



# Занятие 1

## Язык R: введение

19 февраля 2014

# Зачем нужен R?

- Быстрая статистическая обработка данных
- Построение красивых графиков
- Бесплатный, удобный, быстрый для изучения язык



# Структура курса

- Базовый синтаксис
- Генерация данных
- Графики
- Инструменты для статистического анализа данных
  - Анализ биоинформатических данных

# Обучение и контроль

- <http://kodomo.cmm.msu.ru/wiki/2010/8/R>

- Домашние задания:

<https://bioinf.fbb.msu.ru/moodle/>

Необходимо зарегистрироваться и  
прислать письмо на  
fbbrcourse@gmail.com

Контрольные работы

# Среда разработки RStudio:

<http://www.rstudio.com/>

The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains R code for two functions: `getEcdfData` and `getFit`.

```
29 getEcdfData <- function(pos) {  
30   e <- ecdf(top[,pos])  
31   dat <- cbind(seq(0, 1, length.out=1000), e(se  
32   dat <- data.frame(dat)  
33   colnames(dat) <- c("v1", "v2")  
34   return (dat)  
35 }  
36  
37 getFit <- function(pos) {  
38   e <- ecdf(top[,pos])  
39   dat <- cbind(seq(0, 1, length.out=1000), e(se  
40   #dat <- cbind(knots(e), e(knots(e)))  
41   dat <- data.frame(dat)  
42   colnames(dat) <- c("v1", "v2")  
43
```
- Workspace:** Lists data objects: `RF` (1000 obs. of 1002 variables), `aCoeff` (951 obs. of 2 variables), and `b1Coeff` (951 obs. of 2 variables).
- Plots:** A plot window showing a graph with x-axis from 0 to 10 and y-axis from 0 to 60. It features a blue curve and a red curve.
- Console:** Shows the execution output, currently empty.

# Помощь

- Форумы: *Stackoverflow*, *R mailing list*, etc.
- Документация (<http://www.r-project.org>, *help(...)*)
- Полезная шпаргалка:

[R\\_reference.pdf](#) (лежит на kodomo)

- Спросить преподавателя (в самом крайнем случае!)

# R – векторизованный язык

- Основной тип данных – вектор (упорядоченный набор чисел)
- Идея – работать с набором данных как с одним числом (параллельно обрабатывать все значения набора)
  - Это позволяет обходиться (в ряде случаев) без циклов

# Вектор

```
> x<-1:5 ; y<-6:10
```

```
> x
```

```
[1] 1 2 3 4 5
```

```
> y
```

```
[1] 6 7 8 9 10
```

```
> x+y
```

```
[1] 7 9 11 13 15
```

```
> x*2
```

```
[1] 2 4 6 8 10
```

```
> x>4
```

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

```
> y==7
```

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

```
> x*y
```

```
[1] 6 14 24 36 50
```



# Как можно создать вектор?

Оператор `c()`

```
> c(1, 2, 3)
```

```
[1] 1 2 3
```

Последовательности

```
> 1:10
```

```
[1] 1 2 3 4 5 6 7 8 9 10
```

```
> seq(from=1, to=8, by=2)
```

```
[1] 1 3 5 7
```

```
> seq(1, 10, 2)
```

```
[1] 1 3 5 7 9
```

# Как можно создать вектор?

Объединение нескольких векторов

```
> x<-c(1, 2, 3)
```

```
> x<-c(x, 1:3); x
```

```
[1] 1 2 3 1 2 3
```

Повторы

```
> rep(0.5, 6)
```

```
[1] 0.5 0.5 0.5 0.5 0.5 0.5
```

Для целых чисел (работает быстрее)

```
> rep.int(1, 5)
```

```
[1] 1 1 1 1 1
```

# Как можно создать вектор?

## Распределение

- ✓ Нормальное распределение:
- ✓ `dnorm(x)` – плотность распределения
- ✓ `pnorm(q)` – функция распределения
- ✓ `qnorm(p)` – квантильная функция

Случайная генерация из распределения: `> set.seed(100)`

`> rnorm(5)`

`[1] 1.1568405 -0.8248219 0.1428891 -0.4784408 0.7561443`

## Равномерное

```
runif(n, min=0, max=1)
```

```
> runif(5, 0, 1)
```

```
[1] 0.1972687 0.3090867 0.2865924 0.1409635 0.3441481
```

## Биномиальное

```
rbinom(n, size, prob)
```

```
> rbinom(10, 100, 0.5)
```

```
[1] 54 47 55 50 47 45 52 45 58 52
```

## Пуассона

```
rpois(n, lambda)
```

```
> rpois(10, 4)
```

```
[1] 2 3 2 4 10 3 2 3 5 6
```

# Срезы

```
> x<-c(1, 5, 7, 9, 15, 3)
```

```
> x[1]
```

```
[1] 1
```

```
> x[2:4]
```

```
[1] 5 7 9
```

```
> x[c(2, 5)]
```

```
[1] 5 15
```

```
> x[-1]
```

```
[1] 5 7 9 15 3
```

```
> x[-(1:3)]
```

```
[1] 9 15 3
```

```
> x[x>5]
```

```
[1] 7 9 15
```

```
> x[x>5 & x<10]
```

```
[1] 7 9
```

# Простейший статистический анализ

```
> x=rnorm(100)
```

**Среднее**

```
> mean(x)
```

```
[1] -0.04029328
```

**Стандартное отклонение**

```
> sd(x)
```

```
[1] 1.037552
```

# Простейший статистический анализ

## Минимальное и максимальное значения

```
> min(x)
```

```
[1] -2.605444
```

```
> max(x)
```

```
[1] 2.51254
```

## Медиана


```
> median(x)
```

```
[1] -0.1039548
```

## Квантили

```
> quantile(x)
```

	0%	25%	50%	75%	100%
	-2.6054443	-0.6321819	-0.1039548	0.4765935	
					2.5125400



А еще данные удобно  
анализировать с помощью графиков

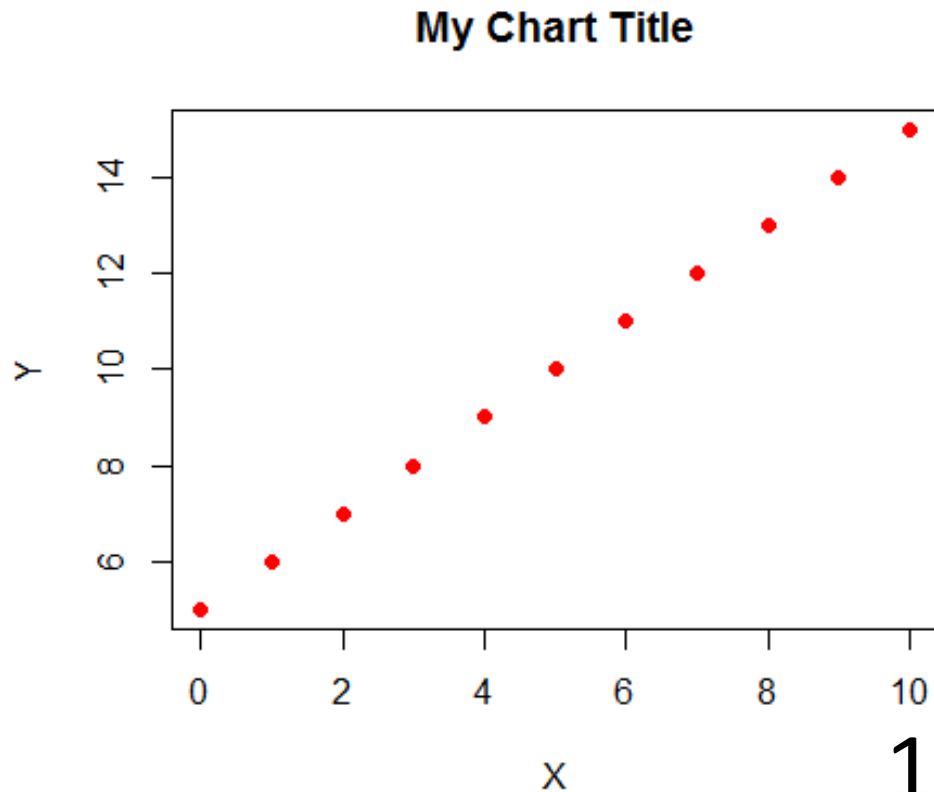


# Самый простой график

```
>x_data <- c(0:10)
```

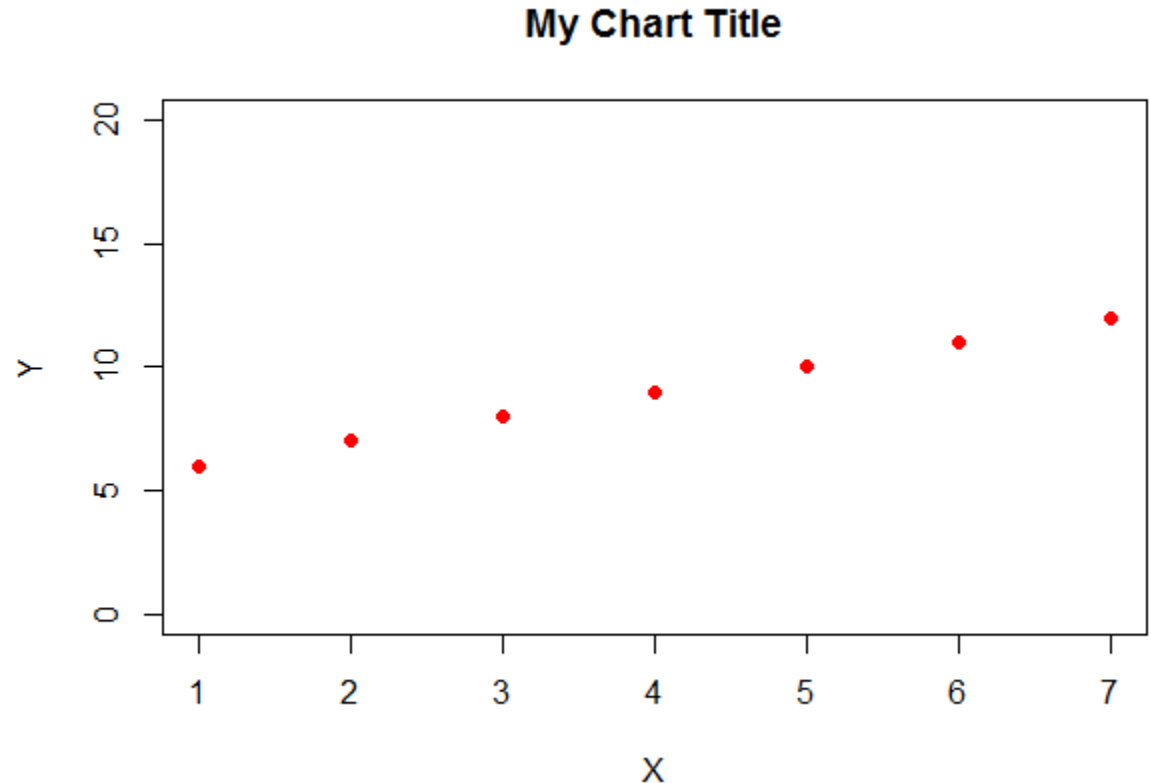
```
>y_data <- x_data +5
```

```
>plot(x_data, y_data, main = "My Chart Title", xlab = "X", ylab = "Y", pch=16, col = "red")
```



# Параметры xlim, ylim

```
>plot(x_data, y_data,  
main = "My Chart  
Title", xlab = "X", ylab =  
"Y", pch=16, col = "red",  
xlim=c(1,7), ylim=c(0,  
20))
```



# Линии

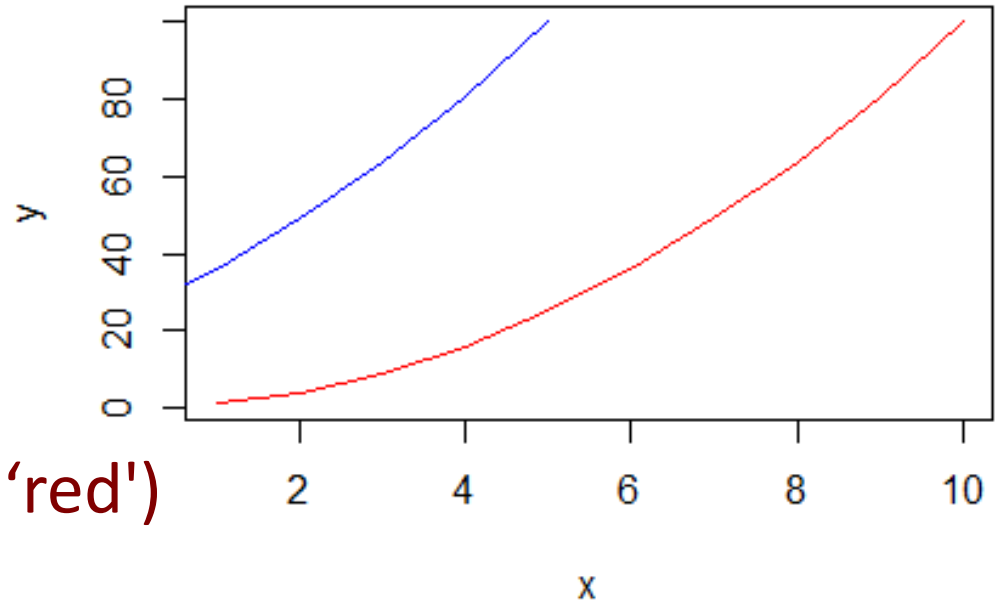
```
> x <- 1:10
```

```
> y <- x*x
```

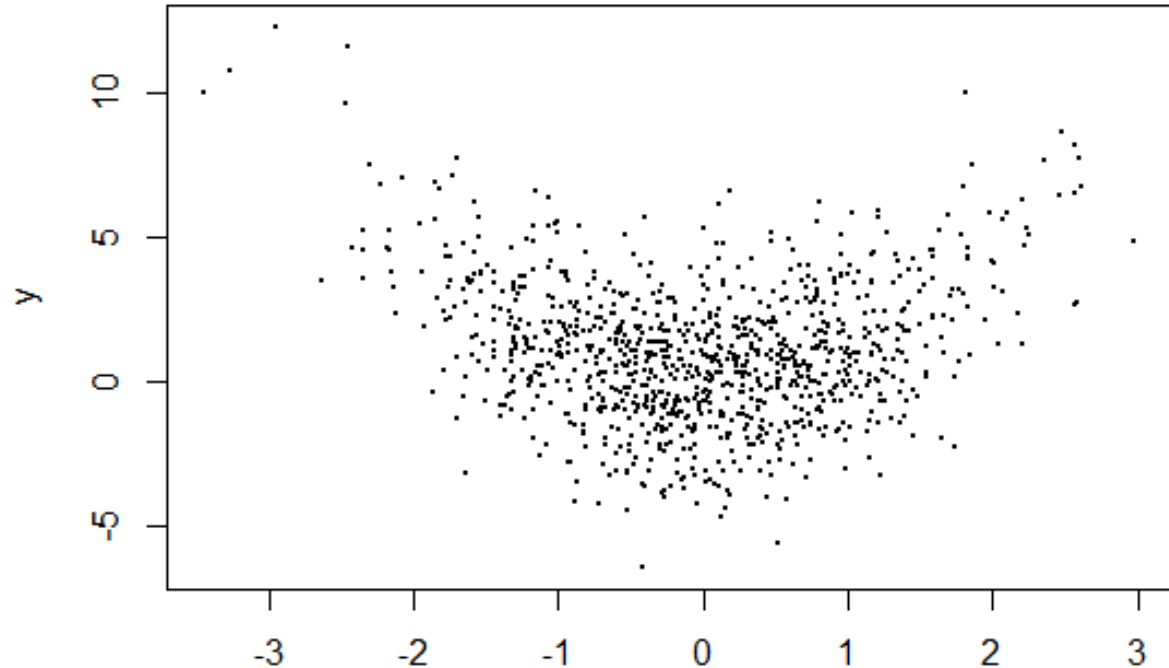
```
> z <- x-5
```

```
> plot(y ~ x, type="l", col = 'red')
```

```
> lines(y ~ z, col = 'blue')
```

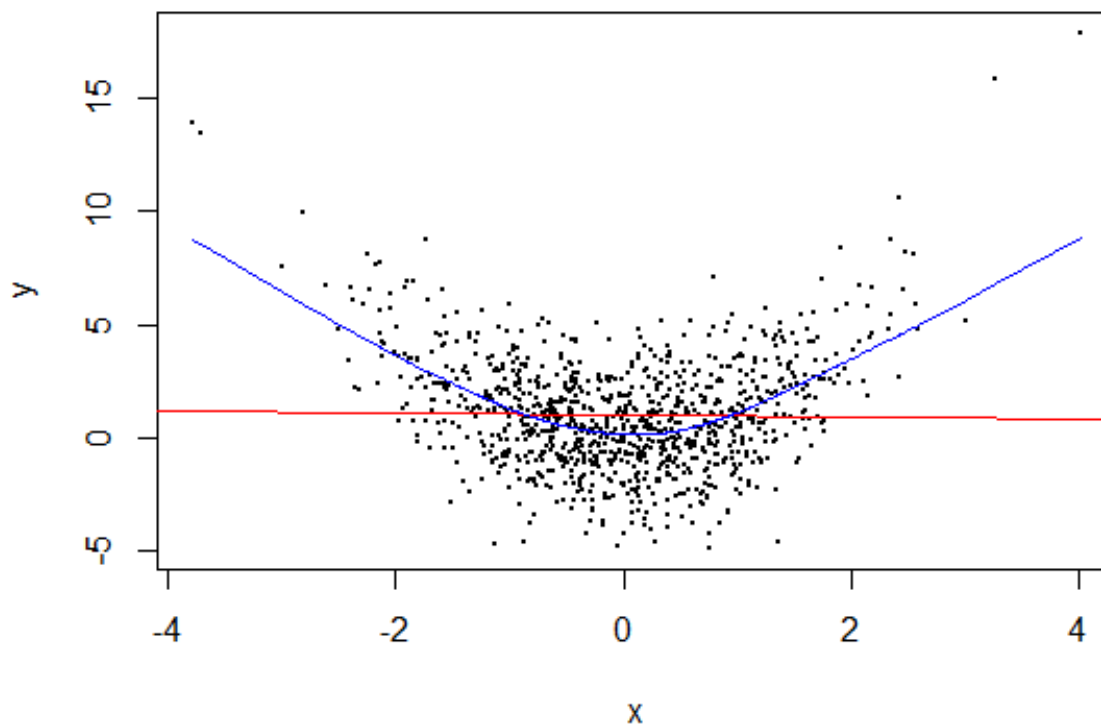


# Scatterplots



```
> x<-rnorm(1000)  
> y<-x*x + rnorm(1000, sd=2)  
> plot(x, y, pch=19, cex=0.3)
```

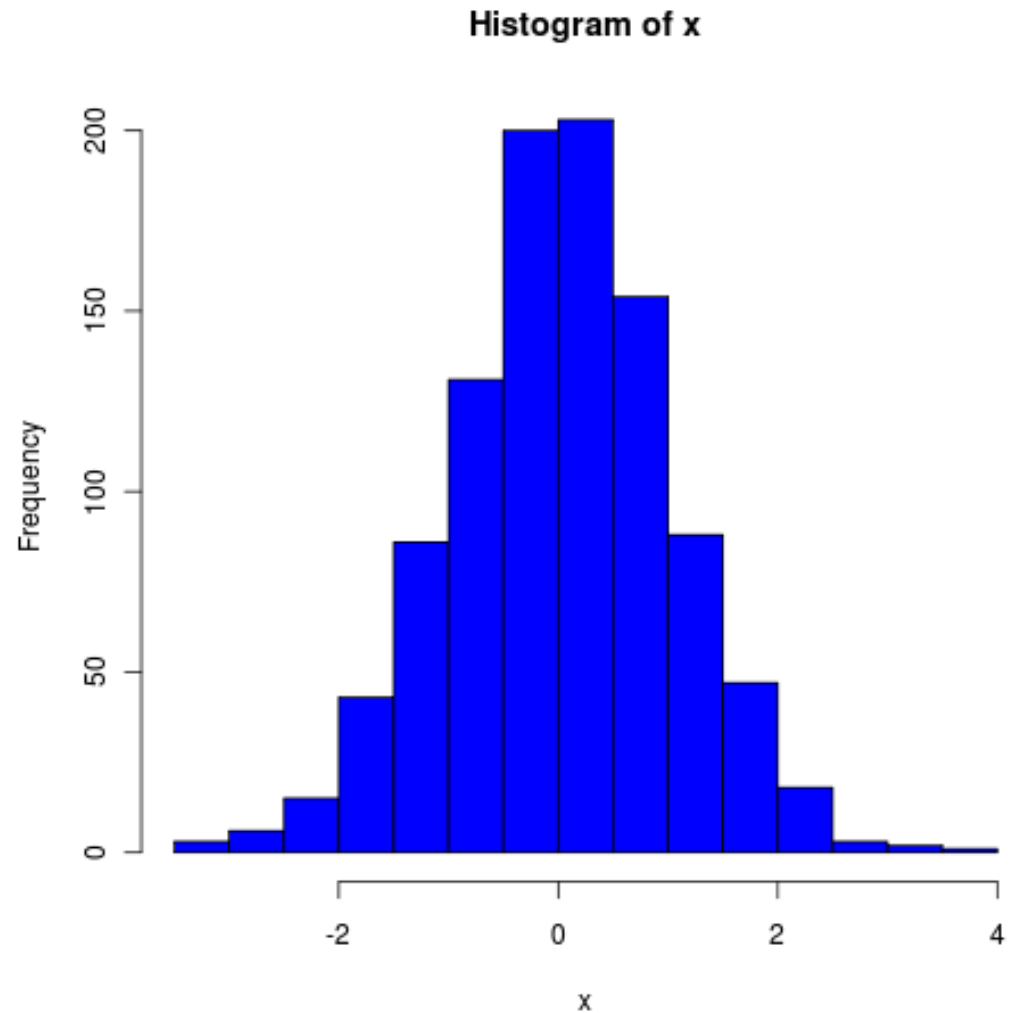
# Scatterplots: добавим линии



```
> abline(lm(y~x), col="red")  
> lines(lowess(y~x), col="blue")
```

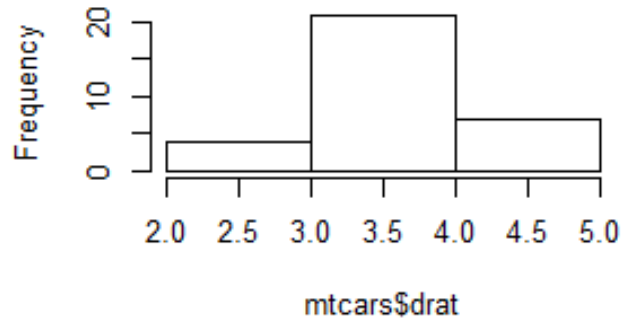
# Гистограммы

```
> x=rnorm(1000)  
> hist(x, col='blue')
```

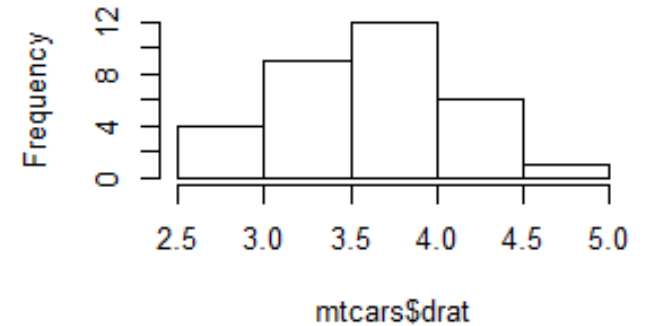


# Гистограммы

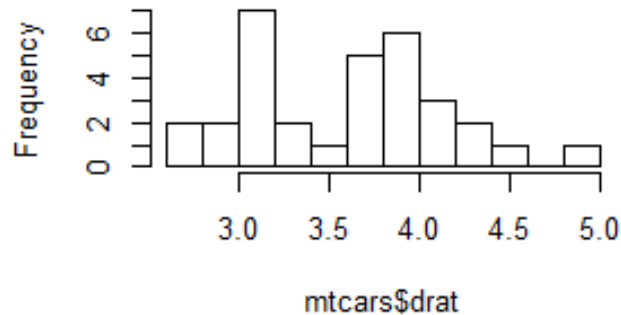
Histogram of mtcars\$drat



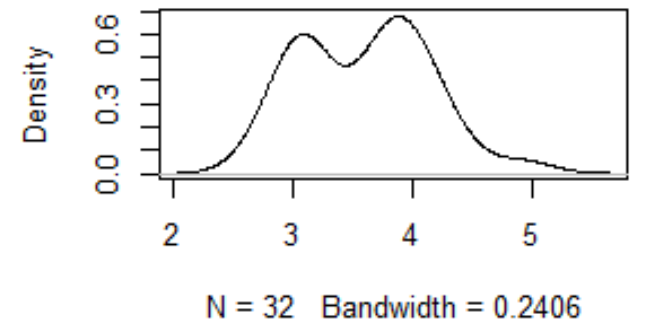
Histogram of mtcars\$drat



Histogram of mtcars\$drat

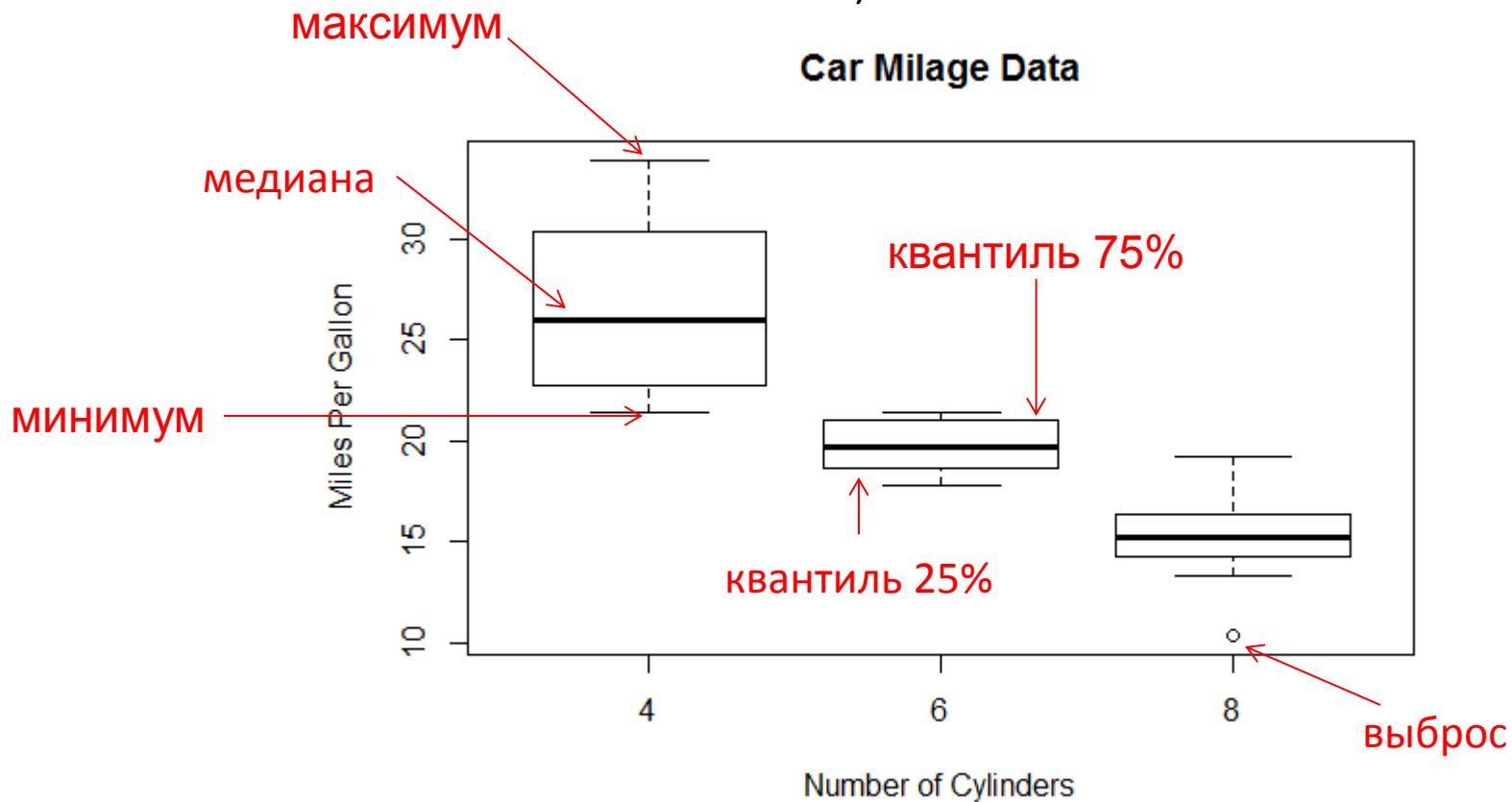


density.default(x = mtcars\$drat)



# Boxplots

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






# Сохранение графика в файл

```
> png(file="Pictures/boxplot.png", width=400,  
height=350, res=72)  
> boxplot(x,y)  
> dev.off()
```

Другие форматы:

<code>pdf("mygraph.pdf")</code>	pdf file: для печати
<code>win.metafile("mygraph.wmf")</code>	windows metafile
<code>png("mygraph.png")</code>	png file: для веба
<code>jpeg("mygraph.jpg")</code>	jpeg file: не рекомендуем
<code>bmp("mygraph.bmp")</code>	bmp file
<code>postscript("mygraph.ps")</code>	postscript file



Вопросы, комментарии, предложения –  
по адресу  
[fbbrcourse@gmail.com](mailto:fbbrcourse@gmail.com)