

Язык R

лекция 2

Елена Ставровская, Анастасия Жарикова
16 сентября 2015

Простая графика

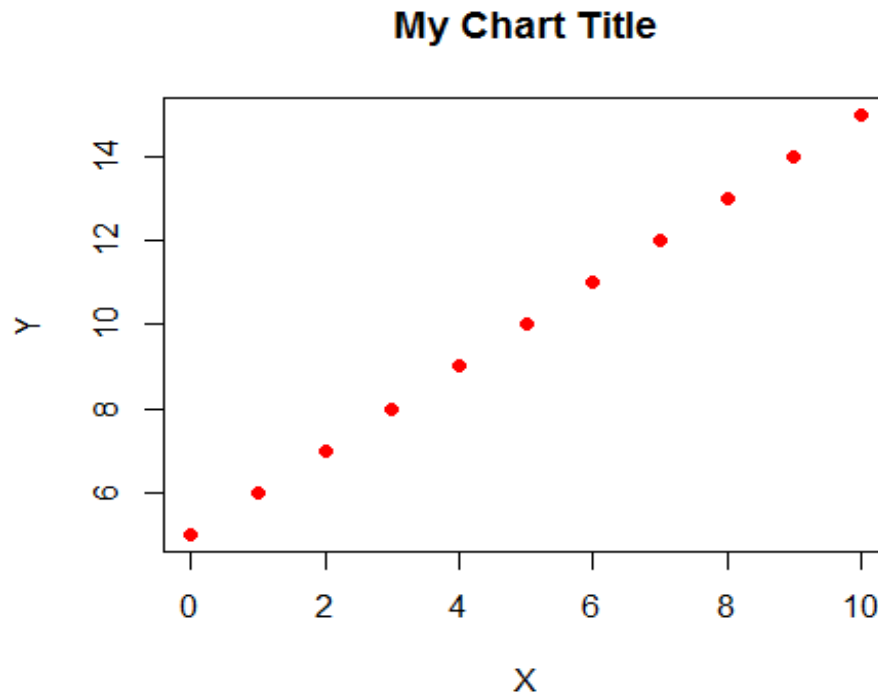
Работа с отсутствующими
данными

Структуры данных (Контейнеры)

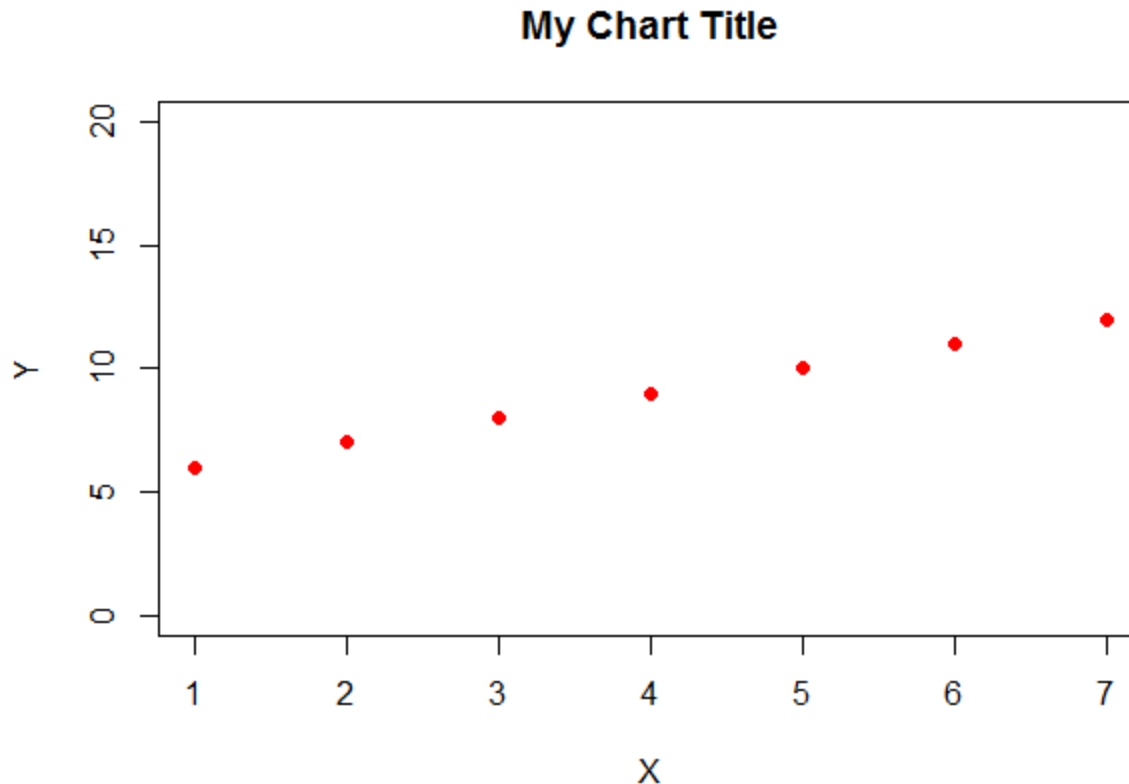
ГРАФИКА

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")
```



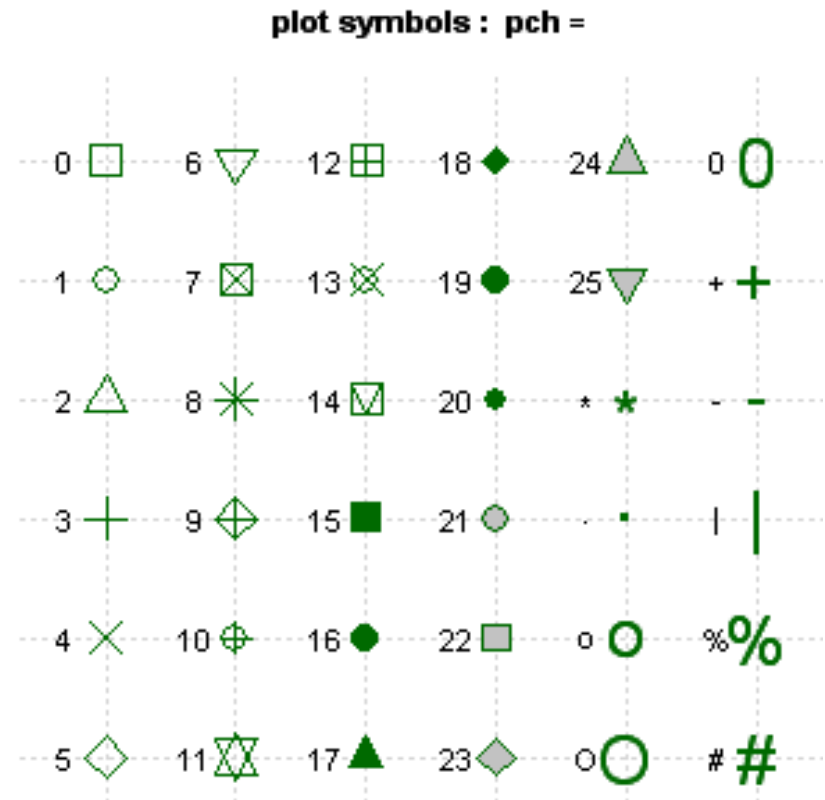
Параметры xlim, ylim



```
>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))
```

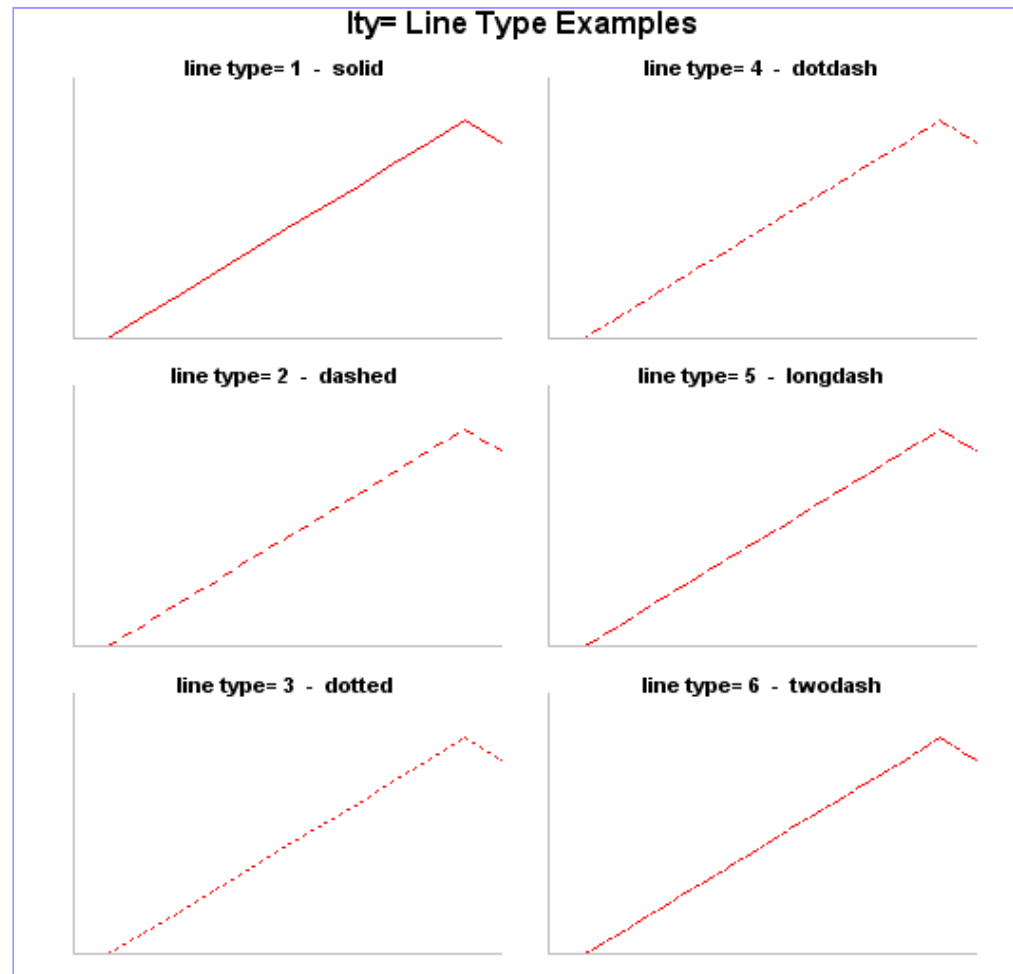
Параметр pch

- В R существует 25 символов для графиков
- Символы 19 – 20 могут быть залиты выбранным цветом
- Символы 21: 25 могут быть залиты выбранным цветом (col) и обведены рамкой (bg)

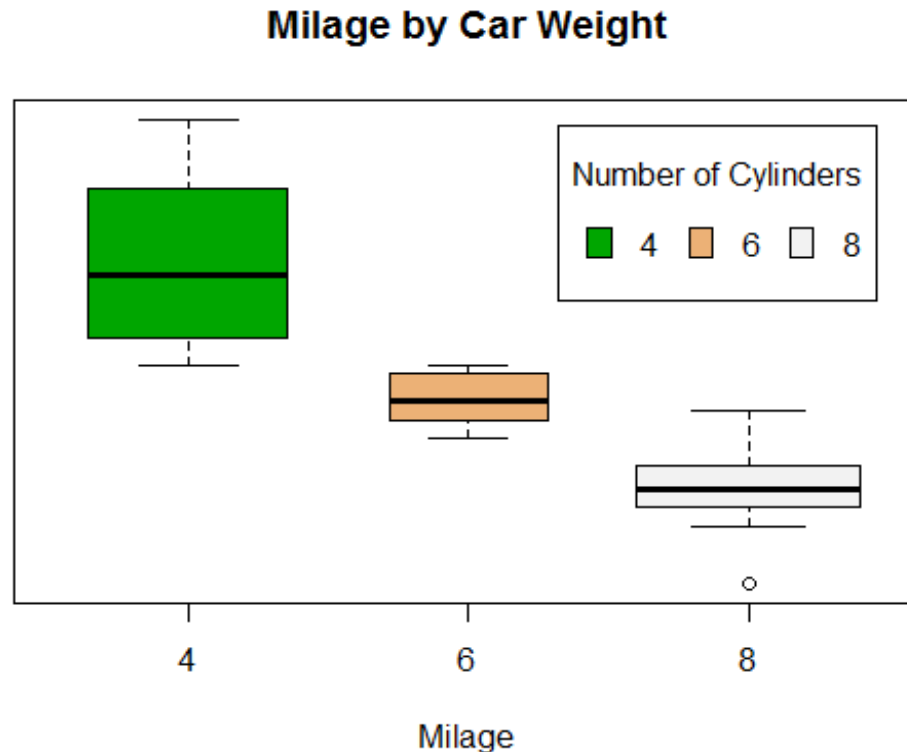


Параметр lty

- В R существует 7 типов линий
- 0 – «прозрачная линия»
- 1 – «сплошная»
- 2 – «пунктирная»
- 3 – «точками»
- 4 – «точка-тире»
- 5 – «длинное тире»
- 6 – «двойное тире»



Параметр legend



```
> boxplot(mtcars$mpg~mtcars$cyl, main="Milage by Car Weight", yaxt="n", xlab="Milage", col=terrain.colors(3), varwidth=T)
```

```
> legend("topright", inset=.05, title="Number of Cylinders", c("4","6","8"), fill=terrain.colors(3), horiz=TRUE)
```


Графический параметр par()

```
> par()
```

```
# посмотреть текущие
```

```
настройки
```

```
> par(col.lab="red")
```

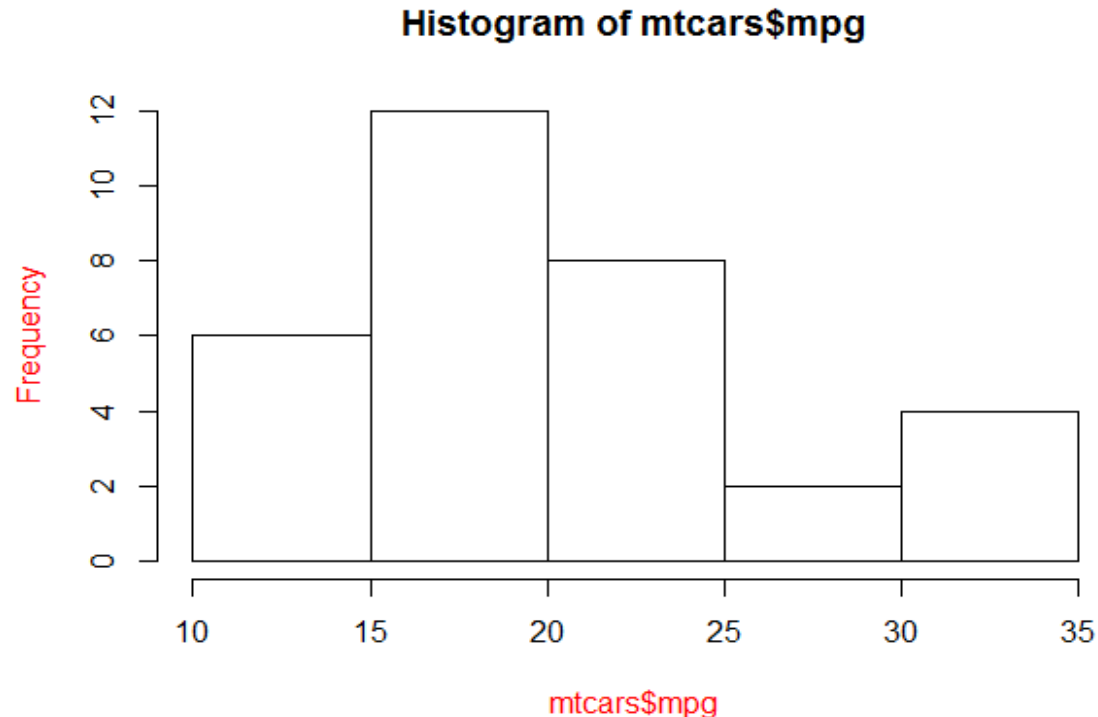
```
# сделать красными подписи к
```

```
осям
```

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

```
# нарисовать график с новыми
```

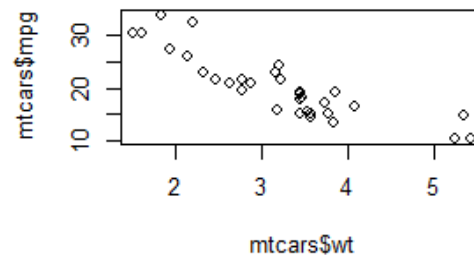
```
настройками
```



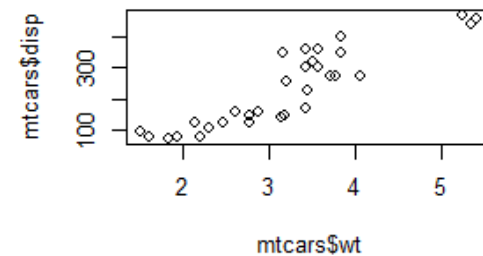
Комбинация графиков

```
> par(mfrow=c(2,2))
> plot(mtcars$wt,mtcars$mpg,
main="Scatterplot of wt vs. mpg")
> plot(mtcars$wt,mtcars$disp,
main="Scatterplot of wt vs disp")
> hist(mtcars$wt,
main="Histogram of wt")
> boxplot(mtcars$wt,
main="Boxplot of wt")
```

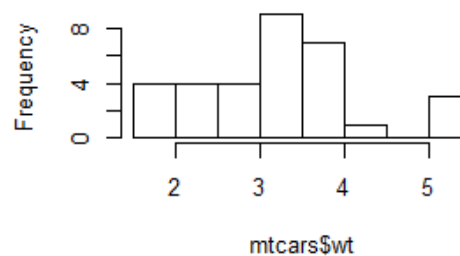
Scatterplot of wt vs. mpg



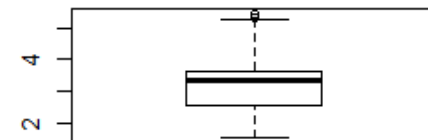
Scatterplot of wt vs disp



Histogram of wt

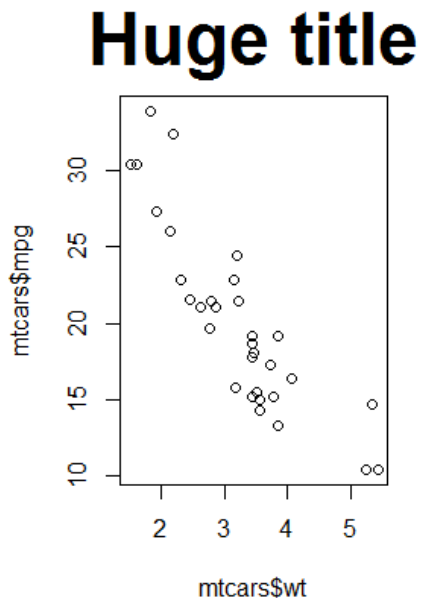
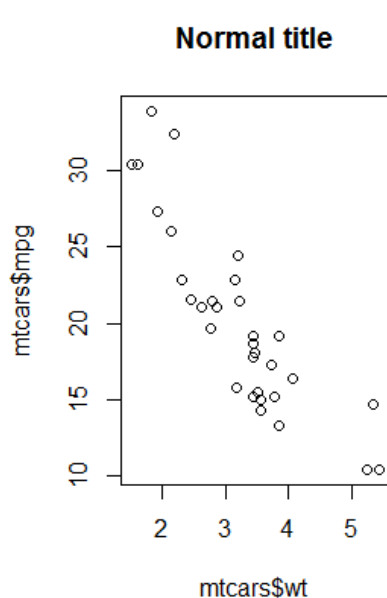


Boxplot of wt



Размер текста и символов

ОПЦИЯ	ОПИСАНИЕ
<code>cex</code>	Размер текста и символов относительно размера по умолчанию
<code>cex.axis</code>	Увеличение текста по осям
<code>cex.lab</code>	Увеличение подписей к осям
<code>cex.main</code>	Увеличение заголовков



```
> par(mfrow=c(1,2))  
> plot(mtcars$mpg ~ mtcars$wt,  
main="Normal title")  
> plot(mtcars$mpg ~ mtcars$wt,  
main="Huge title", cex.main=3)
```

Цвета

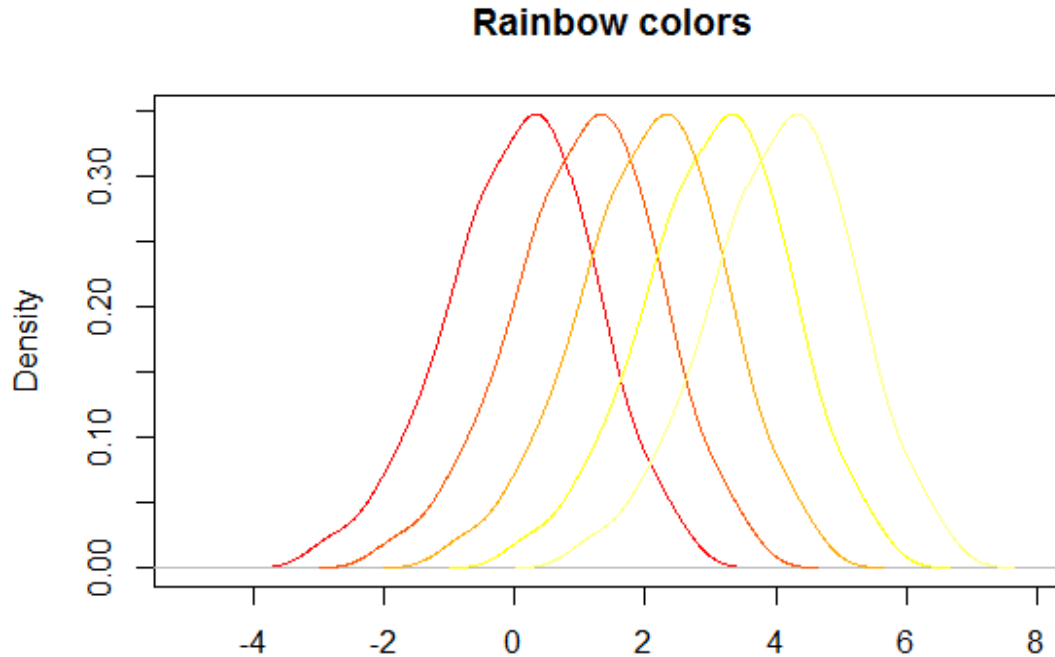
ОПЦИЯ	ОПИСАНИЕ
<code>col</code>	Цвет по умолчанию (может быть вектором)
<code>col.axis</code>	Цвет текста по осям
<code>col.lab</code>	Цвет подписей к осям
<code>col.main</code>	Цвет заголовков

Можно использовать функции

`rainbow(n)`, **`heat.colors(n)`**,
`terrain.colors(n)`, **`topo.colors(n)`** и **`cm.colors(n)`**

для создания вектора цветов

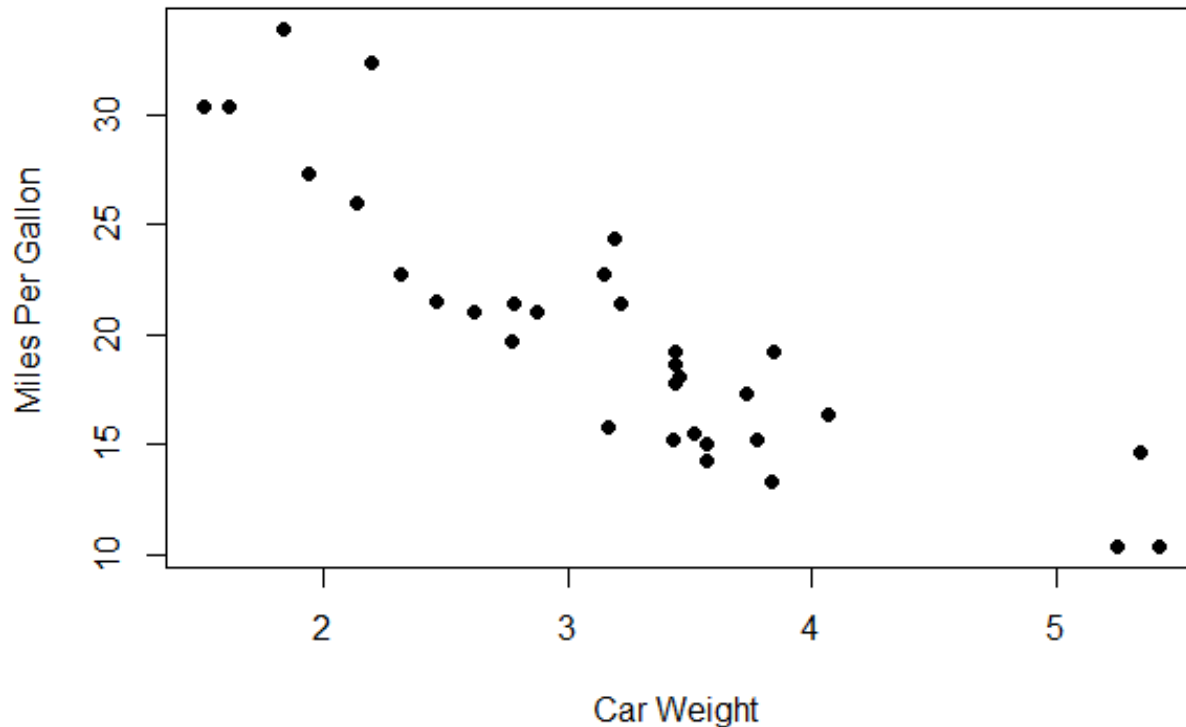
Цвета



```
> x1 <- rnorm(100) ; x2 <- x1+1 ; x3 <- x2+1 ; x4 <- x3+1 ; x5 <- x4+1  
> ourCol <- heat.colors(5)  
> plot(density(x1), col=ourCol[1], xlim=c(-5,8), main="Rainbow colors",  
xlab="")  
> lines(density(x2), col=ourCol[2])  
> lines(density(x3), col=ourCol[3])  
> lines(density(x4), col=ourCol[4])  
> lines(density(x5), col=ourCol[5])
```

Scatterplots

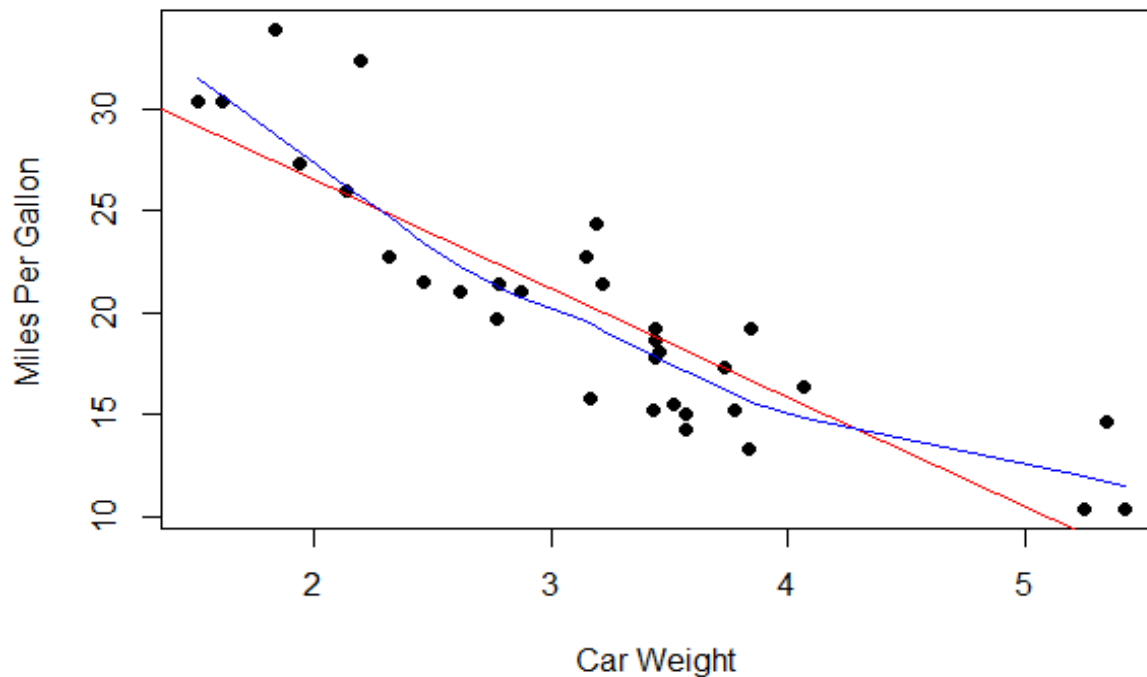
Scatterplot Example



```
> plot(mtcars$wt, mtcars$mpg, main="Scatterplot Example",  
xlab="Car Weight ", ylab="Miles Per Gallon ", pch=19)
```

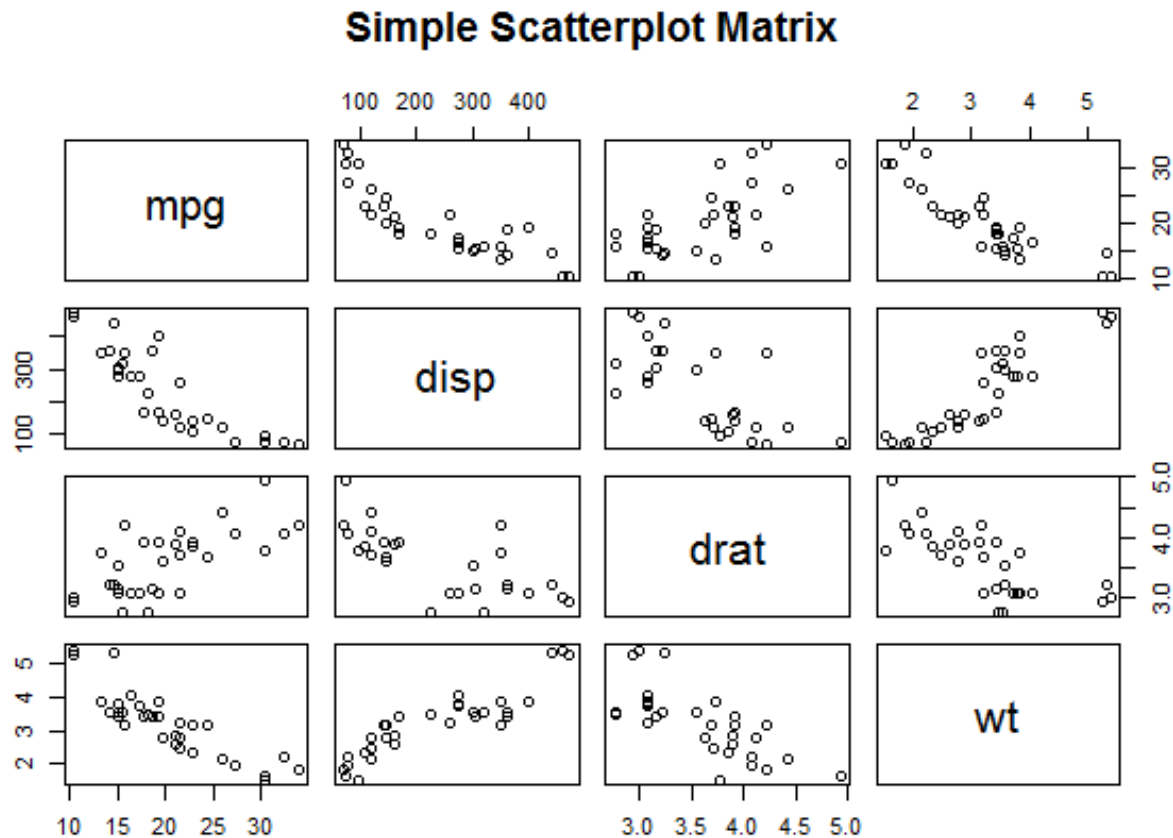
Scatterplots: добавим ЛИНИИ

Scatterplot Example



- > `abline(lm(mtcars$mpg~mtcars$wt), col="red")`
- > `lines(lowess(mtcars$wt, mtcars$mpg), col="blue")`

Scatterplot: матрицы

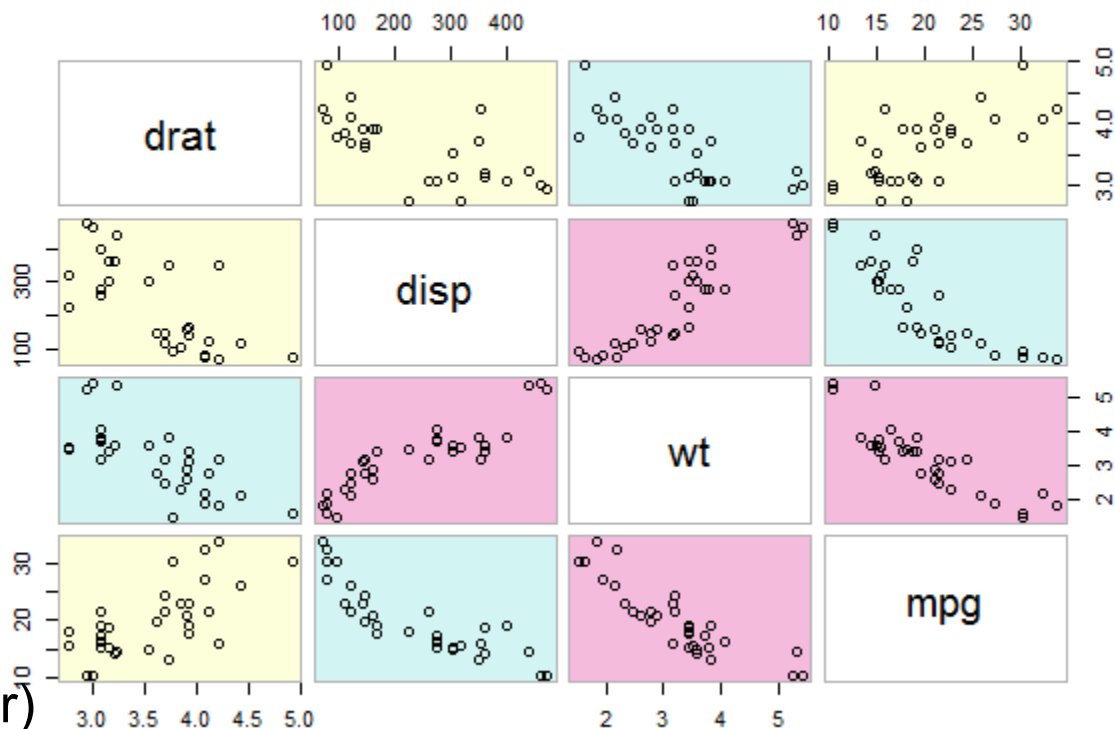


```
> pairs(mtcars[,c(1,3,5,6)]) #или то же самое:  
> pairs(~mpg+disp+drat+wt, data=mtcars,  
main="Simple Scatterplot Matrix")
```


Другие scatterplots

gclus package позволяет группировать переменные таким образом, чтобы переменные с большими корреляциями были ближе к диагонали. Цвета соответствуют коэффициенту корреляции.

Variables Ordered and Colored by Correlation



```
> library(gclus)
> dta <- mtcars[,c(1,3,5,6)]
> dta.r <- abs(cor(dta))
> dta.col <- dmat.color(dta.r)
> dta.o <- order.single(dta.r)
> cpairs(dta, dta.o, panel.colors=dta.col, gap=.5,
main="Variables Ordered and Colored by Correlation" )
```

```
> dta.r
```

```
      mpg      disp      drat      wt
mpg  1.0000000  0.8475514  0.6811719  0.8676594
disp  0.8475514  1.0000000  0.7102139  0.8879799
drat  0.6811719  0.7102139  1.0000000  0.7124406
wt    0.8676594  0.8879799  0.7124406  1.0000000
```

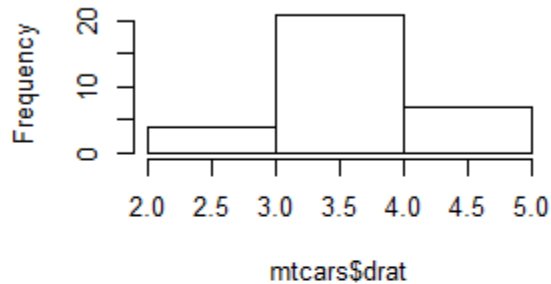
dmat.color: метод, который берет на вход матрицу с корреляциями, возвращает матрицу цветов

```
> dta.col
```

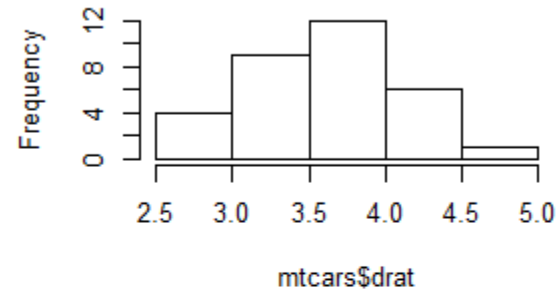
```
      mpg      disp      drat      wt
mpg  NA      "#D2F4F2" "#FDFFDA" "#F4BBDD"
disp "#D2F4F2" NA      "#FDFFDA" "#F4BBDD"
drat "#FDFFDA" "#FDFFDA" NA      "#D2F4F2"
wt   "#F4BBDD" "#F4BBDD" "#D2F4F2" NA
```

Гистограммы

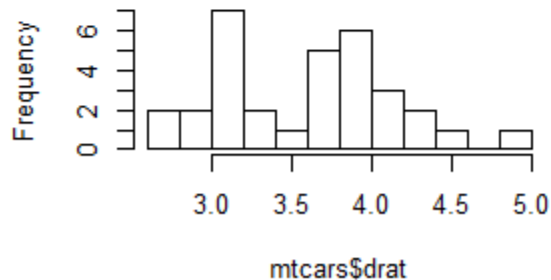
Histogram of mtcars\$drat



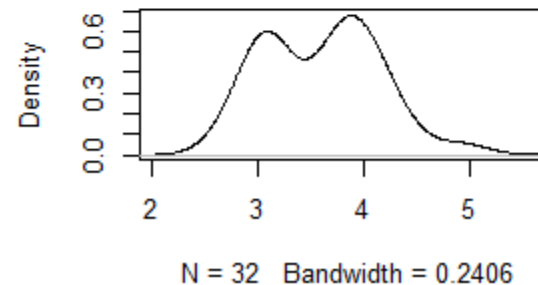
Histogram of mtcars\$drat



Histogram of mtcars\$drat

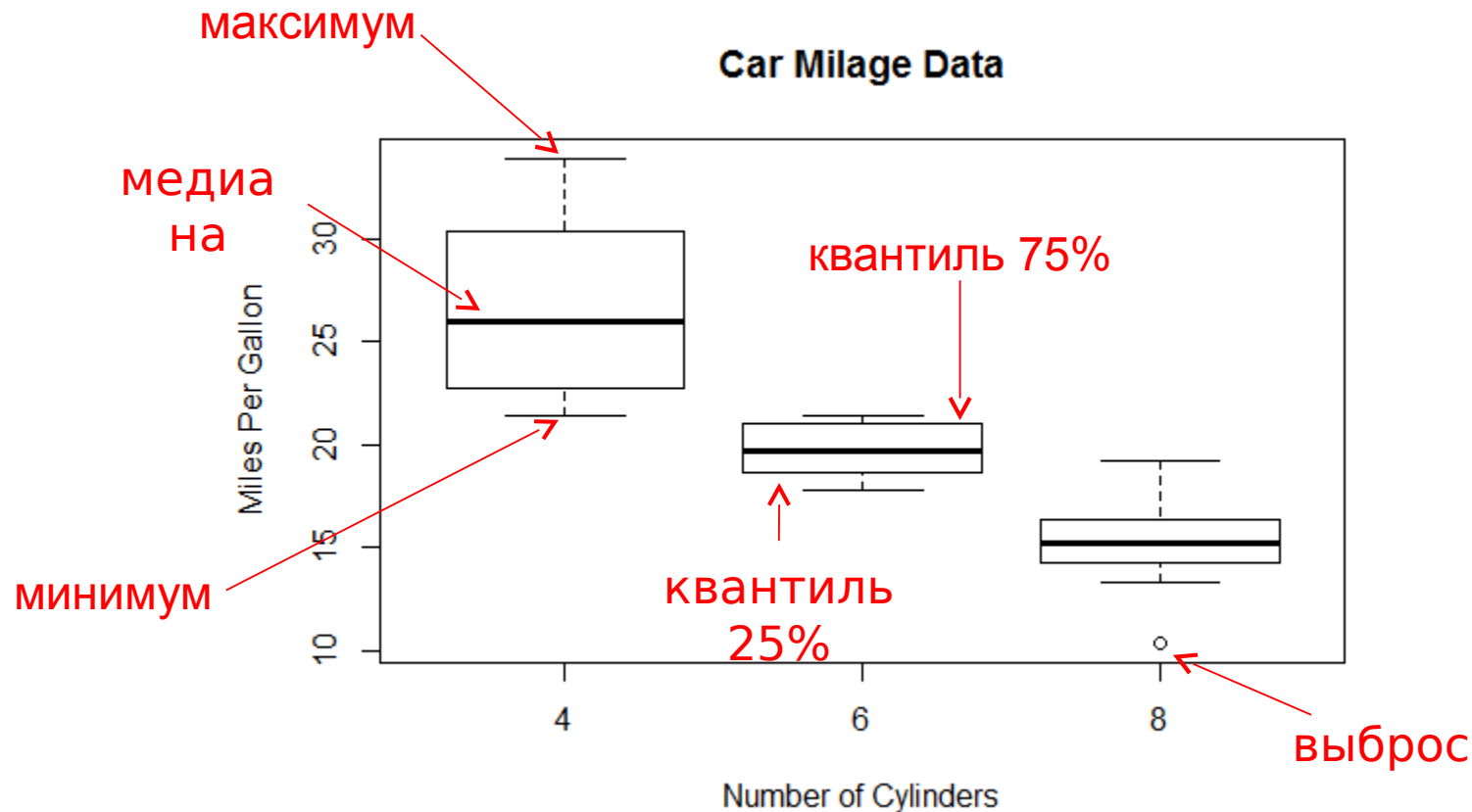


density.default(x = mtcars\$drat)



- > par(mfrow=c(2,2))
- > hist(mtcars\$drat, breaks=3)
- > hist(mtcars\$drat, breaks=5)
- > hist(mtcars\$drat, breaks=12)
- > plot(density(mtcars\$drat))

Boxplots

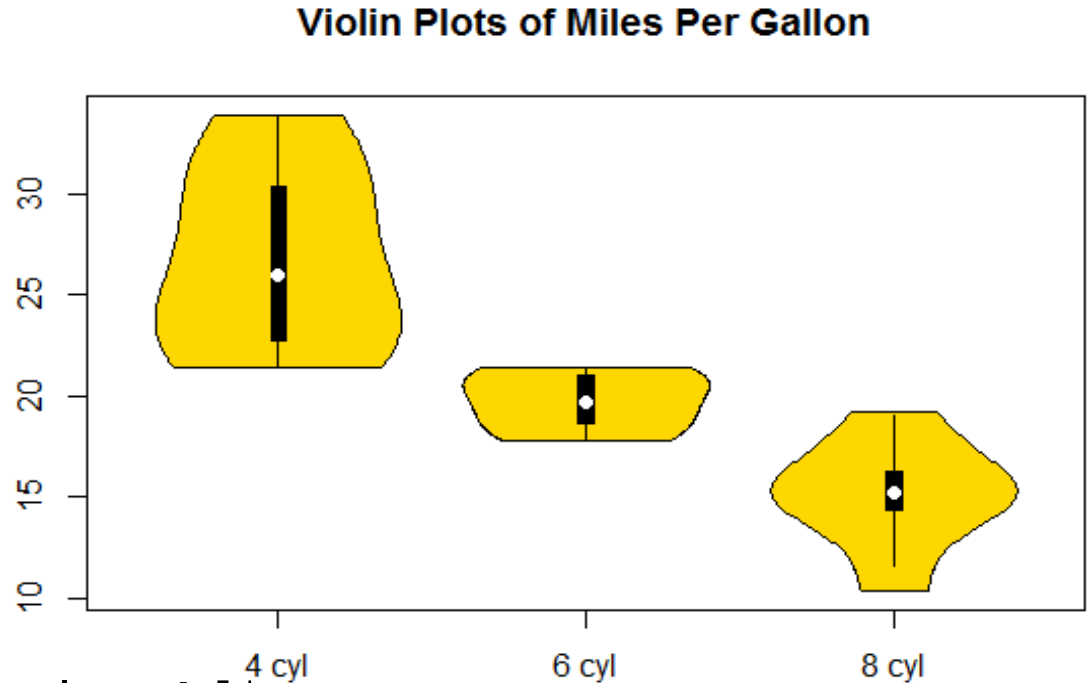


```
> boxplot(mpg~cyl,data=mtcars, main="Car Milage Data",  
xlab="Number of Cylinders", ylab="Miles Per Gallon")
```

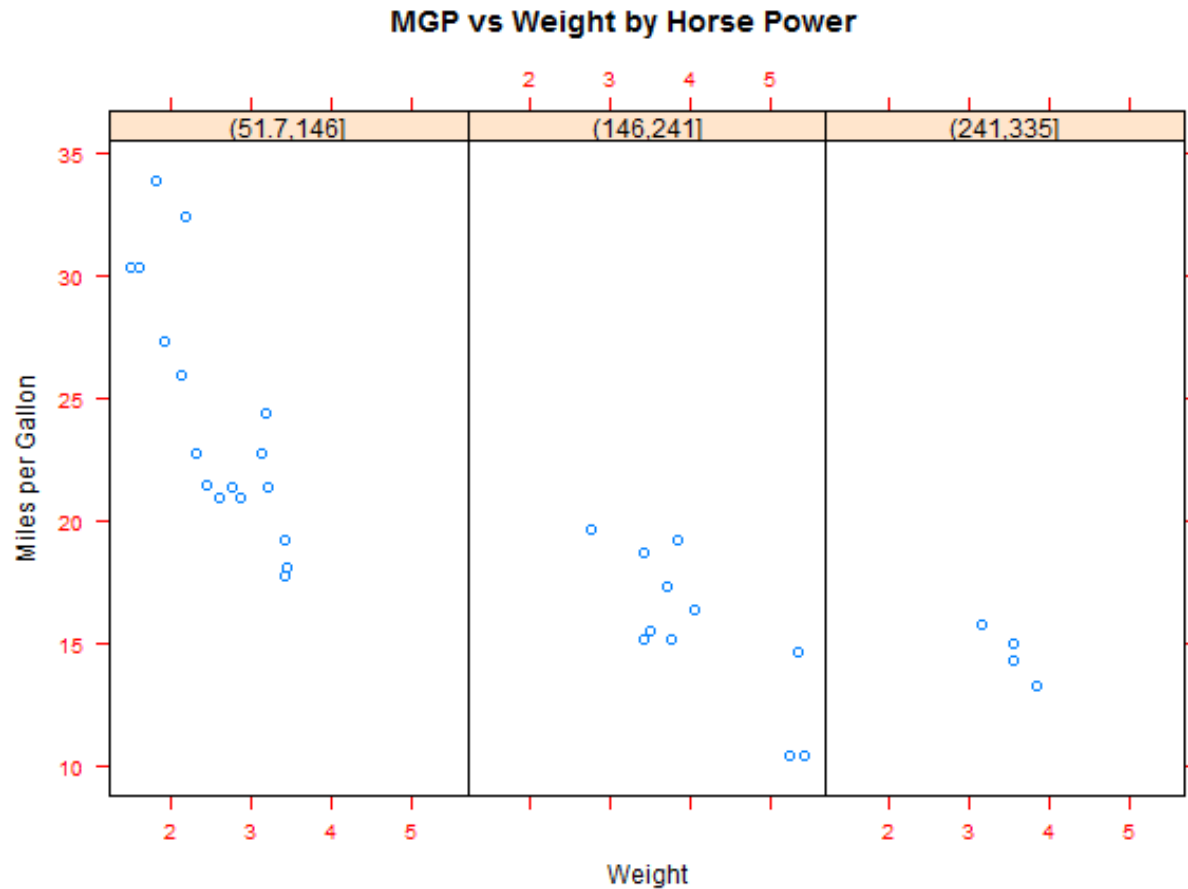
Violin Plot: комбинация boxplot и графика плотности распределения

«The violin plot is like the lovechild between a density plot and a box-and-whisker plot.»

```
> library(vioplot)
> x1 <- mtcars[mtcars$cyl==4,]$mpg
> x2 <- mtcars[mtcars$cyl==6,]$mpg
> x3 <- mtcars[mtcars$cyl==8,]$mpg
> vioplot(x1, x2, x3, names=c("4 cyl", "6 cyl", "8 cyl"),
col="gold")
title("Violin Plots of Miles Per Gallon")
```

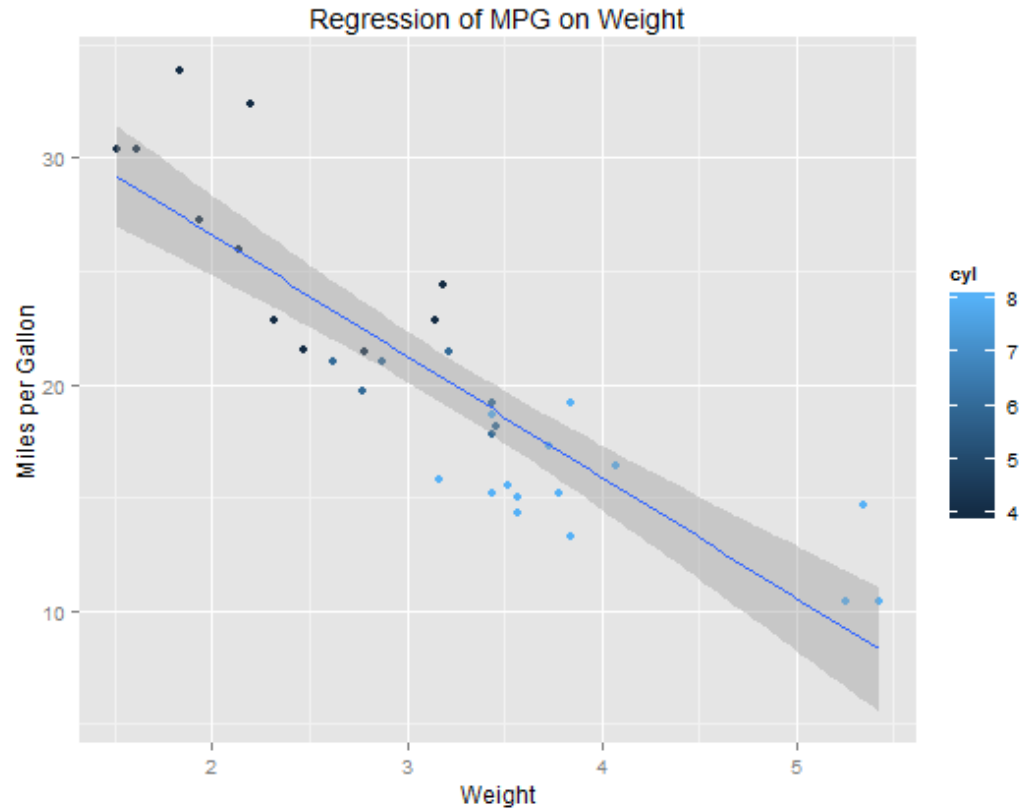


Возможности lattice



```
> library(lattice)
> hp <- cut(mtcars$hp,3) # divide horse power into three bands
> xyplot(mtcars$mpg~mtcars$wt|hp, scales=list(cex=.8, col="red"),
xlab="Weight", ylab="Miles per Gallon", main="MGP vs Weight by Horse
Power")
```

Возможности ggplot2



```
> qplot(wt, mpg, data=mtcars, geom=c("point", "smooth"),  
method="lm", formula=y~x, color=cyl, main="Regression of MPG on  
Weight", xlab="Weight", ylab="Miles per Gallon")
```

Больше графиков по ССЫЛКАМ

- <http://www.statmethods.net/advgraphs/>
- <http://gallery.r-enthusiasts.com/thumbs.php>

Работа с missing data 1/2

```
> newRow <- mtcars[1,]
```

```
> rownames(newRow) <- "Lada"
```

```
> newRow[4] <- NA
```

```
> mtcarsNew <- rbind(mtcars, newRow)
```

```
> mtcarsNew[30:33,]
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Ferrari Dino	19.7	6	145	175	3.62	2.77	15.50	0	1	5	6
Maserati Bora	15.0	8	301	335	3.54	3.57	14.60	0	1	5	8
Volvo 142E	21.4	4	121	109	4.11	2.78	18.60	1	1	4	2
Lada	21.0	6	160	NA	3.90	2.62	16.46	0	1	4	4

```
> mean(mtcarsNew$hp)
```

```
[1] NA
```

```
> any(is.na(mtcarsNew$hp))
```

```
[1] TRUE
```

Работа с missing data 2/2

```
> mean(mtcarsNew$hp, na.rm=TRUE)
[1] 146.6875
```

```
> which(is.na(mtcarsNew$hp))
[1] 33
```

```
! > which(c(FALSE, TRUE, FALSE, TRUE)) #как работает команда which
[1] 2 4
```

```
> mtcarsA <- na.omit(mtcarsNew) #или просто уберем все строки,
                               содержащие NA
```

```
> dim(mtcarsNew)
[1] 33 11
```

```
> dim(mtcarsA)
[1] 32 11
```

```
#проверим, изменилось ли число
                               строк
```

Что еще можно добавить на график

grid(nx, ny)

Add grid lines to current plot. NA stop grid in corresponding direction

axis(side n,)

Add axis at side n to current plot

box(which=,)

Add box around current plot, figure or outer margin area depending on which specified

legend

Add legend to current plot

arrows(x, y)

lines(x, y)

points(x, y)

Add arrow line, line or points to current plot. type = can be used to specify style ("p", "b", "l", etc)

abline(a, b)

abline(h= or v=)

Add line to current plot. a is intercept, b is slope.
h/v for horizontal/ vertical line

segments(x0, x1, y0, y1)

Add line segment(s) between pairs of points

polygon(x, y)

Add polygon defined by vectors x and y

text(x, y, "note")

Add text to current plot at x & y

Матрицы

- На вид трудно отличить от data frame
- «Двумерный вектор»
- Все элементы одного типа

Матрицы заполняются по колонкам:

```
> m <- matrix(c(1:6), nrow=2, ncol=3)
```

```
> m
```

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

Или по рядам (аргумент **byrow=T**)

```
> m <- matrix(c(1:6), nrow=2, ncol=3, byrow=T)
```

```
> m
```

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

```
> as.matrix(1:3)
```

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

Матрицы

- Обращение к элементам – как в data frame

```
> m[2,3]
```

```
[1] 6
```

```
> m[2,]
```

```
[1] 2 4 6
```

```
> m[,3]
```

```
[1] 5 6
```

```
> m[2,3] = 1
```

```
> m
```

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

```
[1,]  1   3   5
```

```
[2,]  2   4   6
```

Матрицы

```
> m <- matrix(nrow=2, ncol=3)
```

```
> m
```

```
      [,1] [,2] [,3]
[1,]  NA  NA  NA
[2,]  NA  NA  NA
```

```
> dim(m) ← Атрибут dim – размер матрицы
```

```
[1] 2 3
```

См. также `nrow`, `ncol`

```
> attributes(m)
```

```
$dim
```

```
[1] 2 3
```

Матрицы

Матрица – это вектор с атрибутом ***dim***

```
> v <- 1:6
```

```
> dim(v)      количество строк
```

```
NULL
```

```
> dim(v) <- c(2,3)  количество столбцов
```

```
> v
```

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

```
> is.matrix(v)
```

```
[1] TRUE
```

Матрицы

```
> a <- c(1:3)
```

```
> b <- c(11:13)
```

```
> cbind(a, b) # "column bind"
```

```
      a  b  
[1,] 1 11  
[2,] 2 12  
[3,] 3 13
```

```
> rbind(a, b) # "row bind"
```

```
  [,1] [,2] [,3]  
a     1     2     3  
b    11    12    13
```


Матрицы

Присваивание имен столбцам и колонкам:

```
> m <- matrix(c(1:6), nrow=2, ncol=3)
```

```
> m
```

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

```
> rownames(m) <- c("r1", "r2")
```

```
> colnames(m) <- c("c1", "c2", "c3")
```

```
> m
```

```
      c1 c2 c3
r1     1  3  5
r2     2  4  6
```

Списки

- Список, в отличие от вектора, может хранить элементы разного типа
 - В частности, элементом может быть список

```
> L = list("A", c(1,2), 30)
Сравните: > v=c(v, 4)
> L1 = list(L, 40)
#->list(list("A",c(1,2),30),40)
> L [[ 4 ]] = 4
> L [[ length( L ) + 1 ]] = 4
```

```
> L
[[1]]
[1] "A"
[[2]]
[1] 1 2
[[3]]
[1] 30
```

СПИСОК
из 3
ЭЛЕМЕНТОВ

Каждый элемент –
вектор (в R 1 элемент
– вектор из самого себя)

СПИСКИ

- Как достать элемент списка

```
> L [ 3 ] #-> список (из 1 элемента)
```

```
[[1]]
```

```
[1] 30
```

Правило: одинарные скобки []
возвращают элемент того же типа,
что и контейнер

```
> L [[ 3 ]]
```

```
#элемент (= вектор из 1 элемента)
```

```
[1] 30
```

- Как достать элемент из списка списков?

```
> L=list(10,list(11,12,13), 20, list(30))
```

```
#L[[2]] - тоже список
```

```
> L[[2]][[3]]
```

```
[1] 13
```

СПИСКИ

Сравните с выдачей любого теста, например t.test

■ Именованные элементы

```
> L=list(10,20)
```

```
> L$abc=123
```

```
> L
```

```
[[1]]
```

```
[1] 10
```

```
[[2]]
```

```
[1] 20
```

```
$abc
```

```
[1] 123
```

```
> names(L)
```

```
[1] "" "" "abc"
```

```
> L[[3]]
```

```
[1] 123
```

```
> L$abc=123
```

```
[1] 123
```

```
> L$abc
```

```
[1] 123
```

```
> L[["abc"]]
```

```
[1] 123
```

Повторение: векторы, списки, матрицы и data frames

	элементы одного типа	элементы разных типов
1D	<p>Вектор</p> <pre>>v = c(1,2,3) >v = c(v, 4) >v [2] #-> вектор (из 1 элемента)</pre>	<p>Список</p> <pre>> L = list(1,2,3) > L1 = list(L, 4) #-> list(list(1,2,3), 4) > L [[length(L) + 1]] = 4 > L [2] #-> список (из 1 элемента) > L [[2]] #элемент (=вектор из 1 элемента)</pre>
2D	<p>Матрица “2D вектор”</p> <pre>>m=matrix(c(1,2,3,4), nrow=2) >m[,1] #колонка матрицы - вектор >m[2,] #ряд матрицы - вектор</pre>	<p>data frame “список векторов (разных типов)”</p> <pre>>df=data.frame(name=c(“A”, “B”, “C”), grade=c(5,5,4)) >df[,1] #колонка d.f- вектор >df[1,] #ряд d.f- тоже data frame (как и d.f, является списком). Почему не вектор?</pre>
nD	<p>array “n-мерный вектор”</p>	