RUDN university

5100

Программирование в среде R

Шевцов Василий Викторович,
директор ДИТ РУДН, shevtsov_vv@rudn.university

mtcars

mpg	расход топлива (количества миль на галлон топлива)
cyl	кол-во цилиндров
disp	объем двигателя
hp	мощность двигателя (лошадиные силы)
drat	передаточное число заднего моста
wt	вес
qsec	значение времени разгона
vs	тип двигателя (v-образный, рядный)
am	тип коробки передач
gear	кол-во передач
carb	число карбюраторов

Функции высокого уровня

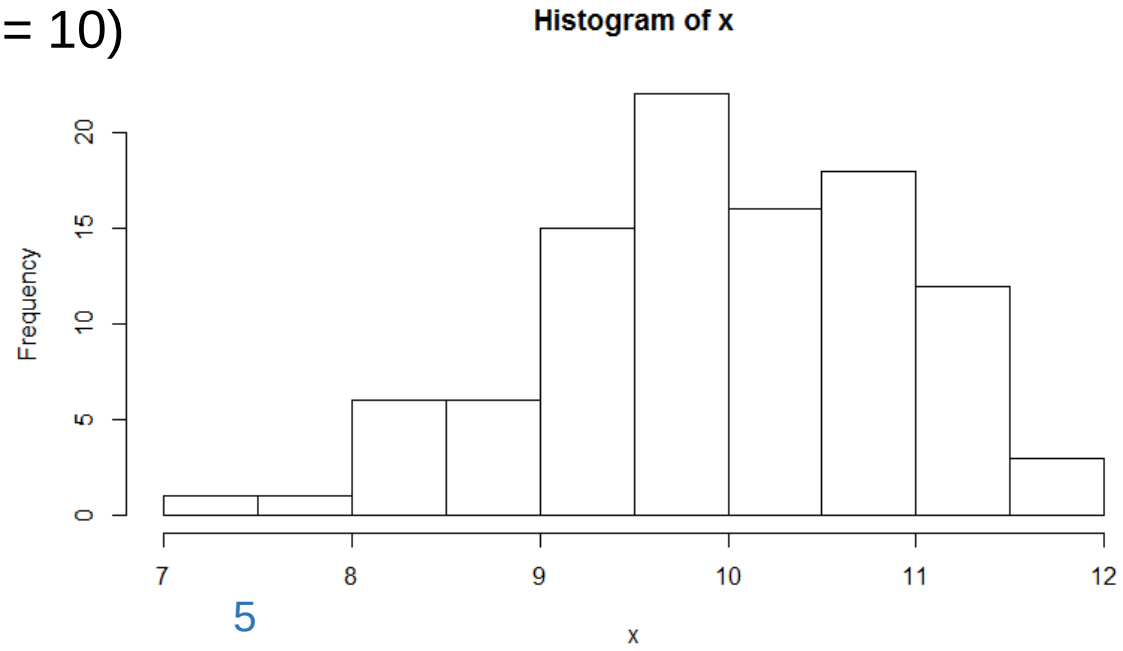
Основные функции высокого уровня

- par() — создание графического окна с заданными параметрами;
- barplot() — построение мозаичной диаграммы (статистика);
- contour() — построение графика с контурными линиями (линиями уровня);
- curve() — построение линии (кривой);
- dotchart() — точечные диаграммы (статистика);
- hist() — построение гистограммы (статистика);
- mosaicplot() — построение мозаичной диаграммы (статистика);
- persp() — построение трёхмерных графиков;
- pie() — построение круговой диаграммы;
- plot() — основная функция построения двумерных графиков;
- plot.data.frame() — графический анализ таблиц данных;
- plot.default() — построение диаграммы рассеивания ;
- plot.factor() — построение диаграммы рассеивания для факторов;
- plot.formula() — построение диаграммы рассеивания с помощью структуры formula;
- plot.histogram() — построение гистограммы (статистика);
- plot.table() — графический анализ таблицы данных (построение мозаичных диаграмм);
- screen — управление графическим окном;
- stem() — построение древесной диаграммы (статистика);

hist()

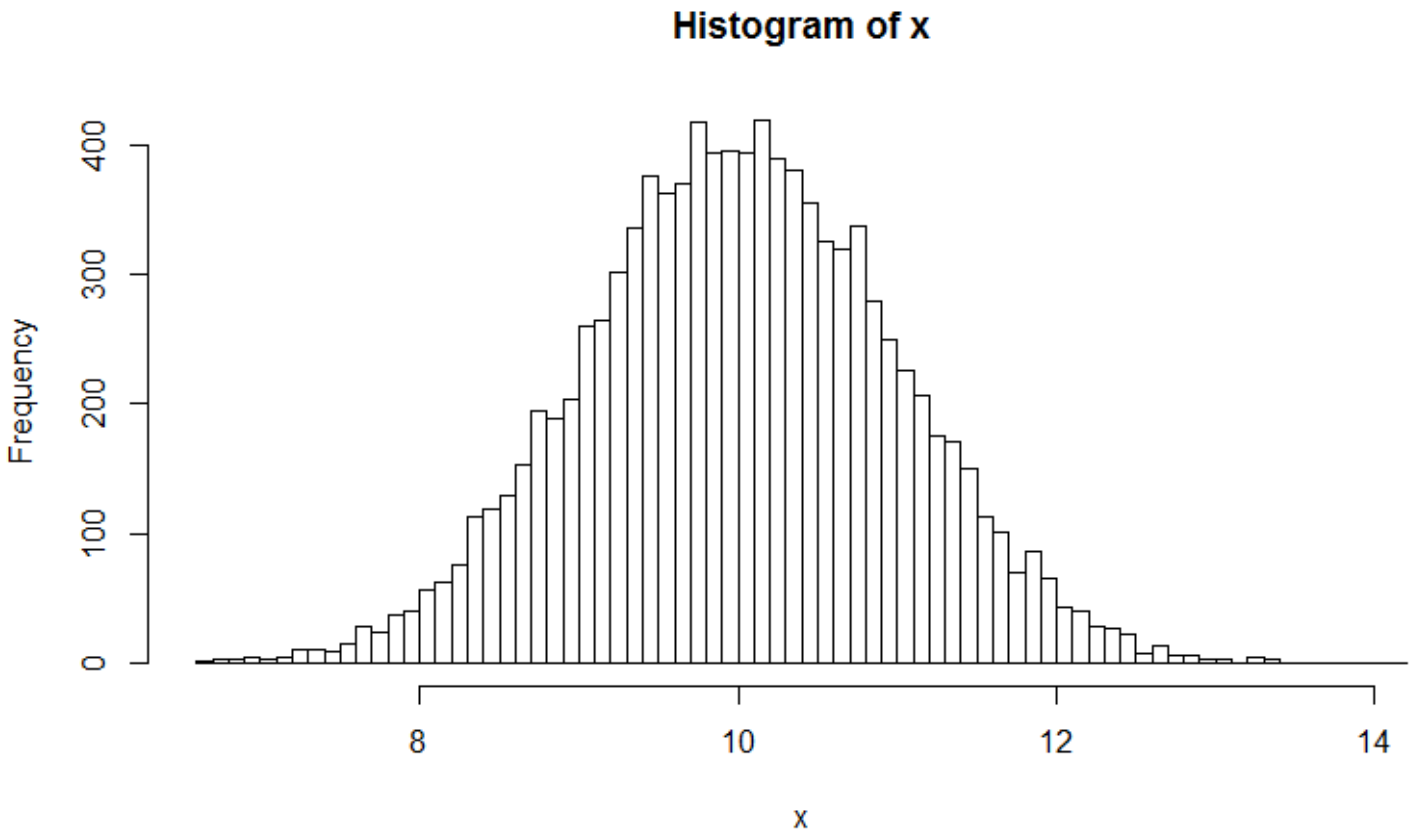
- Гистограмма является важным инструментом статистики, позволяющим наглядно представить распределение значений анализируемой переменной. В системе R для построения гистограмм служит функция hist(). Ее основным аргументом выступает имя анализируемой переменной. В качестве примера создадим нормально распределенную совокупность X из 100 наблюдений со средним значением 15 и стандартным отклонением 5:

```
x<-rnorm(n = 100, mean = 10)
hist(x)
```



hist()

```
x<-rnorm(n = 10000, mean = 10)
hist(x,breaks = 100)
```



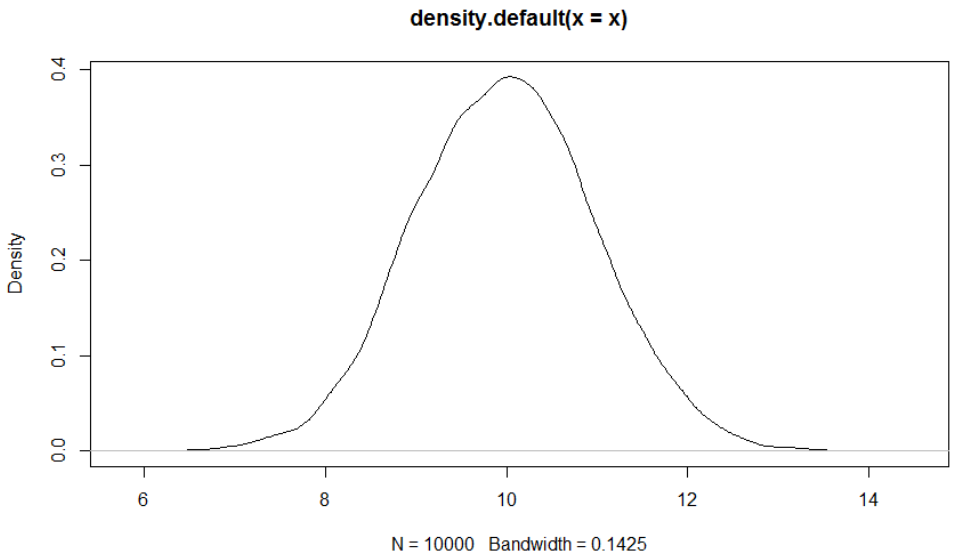
hist()

- По умолчанию функция hist() отображает по оси ординат частоты встречаемости для каждого класса значений X. Такое поведение функции можно изменить, придав аргументу freq (от frequency - частота) значение FALSE. В этом случае ось ординат будет отражать плотность вероятности каждого класса

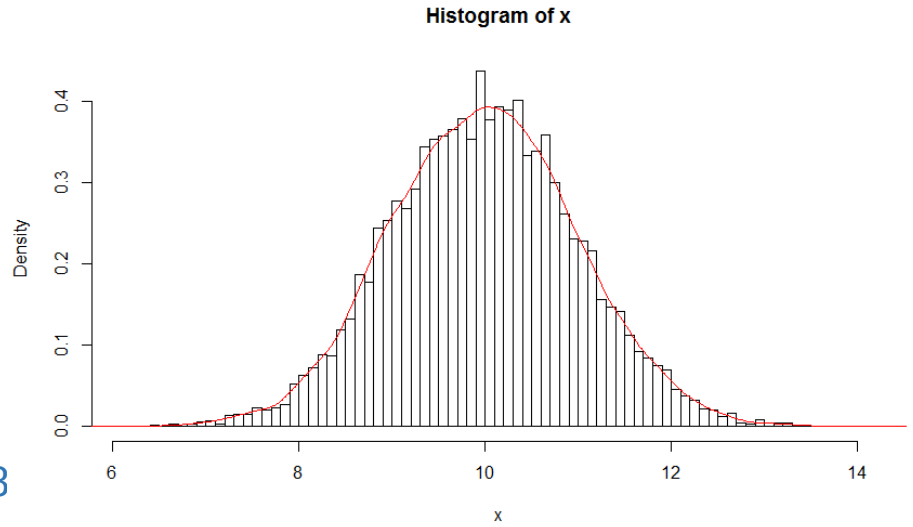
Кривая плотности вероятности

`density(x)` рассчитывает ядерные плотности вероятностей для значений x

```
plot(density(x))
```

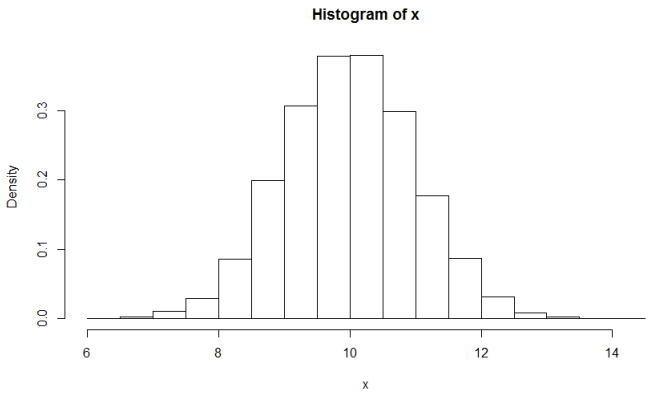



```
> hist(x,breaks = 100, freq=FALSE)
> lines(density(x),col="red")
```

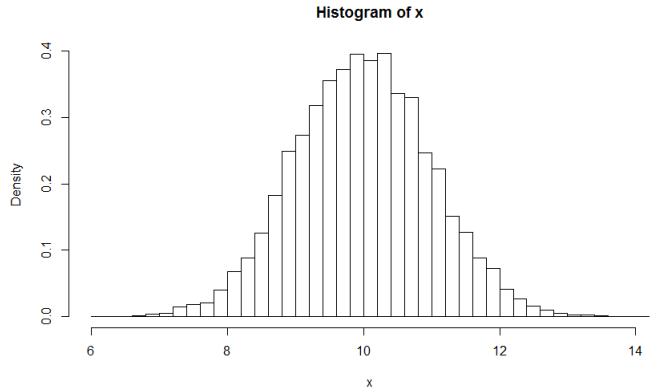


hist()

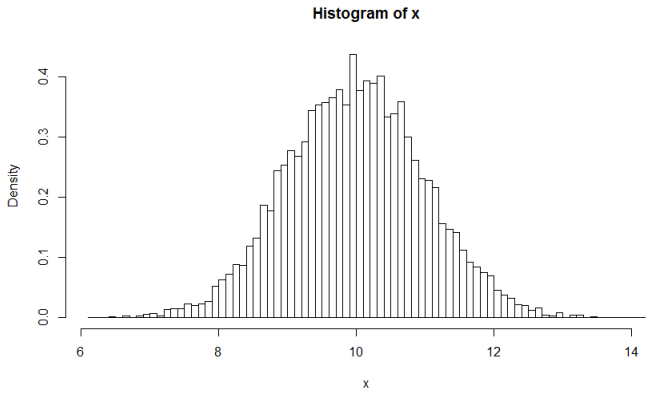
```
hist(x,freq=FALSE)
hist(x, breaks = "Sturges", freq=FALSE)
```




```
hist(x, breaks = "Scott", freq=FALSE)
```

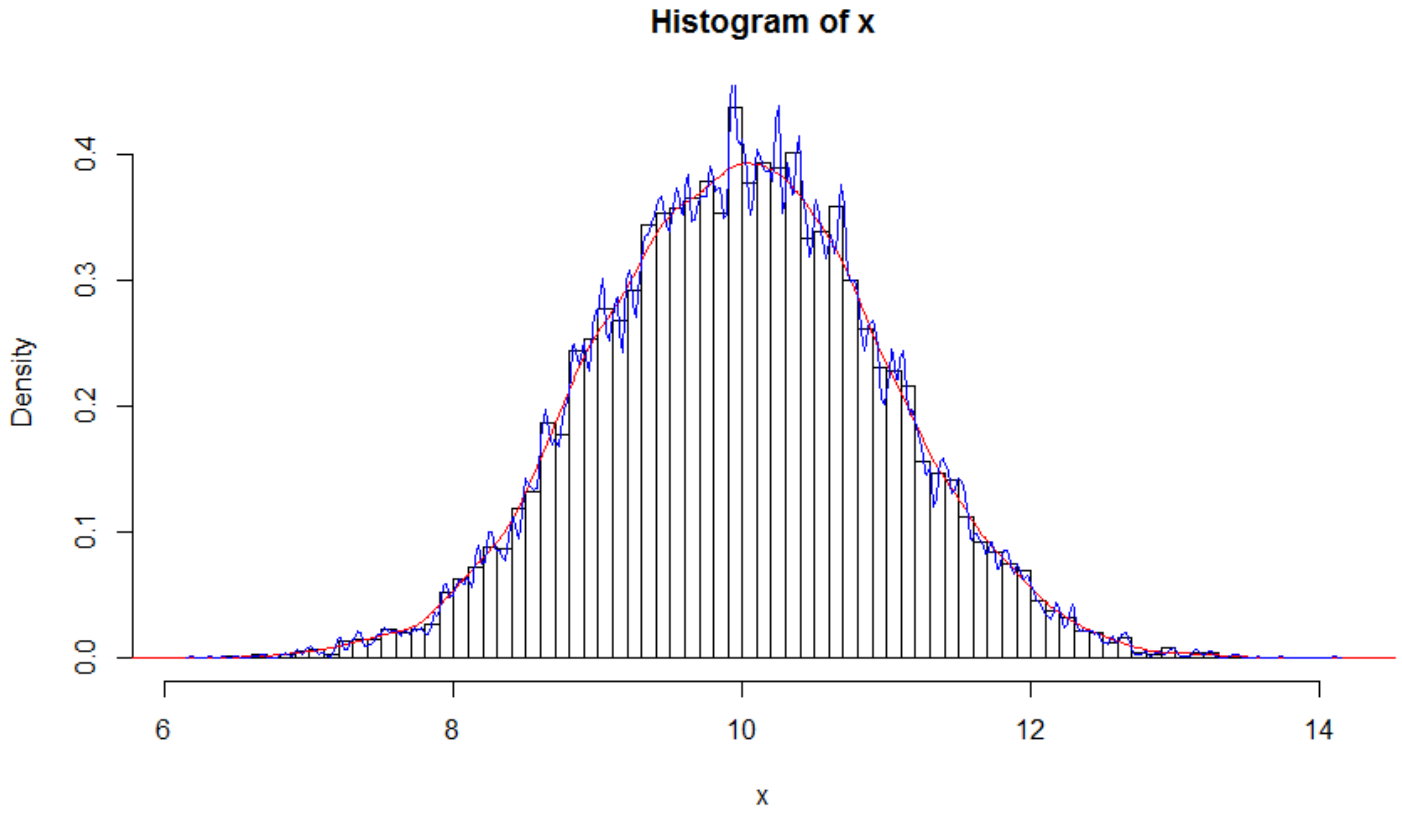



```
> #Freedman-Diaconis
> hist(x, breaks = "FD", freq=FALSE)
```

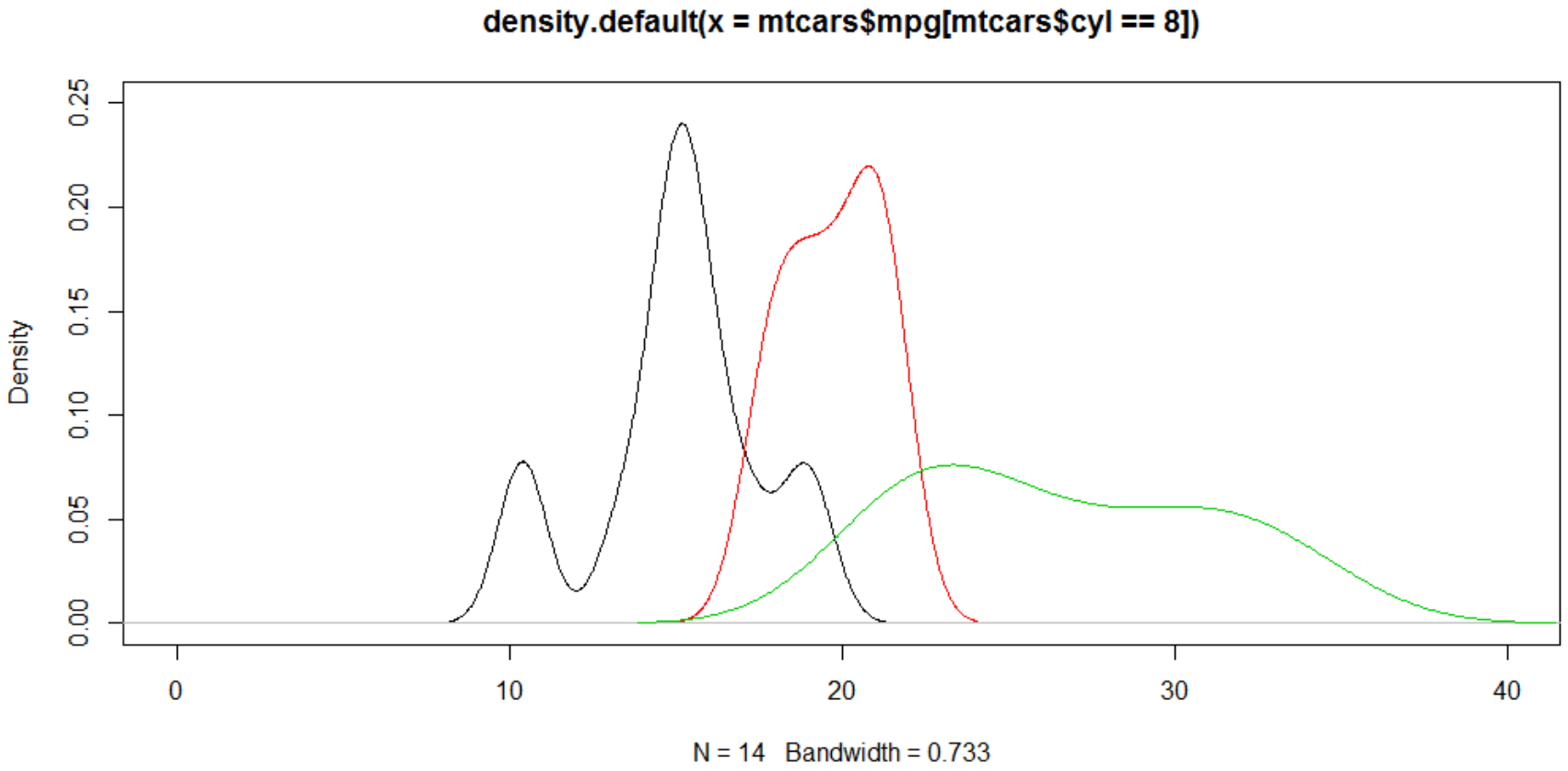


density(adjust = степень сглаживания графика плотности вероятностей

```
> hist(x,breaks = 100, freq=FALSE)
> lines(density(x),col="red")
> lines(density(x,adjust = 0.1),col="blue")
```

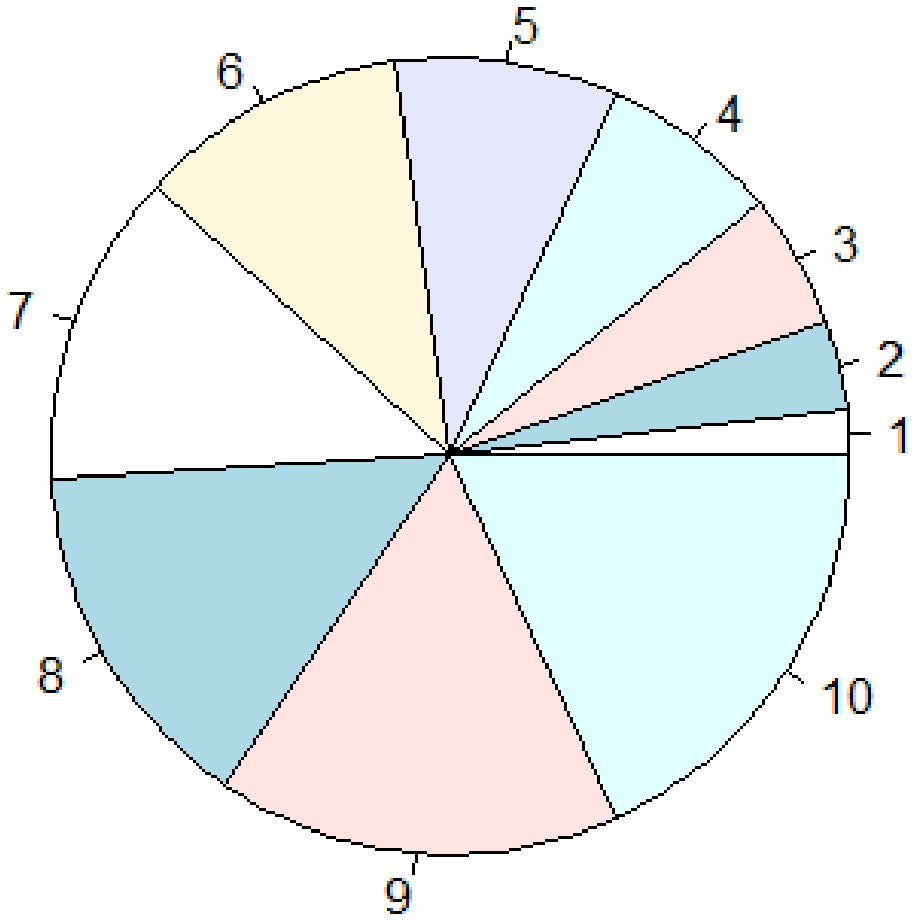


```
> plot(density(mtcars$mpg[mtcars$cyl==8]),xlim=c(0,40),
ylim=c(0,0.25),col=1)

> lines(density(mtcars$mpg[mtcars$cyl==6]),xlim=c(0,40),
ylim=c(0,0.25),col=2)

> lines(density(mtcars$mpg[mtcars$cyl==4]),xlim=c(0,40),
ylim=c(0,0.25),col=3)
```



pie() – круговая диаграмма

```
pie(1:10)
```



barplot()

- Для создания столбиковых (= "*столбчатых*", реже "*линейчатых*"; англ. *bar plots* или *bar charts*) диаграмм в системе R служит функция barplot().
- hight ("высота") - числовой вектор или матрица со значениями, используемыми для построения диаграммы. Если аргумент hight указан в виде вектора, то строится график из последовательно расположенных столбцов, высоты которых соответствуют значениям этого вектора. Если hight указан в виде матрицы и аргумент beside = FALSE, то будет построена столбчая *диаграмма с накоплением*. Если же hight указан в виде матрицы и аргумент beside = TRUE, то столбцы диаграммы будут *сгруппированы* в соответствии со столбцами матрицы.
- width ("ширина") - необязательный параметр, позволяющий регулировать ширину столбцов на диаграмме. Указывается в виде числового вектора, значения которого соответствуют ширине столбцов.

barplot()

- space ("пространство") - величина зазора между столбцами (пропорционально их средней ширине). Может быть указан либо виде одного числа, либо в виде вектора из чисел, соотвествующих каждому столбцу диаграммы.
- names.arg - текстовый вектор, содержащий подписи (вдоль оси OX) для каждого столбца или группы столбцов. Если этот аргумент не указан, в качестве подписей автоматически будут использованы имена элементов вектора height (если таковые имеются), либо заголовки столбцов если height представляет собой матрицу.
- legend.text - вектор, содержащий текстовые элементы легенды графика. Этот аргумент полезен только если height является матрицей. В этом случае метки легенды должны соотвествовать строкам матрицы. Аргументу legend.text можно также присвоить значение TRUE, и тогда имена строк матрицы (если таковые имеются) будут использованы в качестве меток легенды автоматически.

barplot()

- beside - принимает логическое значение и имеет смысл только, если height является матрицей. Значение FALSE приведет к построению диаграммы с накоплением. При значении TRUE столбцы будут сгруппированы.
- horiz - принимает логическое значение: TRUE для горизонтального расположения столбцов и FALSE - для вертикального.
- density - числовой вектор, задающий плотность заштриховки столбцов.
- angle - угол наклона штрихов (в градусах).
- col - вектор цветовых кодов для столбцов или их элементов. По умолчанию столбцы закрашиваются серым цветом если height - вектор, и разными градациями серого если height - матрица.
- border - код цвета для обводки столбцов. Если границу столбцов обводить не предполагается, можно указать border = NA.

InsectSprays

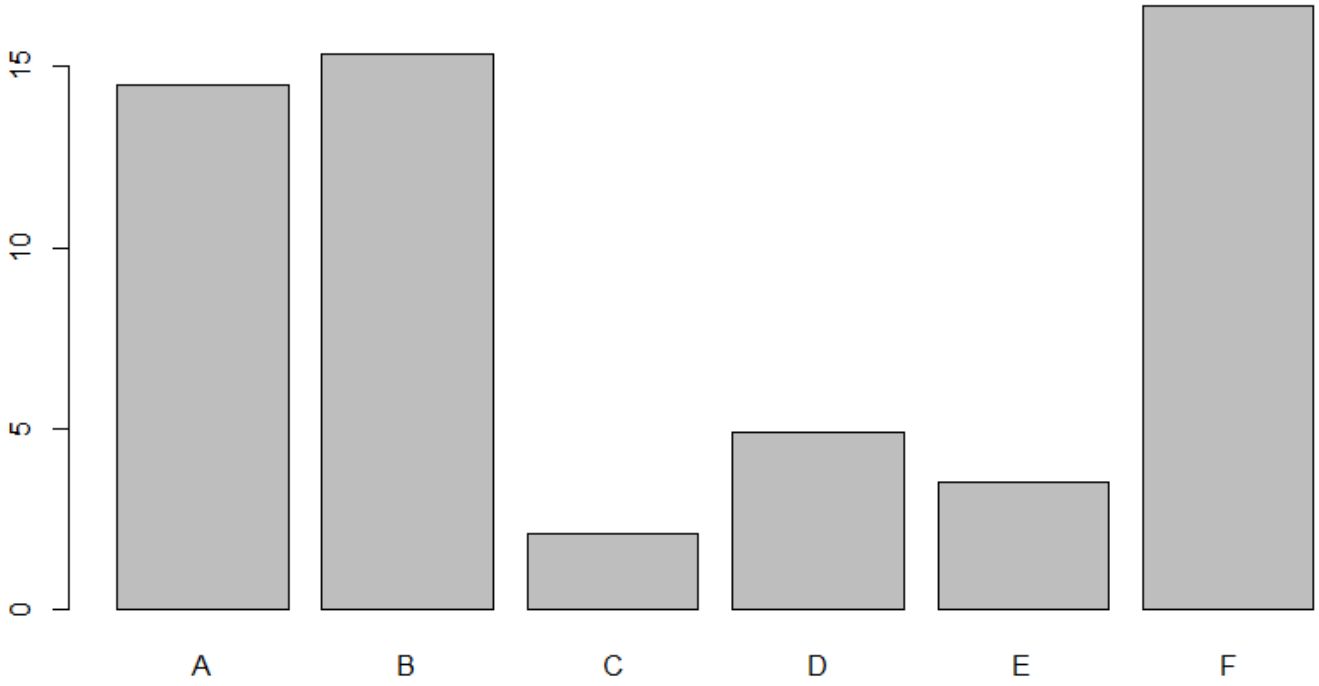
Данные по эффективности шести инсектицидных средств (A - F)

```
> is.factor(InsectSprays$spray)
[1] TRUE
```

```
> InsectSprays
   count spray
1     10     A
2      7     A
3     20     A
4     14     A
5     14     A
6     12     A
7     10     A
8     23     A
9     17     A
10    20     A
11    14     A
12    13     A
13    11     B
14    17     B
15    21     B
16    11     B
17    16     B
18    14     B
19    17     B
20    17     B
21    19     B
22    21     B
23     7     B
24    13     B
25     0     C
26     1     C
```

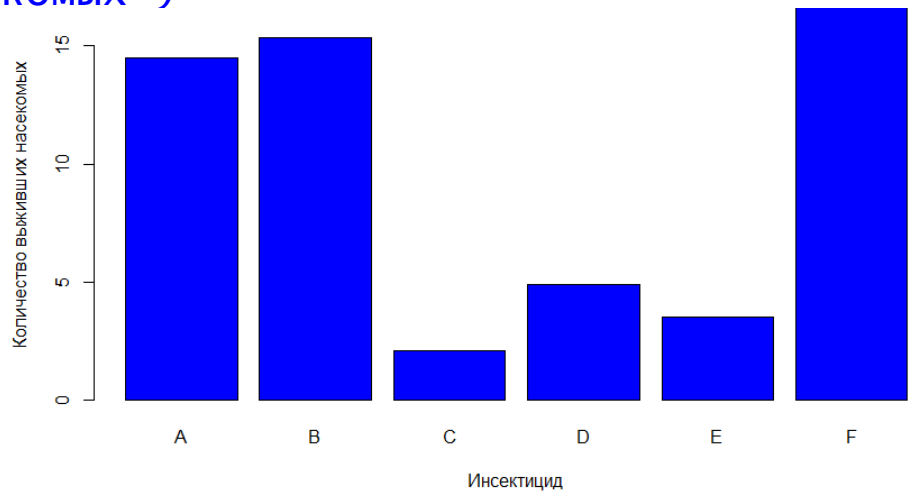
barplot()

```
> t_mean <- tapply(InsectSprays$count, InsectSprays$spray, mean)
> barplot(height = t_mean)
```

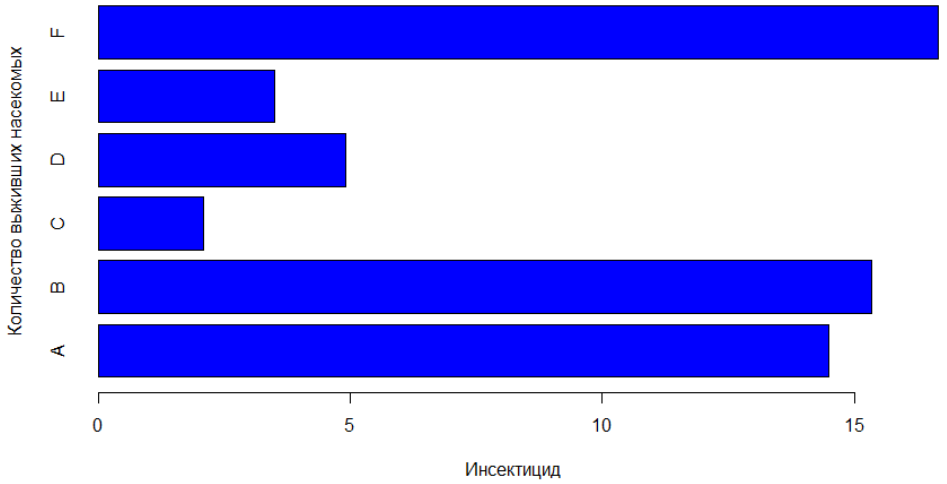


barplot()

```
barplot(t_mean, col = "blue", xlab = "Инсектицид",
ylab = "Количество выживших насекомых")
```

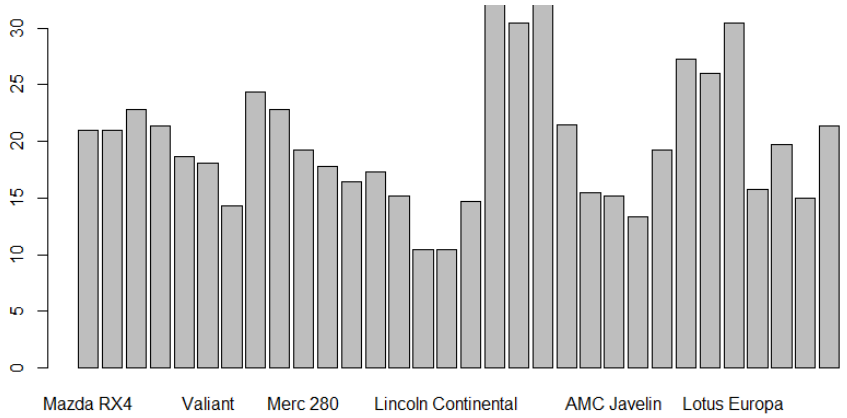



```
barplot(t_mean, col = "blue", xlab = "Инсектицид",
ylab = "Количество выживших насекомых", horiz = TRUE)
```

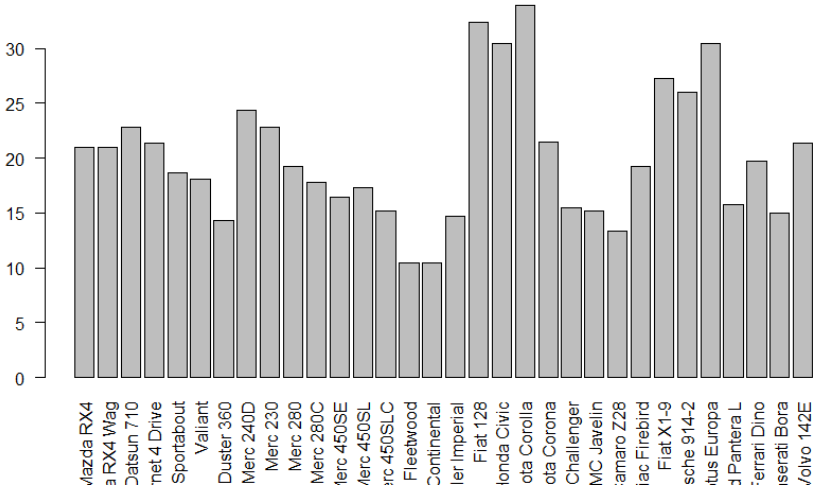


barplot()

```
barplot(mtcars$mpg,names.arg=row.names(mtcars))
```



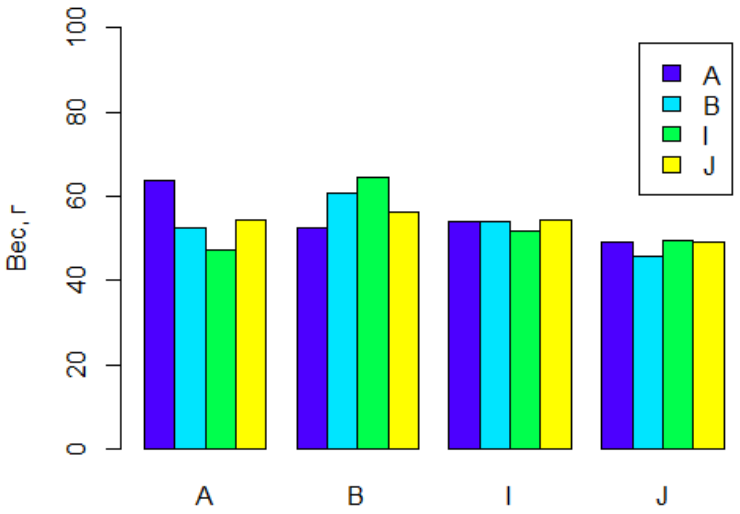

```
barplot(mtcars$mpg,names.arg=row.names(mtcars), las=2)
```



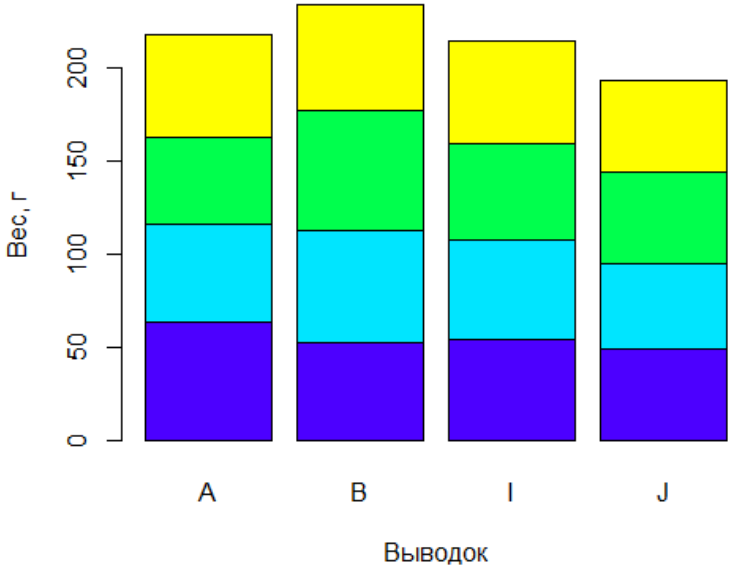
barplot(), beside

```
means
     A     B     I     J
A 63.68 52.40 54.12 48.96
B 52.33 60.64 53.92 45.90
I 47.10 64.37 51.60 49.43
J 54.35 56.10 54.53 49.06
```

```
barplot(means, beside = TRUE,…
```

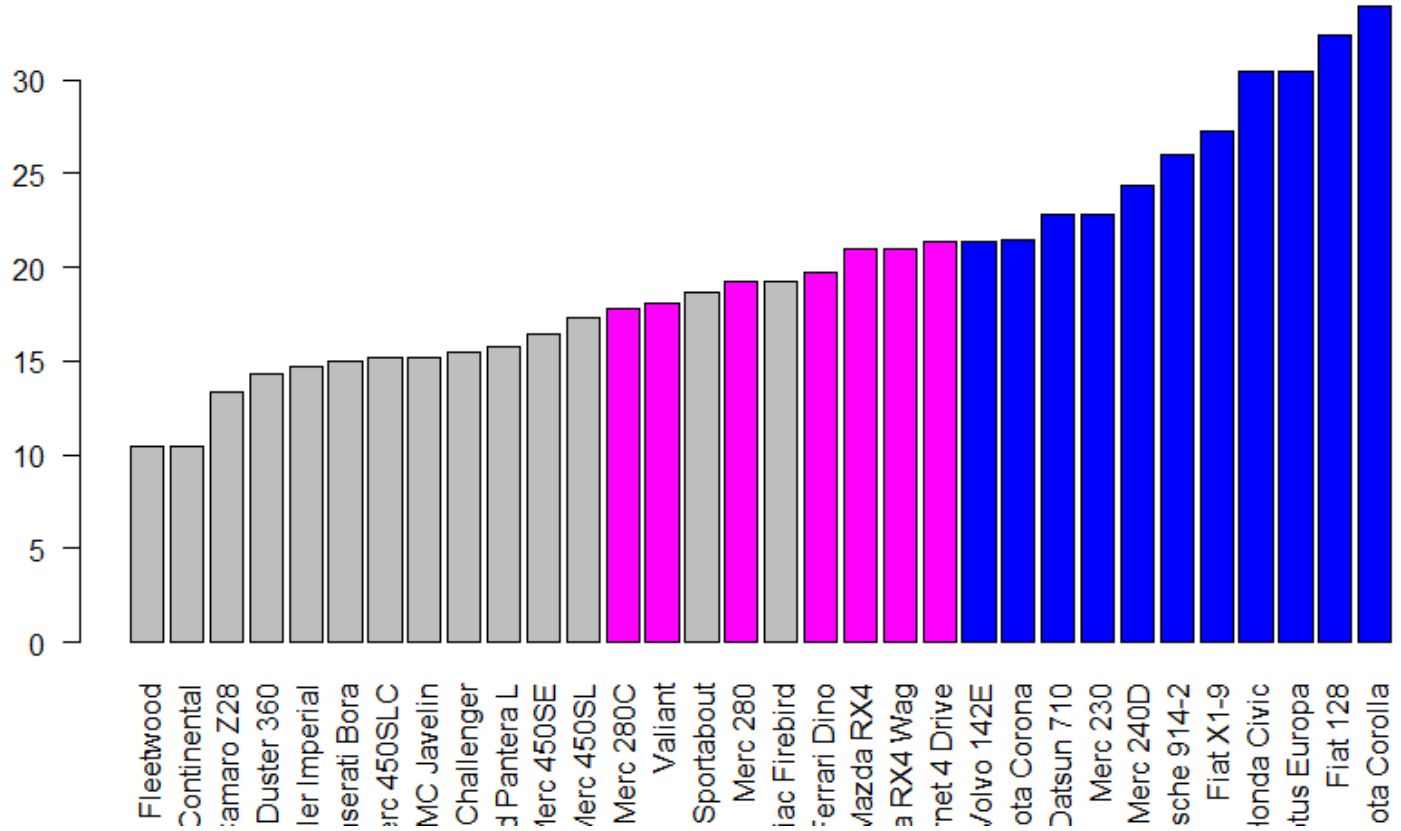



```
barplot(means, beside = FALSE,
```



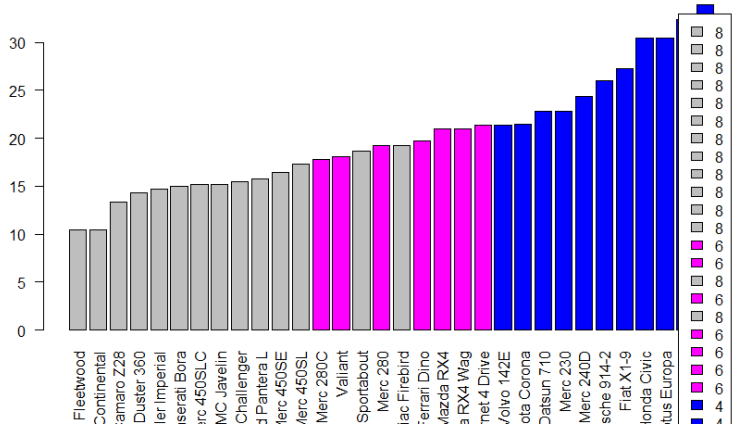
barplot()

```
> t1 <- mtcars[order(mtcars$mpg),]
> barplot(t1$mpg,names.arg=row.names(t1), las=2, col=t1$cyl)
```

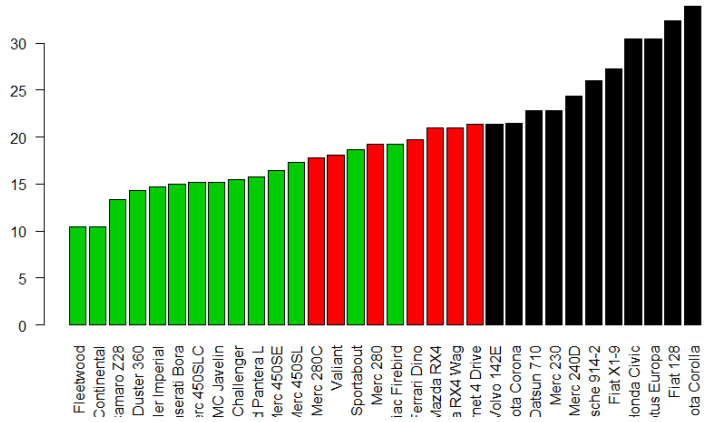


RUDN university

```
barplot(t1$mpg,names.arg=row.names(t1), las=2, col=t1$cyl, legend.text = t1$cyl)
```



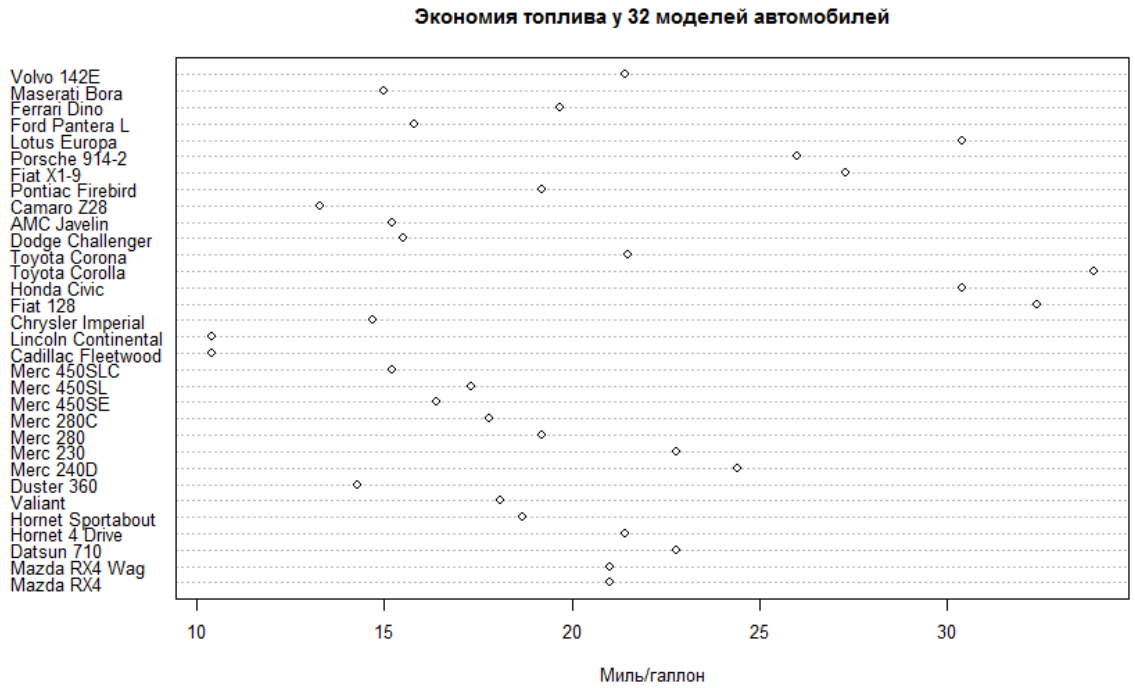

```
barplot(t1$mpg,names.arg=row.names(t1), las=2, col=t1$cyl)
```



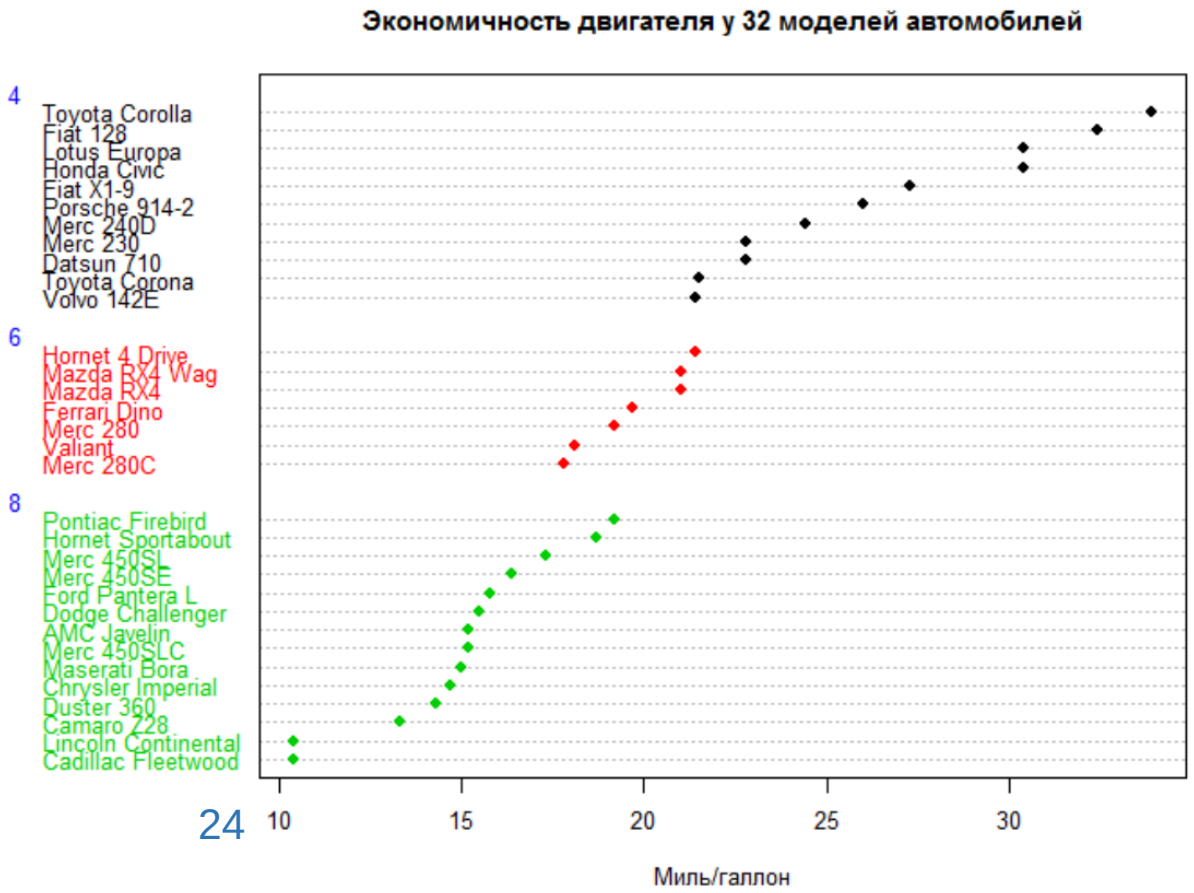
dotchart()

- Точечные диаграммы Кливленда представляют собой графики, на которых точки-маркеры используются для отображения значений некоторой количественной переменной (или переменных), разбитых на группы в соответствии с уровнями некоторой номинальной переменной (или переменных).

```
dotchart(mtcars$mpg, labels = row.names(mtcars),
main="Экономия топлива у 32 моделей автомобилей", xlab="Миль/галлон", cex = 0.8)
```



```
> t1 <- mtcars[order(mtcars$mpg), ]
> t1$cyl <- factor(t1$cyl)
> t1$color[t1$cyl==4] <- 1
> t1$color[t1$cyl==6] <- 2
> t1$color[t1$cyl==8] <- 3
> dotchart(t1$mpg, labels = row.names(t1), groups = t1$cyl,
gcolor = "blue", pch = 16, color = t1$color,
main="Экономичность двигателя у 32 моделей автомобилей",
xlab="Миль/галлон", cex = 0.8)
```



dotchart()

- dotchart(x, labels = NULL, groups = NULL, gdata = NULL, cex = par("cex"), pt.cex = cex, pch = 21, gpch = 21, bg = par("bg"), color = par("fg"), gcolor = par("fg"), lcolor = "gray", xlim = range(x[is.finite(x)]), main = NULL, xlab = NULL, ylab = NULL, …)
- x - either a vector or matrix of numeric values (NAs are allowed). If x is a matrix the overall plot consists of juxtaposed dotplots for each row. Inputs which satisfy is.numeric(x) but not is.vector(x) || is.matrix(x) are coerced by as.numeric, with a warning.
- labels - a vector of labels for each point. For vectors the default is to use names(x) and for matrices the row labels dimnames(x)[[1]].
- groups - an optional factor indicating how the elements of x are grouped. If x is a matrix, groups will default to the columns of x.

dotchart()

- gdata - data values for the groups. This is typically a summary such as the median or mean of each group.
- cex - the character size to be used. Setting cex to a value smaller than one can be a useful way of avoiding label overlap. Unlike many other graphics functions, this sets the actual size, not a multiple of par("cex").
- pt.cex - the cex to be applied to plotting symbols. This behaves like cex in plot().
- pch - the plotting character or symbol to be used.
- gpch - the plotting character or symbol to be used for group values.
- bg - the background color of plotting characters or symbols to be used; use par(bg= *) to set the background color of the whole plot.

dotchart()

- color - the color(s) to be used for points and labels.
- gcolor - the single color to be used for group labels and values.
- lcolor - the color(s) to be used for the horizontal lines.
- xlim - horizontal range for the plot, see plot.window, for example.
- main - overall title for the plot, see title.
- xlab, ylab - axis annotations as in title.

- graphical parameters can also be specified as arguments.

Функции низкого уровня

Функции низкого уровня

- abline() — построение прямых линий в уже существующем графическом окне;
- arrows() — рисование стрелок;
- axis() — построение оси графика;
- box() — построение рамки вокруг графика;
- grid() — задание прямоугольной сетки на графике;
- legend() — задание различных легенд к графику;
- lines() — построение линий, соединяющих заданные точки;
- mtext() — вывод надписей в соответствующей области;
- points() — добавление точек на график;

Функции низкого уровня

- polygon() — построение многоугольников;
- rect() — построение прямоугольников;
- segments() — соединение точек прямыми отрезками;
- symbols() — построение одного из шести видов фигур (круг, квадрат, прямоугольник, звезда, термометр, boxplot) на графике;
- text() — добавление текста к графику;
- title() — добавление заголовков;
- xspline() — построение сплайна относительно заданных контрольных точек.

Функция abline()

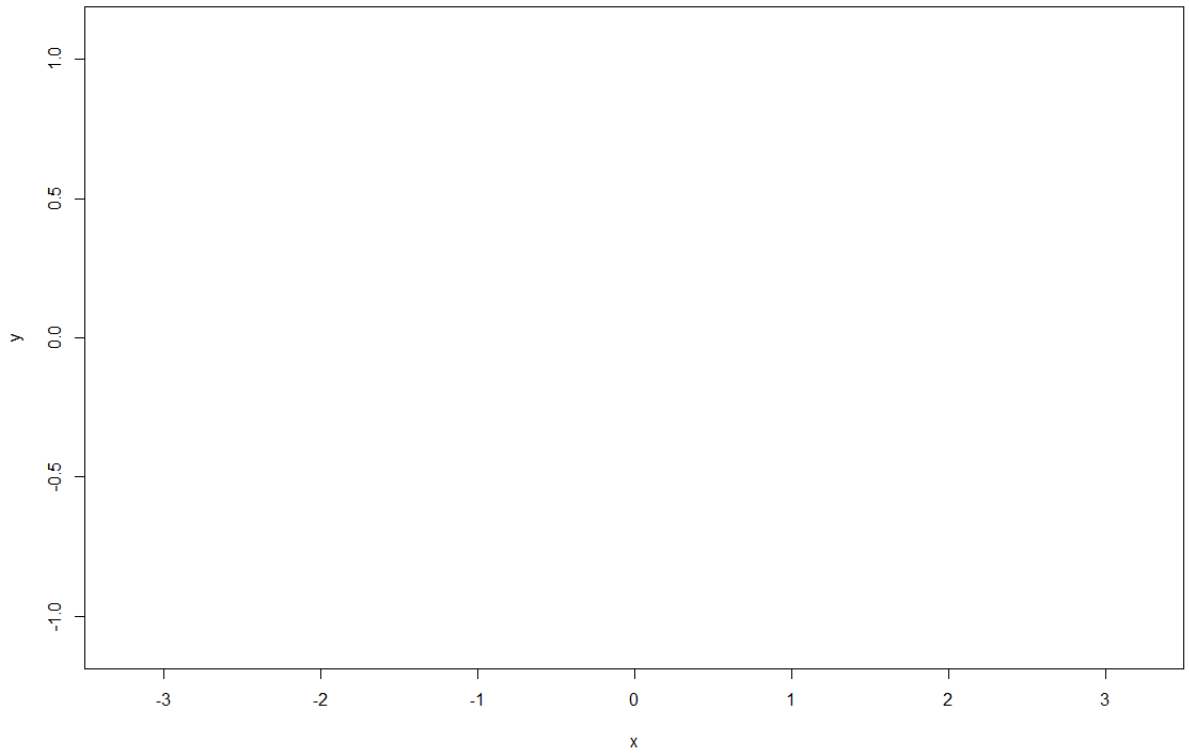
- abline(a = NULL, b = NULL, h = NULL, v = NULL, reg = NULL, coef = NULL, untf = FALSE, ...)
- добавляет одну или несколько прямых линий вида y = a + bx к созданному ранее графику.
- Аргументы функции:
- a и b — числовые аргументы — точка пересечения с осью y и коэффициент наклона, соответственно.
- h — числовой аргумент — значение по оси y для построения горизонтальной линии.
- v — числовой аргумент — значение по оси x для построения вертикальной линии.
- coef — числовой вектор вида c(a,b) — задаёт аргументы a и b.

Функция abline()

- reg — числовой аргумент — определяет вид прямой. Если это – числовой вектор длины 1 — задан коэффициент наклона b прямой y = bx, проходящей через начало координат;
- числовой вектор длины 2 — заданы аргументы a и b.
- untf — логический аргумент — отвечает за приведение строящегося графика в соответствие с заданной системой координат. Если untf=TRUE, а одна из осей (или обе) были переведены в логарифмический формат (аргументы xlog и ylog), то прямая всё равно строится в исходной (не преобразованной) системе координат, в противном случае прямая строится согласно модифицированным координатам.
- col — цвет линии.
- lty — тип линии.
- lwd — ширина линии.

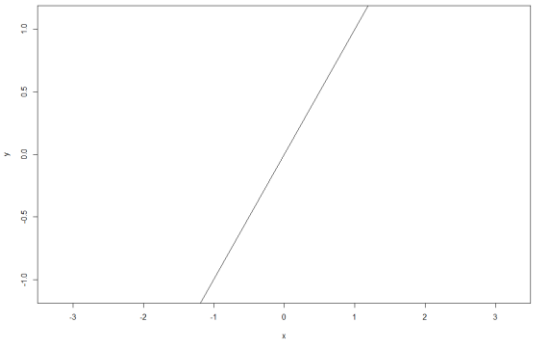
abline()

```
> x=seq(-pi,pi,by=0.01)
> y <- cos(x)
> plot(x,y, type="n",xlab="x", ylab="y",
  xlim=c(-pi-0.1,pi+0.1), ylim=c(-1.1,1.1))
```

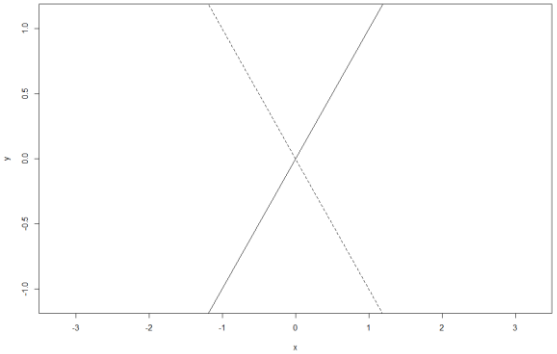


abline()

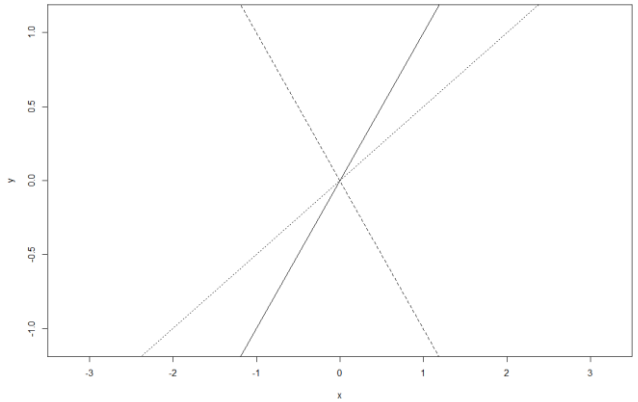
```
> abline(a=0,b=1,lty=1)
```



```
> abline(a=0,b=-1,lty=1)
```

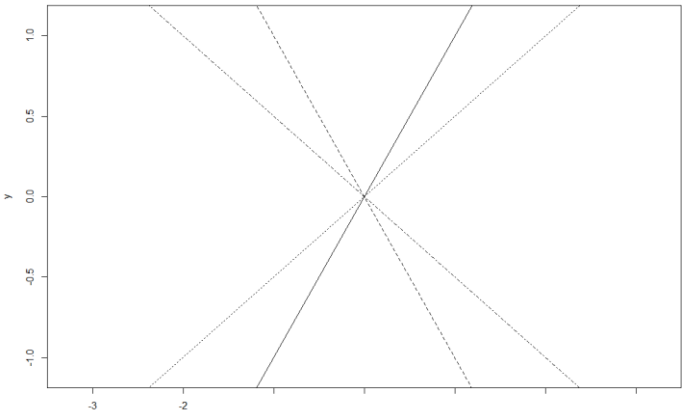


```
> abline(coef=c(0,0.5),lty=3)
```

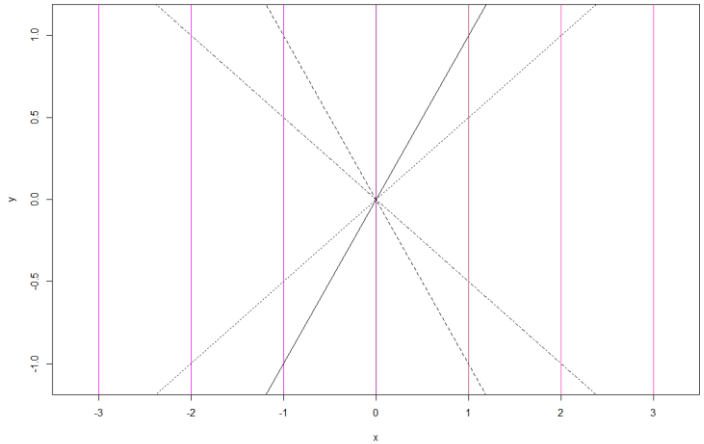


abline()

```
> abline(reg=c(0,-0.5),lty=4)
```

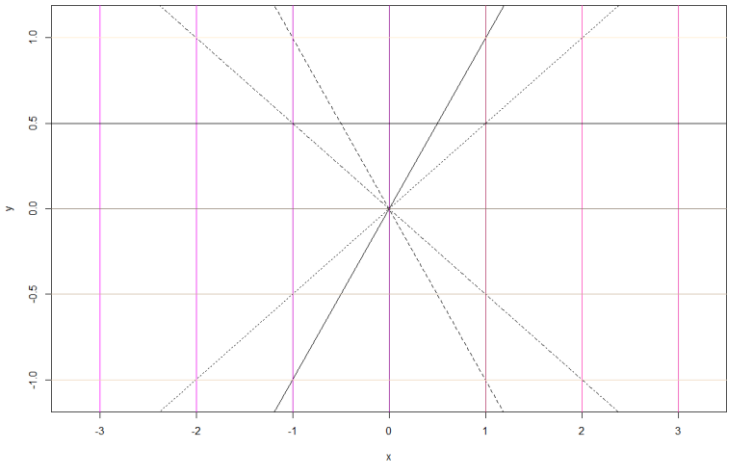



```
> abline(v = -3:3, col=colors()[451:457])
```

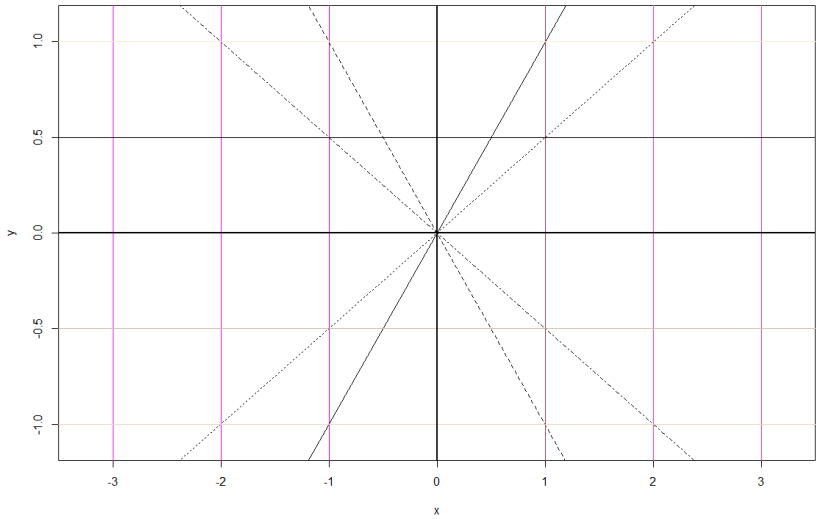


abline()

```
> abline(h=seq(-1,1,by=0.5), col=colors()[21:25])
```




```
> abline(h=0,v=0,lwd=2,col="black")
```

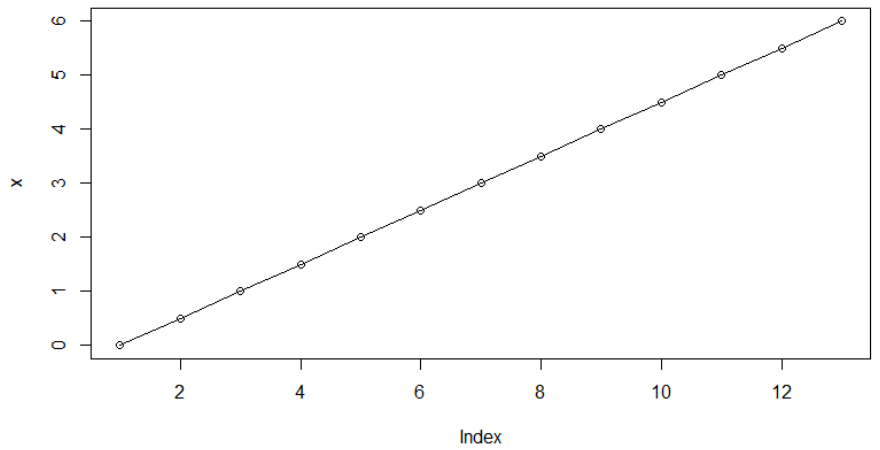


Функция lines()

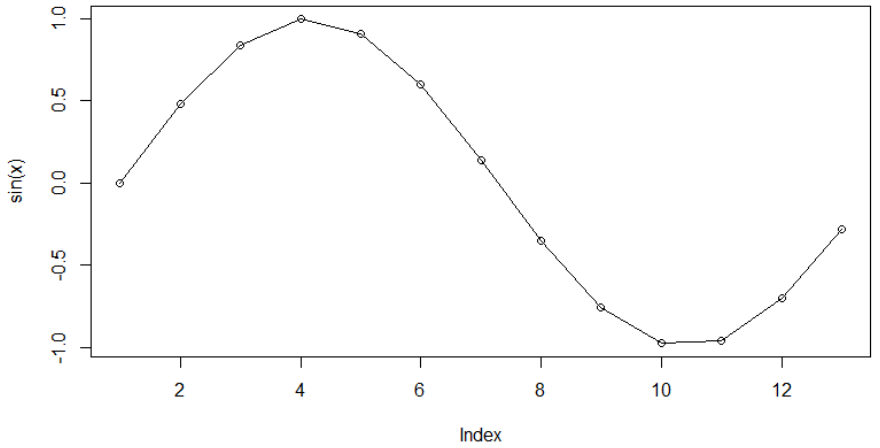
- Функция lines() соединяет заданные точки отрезками. Её вид:
- lines(x, y = NULL, type = "l", ...)
- Единственным аргументом функции является аргумент x — список из двух компонент (первая компонента — координаты по оси 0x, вторая — координаты по оси 0y) или матрица из двух столбцов (первый столбец — координаты по оси 0x, второй — координаты по оси 0y).
- Если аргумент x — числовой вектор — координаты по оси 0x, то должен быть задан аргумент y — координаты по оси 0y.

Функция lines()

```
x<-seq(from=0, to=pi*2, by=0.5)
plot(x)
lines(x)
```




```
x<-seq(from=0, to=pi*2, by=0.5)
plot(sin(x))
lines(sin(x))
```

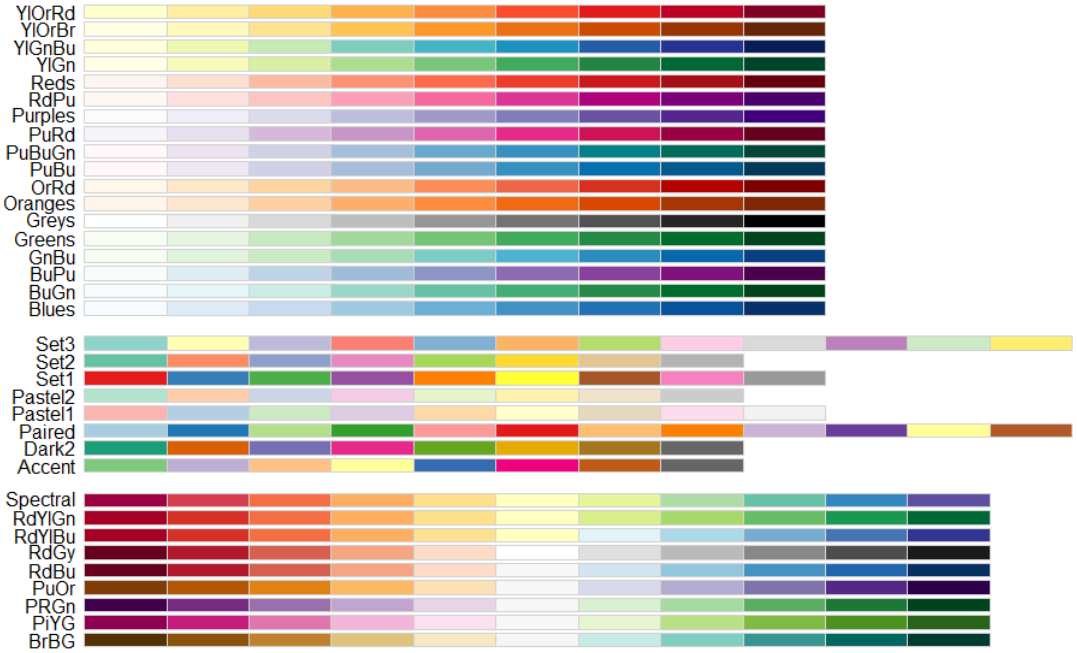


Цветовая палитра в R

В R существует множество стандартных палитр, их список можно найти в справке и документации.

- colors()
- gray()
- rainbow()
- heat.colors()
- topo.colors()
- terrain.colors()

```
install.packages("RColorBrewer")
library(RColorBrewer)
display.brewer.all()
```



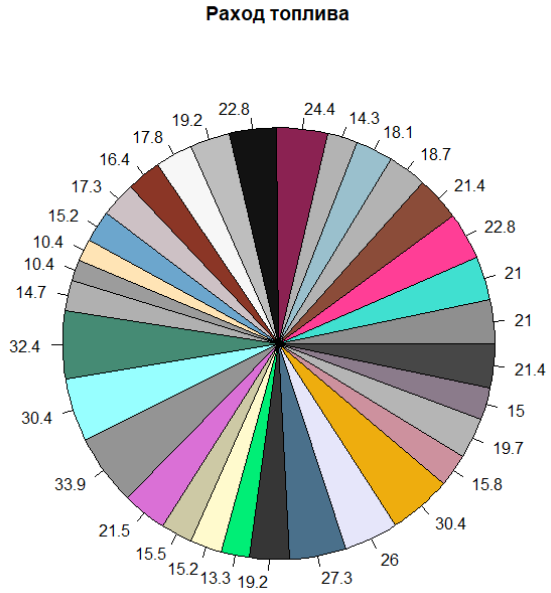
Цветовая палитра в R

```
> cl <- colors()
> cl[10]
[1] "aquamarine2"

> colors()[10]
[1] "aquamarine2"

> gray(0.1)
[1] "#1A1A1A"

> rainbow(5)
[1] "#FF0000FF" "#CCFF00FF" "#00FF66FF" "#0066FFFF" "#CC00FFFF"

> heat.colors(5)
[1] "#FF0000FF" "#FF5500FF" "#FFAA00FF" "#FFFF00FF" "#FFFF80FF"

> topo.colors(5)
[1] "#4C00FFFF" "#004CFFFF" "#00E5FFFF" "#00FF4DFF" "#FFFF00FF"

> terrain.colors(5)
[1] "#00A600FF" "#E6E600FF" "#EAB64EFF" "#EEB99FFF" "#F2F2F2FF"
```

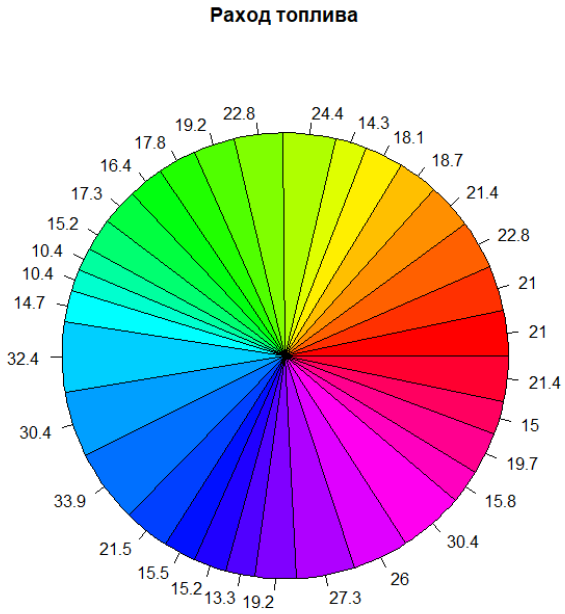
уровень серого, должен быть в [0,1]

Цветовая палитра в R

```
pie(
        mtcars$mpg,
        mtcars$mpg,
        main = "Раход топлива",
        col=sample(colors(),length(mtcars$mpg))
)
```

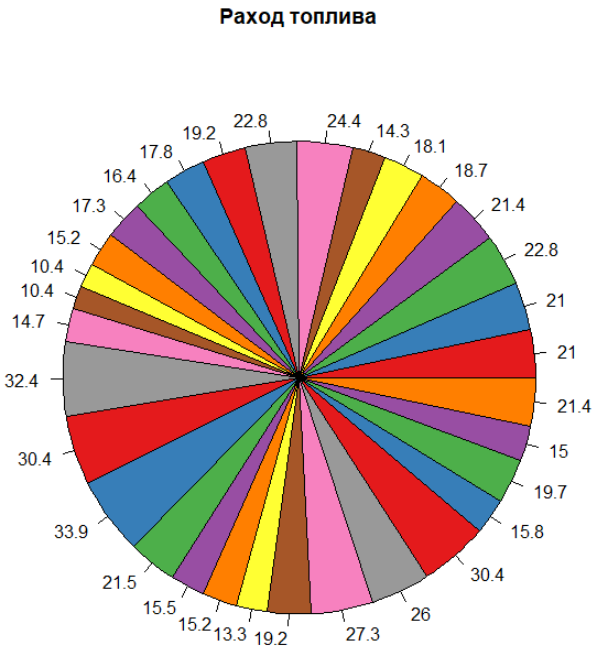



```
pie(
        mtcars$mpg,
        mtcars$mpg,
        main = "Раход топлива",
        col=rainbow(length(mtcars$mpg))
)
```



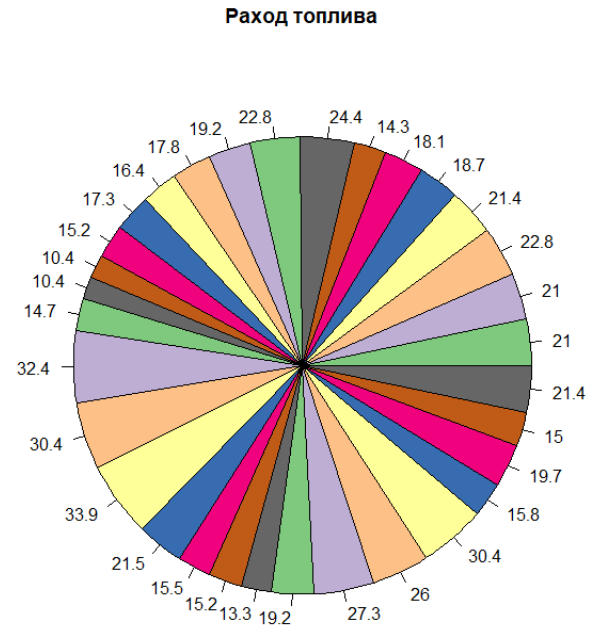
Цветовая палитра в R

```
library(RColorBrewer)
colors = brewer.pal(9,"Set1")
pie(
        mtcars$mpg,
        mtcars$mpg,
        main = "Раход топлива",
        col=colors
)
```



набор цветов Set1 содержит 9 значений.
при превышении будет Warning message

```
colors = brewer.pal(8,"Accent")
pie(
        mtcars$mpg,
        mtcars$mpg,
        main = "Раход топлива",
        col=colors
)
```

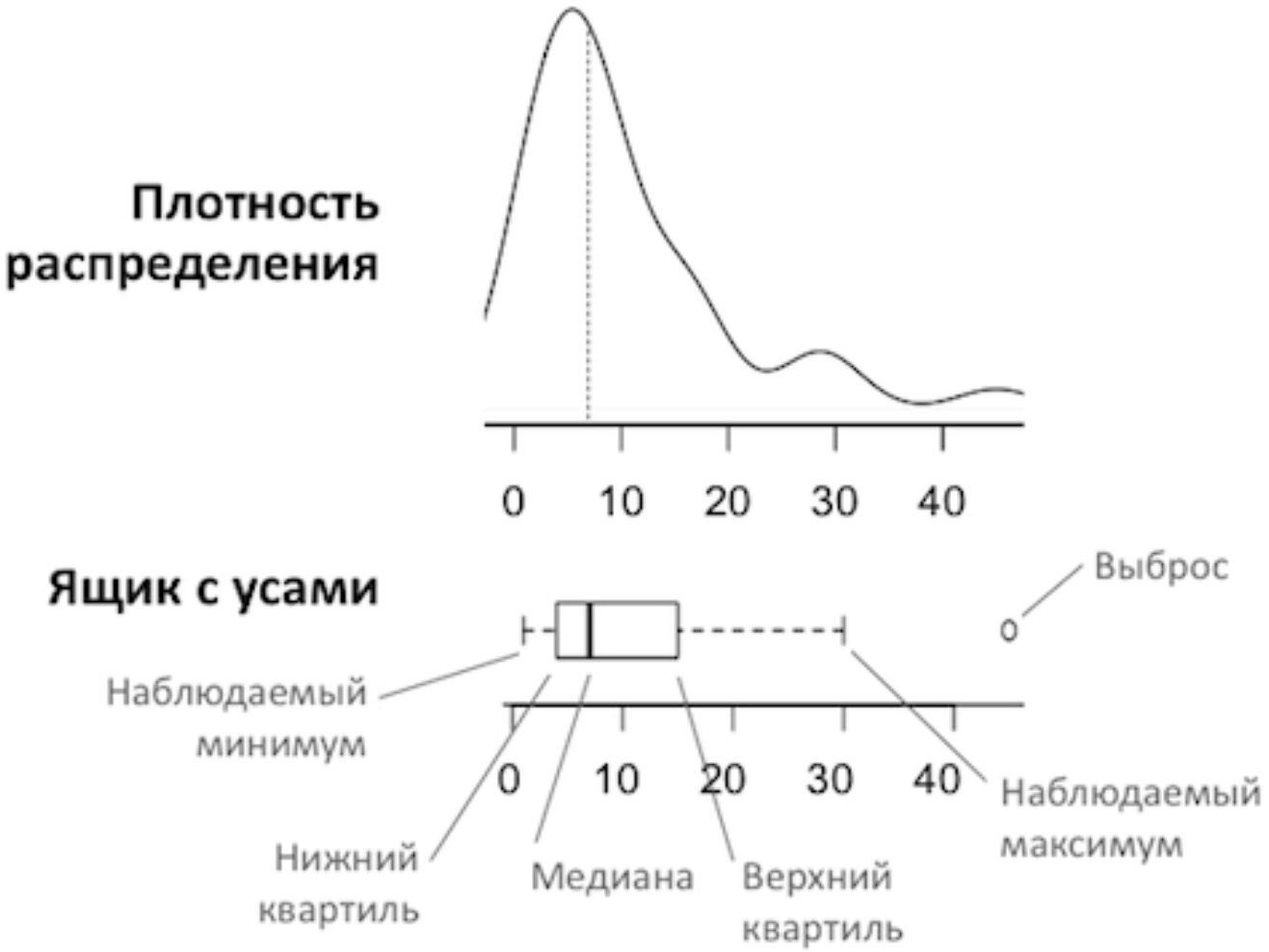


Диаграммы размахов

boxplot()

- Диаграммы размахов (box plot) иллюстрируют распределение значений непрерывной переменной, отображая пять параметров:
  - минимум,
  - нижний квартиль (25-й процентиль),
  - медиану (50-й процентиль),
  - верхний квартиль (75-й процентиль)
  - максимум.
- На этой диаграмме также могут быть отображены вероятные выбросы (значения, выходящие за диапазон в ±1.5 межквартильного размаха, разности верхней и нижней квартилей).
- По умолчанию каждый «ус» продолжается до минимального или максимального значения, которое не выходит за пределы 1.5 межквартильного размаха. Выходящие за эти пределы значения отмечаются точками

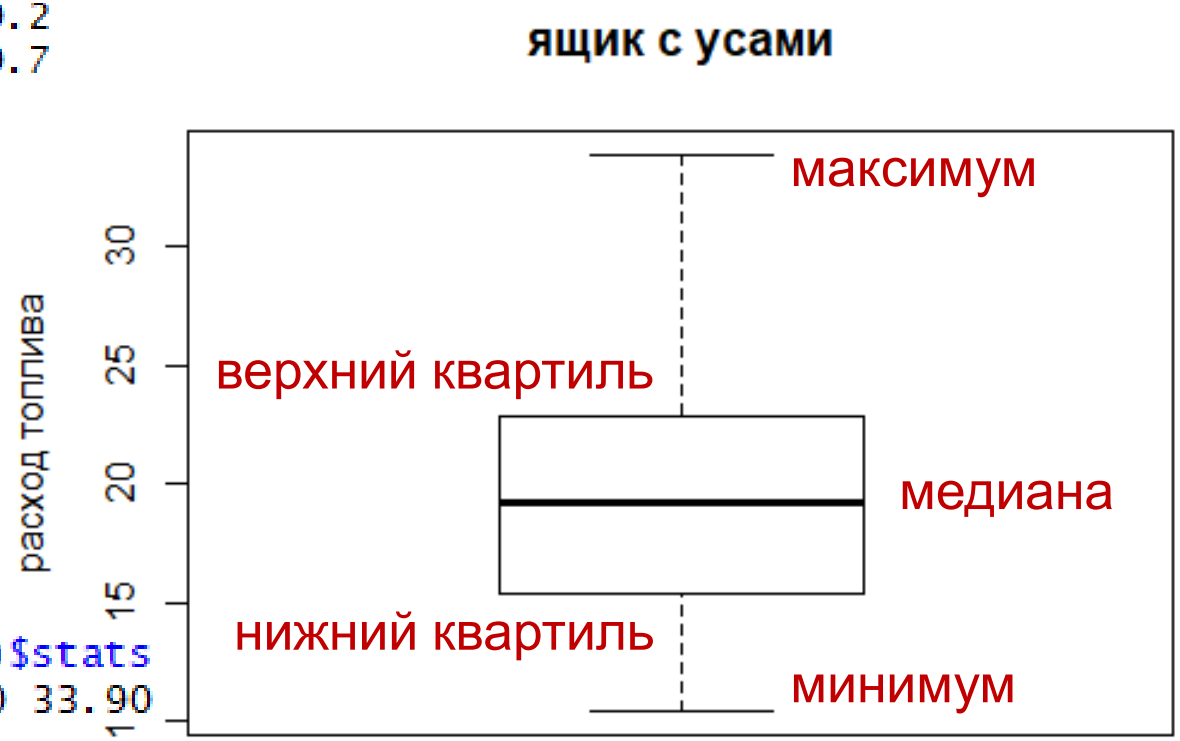
Сравнение плотности распределения и ящика с усами



boxplot()

boxplot(mtcars$mpg, main="ящик с усами", ylab="расход топлива")

```
> mtcars$mpg
 [1] 21.0 21.0 22.8 21.4 18.7
 [6] 18.1 14.3 24.4 22.8 19.2
[11] 17.8 16.4 17.3 15.2 10.4
[16] 10.4 14.7 32.4 30.4 33.9
[21] 21.5 15.5 15.2 13.3 19.2
[26] 27.3 26.0 30.4 15.8 19.7
[31] 15.0 21.4
```




```
> boxplot.stats(mtcars$mpg)$stats
[1] 10.40 15.35 19.20 22.80 33.90
```

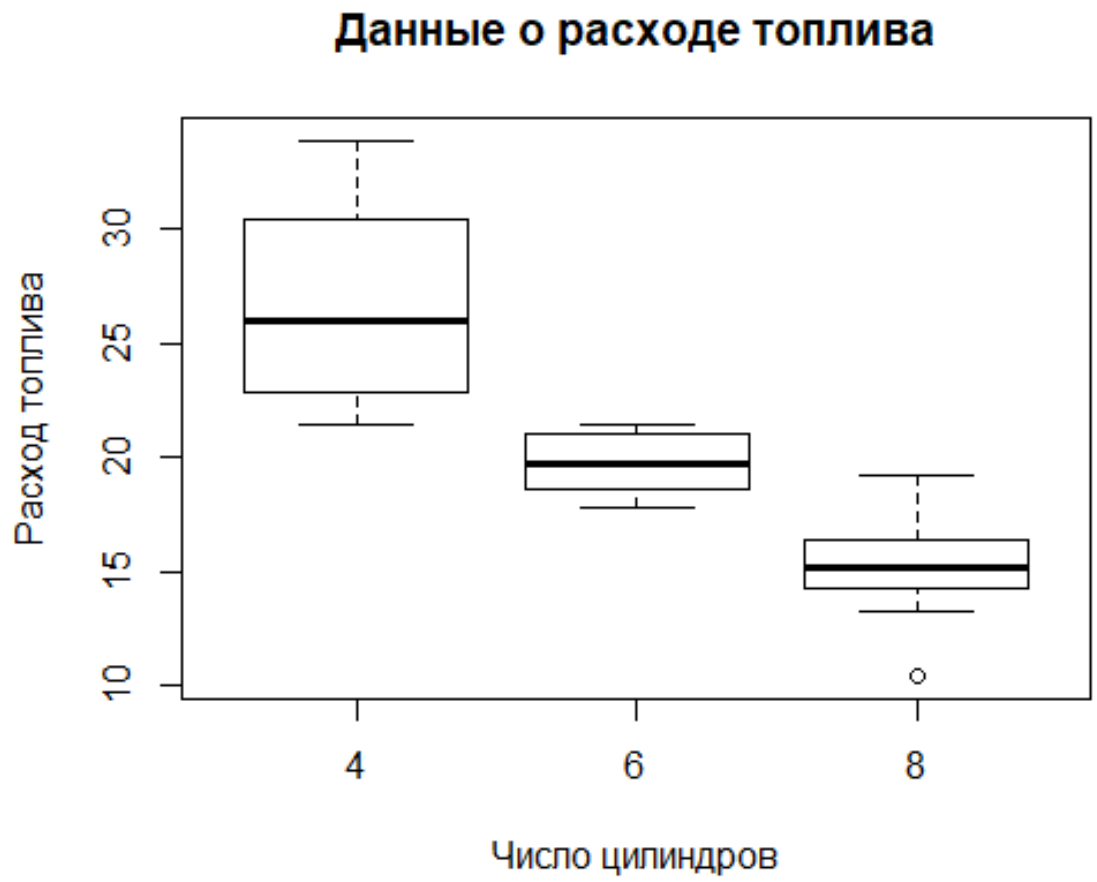
Использование диаграмм размахов для сравнения групп между собой

- Диаграммы размахов можно построить для отдельных переменных или для групп переменных.
- Общий вид команды таков: boxplot(*formula*, data=*dataframe*)
- где *formula* – это формула, а *dataframe* обозначает таблицу данных (или список), где содержатся данные.
- Примером формулы может служить выражение y~A, где для каждого значения категориальной переменной A будет построена отдельная диаграмма размахов для числовой переменной y. Формула y~A*B позволит получить отдельные диаграммы размахов для всех комбинаций значений переменной y, заданных категориальными переменными A и B.

Использование диаграмм размахов для сравнения групп между собой

```
boxplot(mpg ~ cyl, data=mtcars, main="Данные о расходе топлива",
xlab="Число цилиндров", ylab="Расход топлива")
```

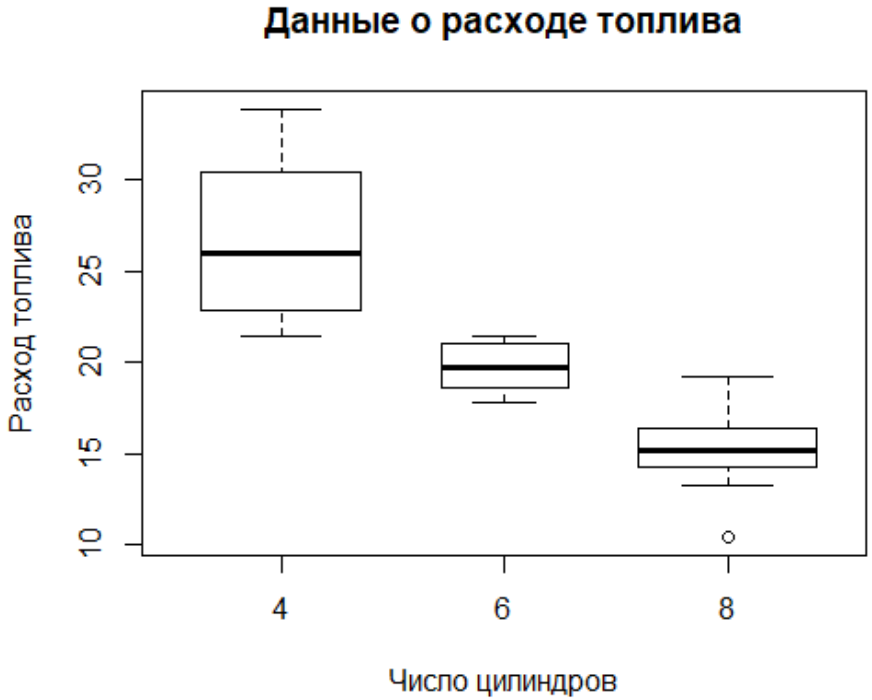
~ способ записи формул, описывающих связь между переменными



Параметры

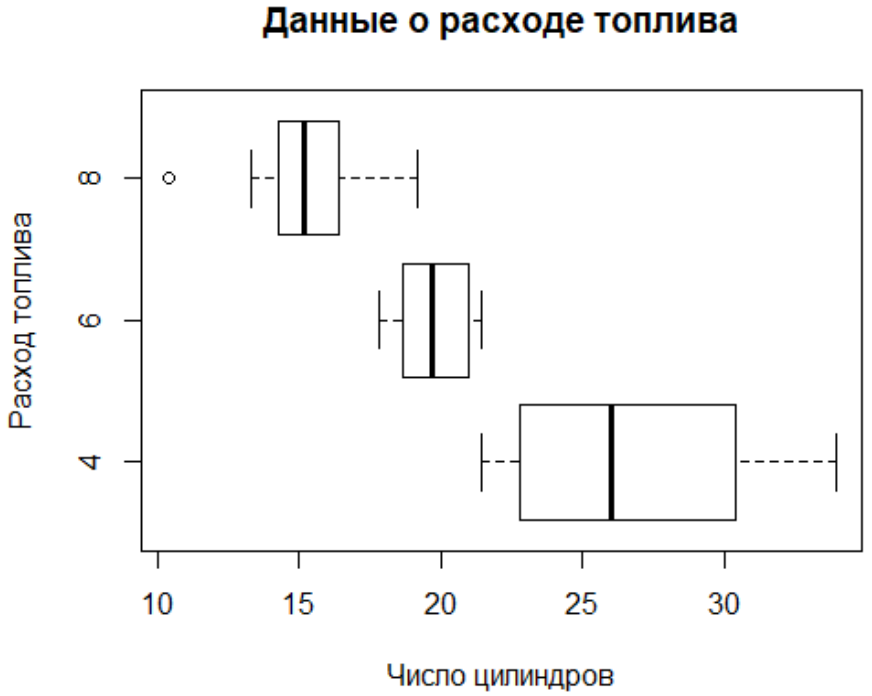
boxplot(mpg ~ cyl, data=mtcars, **varwidth=TRUE**, main="Данные о расходе топлива", xlab="Число цилиндров", ylab="Расход топлива")

varwidth=TRUE
ширина "ящиков" будет пропорциональна квадратному корню из размера выборки



boxplot(mpg ~ cyl, data=mtcars, **horizontal=TRUE**, main="Данные о расходе топлива", xlab="Число цилиндров", ylab="Расход топлива")

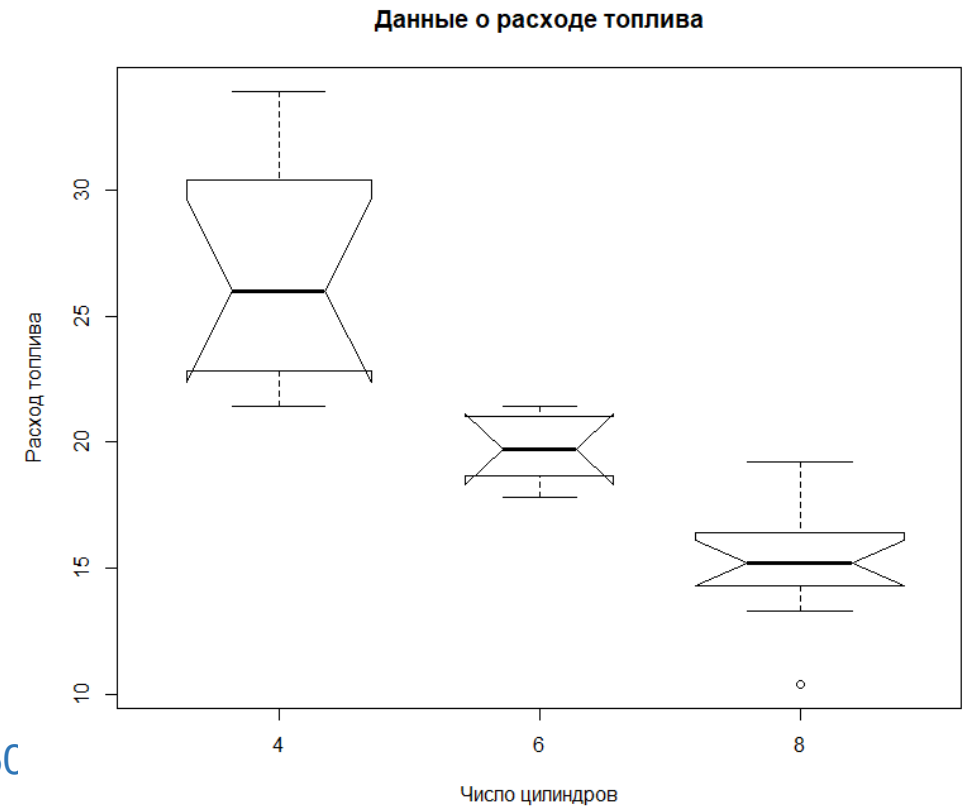
horizontal=TRUE
поменять оси местами



Параметры

notch=TRUE
получатся "ящики" с "насечками". Если "насечки" двух ящиков не перекрываются, высока вероятность того, что медианы соответствующих совокупностей различаются

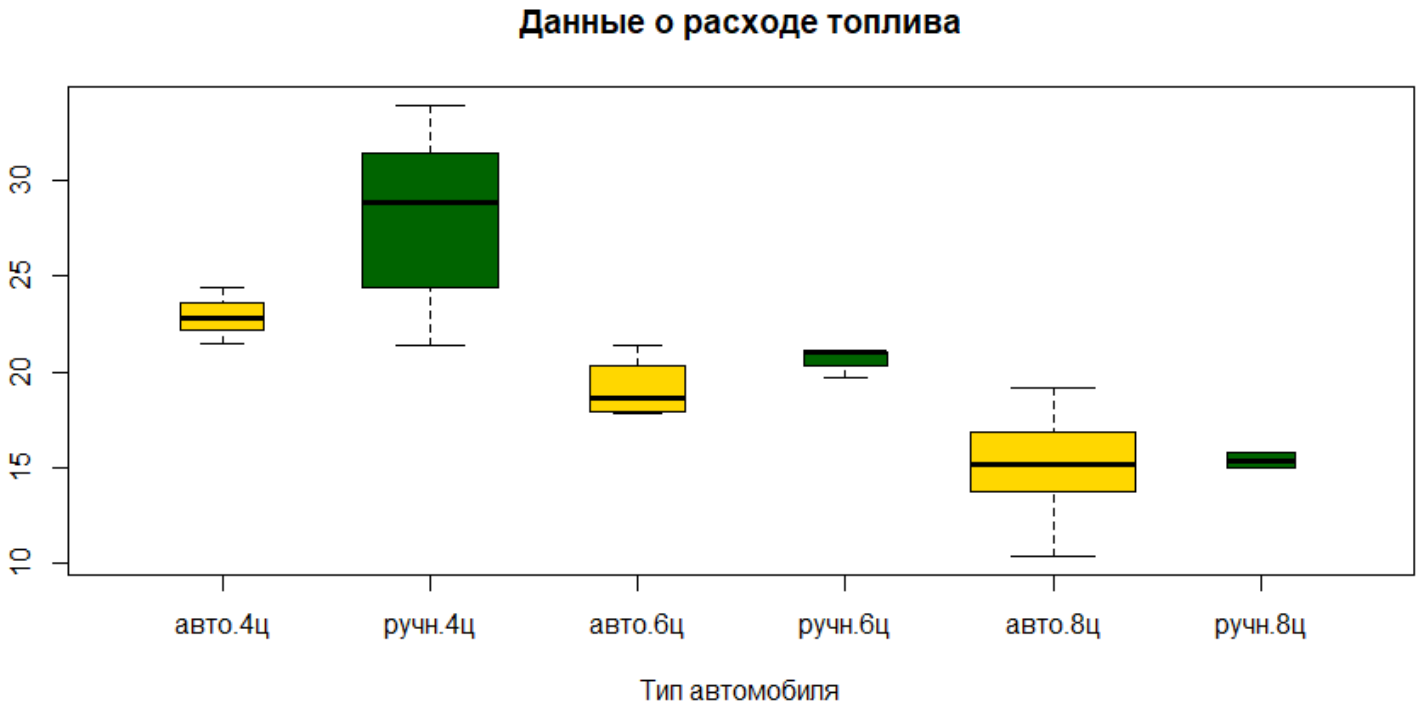
boxplot(mpg ~ cyl, data=mtcars, **notch=TRUE, varwidth=TRUE**, main="Данные о расходе топлива", xlab="Число цилиндров", ylab="Расход топлива")



Диаграммы размахов для нескольких группирующих переменных

Формула y~A*B

```
mtcars$cyl.f <- factor(mtcars$cyl,levels=c(4,6,8),labels=c("4ц","6ц","8ц"))
mtcars$am.f <- factor(mtcars$am,levels=c(0,1),labels=c("авто", "ручн"))
boxplot(mpg ~ am.f *cyl.f,data=mtcars, varwidth=TRUE,
col=c("gold","darkgreen"), main="Данные о расходе топлива",
xlab="Тип автомобиля")
```



Настройки отображения

Размеры элементов графика

Изменять размеры элементов графика можно независимо друг от друга, используя следующие параметры:

- cex — общий масштаб элементов на графике
- cex.axis — масштаб подписей координат на оси
- cex.lab — масштаб подписей названий осей
- cex.main — масштаб заголовка графика
- cex.sub — масштаб подзаголовка графика
- cex.names — масштаб подписей факторов (для некоторых типов диаграмм)

Настройка цвета

Тонкая настройка цвета:

- col цвет графика
- col.axis цвет подписей координат
- col.lab цвет названий осей
- col.main цвет заголовка
- col.sub цвет подзаголовка
- fg цвет элементов переднего плана (оси, рамка и т.д.)
- bg цвет фона графика (background)

Разметка осей

По умолчанию R подбирает оптимальный с точки зрения него шаг разметки осей, в зависимости от разброса значений по осям X и Y, а также размеров графического устройства, на котором производится рисование. Изменяя размер окна прорисовки, вы получите различную разметку осей.

Есть необходимость самостоятельно управлять шагом разметки сетки. Для этого необходимо:

- Вызвать функцию plot(), передав ей дополнительно параметр axes = FALSE (убирает при рисовании обе оси) или один из параметров xaxt="n" / yaxt="n" (убирают оси X и Y соответственно)
- Вызвать столько раз функцию axis(), сколько вы хотите нарисовать осей, передав ей параметры для рисования каждой оси.

Функция axis() принимает следующие параметры:

- side — сторона графика, на которой будет нарисована ось (1=bottom, 2=left, 3=top, 4=right)
- at — вектор значений, в которых должны быть нарисованы метки оси
- labels — вектор подписей, которые будут нарисованы в местоположениях, указанных в параметре at. Этот параметр можно пропустить, если подписи совпадают с местоположениями меток
- pos — координата, вдоль которой будет нарисована ось
- lty — тип линии
- col — цвет линии и меток
- las — расположение подписей параллельно (0) или перпендикулярно (2) оси
- tck — длина метки относительно размера графика. Отрицательные значения дают метки, выходящие за пределы графика. положительные — внутрь графика. 0 убирает метки, 1 рисует линии сетки.

Рамка вокруг графика - функция box()

Сетка координат

Для размещения сетки координат существует функция
grid(nx = NULL, ny = nx, col = "lightgray", lty = "dotted", lwd = par("lwd"), equilogs = TRUE).
Сетка определяется количеством линий в горизонтальном и вертикальном направлении. Это не всегда бывает удобно, поскольку как правило надо задать шаг сетки конкретной величины. По умолчанию линии сетки выбираются автоматически.

Можно поменять количество линий, однако R не будет согласовывать шаг сетки и шаг меток осей, поскольку метки генерируются на стадии рисования plot() или axis() и не запоминаются.
Линии сетки будут равномерно распределены и не совпадать с метками.

Функция grid() на самом деле является оберткой функции abline(), которая позволяет рисовать произвольные линии на графике.

При построении пользовательских сеток следует пользоваться функцией abline()

Аннотации данных (текст на графике, метки данных)

Аннотации данных добавляются на график с помощью функции text(). В качестве трех обязательных аргументов ей необходимо передать координаты точек размещения текста, и вектор подписей.

Параметр pos=, отвечает за размещение аннотации относительно точки.
Значения pos, равные 1, 2, 3 и 4, соответствуют размещению снизу, слева, сверху и справа от точки

```
text(
        X_vector,
        Y_vector,
        labels = labels_vector,
        cex = 0.75,
        pos = 3
)
```

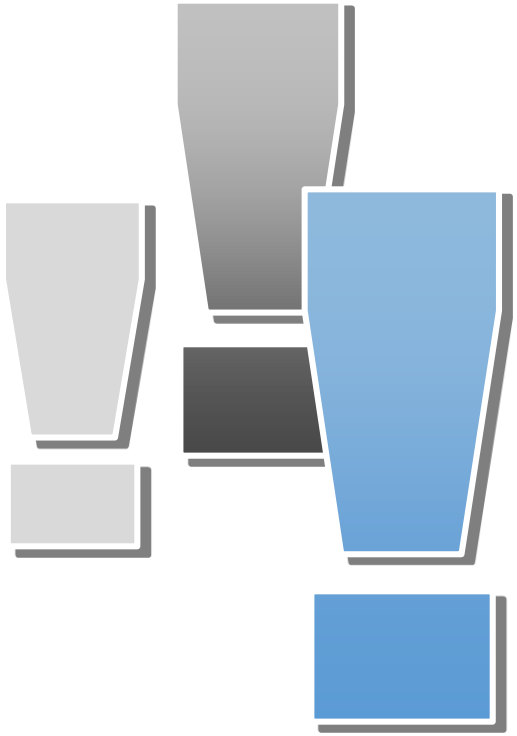
Легенда

Легенда к графику размещается с помощью функции legend().
Эта функция принимает несколько аргументов, включая: местоположение, заголовок, названия элементов, графические параметры. Местоположение может быть задано координатами (x,y) в системе координат графика, но удобнее пользоваться следующими предопределенными константами:

- "bottomright",
- "bottom",
- "bottomleft",
- "left",
- "topleft",
- "top",
- "topright",
- "right",
- "center".

Чтобы в легенде появились точки, необходимо задать параметр pch=.
Для линейной легенды, следует задать, соответственно, параметр lty = и/или lwd =. Каждый из этих параметров должен быть вектором по количеству элементов легенды.

Спасибо за внимание!



Шевцов Василий Викторович

shevtsov_vv@rudn.university
+7(903)144-53-57