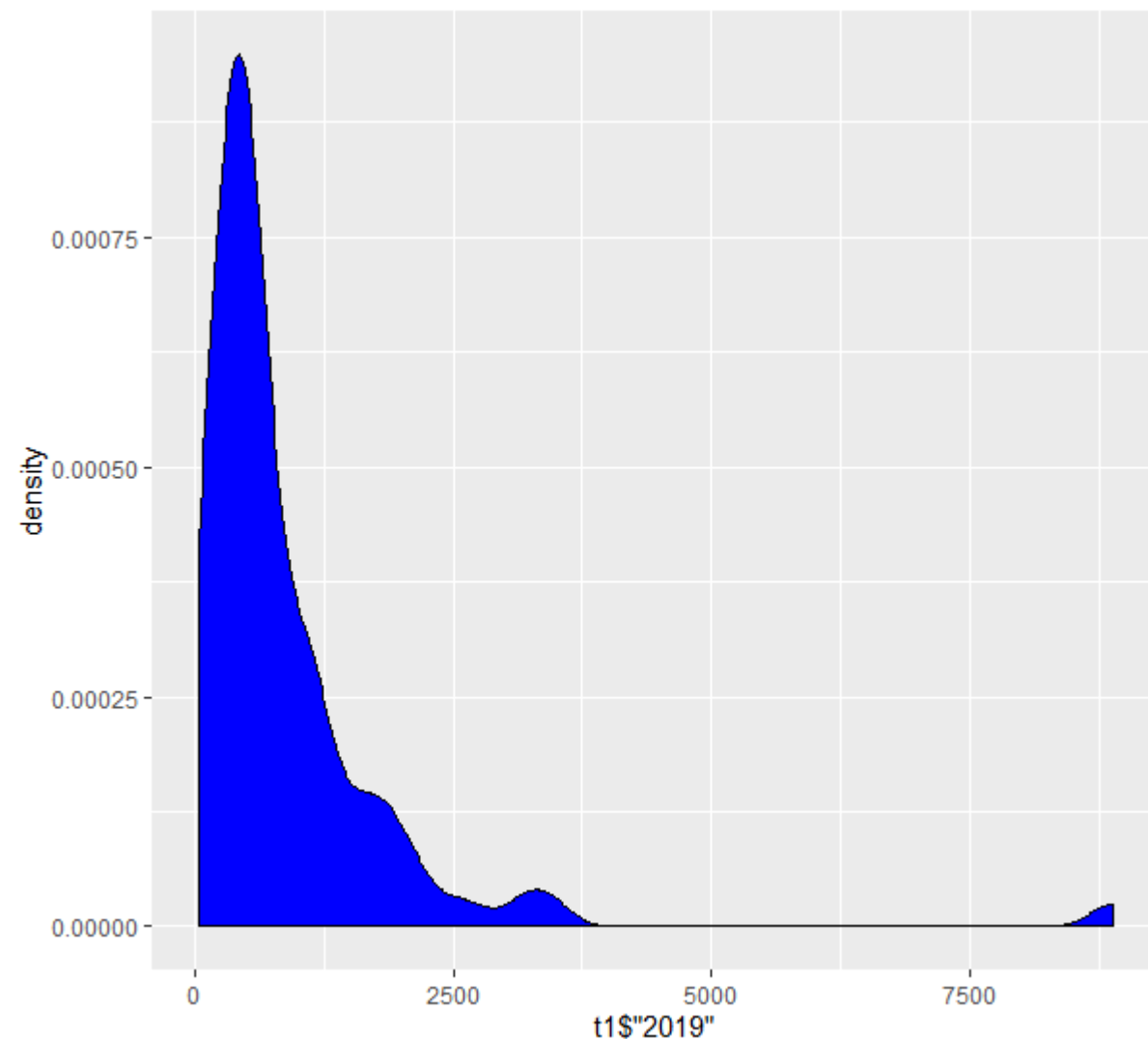
```

## Занятое население, тыс. чел.



# Построение полигона распределения по данным о занятом населении в регионах за 2019г

```
#построим полигон распределения по 2019 г  
library(ggplot2)  
ggplot( t1, aes(x = t1$'2019'))+geom_density(fill = "blue")
```



# Построение полигонов распределения по данным о занятом населении в регионах за 2014-2019гг (меняем структуру данных)

```
library(readxl)
types = c("text", rep("numeric", 2))
t2 <- as.data.frame(read_excel("C:/Users//компьютер/Documents/employed2.xlsx", 1,
                             col_types = types))

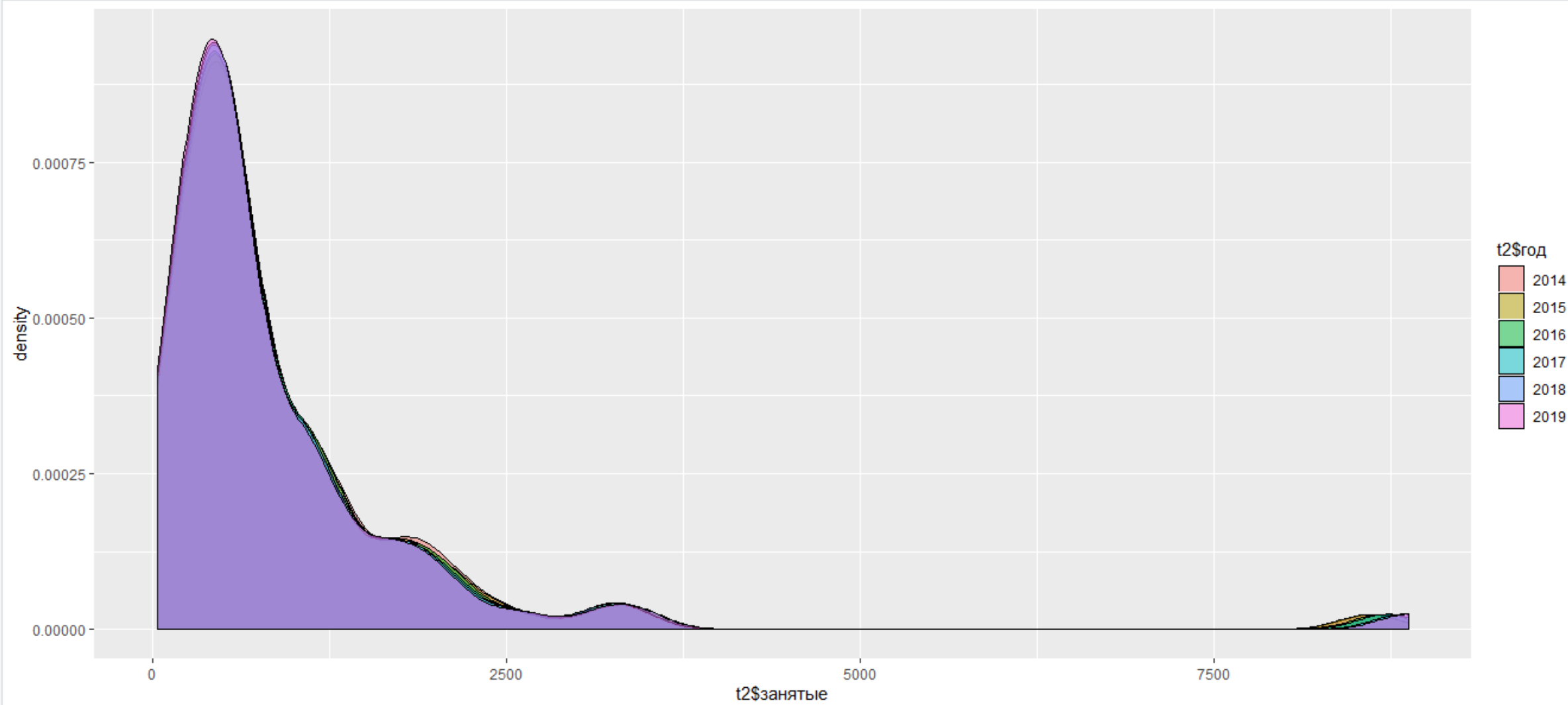
#смотрим структуру переменной
str(t2)
t2$год <- as.factor(t2$год)
#уровни фактора называем в переменной
levels(t2$год) <- c("2014", "2015", "2016", "2017", "2018", "2019")
ggplot(t2, aes(t2$занятые, fill = t2$год)) + geom_density(alpha = 0.5)
```

	A	B	C
1	регионы	занятые	год
2	Белгородская область	764,455	2014
3	Брянская область	557,766	2014
4	Владимирская область	669,711	2014
5	Воронежская область	1117,774	2014
6	Ивановская область	455,875	2014
7	Калужская область	518,995	2014
8	Костромская область	307,561	2014
9	Курская область	529,53	2014
10	Липецкая область	599,317	2014
11	Московская область	3405,262	2014
12	Орловская область	340,605	2014
13	Рязанская область	513,557	2014
14	Смоленская область	470,036	2014
15	Тамбовская область	502,179	2014
16	Тверская область	640,287	2014
17	Тульская область	749,993	2014
18	Ярославская область	633,407	2014
19	г.Москва	8613,866	2014
20	Республика Карелия	290,066	2014
21	Республика Коми	446,783	2014
22	Архангельская область	562,158	2014

Лист1

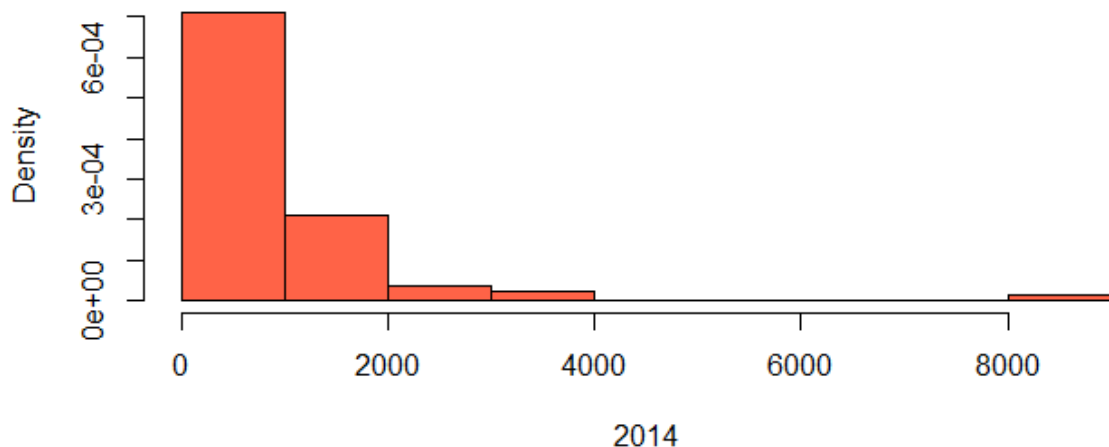


# Результат



# Построение гистограммы распределения по данным о занятом населении в регионах за 2014г

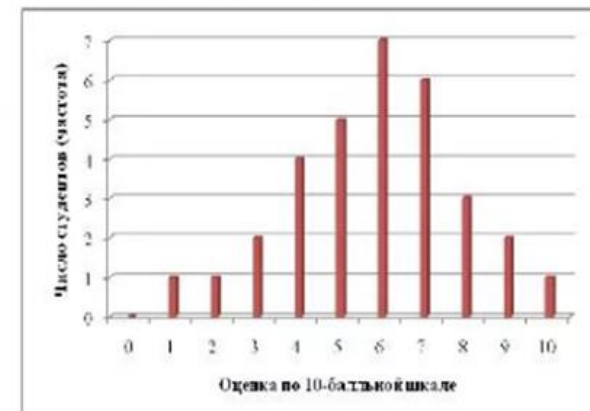
Histogram of занятое население



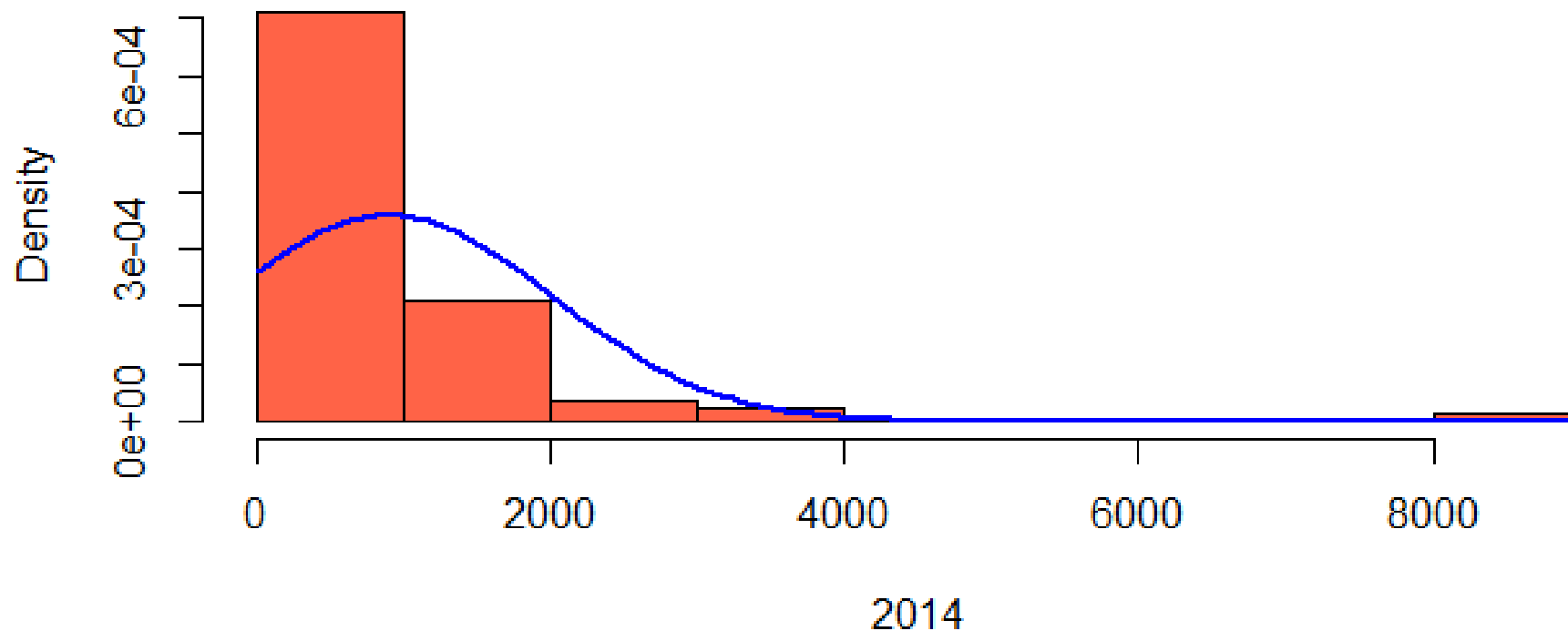
```
hist(t1$`2014`, xlab="2014", main = 'Histogram of занятое население', col = 'tomato', freq = FALSE)
```

**Гистограмма** (histogram) - диаграмма в виде столбцов, по оси абсцисс которой отображаются все возможные значения переменной, по оси ординат – частоты встречаемости  $m_i$  каждого значения или относительные частоты – доли, частоты ( $m_i/n$ ).

Гистограмма была введена в статистическую практику Карлом Пирсоном в 1895 г.



## Histogram of занятое население



```
hist(t[,1], xlab="2014", main = 'Histogram of занятое население', col = 'tomato', freq = FALSE)
#наложим на нее кривую нормального распределения
# na.rm = TRUE - не учитываем пропуски (NA)
# lwd - line width, толщина линии
curve(dnorm(x, mean = mean(t1$`2014`, na.rm = TRUE),
              sd = sd(t1$`2014`, na.rm = TRUE)), col = 'blue', lwd = 2, add = TRUE)
```

- Одним из способов проверки распределения экспериментальных данных на нормальность является расчёт показателей асимметрии и эксцесса и сопоставление их с критическими значениями (метод Е.И. Пустыльника).

$$A_{кр} = 3 \cdot \sqrt{\frac{6 \cdot (n-1)}{(n+1) \cdot (n+3)}}$$

$$E_{кр} = 5 \cdot \sqrt{\frac{24 \cdot n \cdot (n-2) \cdot (n-3)}{(n+1)^2 \cdot (n+3) \cdot (n+5)}}$$

Один из статистических критериев, позволяющих проверить нормальность распределения данных, это **критерий Шапиро-Уилка**. С помощью этого критерия проверяется нулевая гипотеза, которая состоит в том, что *данные распределены нормально*. Данный тест для нормальности с неопределенным средним и дисперсией. Функция для выполнения теста Шапиро-Уилка действительно принимает только  $\leq 5000$  значений. (Shapiro S. S., Wilk M. B. An analysis of variance test for normality. — Biometrika, 1965, 52, №3 — p. 591-611.).

Скрипт

`shapiro.test(t1$`2014`)`

Ответ:

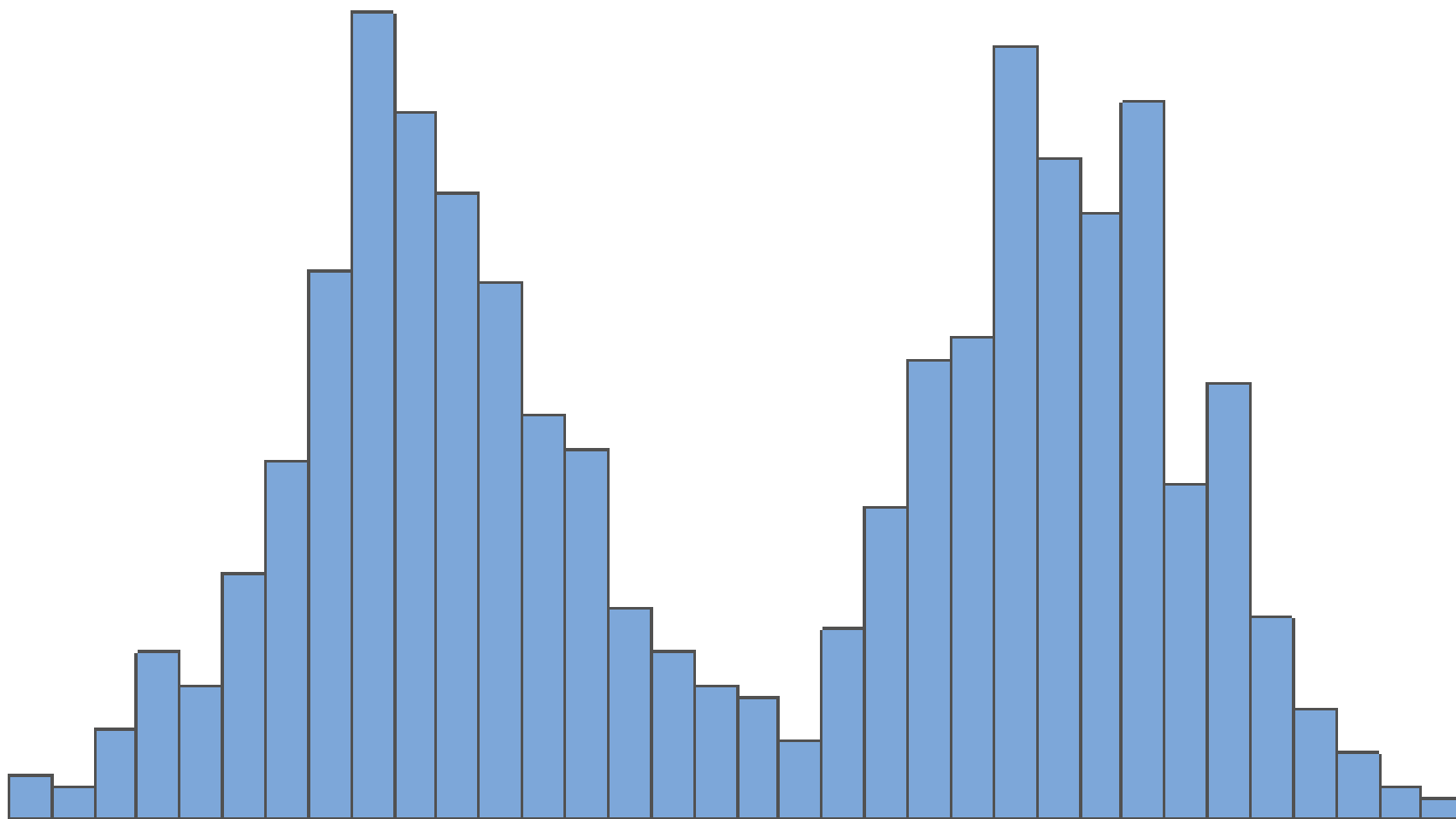
Shapiro-Wilk normality test

data: t1\$`2014`

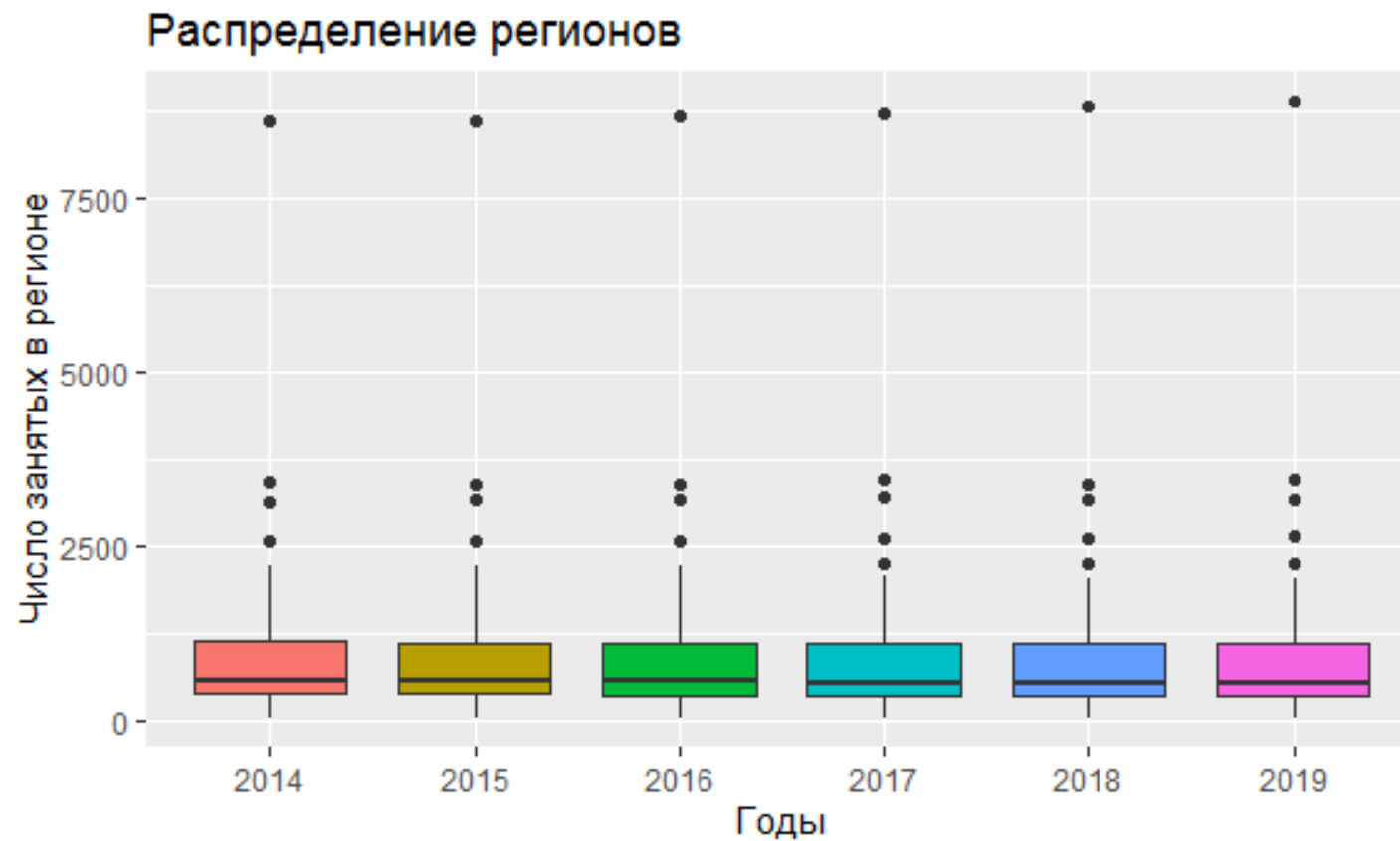
W = 0.57756, p-value = 8.047e-14

Вывод: P-value < 0.05, следовательно, «жизнеспособность» нулевой гипотезы, оценённая на основе имеющихся данных, мала. На имеющихся данных на уровне значимости 5% (0.05) есть основания отвергнуть нулевую гипотезу о том, что данные распределены нормально. Переменная y не распределена нормально.

## Гистограмма с двумя модами



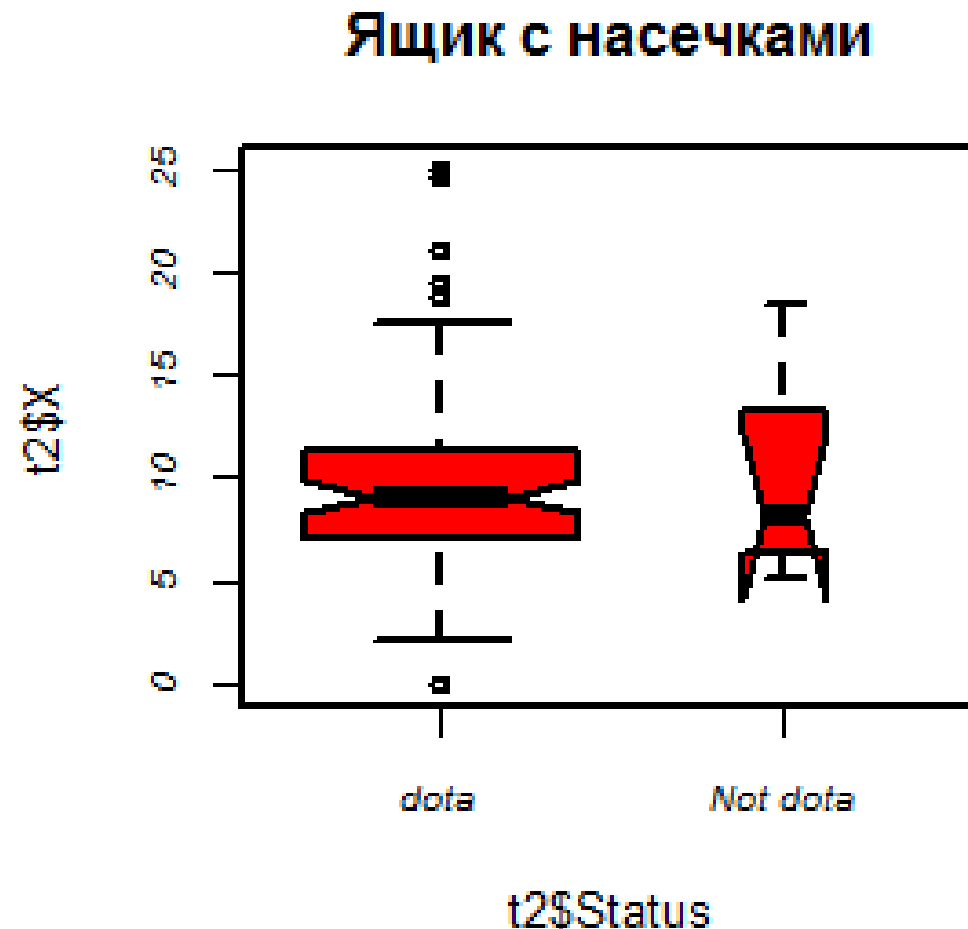
## Построение box-plot



```
#построим boxplot распределения занятых в регионах с учетом года
ggplot(t2, aes(x = t2$год, y = t2$занятые, fill = t2$год))+
  geom_boxplot()+
  xlab("Годы")+
  ylab("число занятых в регионе")+
  ggtitle("Распределение регионов")
```

```
boxplot(t2$X ~ t2$Status, notch=TRUE,  
varwidth=TRUE,  
col="red", main="Ящик с  
насечками")
```

Если «насечки» двух ящиков не перекрываются, высока вероятность того, что медианы соответствующих совокупностей различаются



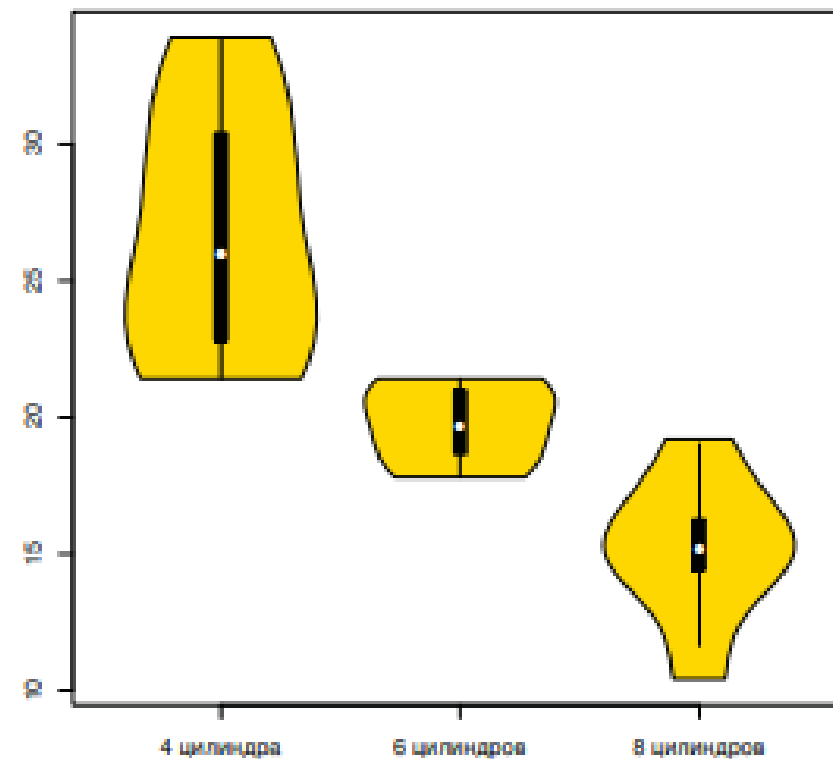


## Скрипичные диаграммы (violin plot)

Скрипичные диаграммы представляют собой симметричные диаграммы ядерной оценки функции плотности, наложенные на диаграммы размахов. Здесь белая точка – медиана, черный прямоугольник – межквартильный размах, а тонкие черные линии – «усы».

Внешний контур фигуры – это диаграмма ядерной оценки функции Плотности.

Такую диаграмму можно создать при помощи функции `vioplot()` из пакета `vioplot`



Скрипичные диаграммы, отражающие расход топлива у автомобилей с разным числом цилиндров

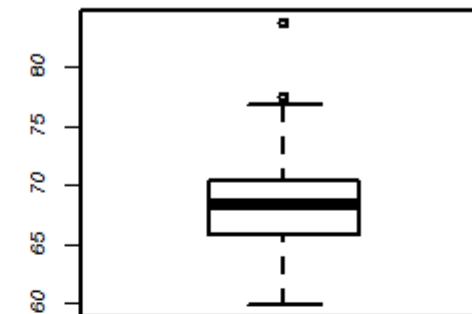
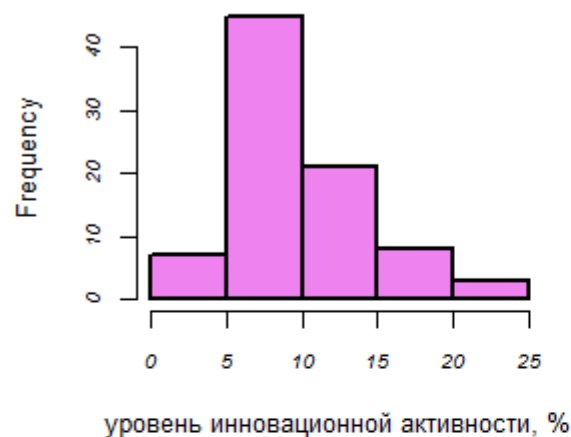
# Построение матрицы диаграмм

```
attach(t2)
opar <- par(no.readonly=TRUE)
par(mfrow=c(2,2))
hist(t2$X, col="violet",
main="Распределение регионов",
xlab="уровень инновационной
активности, %" )
boxplot(t2$Y)
hist(t2$Y, col="orange",
main="Распределение регионов",
xlab="уровень участия в рабочей
силе, %" )
boxplot(t2$X)
```

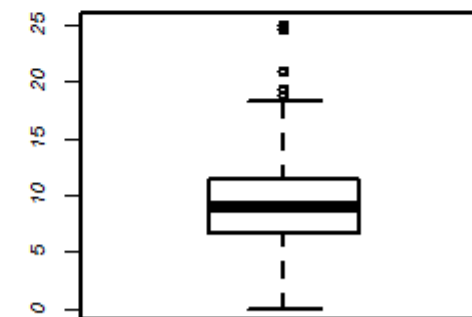
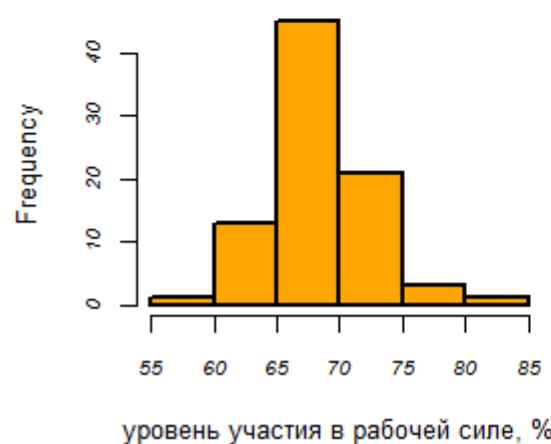


Число строк и столбцов  
в матрице диаграмм

Распределение регионов



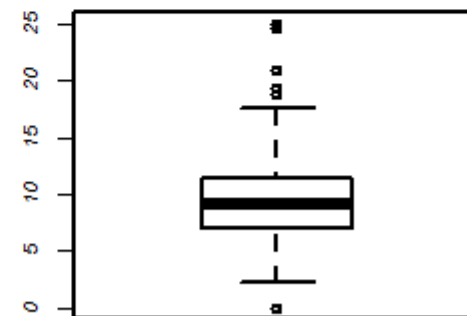
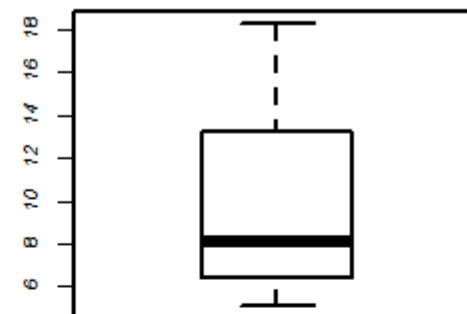
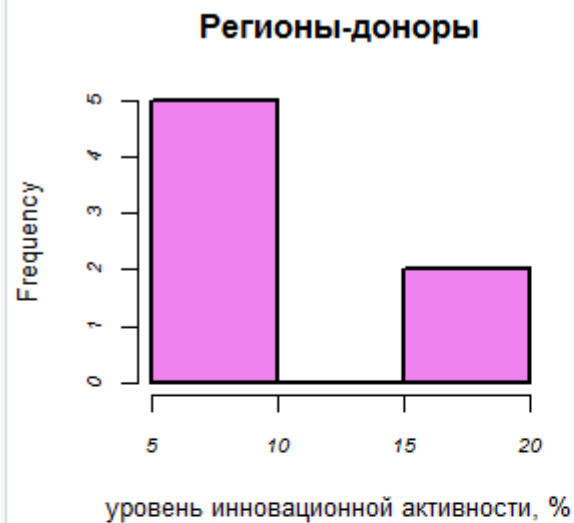
Распределение регионов



```

attach(t2)
opar <- par(no.readonly=TRUE)
par(mfrow=c(2,2))
hist(t2$X[t2$Status=="Not dota"], col="violet",
main="Регионы-доноры", xlab="уровень инновационной
активности, %" )
boxplot(t2$X[t2$Status=="Not dota"])
hist(t2$X[t2$Status=="dota"], col="orange",
main="Дотационные регионы", xlab="уровень
инновационной активности, %" )
boxplot(t2$X[t2$Status=="dota"])

```



```
par (mfrow=c(2,2) )
```

```
hist (mtcars$mpg)
```

1 Простая  
гистограмма

```
hist (mtcars$mpg,
```

```
breaks=12,
```

```
col="red",
```

```
xlab="Расход топлива",
```

```
main="Цветная гистограмма с 12 столбцами")
```

2 Гистограмма с заданным  
числом интервалов и  
раскрашенными столбцами

```
hist (mtcars$mpg,
```

```
freq=FALSE,
```

```
breaks=12,
```

```
col="red",
```

```
xlab="Расход топлива",
```

```
main="Гистограмма, график-щетка и кривая плотности распределения")
```

3 С кривой плотности  
распределения точек

```
rug (jitter (mtcars$mpg) )
```

```
lines (density (mtcars$mpg) , col="blue", lwd=2)
```

```
x <- mtcars$mpg
```

```
h<-hist (x,
```

```
breaks=12,
```

```
col="red",
```

```
xlab="Расход топлива",
```

```
main="Гистограмма с кривой нормального распределения в рамочке")
```

4 С кривой нормального  
распределения и в рамочке

```
xfit<-seq (min (x) , max (x) , length=40)
```

```
yfit<-dnorm (xfit, mean=mean (x) , sd=sd (x) )
```

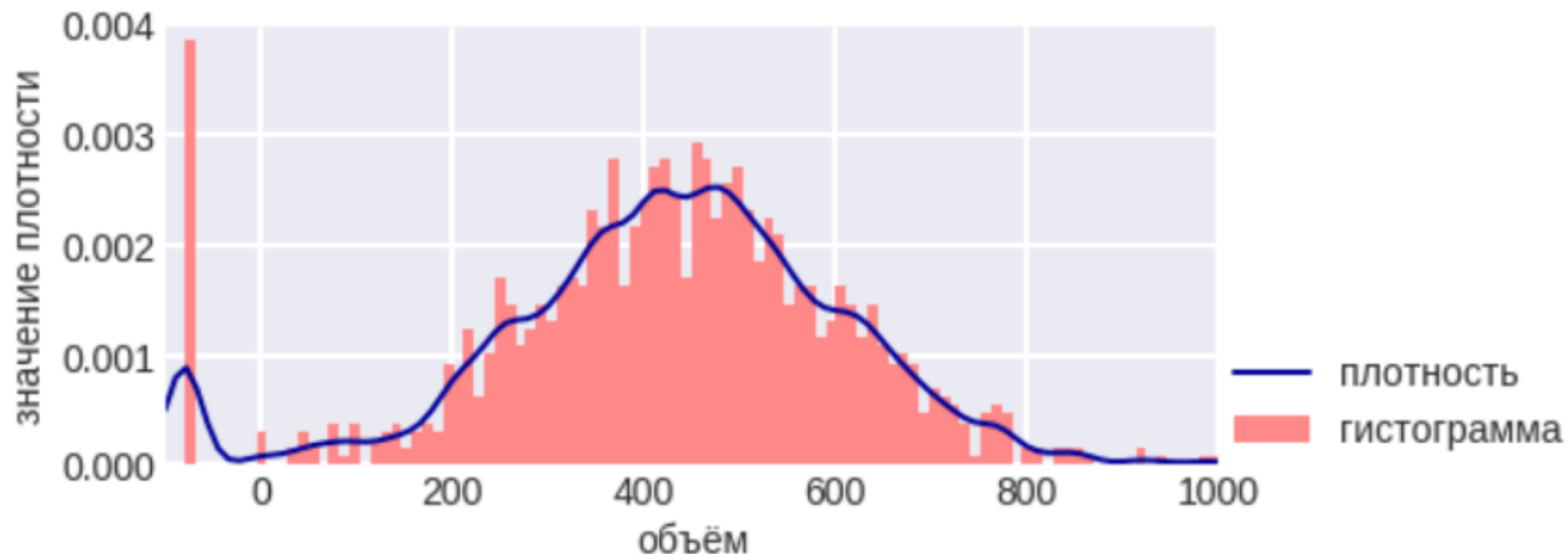
```
yfit <- yfit*diff (h$mids [1:2]) *length (x)
```

```
lines (xfit, yfit, col="blue", lwd=2)
```

```
box ()
```

## Пропуски – как выглядят в данных

- пустые значения
- специальные значения (NA, NaN, null, ...)
- специальный код (-999, mean, число за пределами значения признака)



```
df[name].isnull().sum() # число "нанов"  
df[name].count() # число не "нанов"
```

## Пропуски – что делать

- **оставляем**  
(но не все модели могут работать с пропусками)
- **удаляем описания объектов с пропусками / признаки**  
(радикальная мера, которая редко используется)
- **заменяем на фиксированное значение**  
(например, если признак бинарный, то на 0.5)
- **заменяем на легковычисляемое значение**  
(среднее, медиана, мода)
- **восстановление значения**  
(построение специальной модели для восстановления)
- **экспертная замена**

## **Зашумлённые данные (Noisy Data)**

### **Аналогия с пропусками**

#### **Что делать**

- **оставляем (но будет погрешность при моделировании)**
- **удаляем сильно зашумлённые признаки**
- **удаляем сильно зашумлённые объекты**
- **замена аномальных значений (ex: clipping)**

**могут нести важную информацию!**

**Главный вопрос: «Почему в данных есть это?»**

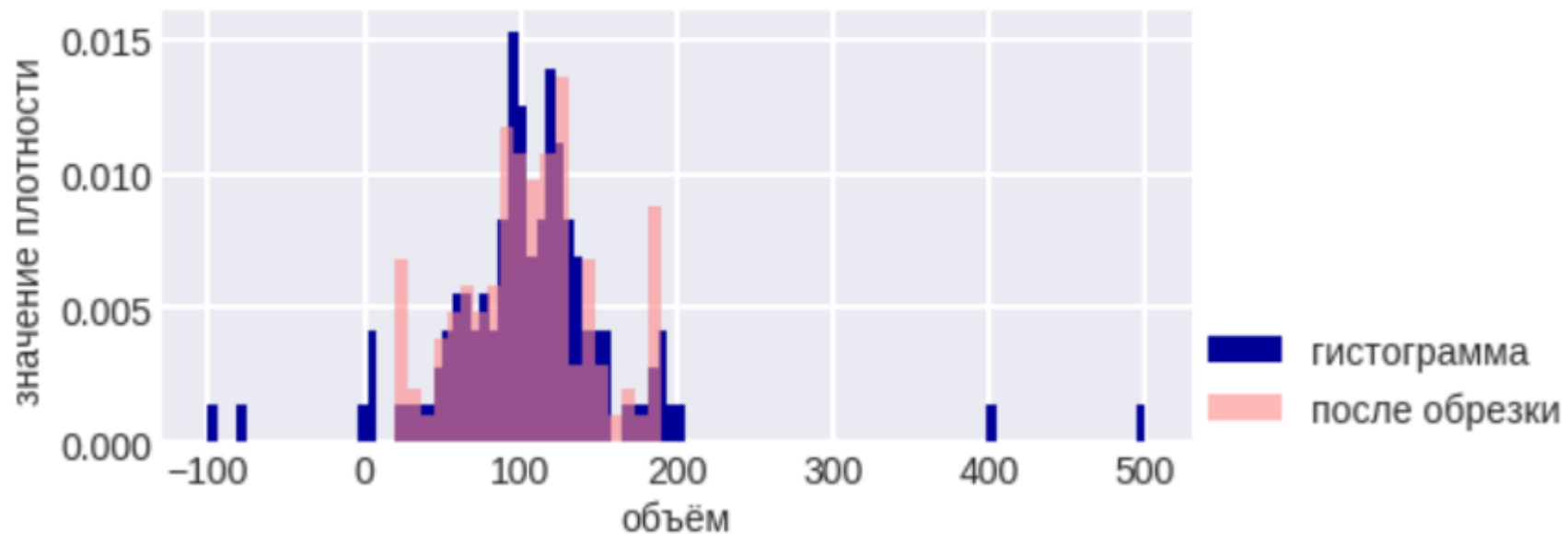
#### **Причины**

- **ошибка сбора данных (ex: погрешность прибора, ввода и т.п.)**
- **ошибка обработки данных**
- **свойство данных (ex: выброс – зарплата CEO)**

**Винсоризация** — это серия трансформаций, направленных на ограничения влияния выбросов.

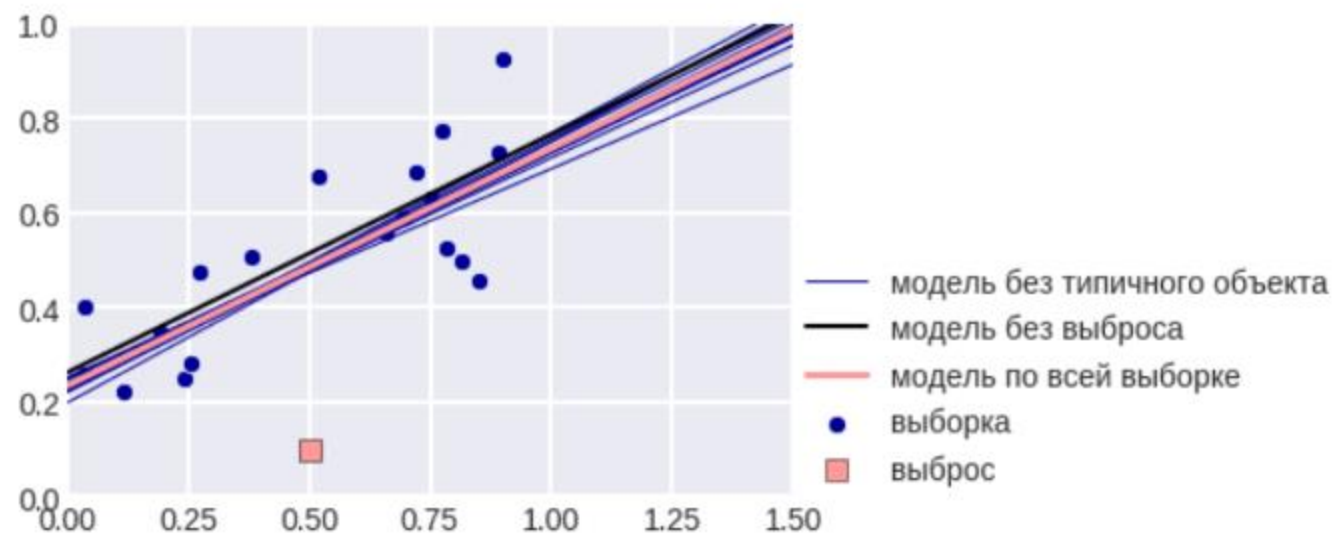
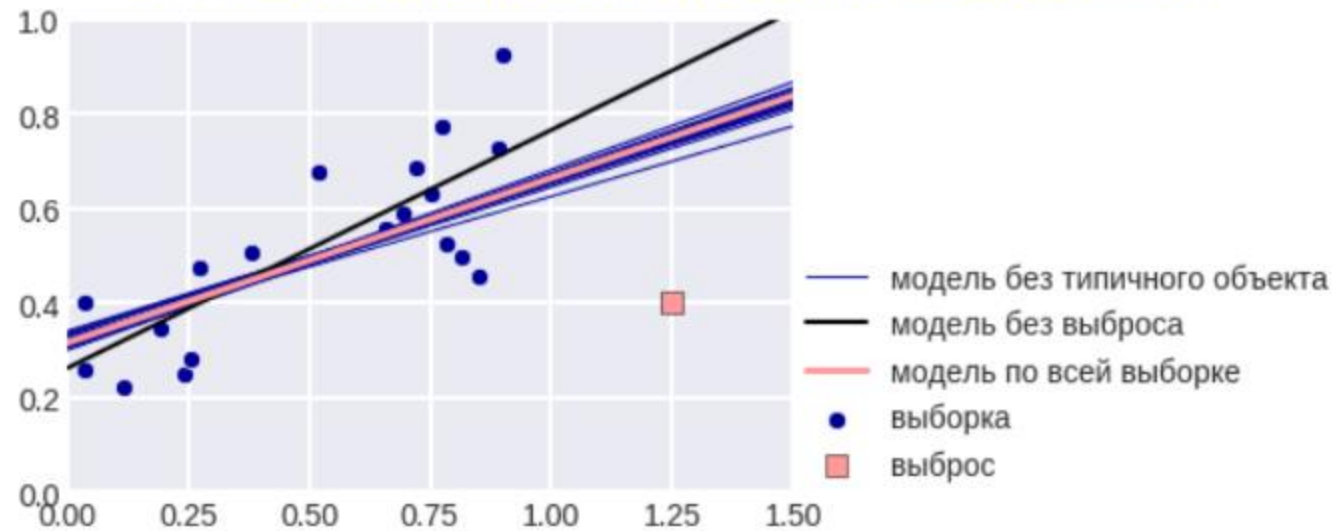
90%-ая **винсоризация** означает, что мы берём значения меньше 5% перцентиля и выше 95% перцентиля и приравниваем их к значениям на 5-м и 95-м перцентилях соответственно.

### Зашумлённые данные – Винсоризация (Winsorizing)





## В чём разница между этими выбросами?



Для большинства алгоритмов машинного обучения необходимо, чтобы все признаки были вещественными и «в одной шкале».

- **Стандартизация**  
(Z-score Normalization / Variance Scaling)

$$\{u_i\}_{i \in I} \rightarrow \left\{ \frac{u_i - \text{mean}\{u_t\}_{t \in I}}{\text{std}\{u_t\}_{t \in I}} \right\}_{i \in I}$$

- **Нормировка на отрезок**  
(Min-Max Normalization)

$$\{u_i\}_{i \in I} \rightarrow \left\{ \frac{u_i - \min\{u_t\}_{t \in I}}{\max\{u_t\}_{t \in I} - \min\{u_t\}_{t \in I}} \right\}_{i \in I}$$

- **Нормировка по максимуму**

$$\{u_i\}_{i \in I} \rightarrow \left\{ \frac{u_i}{\max\{u_t\}_{t \in I}} \right\}_{i \in I}$$

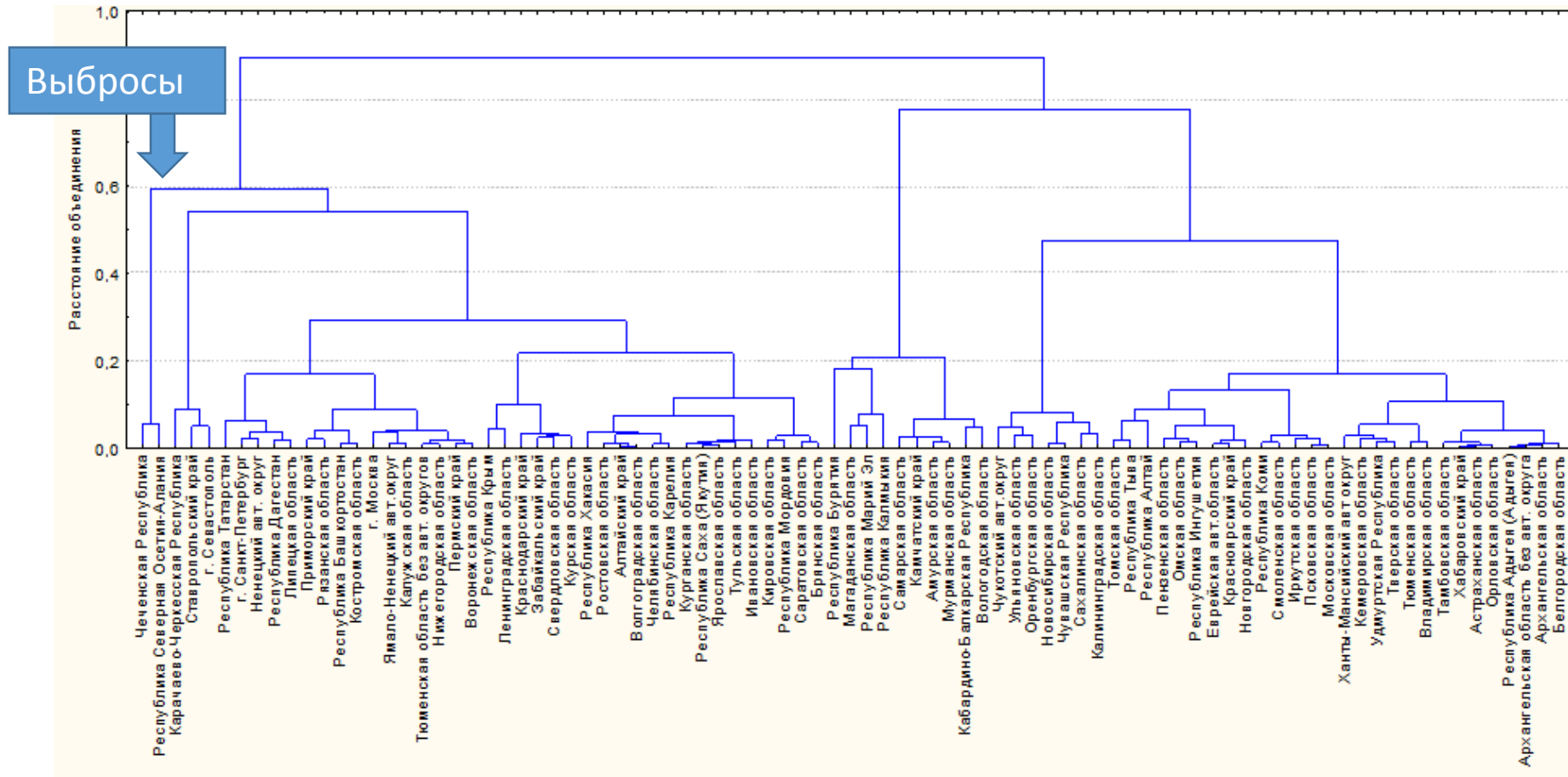
- **Decimal Scaling Normalization**

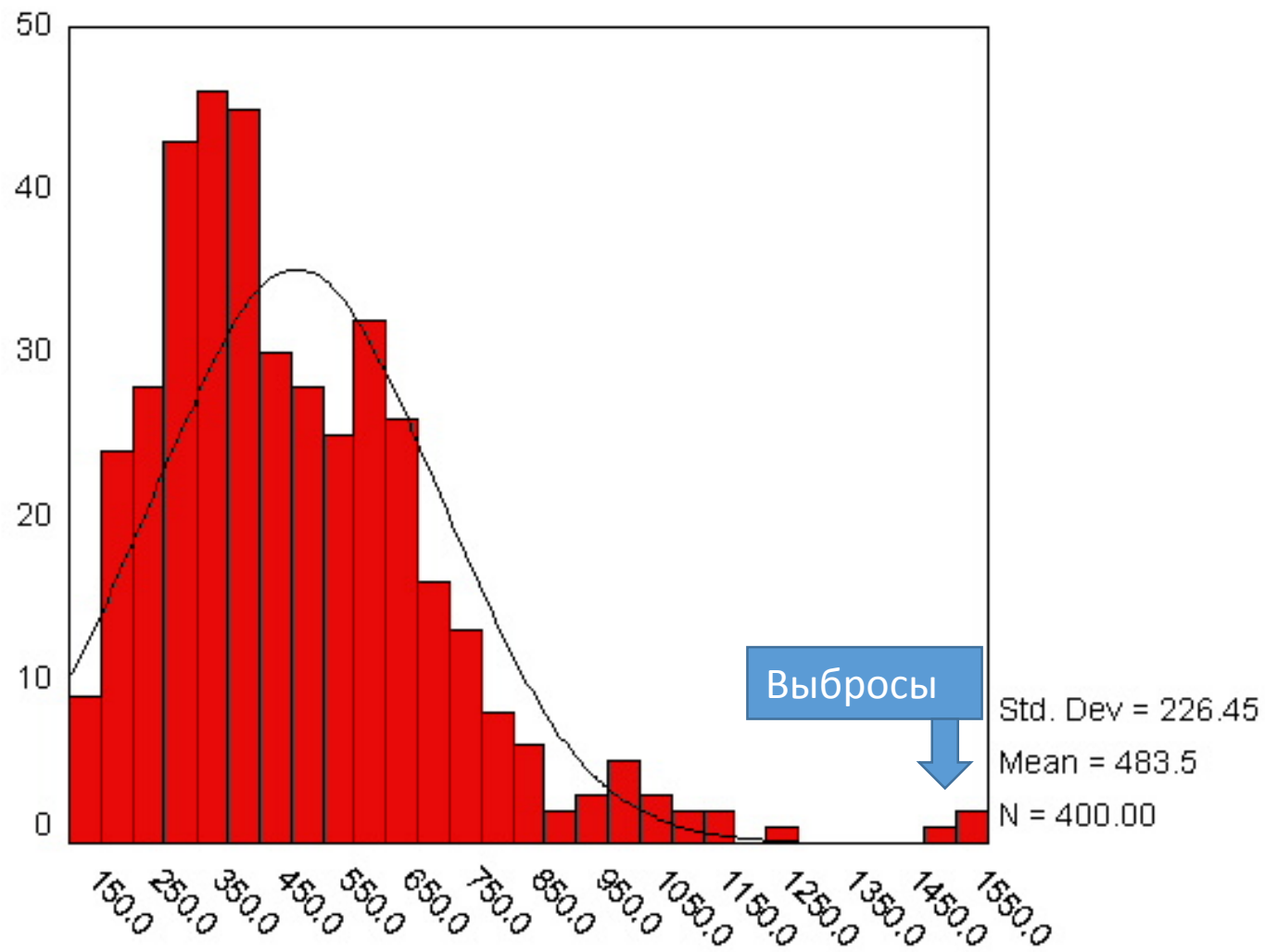
$$N_{ds}(x) = \frac{x}{10^{\min\{i: 10^i > x\}}}$$

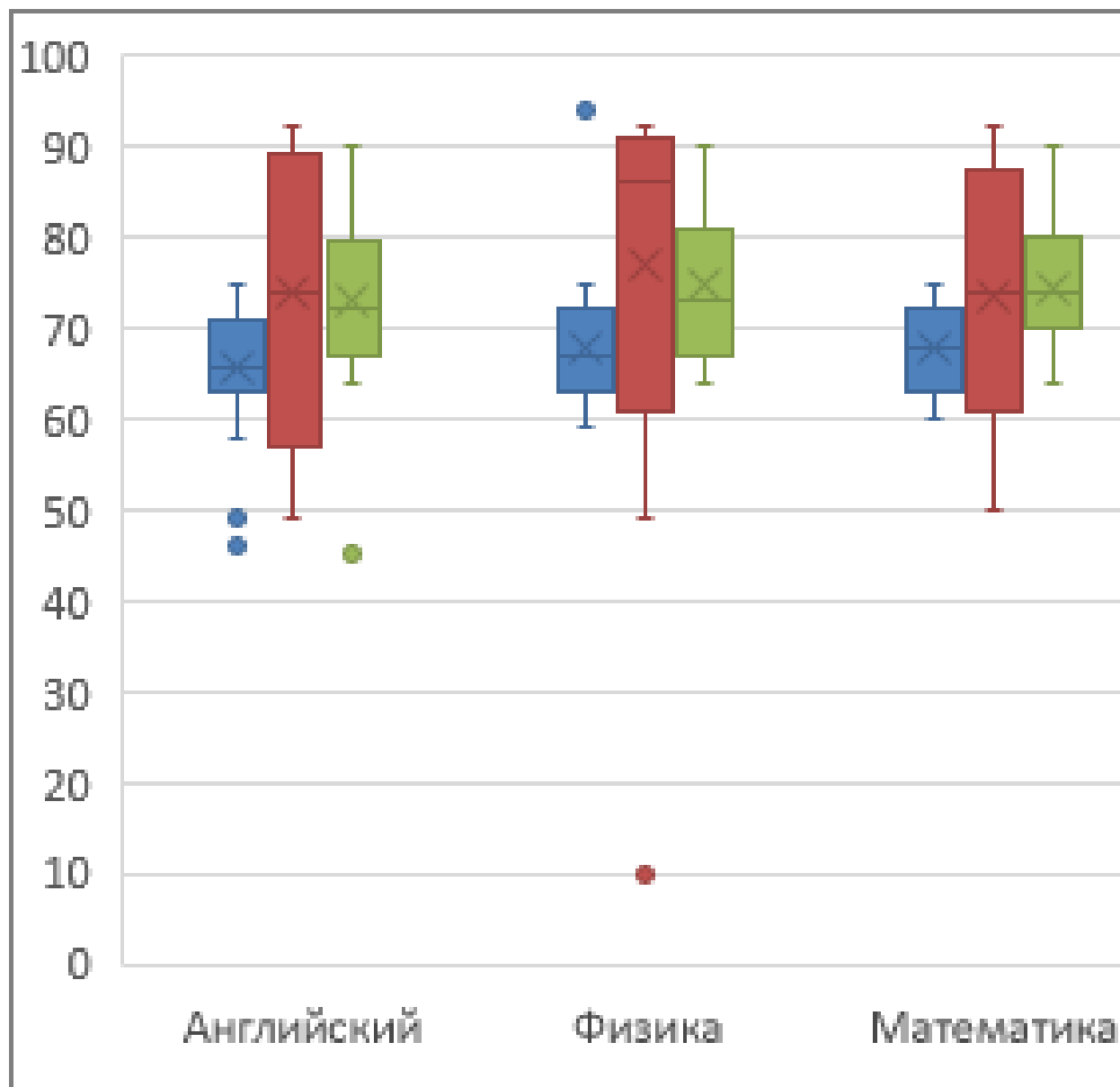
- **Ранговая нормировка**  
(tiedrank, rankdata)

- функция **scale()** в **R** нормирует данные так, чтобы его среднее арифметическое было равно нулю, а стандартное отклонение – единице (делает стандартизацию)

# Дендограмма регионов по нерегулируемым факторам РПТ



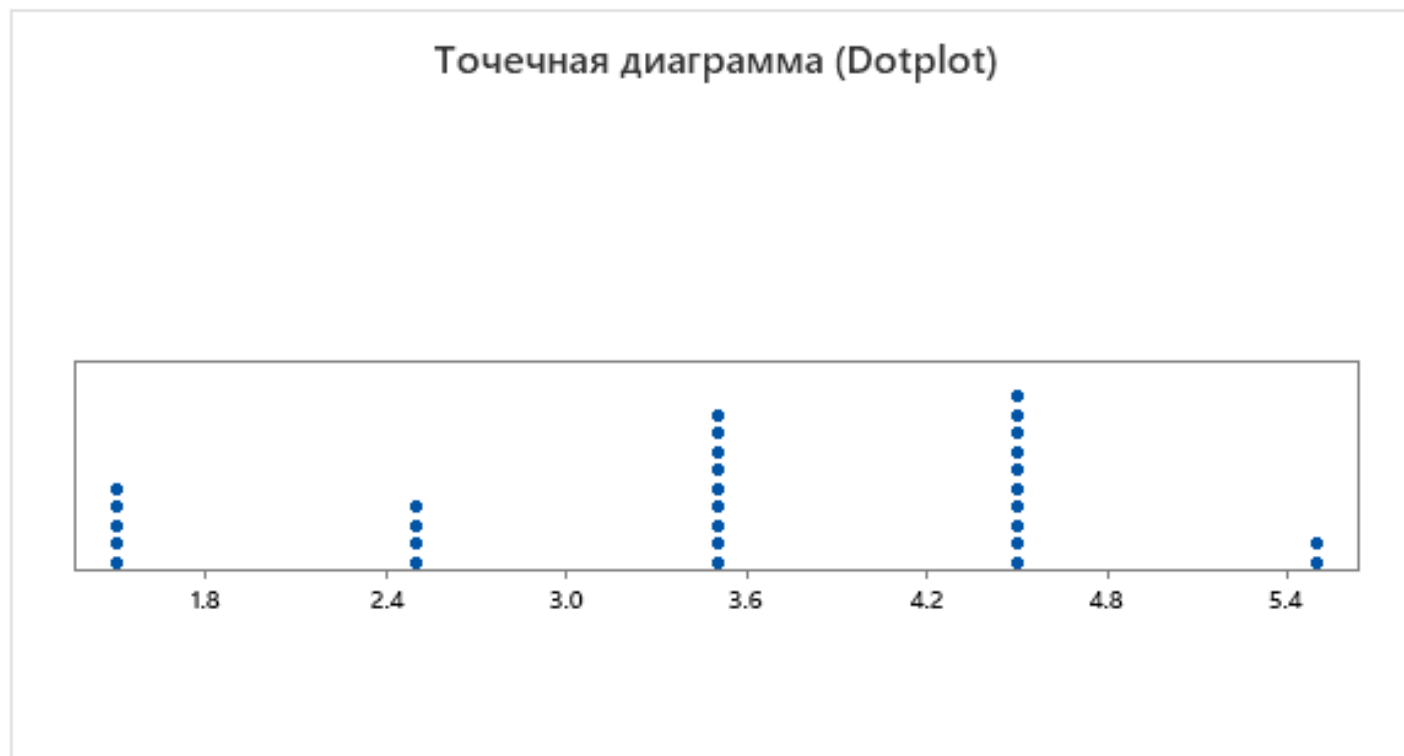




Выбросы

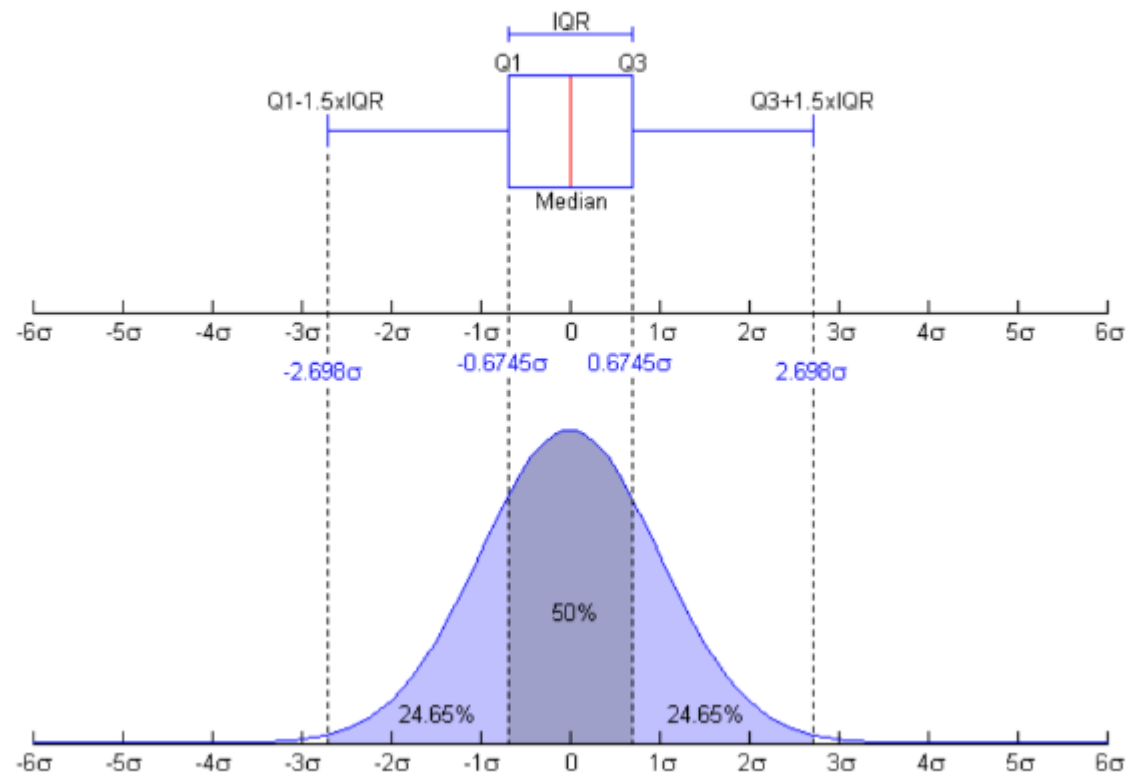
Точечные диаграммы дают нам немного больше представления об индивидуальных наблюдениях, так как отражают каждое из них. Гистограммы, в отличие от них, объединяют наблюдения, которые попадают в один интервал, под одним столбцом. Это преимущество, однако, теряется с увеличением количества наблюдений, так как с увеличением количества единичных наблюдений близкие значения также группируют в точки.

Еще с помощью точечных диаграмм легче заметить “гранулы” – одинаковые значения:



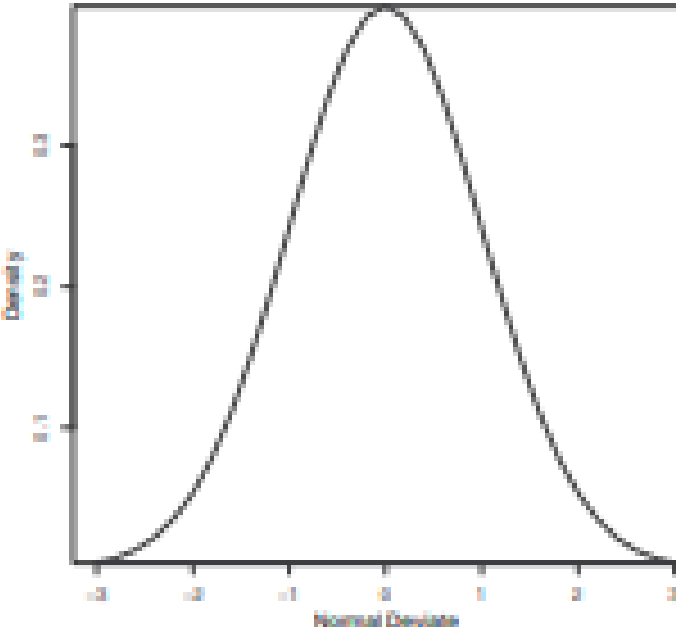
# Правило трёх сигм

**Описательные статистики – характерные элементы**



- В качестве критерия при подобном выборе обычно предполагается достижение наиболее робастной оценки. Хогг [20] предложил простую схему для выбора робастной оценки, основанную на тестировании набора альтернативных оценок для широкого ряда распределений методами Монте-Карло:
- если эксцесс находится между 2 и 4, то среднеарифметическое является рекомендуемой оценкой;
- если эксцесс находится между 4 и 5,5, тогда лучше использовать 25-процентное усеченное среднее;
- если эксцесс превышает 5,5, тогда лучшей оценкой является медиана.
- Результаты Коенкера и Бассетта [21] подтвердили схему Хогга: чем больше эксцесс, тем тяжелее хвосты и тем меньший вес должен быть для экстремальных наблюдений (выбросов), для того чтобы получить более устойчивую оценку.



Задача	Решение
<p>Как нарисовать кривую стандартного нормального распределения в диапазоне значений <math>[-3, 3]</math> (см. ниже)?</p> 	<pre>x &lt;- pretty(c(-3,3), 30) y &lt;- dnorm(x) plot(x, y,      type = "l",      xlab = "Normal Deviate",      ylab = "Density",      yaxp = "i" )</pre>
<p>Какова площадь под кривой стандартного нормального распределения слева от <math>z=1.96</math>?</p>	<p><code>pnorm(1.96)</code> равно 0.975</p>
<p>Каково значение 90-го перцентиля нормального распределения со средним значением 500 и стандартным отклонением 100?</p>	<p><code>qnorm(.9, mean=500, sd=100)</code> равно 628.16</p>
<p>Как создать 50 случайных чисел, принадлежащих нормальному распределению со средним значением 50 и стандартным отклонением 10?</p>	<p><code>rnorm(50, mean=50, sd=10)</code></p>

В программе R функции распределения имеют вид:

[dpqr](сокращенное название распределения), где первые буквы означают параметры распределения данных:

- d = плотность;
- p = функция распределения;
- q = функция, определяющая квантили;
- r = генератор случайных отклонений

Распределение	Сокращенное название
Бета	<code>beta</code>
Биномиальное	<code>binom</code>
Коши	<code>Cauchy</code>
Хи-квадрат (асимметричное)	<code>chisq</code>
Экспоненциальное	<code>exp</code>
F	<code>f</code>
Гамма	<code>gamma</code>
Геометрическое	<code>geom</code>
Гипергеометрическое	<code>hyper</code>
Логнормальное	<code>lnorm</code>
Логистическое	<code>logis</code>
Мультиномиальное	<code>multinom</code>
Отрицательное биномиальное	<code>nbinom</code>
Нормальное	<code>norm</code>
Пуассоновское	<code>pois</code>
Знаковых рангов Вилкоксона	<code>signrank</code>
T	<code>t</code>
Равномерное	<code>unif</code>
Вейбулла	<code>weibull</code>
Суммы рангов Вилкоксона	<code>wilcox</code>

# Типы распределений в R

```
> library(MASS)
> options(digits=3)
> set.seed(1234)
```

← ① Определяем случайное начальное число

```
> mean <- c(230.7, 146.7, 3.6)
> sigma <- matrix(c(15360.8, 6721.2, -47.1,
                    6721.2, 4700.9, -16.5,
                    -47.1, -16.5, 0.3), nrow=3, ncol=3)
```

← ② Назначаем вектор средних значений и ковариационную матрицу

```
> mydata <- mvrnorm(500, mean, sigma)
> mydata <- as.data.frame(mydata)
> names(mydata) <- c("y", "x1", "x2")
```

← ③ Генерируем данные

```
> dim(mydata)
> head(mydata, n=10)
```

	y	x1	x2
1	98.8	41.3	4.35
2	244.5	205.2	3.57
3	375.7	186.7	3.69
4	-59.2	11.2	4.23
5	313.0	111.0	2.91
6	288.8	185.1	4.18
7	134.8	165.0	3.68
8	171.7	97.4	3.81
9	167.3	101.0	4.01
10	121.1	94.5	3.76

← ④ Смотрим результаты

- Генерация данных, принадлежащих многомерному нормальному распределению в R

# Процедуры предварительной статистической обработки данных

- Заполнение пропущенных значений в имеющихся данных
- Выявление и исключение аномальных единиц наблюдения
- Обобщение исходных данных и приведение их в наглядную форму с применением методов описательной статистики

## Методы работы с аномальными данными

Статистические методы распознавания аномалий:

- **Непараметрические:** не требуют формализации заранее определённого закона распределения и реализуются с помощью алгоритмов сопоставления имеющихся значений переменной (гистограмма, box-plot, дендограмма)
- **Параметрические:** применение статистических критериев

### **Гистограммы очень хороши**

- быстро оценить форму распределения
  - придумать деформацию

но надо настраивать вручную (впрочем, любую визуализацию)

**Есть много описательных статистик**  
хороши как признаки

**Смотреть по признакам**  
распределения, распределения обучение / тест, распределения целевой переменной,  
аномальности в распределении, пропуски, естественность порядка значений

**Приёмы:**  
**деформация признака (чаще логарифмирование)**  
**масштабирование**

## 2. Команды статистического анализа

Знак	Описание
<	Меньше чем
<=	Меньше или равно
>	Больше чем
>=	Больше или равно
==	Тождественно равно
!=	Не равно
!x	Не x
x y	x или y
x&y	x и y
isTRUE(x)	Проверяет, выполняется ли x

Оператор	Описание
+	Сложение
-	Вычитание
*	Умножение
/	Деление
^ или **	Возведение в степень
x%y	Остаток от деления x на y: 5%2=1
x%/y	Целая часть при делении x на y: 5%/2=2

Функция	Описание
abs(x)	Модуль abs(-4) равно 4
sqrt(x)	Квадратный корень sqrt(25) равно 5 Это то же, что и 25^(0.5)
ceiling(x)	Наименьшее целочисленное значение, не меньшее, чем x ceiling(3.457) равно 4
floor(x)	Наибольшее целочисленное значение, не большее, чем x floor(3.457) равно 3
trunk(x)	Целое число, полученное при округлении x в сторону нуля trunk(5.99) равно 5
round(x, digits=n)	Округляет x до заданного числа знаков после запятой round(3.475, digits=2) равно 3.48

## 2. Команды статистического анализа

Функция	Описание
<code>signif(x, digits=n)</code>	Округляет $x$ до заданного числа значащих цифр <code>signif(3.475, digits=2)</code> равно 3.5
<code>cos(x), sin(x), tan(x)</code>	Косинус, синус и тангенс <code>cos(2)</code> равно -0.416
<code>acos(x), asin(x), atan(x)</code>	Арккосинус, арксинус и арктангенс <code>acos(-0.416)</code> равно 2
<code>cosh(x), sinh(x), tanh(x)</code>	Гиперболические косинус, синус и тангенс <code>sinh(2)</code> равно 3.627
<code>acosh(x), asinh(x), atanh(x)</code>	Гиперболические арккосинус, арксинус и арктангенс <code>asinh(3.627)</code> равно 2
<code>log(x, base=n)</code> <code>log(x)</code> <code>log10(x)</code>	Логарифм $x$ по основанию $n$ Для удобства: <code>log(x)</code> – натуральный логарифм <code>log10(x)</code> – десятичный логарифм <code>log(10)</code> равно 2.3026 <code>log10(10)</code> равно 1
<code>exp(x)</code>	Экспоненциальная функция <code>exp(2.3026)</code> равно 10

Функция	Описание
<code>mean(x)</code>	Среднее арифметическое <code>mean(c(1, 2, 3, 4))</code> равно 2.5
<code>median(x)</code>	Медиана <code>median(c(1, 2, 3, 4))</code> равно 2.5
<code>sd(x)</code>	Стандартное отклонение <code>sd(c(1, 2, 3, 4))</code> равно 1.29
<code>var(x)</code>	Дисперсия <code>var(c(1, 2, 3, 4))</code> равно 1.67
<code>mad(x)</code>	Абсолютное отклонение медианы <code>mad(c(1, 2, 3, 4))</code> равно 1.48
<code>quantile(x, probs)</code>	Квантили, где $x$ – числовой вектор, для которого нужно вычислить квантили, а $probs$ – числовой вектор с указанием вероятностей в диапазоне [0; 1] # 30-й и 84-й процентиля $x$ <code>y &lt;- quantile(x, c(.3, .84))</code>
<code>range(x)</code>	Размах значений <code>x &lt;- c(1, 2, 3, 4)</code> <code>range(x)</code> равно <code>c(1, 4)</code> <code>diff(range(x))</code> равно 3
<code>sum(x)</code>	Сумма <code>sum(c(1, 2, 3, 4))</code> равно 10
<code>diff(x, lag=n)</code>	Разность значений в выборке, взятых с заданным интервалом ( $lag$ ). По умолчанию интервал равен 1. <code>x &lt;- c(1, 5, 23, 29)</code> <code>diff(x)</code> равно <code>c(4, 18, 6)</code>
<code>min(x)</code>	Минимум <code>min(c(1, 2, 3, 4))</code> равно 1
<code>max(x)</code>	Максимум <code>max(c(1, 2, 3, 4))</code> равно 4
<code>scale(x, center=TRUE, scale=TRUE)</code>	Значения объекта $x$ , центрованные ( <code>center=TRUE</code> ) или стандартизованные ( <code>center=TRUE, scale=TRUE</code> ) по столбцам. Пример дан в программном коде 5.6



Функция	Описание
<code>length(x)</code>	Число элементов объекта <code>x</code> . <code>x &lt;- c(2, 5, 6, 9)</code> <code>length(x)</code> равно 4
<code>seq(from, to, by)</code>	Создание последовательности элементов. <code>indices &lt;- seq(1,10,2)</code> <code>indices</code> равно <code>c(1, 3, 5, 7, 9)</code>
<code>rep(x, n)</code>	Повторяет <code>x</code> <code>n</code> раз. <code>y &lt;- rep(1:3, 2)</code> <code>y</code> равно <code>c(1, 2, 3, 1, 2, 3)</code>
<code>cut(x, n)</code>	Преобразует непрерывную переменную <code>x</code> в фактор с <code>n</code> уровнями. Для создания упорядоченного фактора добавьте опцию <code>ordered_result = TRUE</code>
<code>pretty(x, n)</code>	Создает «красивые» пограничные значения. Разделяет непрерывную переменную <code>x</code> на <code>n</code> интервалов, выбрав <code>n+1</code> одинаково отстоящих друг от друга округленных значений. Часто используется при построении диаграмм
<code>cat(... , file = "myfile", append = FALSE)</code>	Объединяет объекты в ... и выводит их на экран или в файл (если указано его название). <code>firstname &lt;- c("Jane")</code> <code>cat("Hello" , firstname, "\n")</code>

Функция `subset()` – способ создания новых переменных из данных посредством выбора переменных и наблюдений

## Пример:

```
> newdata <- subset(t2, X >= 5 & Y > 80,
+                   select=c(регионы, X, Y, Status))
> newdata
```

	регионы	X	Y	Status
83	Чукотский автономный округ	25	83.9	dota

```
> str(t2)
```

```
'data.frame': 84 obs. of 4 variables:
```

```
$ регионы: chr "Иркутская область" "Карачаево-Черкесская Республика"
"Иркутская область" "Сахалинская область" ...
```

```
$ Y : num 68.1 65.2 64.4 72.3 69.7 69.1 71.3 77.5 66.8 65.8 ...
```

```
$ X : num 2.2 2.7 3.3 3.4 4.6 4.8 5.1 5.1 5.3 5.3 ...
```

```
$ Status : chr "dota" "dota" "dota" "dota" ...
```

## Создание новых переменных как выборок из данных (sample)

```
> v1 <- t2[sample(1:nrow(t2), 3, replace=FALSE), 1:4]  
> v1
```

	регионы	У	X	Status
37	Орловская область	65.6	8.4	dota
39	Тамбовская область	63.8	8.8	dota
72	Республика Саха (Якутия)	70.6	14.6	dota

Из таблицы t2 из 84 строк и 4 столбцов выбрали случайно 3 строки

**Что можно увидеть в данных («признак» – «признак»)**  
**корреляцию**  
при правильном масштабе и небольшом шуме

**зависимость признаков**  
при малом шуме и «достаточно равномерном» распределении

**независимость признаков**  
часто это «ложное видение»

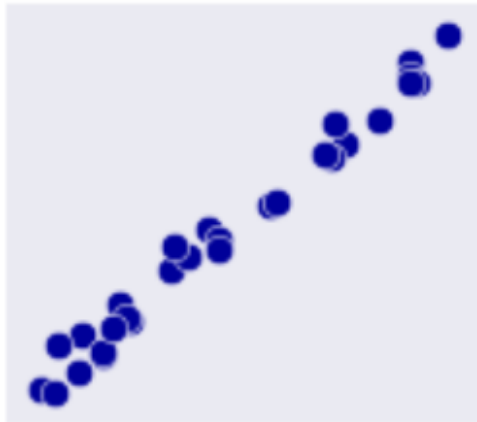
**типичные значения**  
сложно при большом объёме данных

**выбросы**  
при правильном масштабе

**кластеры**  
при правильном масштабе

**Самый распространённый способ –  
диаграмма рассеивания («скатерплот»)**

**Что можно увидеть в данных («признак» – «признак»)**



корреляция



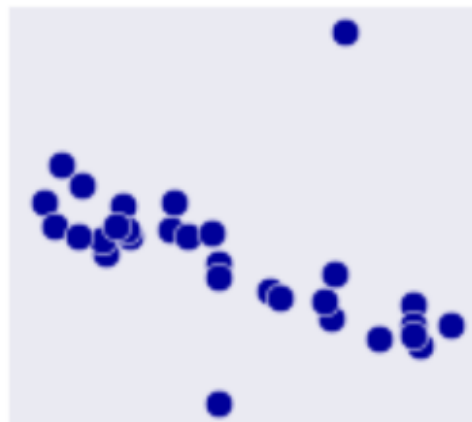
зависимость



независимость



типичные значения



выбросы



кластеры

1. Какие задачи решает разведочный анализ данных?
2. Что такое описательные статистики?
3. Как с помощью визуализации оценить данные?
4. Какие возможности визуализации данных есть в R?
5. Как можно вводить данные в R?
6. Какие специальные возможности есть в R для проверки данных на нормальность?
7. Как в R найти выбросы?
8. Как визуализировать взаимосвязи в R?

Роберт И. Кабаков R в действии. Анализ и визуализация данных в программе R / пер. с англ. Полины А. Волковой. – М.: ДМК Пресс, 2014. – 588 с.: ил. ISBN 978-5-947060-077-1