Въведение в Статистиката с R – Bivariate Data

В предната тема работихме с едномерни качествени и количествени данни. Сега, по аналогичен начин ще работим с двумерни величини. Ще разгледаме случаите: качествена/качествена, качествена/количествена, количествена/количествена.

1. Работа с две категорийни(категорни, качествени) величини –

За да демонстрираме основните възможности на R за описване на зависимостта между два категорийни признака ще използваме таблиците с данни SoftwareEngineering и tires. Нека първо да зададем работната си директория и да заредим данните от тези два файла.

```
> setwd("C:\\Users\\Monika\\Desktop\\R\\2016-2017\\Lecture2")
> dir()
[1] "SoftwareEngineering.csv"              "tires.csv"
> SoftwareEngineering = read.csv(file = "SoftwareEngineering.csv")
> tires = read.csv(file = "tires.csv", header = TRUE, sep = ";", dec = ",")
```

До векторите, които съдържат резултатите от наблюденията, можем да достигнем чрез функцията

```
> attach(SoftwareEngineering)
> attach(tires)
```

или чрез оператора "$" като преди него напишем името на таблицата с данни. Т.к. ние ще работим изцяло с тези таблици с данни ще използваме функцията attach.

В таблицата с данни SoftwareEngineering категорни признаци са:

Gender, Region, HairColor, EyesColor.

В таблицата с данни tires категорни променливи са:

X1, X2, X3, X7, X8, X10, X11_1, X11_2, X11_3, X12_1, X12_2, X12_3

1.1 Групировка в абсолютни честоти

В резултат от групировката на данните според наблюдаваните при тях значения на два признака се получават така наречените кръстосани таблици. Ако те съдържат броя на наблюденията, които попадат в непресичащите се групи, определени от различните регистрирани значения на двата признака при наблюдаваните статистически единици, говорим за групировка в абсолютни честоти.

Кръстосаната таблица в абсолютни честоти, по двата признака Gender и HairColor на таблицата с данни SoftwareEngineering се извежда в R с използването на функцията ***table***. Като параметри подаваме имената на разглежданите два признака.

```
> table(Gender, HairColor)
        HairColor
Gender   Black Blonde Blue Brown
  Female     0      1    1    19
  Male       9      0    0     9
```

По аналогичен начин, в таблицата с данни tires, кръстосаната таблица, която се получава след групировка по

X1 – "Вид гума, за която ще бъде попълнена анкетата" и

X2 – "Производител"

се извежда в R с:

```
> table(X1, X2)
X2
X1   P1  P2  P3  P4  P5  P6
```

```
T1   3   3   1   4   5   6
T2   3   6   5   4   2   4
T3   4   4   5   3   3   4
T4   5   5   3   3   4   6
T5   6   3   2   4   7   6
T6   6   8  13  13  11  12
T7  11  10  10  11  11  11
```

С помощта на функцията names можем да зададем имена на двата разглеждани признака и да ги използваме вместо X1 и X2. Добре е да ги зададем като отделен вектор.

```
> names = c("Вид", "Производителl")
> table(X1, X2 , dnn = names)
          Производител
Вид  P1 P2 P3 P4 P5 P6
T1    3  3  1  4  5  6
T2    3  6  5  4  2  4
T3    4  4  5  3  3  4
T4    5  5  3  3  4  6
T5    6  3  2  4  7  6
T6    6  8 13 13 11 12
T7   11 10 10 11 11 11
```

Вместо етикетите на видовете гумите, за които е попълнена анкетата можем да напишем марките им. Това става с помощта на функцията ***rownames***. Тази марки също е добре е бъдат зададени като отделен вектор.

```
> rnames = c("Audi", "BMW","Honda", "Mercedes", "Nissan","Peugeot", "Citroen")
> names = c("Вид", "Производител")
> t = table(X1, X2 , dnn = names)
> rownames(t) = rnames
t
            Производител
Вид         P1  P2  P3  P4  P5  P6
Audi         3   3   1   4   5   6
BMW          3   6   5   4   2   4
Honda        4   4   5   3   3   4
Mercedes     5   5   3   3   4   6
Nissan       6   3   2   4   7   6
Peugeot      6   8  13  13  11  12
Citroen     11  10  10  11  11  11
```

По аналогичен начин, с помощта на функцията ***colnames*** можем да сменим имената на колоните.

Функцията ***prop.table*** в R, с втори параметър 1, ни позволява да определим какъв е процента на единиците в съответната група от общата сума в реда.

```
> GenderHair = table(Gender, HairColor)
> prop.table(GenderHair, 1)
        HairColor
 Gender       Black      Blonde        Blue       Brown
 Female 0.00000000 0.04761905 0.04761905 0.90476190
```

Male	0.50000000	0.00000000	0.00000000	0.50000000

Може да закръглим числата до втория знак след десетичната запетая

```
> round(prop.table(GenderHair, 1)*100,2)
```

Gender	HairColor Black	Blonde	Blue	Brown
Female	0.00	4.76	4.76	90.48
Male	50.00	0.00	0.00	50.00

Това означава например, че:

- 50.00% от Мъжете са с черна коса.
- 4.76% от Жените са руси.
- 4.76% от Жените са със синя коса.
- 90.48% от Жените са брюнетки.

По аналогичен начин, в примера с таблицата с данни tires, кръстосаната таблица, която съдържа пропорциите, като процент от общата сума в реда се получава с:

```
> prop.table(t, 1)*100
```

	Производител					
Вид	P1	P2	P3	P4	P5	P6
Audi	13.636364	13.636364	4.545455	18.181818	22.727273	27.272727
BMW	12.500000	25.000000	20.833333	16.666667	8.333333	16.666667
Honda	17.391304	17.391304	21.739130	13.043478	13.043478	17.391304
Mercedes	19.230769	19.230769	11.538462	11.538462	15.384615	23.076923
Nissan	21.428571	10.714286	7.142857	14.285714	25.000000	21.428571
Peugeot	9.523810	12.698413	20.634921	20.634921	17.460317	19.047619
Citroen	17.187500	15.625000	15.625000	17.187500	17.187500	17.187500

Добре е да закръглим числата до втория знак след десетичната запетая

```
> t1 = round(prop.table(t,1)*100,2)
```

	Производител					
Вид	P1	P2	P3	P4	P5	P6
Audi	13.64	13.64	4.55	18.18	22.73	27.27
BMW	12.50	25.00	20.83	16.67	8.33	16.67
Honda	17.39	17.39	21.74	13.04	13.04	17.39
Mercedes	19.23	19.23	11.54	11.54	15.38	23.08
Nissan	21.43	10.71	7.14	14.29	25.00	21.43
Peugeot	9.52	12.70	20.63	20.63	17.46	19.05
Citroen	17.19	15.62	15.62	17.19	17.19	17.19

Това означава например, че:

- 13.63% от гумите Ауди са произведени от P1.
- 13.64% от гумите Ауди са произведени от P2.
- 4.55% от гумите Ауди са произведени от P3.
- 25% от гумите BMW са произведени от P2.
- Ако избраната гума е Нисан, с 25% шанс тя е произведена от P5.

С помощта на функцията apply можем да проверим дали сумите в редовете са по 100.

```
> apply(t1, 1, sum)
```

Audi	BMW	Honda	Mercedes	Nissan	Peugeot	Citroen
100.01	100.00	99.99	100.00	100.00	99.99	100.00

По аналогичен начин, с помощта на функцията ***prop.table*** с втори параметър 2, R ни позволява да определим какъв е процентът на единиците в съответната група, от общата сума в колоната.
От първия ни пример за студентите.

```
> GenderHair = table(Gender, HairColor)
> prop.table(GenderHair, 2)
        HairColor
Gender       Black       Blonde      Blue        Brown
  Female 0.0000000 1.0000000 1.0000000 0.6785714
  Male   1.0000000 0.0000000 0.0000000 0.3214286
```

Може да превърнем пропорциите в проценти и да закръглим числата до втория знак след десети чната запетая

```
> round(prop.table(GenderHair, 2)*100,2)
        HairColor
Gender      Black Blonde    Blue  Brown
  Female     0.00 100.00  100.00  67.86
  Male     100.00   0.00    0.00  32.14
```

Това означава например, че:

- 100.00% от хората с черна коса са мъже, т.е. с черна коса имаме само мъже.
- 100.00% от хората с руса коса са жени, т.е. с руса коса имаме само жени.
- 100.00% от хората със синя коса са жени, т.е. със синя коса имаме само жени.
- 67.86% от брюнетите са жени, т.е. имаме повече момичета брюнетки.

В примера с таблицата с данни tires, кръстосаната таблица, която съдържа пропорциите, като пр оцент от общата сума в колоната се получава с

```
> t2=round(prop.table(t,2)*100,2)
```

	Производител					
Вид	P1	P2	P3	P4	P5	P6
Audi	7.89	7.69	2.56	9.52	11.63	12.24
BMW	7.89	15.38	12.82	9.52	4.65	8.16
Honda	10.53	10.26	12.82	7.14	6.98	8.16
Mercedes	13.16	12.82	7.69	7.14	9.30	12.24
Nissan	15.79	7.69	5.13	9.52	16.28	12.24
Peugeot	15.79	20.51	33.33	30.95	25.58	24.49
Citroen	28.95	25.64	25.64	26.19	25.58	22.45

Това означава например, че:

- 7.89% от гумите произведени от P1, за които са попълнени анкетите са Аудита.
- 7.69% от гумите произведени от P2, за които са попълнени анкетите са Аудита.
- 22.45% от гумите произведени от P6, за които са попълнени анкетите са Ситроени.
- Ако избраната гума е произведена от P1, най-голям е шансът тя да е Ситроен.

Можем да проверим дали сумите в редовете са по 100.

```
> apply(t2,2,sum)
   P1    P2    P3    P4    P5    P6
100.00  99.99  99.99  99.98 100.00  99.98
```

Разликите се дължат на грешки от закръгляния.

Функцията ***prop.table*** без втори параметър ни позволява да определим какъв е процентът на единиците в съответната група, от обема на извадката, т.е. от общата сума в таблицата с абсолютни честоти.

Пропорциите на студентите в отделните групи, образувани по „Пол“ и „Цвят на косата“ могат да бъдат получени с:

```
> GenderHair = table(Gender, HairColor)
> prop.table(GenderHair)
        HairColor
Gender       Black       Blonde         Blue        Brown
  Female 0.00000000 0.02564103  0.02564103 0.48717949
  Male   0.23076923 0.00000000 0.00000000  0.23076923
```

Може да превърнем тези пропорции в проценти, и да закръглим числата до втория знак след де сетичната запетая.

```
> round(prop.table(GenderHair)*100,2)
        HairColor
Gender   Black Blonde  Blue Brown
  Female  0.00   2.56  2.56 48.72
  Male   23.08   0.00  0.00 23.08
```

Това означава например, че:

- 23.08% от всички хора в извадката ни са мъже с черна коса.
- 2.56% от всички хора в извадката ни са жени и са руси.
- 2.56% от всички хора в извадката ни са жени и със синя коса.
- 48.72% от всички хора в извадката са жени и са брюнетки .

Пропорциите на анкетираните в отделните групи по „Вид на гумата“ и „Производител“, измерени в проценти от обема на извадката, могат да бъдат получени с:

```
> t3 = round(prop.table(t)*100,2)
                Производител
Вид          P1   P2   P3   P4   P5   P6
Audi        1.2  1.2  0.4  1.6  2.0  2.4
BMW         1.2  2.4  2.0  1.6  0.8  1.6
Honda       1.6  1.6  2.0  1.2  1.2  1.6
Mercedes    2.0  2.0  1.2  1.2  1.6  2.4
Nissan      2.4  1.2  0.8  1.6  2.8  2.4
Peugeot     2.4  3.2  5.2  5.2  4.4  4.8
Citroen     4.4  4.0  4.0  4.4  4.4  4.4
```

Това означава например, че:

- 1.2% от анкетираните са ползватели на гуми Ауди, с производител P1.
- 2.4% от анкетираните са ползватели на гуми BMW, с производител P2.
- 4.4% от анкетираните са ползватели на гуми Ситроен, с производител P6.

Можем да проверим дали сумата в таблицата е 100.

```
> sum(t3)
[1] 100
```

Задачи за упражнение.

1. Използвайки таблицата SoftwareEngineering направете кръстосани таблици, които да отразяват резултата от групировките по признаците

Gender – EyesColor, Gender – AverageGrade, HairColor - AverageGrade

Представете групировката в

- абсолютни числа
- процент от общото
- процент от сумата в реда
- процент от сумата в колоната

и анализирайте резултатите.

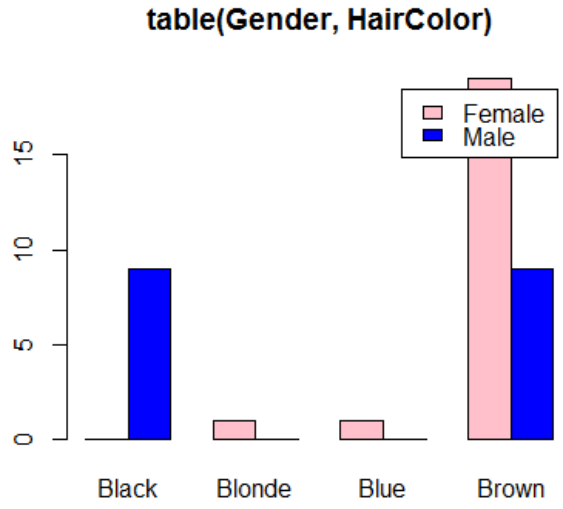
2. Използвайки таблицата tires направете кръстосани таблици, които да отразяват резултата от групировка по признаците
X2 - Производител
X3 – Доставчик.
Сменете имената на променливите. Сменете имената на редовете. Сменете имената на колоните.
Представете групировката в

- абсолютни числа
- процент от общото
- процент от сумата в реда
- процент от сумата в колоната

и анализирайте резултатите.

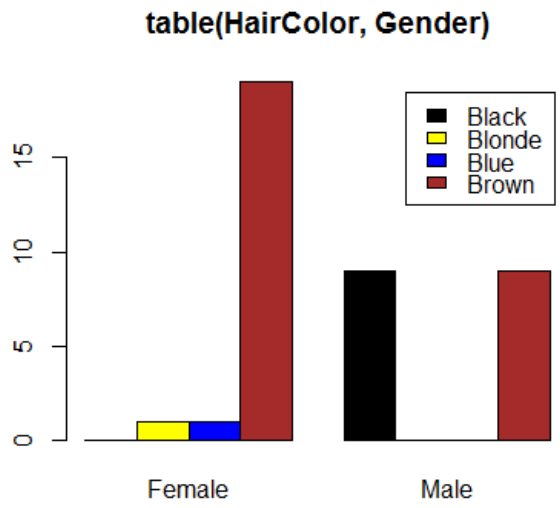
1.2 Графични изображения за онагледяване на структурата на извадката според наблюдаваните значения на два категорийни признака.

За да онагледим структурата на извадката според наблюдаваните значения на два категорийни признака и по този начин до направим изводи за зависимостта между тях можем да използваме функцията ***barplot***.
Следващите две графики показват, че жените студенти с кафява коса са много по-често срещани от мъжете студенти с кафява коса. При това, само мъжете са с черна коса и само жените са с руса или синя коса.
> barplot (table(Gender, HairColor), col = c("pink", "blue"), main = "table(Gender, HairColor)", beside = TRUE, legend.text = c("Female", "Male"))

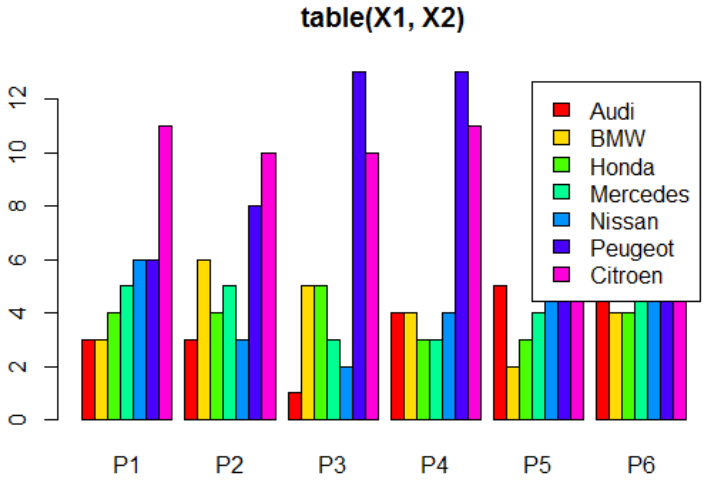


Или

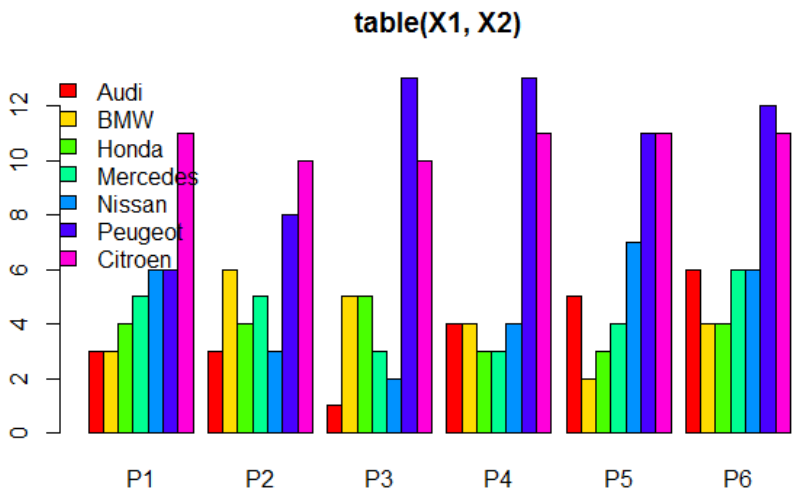
> barplot (table(HairColor, Gender), col = c("black", "yellow", "blue", "brown"), main = "table(HairColor, Gender)", beside = TRUE, legend.text = c("Black", "Blonde", "Blue", "Brown"))



По данните от таблицата tires, структурата на съвкупността от анкетираните според вида на използваната гума и нейния производител може да бъде наблюдавана с:
> rnames = c("Audi", "BMW","Honda", "Mercedes", "Nissan","Peugeot", "Citroen")
> names = c("Вид", "Производител")
> t = table(X1, X2 , dnn = names)
> rownames(t) = rnames
> t
> barplot(t , col = rainbow(7), main = "table(X1, X2)", beside = TRUE, legend.text = rnames)

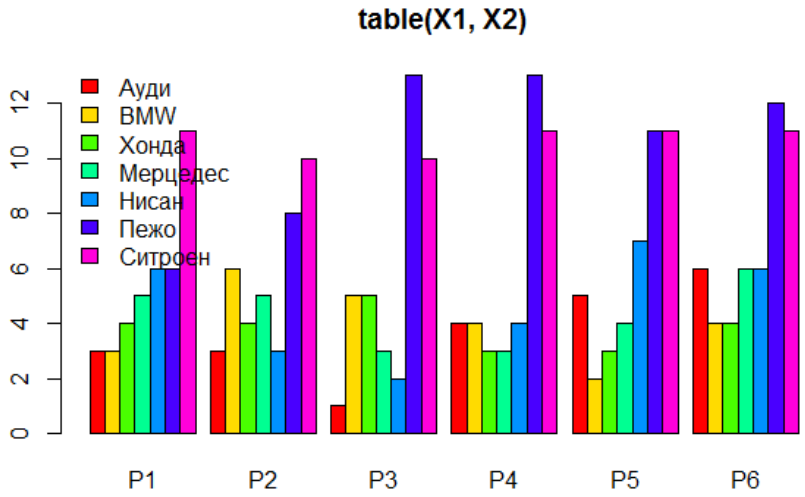


Можем да сложим легендата на мястото, където кликнем с мишката и да махнем контура й.
> barplot(t , col=rainbow(7), main="table(X1, X2)" , beside=TRUE)
> legend(locator(1), rnames, cex=1, fill= rainbow(7),bty="n")



С командата ***legend.text*** можем да сменим текста в легендата

```
> barplot(t ,  col = rainbow(7), main = "table(X1, X2)" , beside = TRUE)
> legend.text = c("Ауди","BMW","Хонда","Мерцедес","Нисан", "Пежо","Ситроен")
> legend(locator(1), legend.text, cex = 1, fill =  rainbow(7), bty = "n")
```

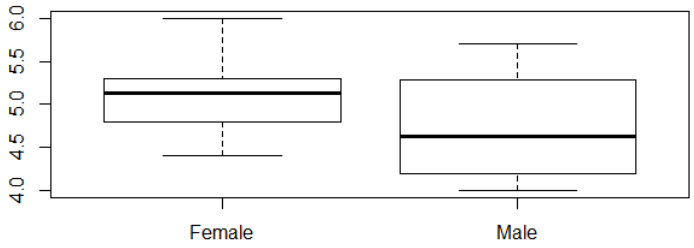


2. Работа с един категориен и един количествен признак –

В тази точка ще предполагаме, че имаме един категориен признак и един количествен и ще се интересуваме от начините за описване на зависимостта между тях.

Нека наблюдаваме зависимостта на качествения признак Gender и количествения AverageGrade от таблицата с данни SoftwareEngineering. За онагледяване на влиянието на категорийния признак (Gender - Пол) върху количествения признак (AverageGrade – среден успех за годината) може да направим boxplot, представящ данните на количествения признак, поотделно по подгрупите на категорийния признак. За целта използваме оператора “~”. От лявата му страна се пише зависимата променлива, количествения признак AverageGrade. От дясната страна на “~” се пише независимата променлива, категорийния признак Gender. Така можем да наблюдаваме влиянието на Gender върху AverageGrade.

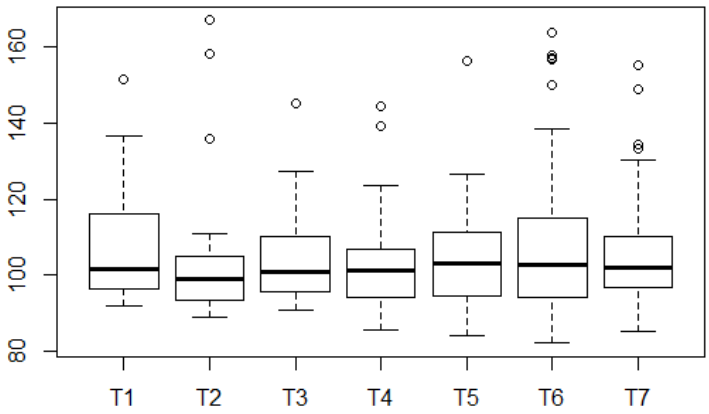
```
> boxplot(AverageGrade ~ Gender)
```



Виждаме, че момичетата в анкетирания курс имат по-висок среден успех от момчетата в същия курс.

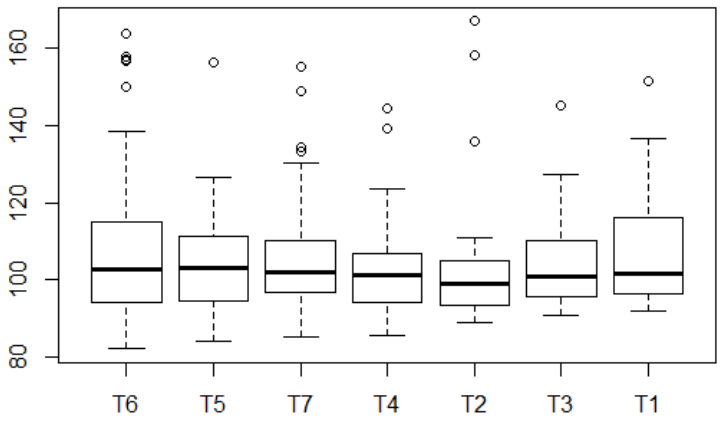
Сега да разгледаме данните от таблицата tires и по-точно качествените признаци:
X1 – вид на колата
X2 – производител
и количествените признаци:
X4 - Цена на закупуване, актуализирана към днешна дата
X5 - Пробег
X6 - Продължителност на живот в дни
X9 - Диаметър на джантата в цолове
За по-лесно установяване на влиянието на категорийния признак (X1 – вид на колата) върху количествения признак (X4 - Цена на закупуване, актуализирана към днешна дата), може да направим резюме (например във формата на boxplot) на данните от количествения признак, поотделно по подгрупите на категорийния признак. За целта използваме оператора “~”. От лявата му страна се пише зависимата променлива. По-долу това е количественият признак X4. От дясната страна на “~” се пише независимата променлива. По-долу това е категорийната променлива X1. Т.е. така визуализираме влиянието на X1 върху X4 или, което е все едно - зависимостта на X4 от X1.

```
> boxplot(X4 ~ X1)
```

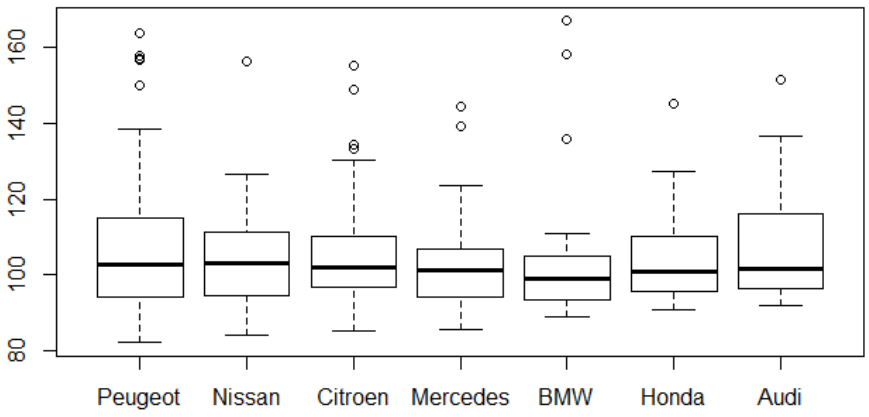


В предната тема вече видяхме, че преди да изчертаем тези графики може, с функциите ***sort*** и ***tapply*** да подредим групите в X1 например според минимумите в X4. За целта виждаме реда на подгрупите в X1 според медианите по признака X4 в тях

```
> sort(tapply(X4, X1, min))
```
```
T6       T5       T7       T4       T2       T3       T1
82.13 84.03 85.29 85.73 88.95 91.03 91.94
```
и дефинираме нова променлива ord, която е таблица с групировката по X1, но нивата са подредени според реда, който сме задали.
```
> ord = ordered(X1, levels = c("T6","T5","T7","T4","T2","T3","T1"))
```
Изчертаваме графики с мустачки на зависимостта на X4 от новата, подредена по наше желание променлива.
```
> boxplot(X4 ~ ord)
```



Можем да сменим имената за по-лесно разчитане на резултатите. При T1 – T7 те бяха съответно "Audi", "BMW","Honda", "Mercedes", "Nissan","Peugeot", "Citroen"
Значи сега ще са
```
>  names = c("Peugeot", "Nissan", "Citroen", "Mercedes", "BMW", "Honda", "Audi")
```
Първо изтриваме етикетите по абсцисната ос на boxplot
```
> boxplot(X4 ~ ord, xaxt="n")
```
След това добавяме новите имена
```
> axis(side = 1, at = c(1,2,3,4,5,6,7), labels = names)
```



Виждаме, че Пежо имат най-голямо разнообразие в цените на гумите си в извадката. При тях е наблюдавана най-ниската цена на гуми, но и най-високата. Според горните графики, минималните цени нарастват от ляво на дясно.

Можем за превърнем количествения признак в качествен, посредством функцията ***cut*** и след това да продължим анализа, така все едно имаме два категорийни признака. Например:

```
> table(cut(X4, quantile(X4)), X2)
```

	X2					
	P1	P2	P3	P4	P5	P6
(82.1, 94.5]	14	12	11	6	9	10
(94.5, 102.0]	10	10	7	11	13	11
(102, 111.0]	7	7	7	14	9	18
(111, 167.0]	7	10	13	11	12	10

Този начин, обаче не винаги е за предпочитане, защото при групировката по количествения признак се губи част от информацията и по-късно това ограничава възможностите за анализ.

3. Работа с два количествени признака –

При извършване на анализи на данни най-удобно се работи с количествени признаци т.к. можем да използваме различни видове мерки.

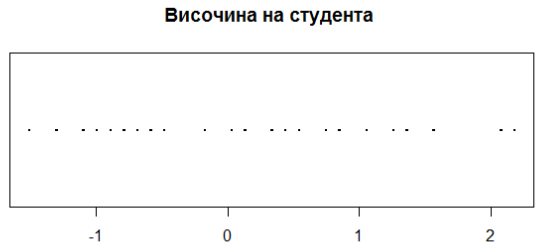
3.1 Сравняване чрез подреждане на наблюденията и визуализирането им върху реалната права.

За да сравним стойностите на наблюдаваните значения на признаците най-лесно е да отбележим наредените наблюдения върху правата и да ги сравним. Това става с помощта на функцията ***stripchart.***

В таблицата с данни SoftwareEngineering количественна данни са Height, Weight, AverageGrade.

Да сравним, например измерените значения на признаците Height и Weight. Т.к. тези два признака са измерени в различна мярка, за да ги сравним трябва първо да ги центрираме и нормираме. Т.е. от всяко наблюдение трябва да извадим средната аритметична и да разделим на стандартното отклонение в извадката. Новите величини вече не са именувани.

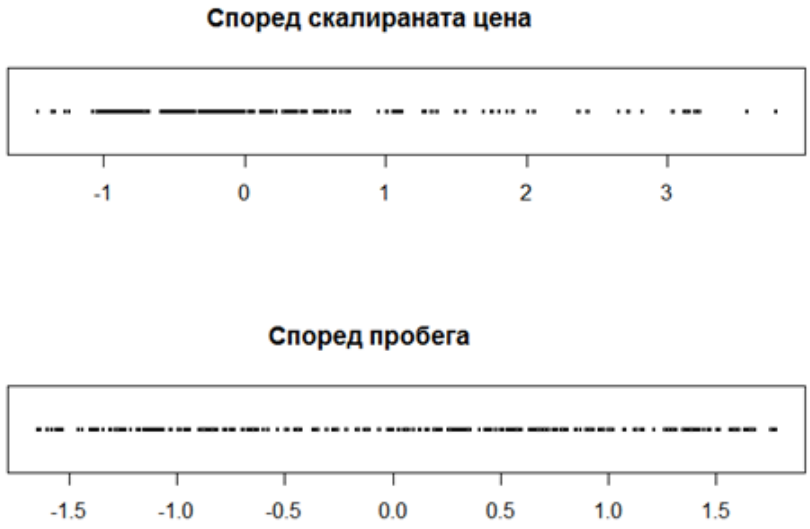
```
> stripchart(scale(Height), cex = 0.2, main = "Височина на студента ", xlab = "см")
> stripchart(scale(Weight), cex = 0.2, main = "Тежест на студента ", xlab = "кг")
```



И по двата признака имаме съвсем слабо изразена асиметрия, като студентите са почти равномерно разпределени в наблюдаваните интервали.

От таблицата с данни tires да сравним, например разпределението на гумите според X4 и X5.

```
> stripchart(scale(X4),cex = 0.2, main = "Според скалираната цена в лв. ", xlab = "лв.")
> stripchart(scale(X5),cex = 0.2, main = "Според пробега в км. ", xlab = "лв.")
```

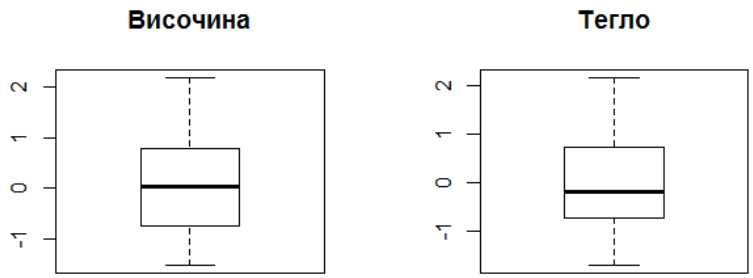


При разпределението на гумите според тяхната цена имаме дясна асиметрия, т.е. струпване на наблюденията около по-малките значения на признака, докато според пробега разпределението е почти симетрично и наблюденията са почти равномерно разпределения в целия интервал от наблюдавани стойности.

3.2 Сравняване чрез графики с мустачки.

Вече знаем, че единият начин за сравняване на разпределения е посредством техните графики с мустачки. Добре е преди това да сме центрирали и нормирали случайните величини. Т.е. от всяко наблюдение да сме извадили средното на извадката и да сме разделили на стандартното отклонение. Това може да бъде направено с функцията ***scale***. Например:

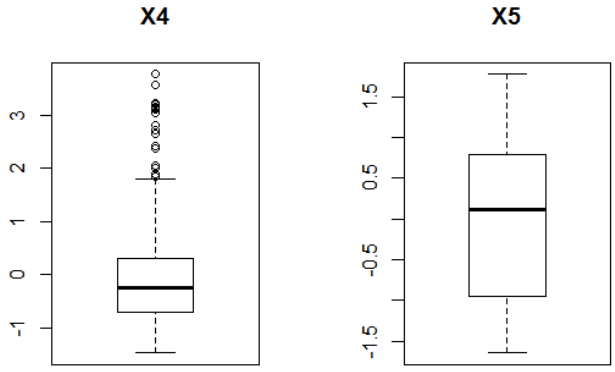
```
> par(mfrow = c(1, 2))     # построяваме си матрица от 1 ред и два колони, в която ще разположим
                           # графиките си.
> boxplot(scale(Height), main = "Височина")
> boxplot(scale(Weight), main = "Тегло")
```



Тук виждаме, че разпределенията на центрираните и нормирани височина и тегло са почти еднакви.

По аналогичен начин ще наблюдаваме, че разпределението на гумите, според цената им е много по-дясноасиметрично от разпределението на гумите според техния пробег.

```
> par(mfrow = c(1, 2))     # построяваме си матрица от 1 ред и два колони, в която ще разположим
                           # графиките си.
> boxplot(scale(X4), main = "X4")
> boxplot(scale(X5), main = "X5")
```



3.3 Сравняване чрез емпирични функции на разпределение.

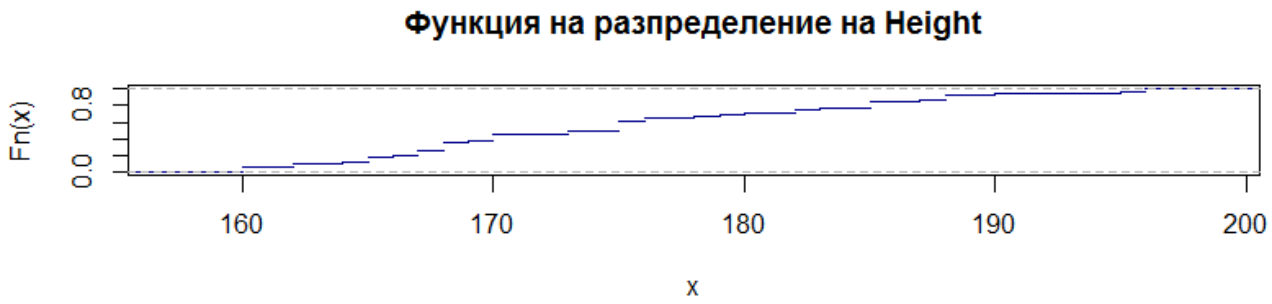
Сравняване на количествени величини и техните разпределения обикновено става чрез изчертаването на емпиричните им функции на разпределение. Да припомним, че емпирична функция на разпределение наричаме

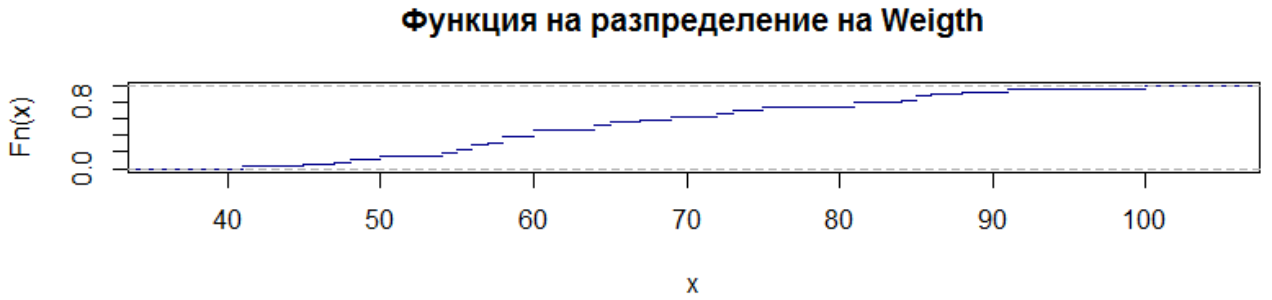
$$F_n(x) = \frac{\text{Брой наблюдения със стойност по – малка от x}}{\text{Обем на извадата}}$$

Стойностите на емпиричната функция на разпределение се определяха с помощта на функцията ***ecdf***.

Да сравним, например измерените значения на признаците Height и Weight чрез техните емпирични функции на разпределение.

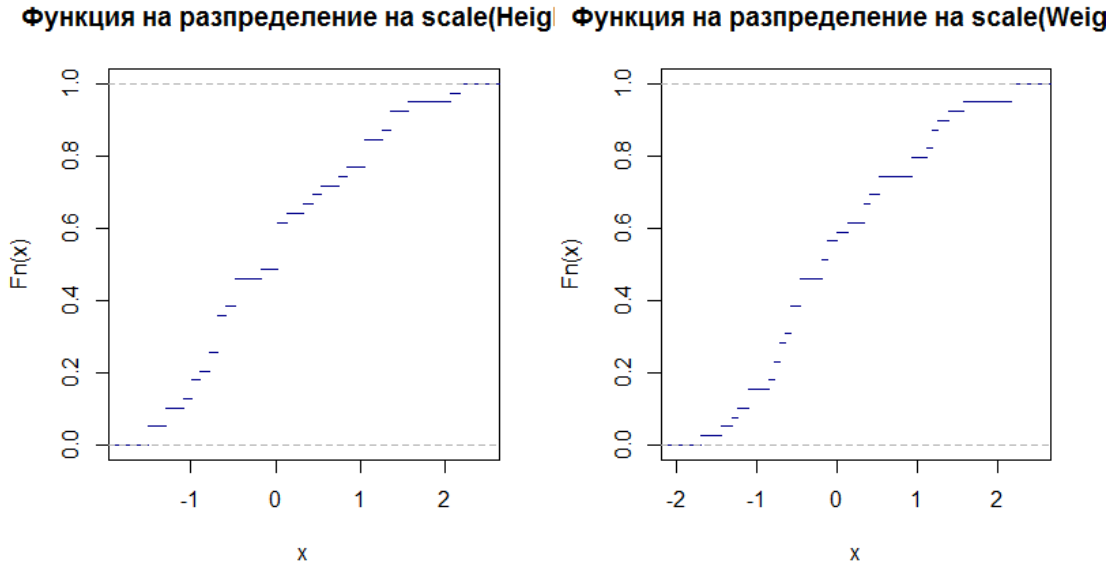
```
> par(mfrow = c(2, 1))
> plot(ecdf(Height), verticals = FALSE, col = "darkblue", do.points = FALSE, lwd = 1, main = "Функц
ия на разпределение на Height")
> plot(ecdf(Weight), verticals = FALSE, col = "darkblue", do.points = FALSE, lwd = 1, main = "Функ
ция на разпределение на Weigth")
```





При построяването на тези графики не сме центрирали и нормирали случайните величини, за които те са изчертани. Ако ще сравняваме типовете на разпределенията е добре преди това да центрираме и нормираме случайните си величини. Така по абсцисната ос ще получим по-близки значения на наблюдаваните признаци.

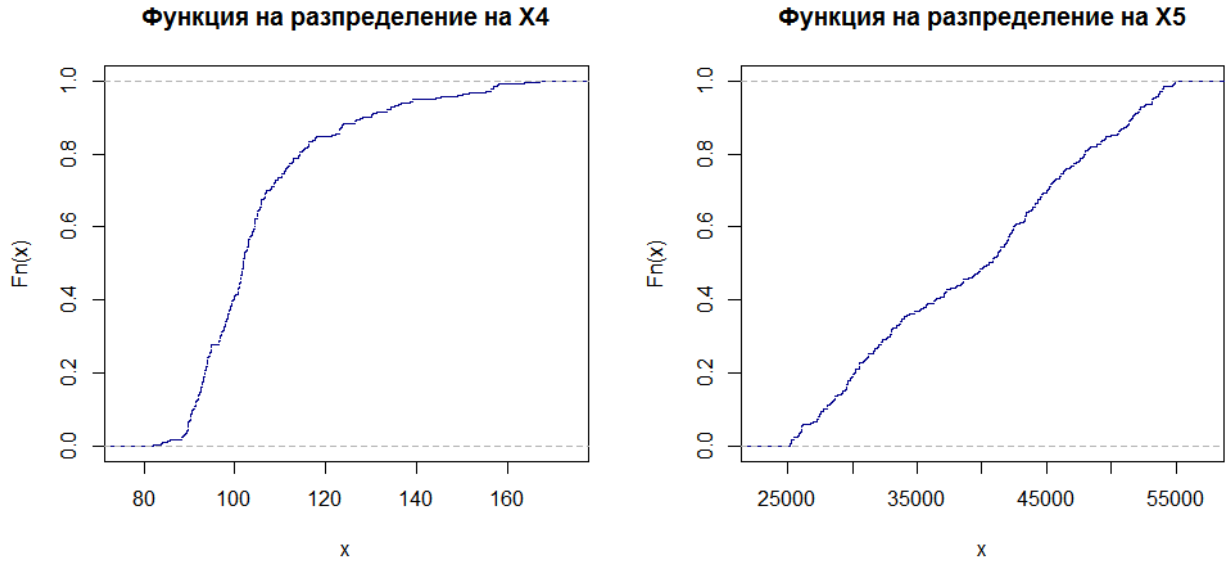
```
> par(mfrow = c(1, 2))
> plot(ecdf(scale(Height)), verticals = FALSE, col = "darkblue", do.points = FALSE, lwd = 1, main = "Функция на разпределение на scale(Height)")
> plot(ecdf(scale(Weight)), verticals = FALSE, col = "darkblue", do.points = FALSE, lwd = 1, main = "Функция на разпределение на scale(Weight)")
```



Отново наблюдаваме, че разпределенията на центрираните и нормирани височина и тегло са почти еднакви.

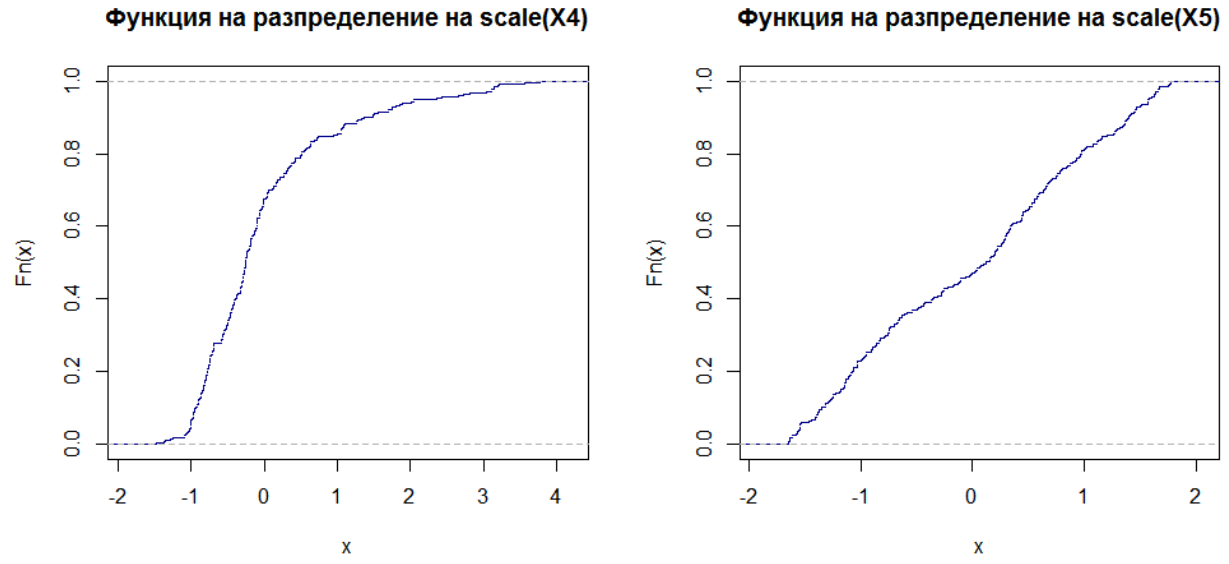
Да сравним сега X4 и X5 чрез техните емпирични функции на разпределение.

```
> par(mfrow = c(2, 1))
> plot(ecdf(X4), verticals = FALSE, col = "darkblue", do.points = FALSE, lwd = 1, main = "Функция на разпределение на X4")
> plot(ecdf(X5), verticals = FALSE, col = "darkblue", do.points = FALSE, lwd = 1, main = "Функция на разпределение на X5")
```



На тези графики не сме центрирали и нормирали случайните величини, за които те са изчертани. Ако ще сравняваме разпределенията на X4 и X5, и по-точно типовете им, трябва преди това да центрираме и нормираме случайните си величини. Така по абсцисната ос ще получим по-близки значения на наблюдаваните признаци. Това може да стане с:

```
> par(mfrow = c(1, 2))
> plot(ecdf(scale(X4)), verticals = FALSE, col = "darkblue", do.points = FALSE, lwd = 1, main = "Функция на разпределение на scale(X4)")
> plot(ecdf(scale(X5)), verticals = FALSE, col = "darkblue", do.points = FALSE, lwd = 1, main = "Функция на разпределение на scale(X5)")
```



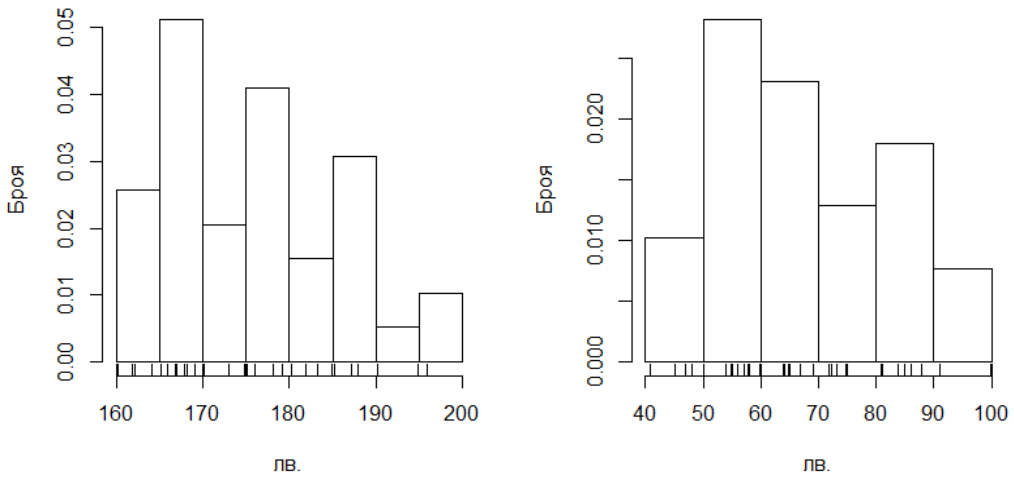
Забелязва се, че нарастването на емпиричната функция на разпределение на цената на гумите е много по-бързо при малките стойности и много по-бавно при големите стойности. Това показва,

че наблюдаваните гуми имат много по-често срещани по-малки цени, в сравнение със средното и по-рядко се случват по-големи цени, които обаче значително се отличават от средното за да компенсират струпванията около по-малките цени. Т.е. имаме дясно асиметрично разпределение. Същия извод направихме и чрез графиката с мустачки.
Функцията на разпределение на пробега нараства много по-равномерно. Това означава, че колите са почти равномерно разпределени според своя пробег в интервала от минимума до максимума.

3.4 Сравняване чрез хистограми.

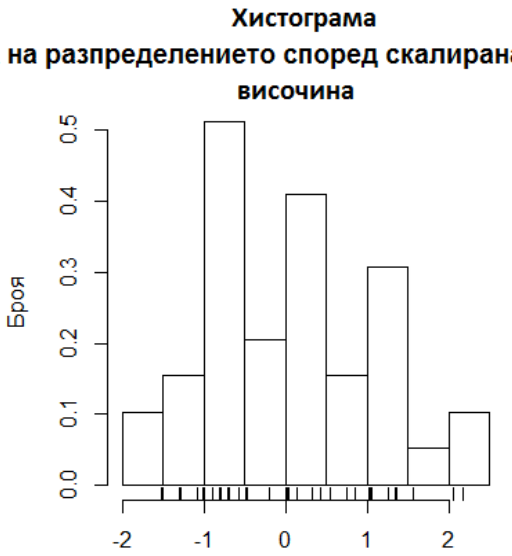
Сравняване на два количествени признака може да стане и чрез техните хистограми, като е добре да добавим и маркери за точните стойности на наблюденията.

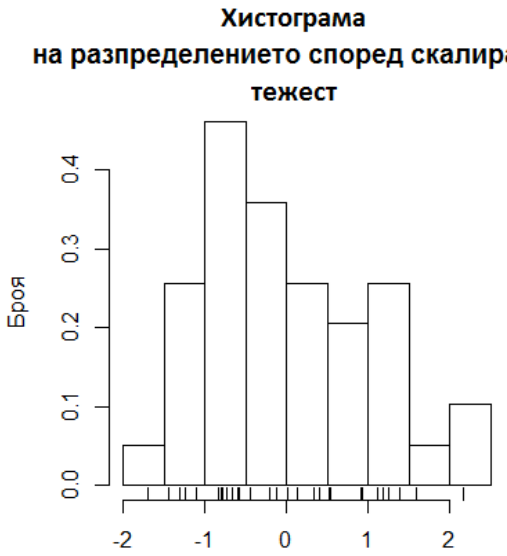
```
> par(mfrow = c(1, 2))
> hist(Height, probability = TRUE, right = FALSE, main = "Хистограма на разпределението според височината в см. ", xlab = "лв.", ylab = "Броя")
> rug(jitter(Height))
> hist(Weight, probability = TRUE, right = FALSE, main = "Хистограма на разпределението според тежестта в кг. ", xlab = "лв.", ylab = "Броя")
> rug(jitter(Weight))
```



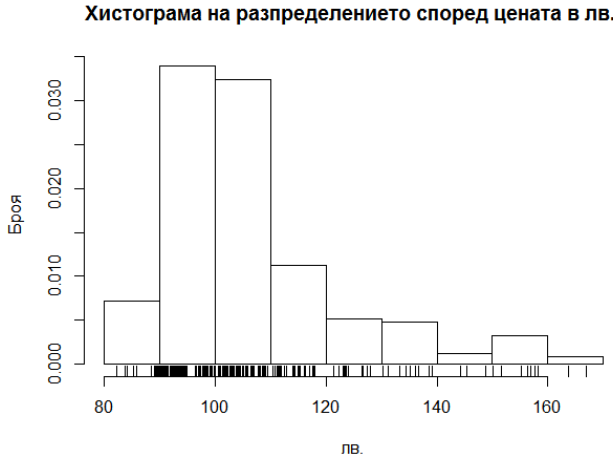
За да имаме еднакви скали по абсцисната ос е добре първо да центрираме и нормираме данните с функцията ***scale***.

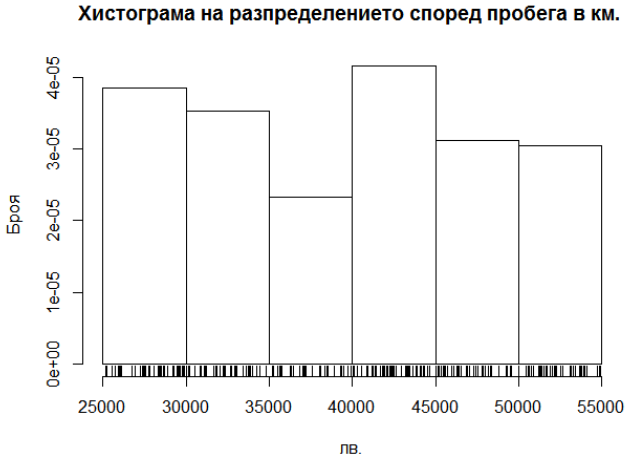
```
> par(mfrow = c(1, 2))
> hist(scale(Height), probability = TRUE, right = FALSE, main = "Хистограма на разпределението според скалираната височина.", xlab = " ", ylab = "Броя")
> rug(jitter(scale(Height)))
> hist(scale(Weight), probability = TRUE, right = FALSE, main = "Хистограма на разпределението според скалираната тежест. ", xlab = " ", ylab = "Броя")
> rug(jitter(scale(Weight)))
```






```
> par(mfrow = c(1, 2))
> hist(X4, probability = TRUE, right = FALSE, main = "Хистограма на разпределението според цената в лв. ", xlab = "лв.", ylab = "Броя")
> rug(jitter(X4))
> hist(X5, probability = TRUE, right = FALSE, main = "Хистограма на разпределението според пробега в км. ", xlab = "лв.", ylab = "Броя")
> rug(jitter(X5))
```

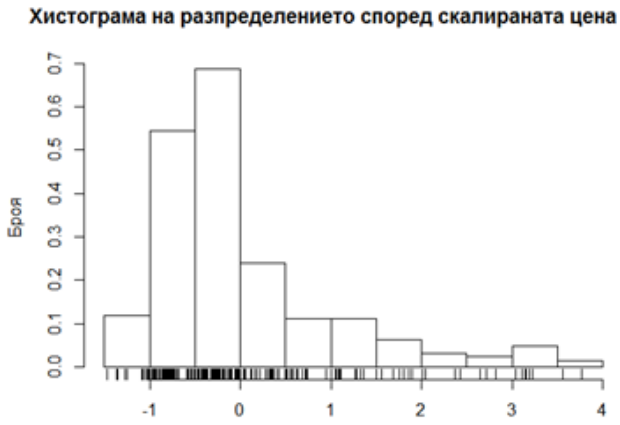


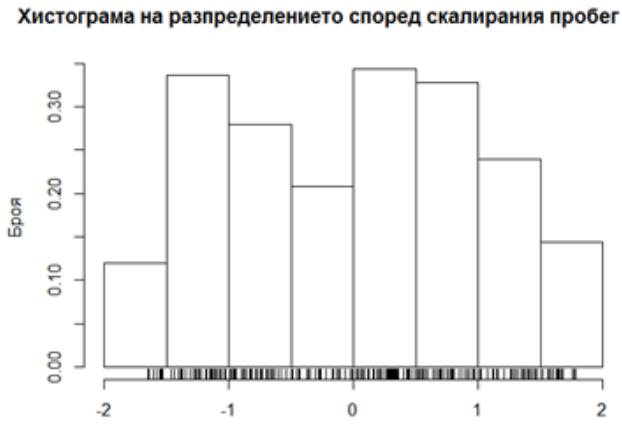


Отново забелязваме дясна асиметрия при разпределението според цената и почти равномерно разпределение на гумите според техния пробег.

Както в предните точки, за да имаме еднакви скали по абсцисната ос първо е добре да центрираме и нормираме данните с функцията ***scale***.

```
> par(mfrow = c(1, 2))
> hist(scale(X4), probability = TRUE, right = FALSE, main = "Хистограма на разпределението според скалираната цена. ", xlab = " ", ylab = "Броя")
> rug(jitter(scale(X4)))
> hist(scale(X5), probability = TRUE, right = FALSE, main = "Хистограма на разпределението според скалирания пробег. ", xlab = " ", ylab = "Броя")
> rug(jitter(scale(X5)))
```



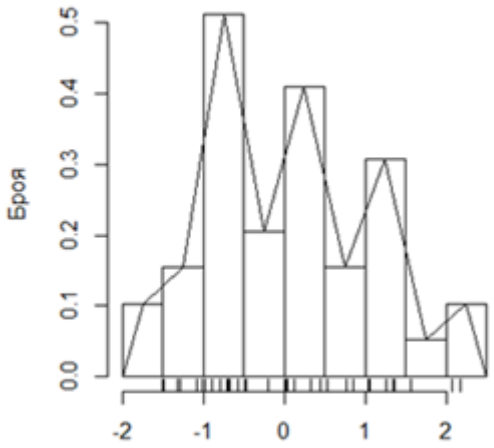


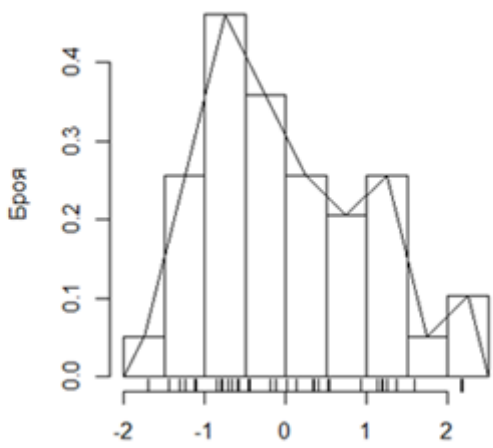
Така успяваме да забележим не само дясната асиметрия на разпределението на гумите според цената, но и да оформим хипотезата, че това разпределение ще има тежка дясна опашка. При пробегът разпределението е по-скоро симетрично и почти равномерно.

3.5 Сравняване чрез полигони.

Можем да сравним две емпирични разпределения е чрез сравняване на техните полигони.

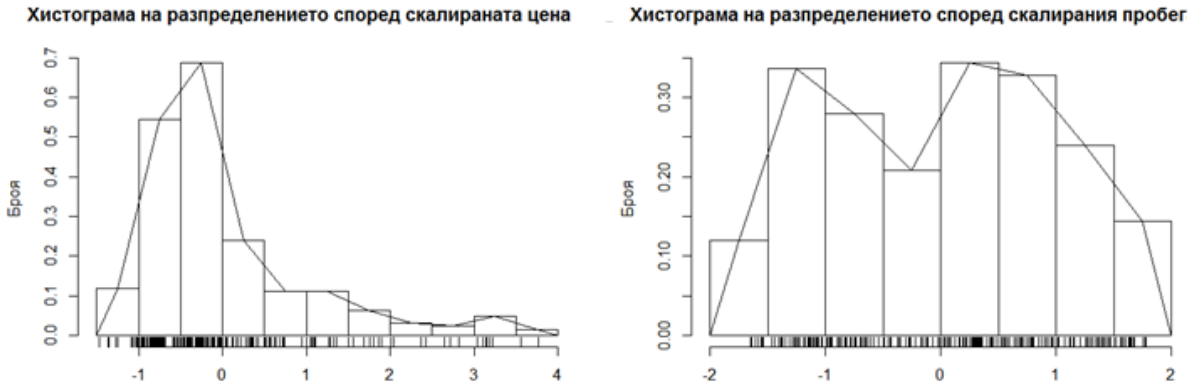
```
> par(mfrow = c(1, 2))
> tmp  = hist(scale(Height), probability = TRUE, right = FALSE, main = "Хистограма на разпределението според скалираната височина. ", xlab = " ", ylab = "Броя")
> rug(jitter(scale(Height)))
> lines(c(min(tmp$breaks), tmp$mids, max(tmp$breaks)), c(0, tmp$density, 0), type="l")

> tmp  = hist(scale(Weight), probability = TRUE, right = FALSE, main = "Хистограма на разпределението според скалираната тежест. ", xlab = " ", ylab = "Броя")
> rug(jitter(scale(Weight)))
> lines(c(min(tmp$breaks), tmp$mids, max(tmp$breaks)), c(0, tmp$density, 0), type="l")
```






```
> par(mfrow = c(1, 2))
> tmp  = hist(scale(X4), probability = TRUE, right = FALSE, main = "Хистограма на разпределението според скалираната цена. ", xlab = " ", ylab = "Броя")
> rug(jitter(scale(X4)))
> lines(c(min(tmp$breaks), tmp$mids, max(tmp$breaks)), c(0, tmp$density, 0), type="l")
```

```
> tmp  = hist(scale(X5), probability = TRUE, right = FALSE, main = "Хистограма на разпределението според скалирания пробег. ", xlab = " ", ylab = "Броя")
> rug(jitter(scale(X5)))
> lines(c(min(tmp$breaks), tmp$mids, max(tmp$breaks)), c(0, tmp$density, 0), type="l")
```



3.6 Сравняване чрез плътности.

Подобно сравнение може да бъде направено и с помощта на плътностите. Те се изчертаваха чрез функциите

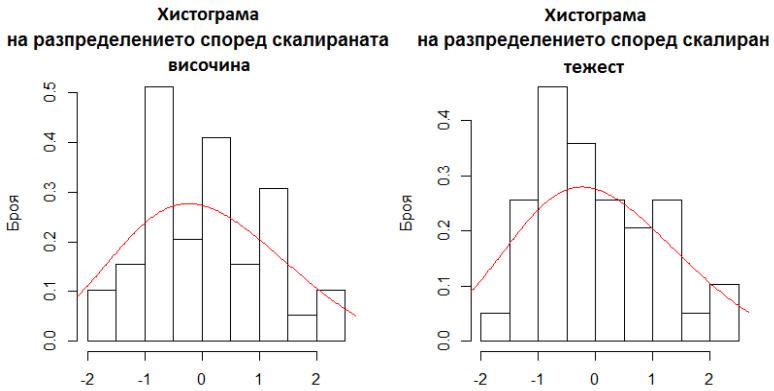
lines и ***density***,

като преди това трябва да сме използвали функцията

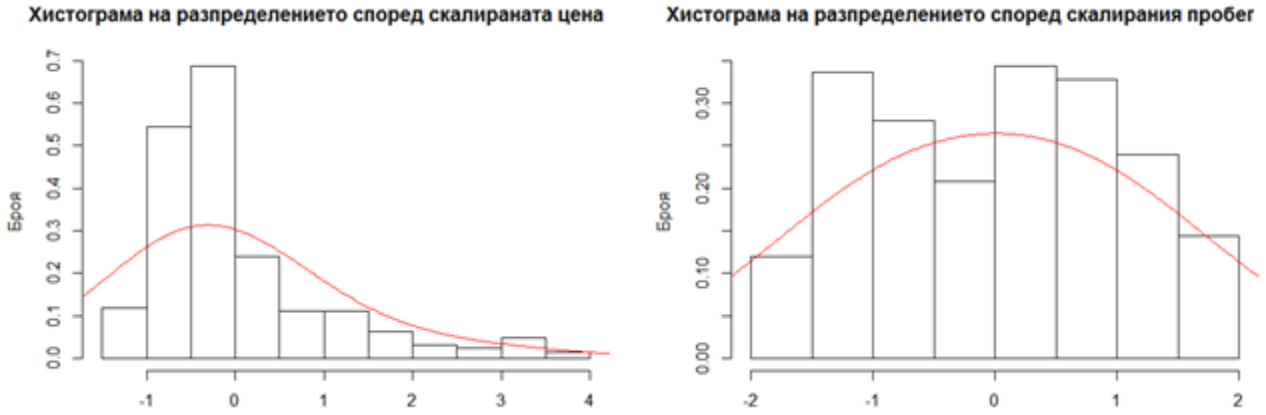
hist.

При оценяването на плътността на наблюдаваната величина се използва осредняване по подинтервали. Ширините на тези подинтервали се задават в параметъра ***bw***, чието съкращение идва от bandwidth. Колкото ширината на подинтервалите е по-голяма, толкова приближаващата плътността крива е по-гладка. Когато правим сравнения трябва да използваме един и същ параметър ***bw***. (Самият алгоритъм за построяването на гладката крива, понякога е доста сложен).

```
> par(mfrow = c(1, 2))
> hist(scale(Height), prob = T, main = "Хистограма на разпределението според скалираната височина. ", xlab = " ", ylab = "Броя")
> lines(density(scale(Height), bw = 1), col = 'red')
> hist(scale(Weight), prob = T, main = "Хистограма на разпределението според скалираната тежест. ", xlab = " ", ylab = "Броя")
> lines(density(scale(Weight),bw = 1),  col = 'red')
```



```
> par(mfrow = c(1, 2))
> hist(scale(X4), prob = T, main = "Хистограма на разпределението според скалираната цена. ", xlab = " ", ylab = "Броя")
> lines(density(scale(X4), bw = 1), col = 'red')
> hist(scale(X5), prob = T, main = "Хистограма на разпределението според скалирания пробег. ", xlab = " ", ylab = "Броя")
> lines(density(scale(X5),bw = 1),  col = 'red')
```



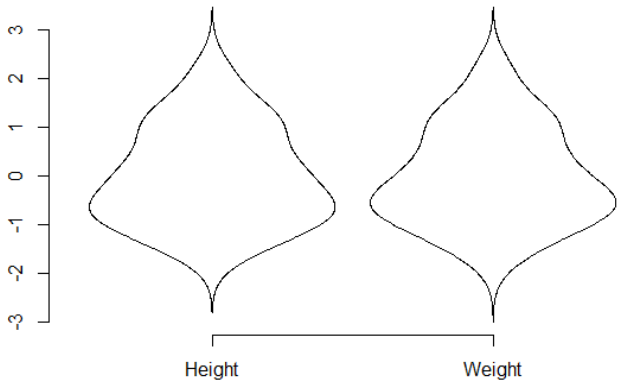
Към края на нашия курс ще научим и начини за сравняване на теоретични и емпирични разпределения.

За да се подчертаят различията във плътността тя може да бъде изчертана огледално поотделно за всяка наблюдавана величина. Това става с функцията ***simple.violinplot*** от библиотеката ***UsingR***.

```
> simple.violinplot(scale(Height), scale(Weight), xaxt = "n")
```

След това добавяме новите имена

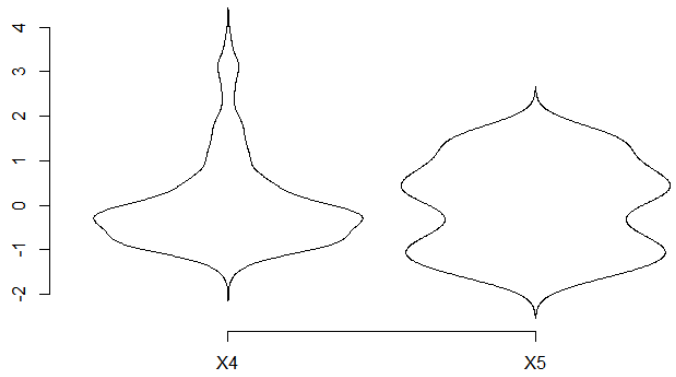
```
> axis(side = 1, at = c(1,2), labels = c("Height","Weight"))
```




```
> simple.violinplot(scale(X4), scale(X5), xaxt = "n")
```

След това добавяме новите имена

```
> axis(side = 1, at = c(1,2), labels = c("X4","X5"))
```



4. Линейна регресия

При наблюдаване на количествени признаци често възниква въпросът за моделиране на формата на зависимост между тях. Той може да бъде решен със средствата на регресионния анализ.

Регресионният анализ започва с избор на линия на регресия. Да приемем, че регресионният модел е

$$Y = f(X, \beta) + \varepsilon, \qquad (*)$$

където β е r-мерен вектор, чиито координати са неизвестни параметри на функцията f, а ε е стохастичната грешка. Т.е. грешка, която се дължи на случайния характер на извадката. Ако моделът е добър, грешките са случайни величини, които са некорелирани с $E\varepsilon = 0$ и $D\varepsilon = \sigma^2$.

X може да бъде вектор от признаци или един единствен признак. Той се нарича независима променлива/и или фактор-признак/ци, а Y се нарича резултативна величина или зависима променлива. Ако фактор-признакът X е един, говорим за единична регресия. Иначе говорим за множествена регресия.

При различните наблюдения имаме реализации на случайните величини X и Y. Ще предполагаме, че наблюденията върху признаците X и Y са независими и извършени при непроменени условия на експеримента. При това, от (*), ако X и ε независими

$$E(Y/X = x) = f(x, \beta). \qquad (**)$$

След построяването на регресионния модел, т.е. след определянето на оценките $\hat{\beta}$ на коефициентите β в уравнение (*), ако знаем X, ще можем да предскажем Y с известна грешка. Получената оценка на E(Y/X = x) ще означаваме с

$$\hat{Y} = f(X, \hat{\beta})$$

и ще наричаме „изгладена стойност на Y“.

Ако функцията f е линейна относно неизвестните параметри β, но не обезателно линейна относно независимите променливи X, говорим за линеен регресионен модел. Иначе моделът се нарича нелинеен.

Оценката на линията на регресията се прави в клас от функции. Това е онази функция g от разглеждания клас, която минимизира средно-квадратичната грешка на Y относно g(X) в класа от функции G.

Да припомним, че ***средноквадратична грешка (Mean Square Error) на η относно g(ξ)*** означаваме с ***MSE(g)*** и това е

$$\boldsymbol{MSE(g)} = E(\eta - g(\xi))^2.$$

От математическа гледна точка, първата задача на регресионния анализ е да се построят най-добри оценки на параметрите на регресия β така, че измежду всички линии с това аналитично представяне, при получените оценки на параметрите, да имаме най-малка сума от квадратите на грешките.

По данните от извадката, използвайки метода на най-малките квадрати, правим оценка на вектора β. Ще я означаваме с $\hat{\beta}$. Тя е такава, че да минимизира сумата от квадратите на грешките

$$\sum_{i=1}^{n} \varepsilon_i^{\,2} = \sum_{i=1}^{n} (Y_i - f(X_i, \beta))^2$$

Намира се като решим относно β, следната система

$$\left| \begin{array}{l} \dfrac{\partial \sum_{i=1}^{n} (Y_i - f(X_i, \beta_0, \ldots, \beta_r))^2}{\partial \beta_i} = 0, \\ i = 1, \ldots, r. \end{array} \right.$$

наречена ***система нормални уравнения***.

Заместваме тези коефициенти $\hat{\beta}$ в уравнението на регресия. От полученото уравнение на регресия пресмятаме оценките на стойностите на зависимата променлива, т.е.

$$\hat{Y} = f(X, \hat{\beta}).$$

След като се определят оценките, $\hat{\beta}$ на параметрите в избрания модел, се прави анализ на остатъците. Това става най-бързо от тяхната диаграма на разсейване. По-точно проверява се дали отклоненията $\varepsilon = Y - \hat{Y}$, на фактическите стойности, Y от техните оценки $\hat{Y}$ имат случаен характер. Дали тези остатъци ε са еднакво разпределени. Дали имат равни дисперсии. С някои от критериите за съгласие, които ще разгледаме по-нататък се проверява дали разпределението им е нормално. Проверява се хипотезата за липса на корелация в остатъчния компонент. Алгоритмите на всички тези проверки ще ги учим в следващите теми.

Ако тези условия са удовлетворени, се прави проверка на хипотезата за статистическата значимост на коефициентите в уравнението на регресия.

Величината

$$S_\varepsilon = \sqrt{\frac{\sum_{i=1}^{n} (\hat{Y}_i - Y_i)^2}{n - r}}$$

се нарича ***стандартна грешка на модела***. Тук r е броят на неизвестните параметри в уравнението на регресия.

Може да тестваме повече от една функция *f*. При всяка от тях ще получаваме различни оценки $\hat{Y}$ и различни стандартни грешки S_ε. Най-добър модел за съответните данни ни дава тази линия, за която сумата от квадратите на отклоненията $\varepsilon_i = Y_i - \hat{Y}_i$ на фактическите (измерените) значения на резултативната величина Y_i от техните оценки $\hat{Y}_i$ е минимална и остатъците удовлетворяват условията:

- $E\varepsilon = 0;$
- ε да имат равни дисперсии;
- да са некорелирани.

Тогава моделът с най-малка стандартна грешка S_ε е най-подходящ за нашите данни.

След намирането на уравнението на регресия можем да получим най-добра оценка за *Y* по зададено значение на *X*.

Този анализ не отчита, че изменението на разглежданите величини може да се дължи на външни, невключени в модела признаци, но измерва силата на зависимост между включените в модела фактори.

Най-общ алгоритъм на простата линейна регресия.

Нека изследваме влиянието на фактора *X* върху резултативния признак *Y*.

- *Изчертаване на корелационно поле на данните*

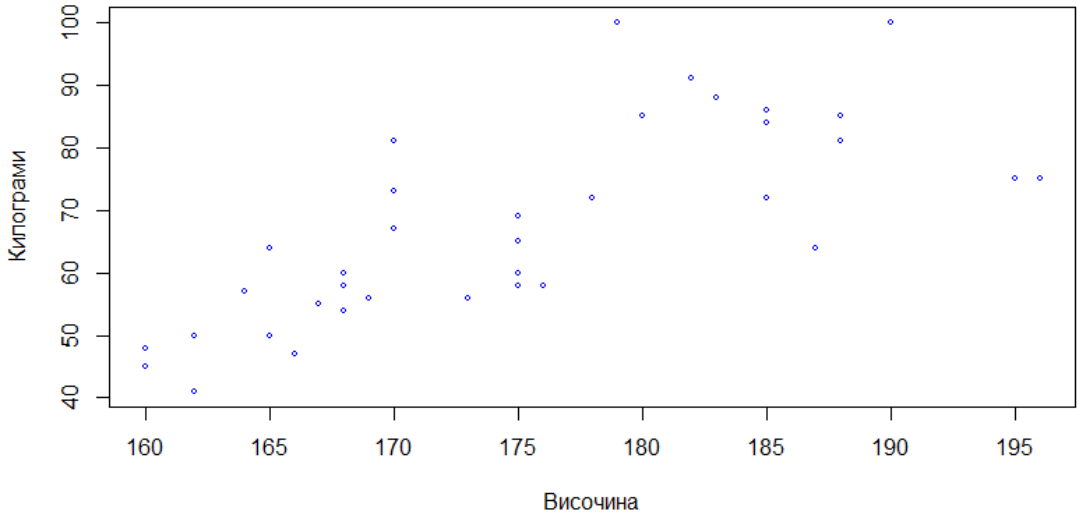
Обикновено се започва с изчертаване на корелационно поле на данните. По абсцисната ос се нанасят измерените значения на фактор-признака *X*, а по ординатната, измерените значения на резултативния признак *Y*. Да предположим, че разполагаме с *n* на брой двойки от наблюдения

$$(X_1,Y_1), (X_2,Y_2), \ldots, (X_n,Y_n).$$

По данните от таблицата SoftwareEngineering и да моделираме влиянието на височината (Height) на студентите, върху тежестта (Weigth) на студентите.

Изчертаване на корелационно поле на данните

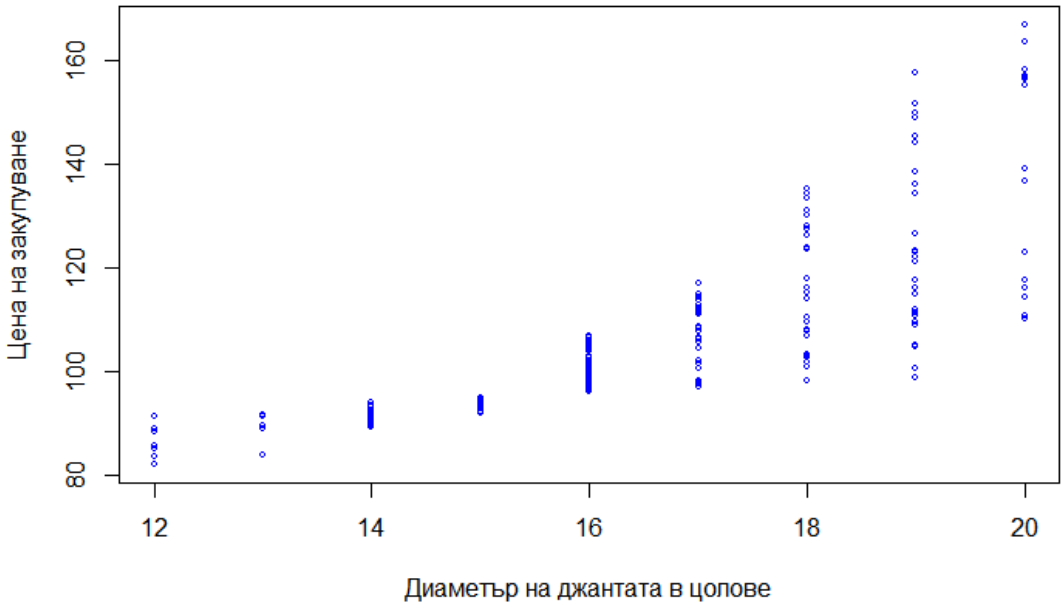
```
> plot(Height, Weight, xlab = " Височина ", ylab = " Килограми ", col = "blue", cex  = 0.5)
```



Нека сега да използваме данните от таблицата tires и да моделираме влиянието на диаметъра (X9) на джантата в цолове, върху цената на гумата (X4).

Изчертаване на корелационно поле на данните

```
> plot(X9, X4, xlab = " Диаметър на джантата в цолове ", ylab = " Цена на закупуване ", col = "blue",
cex  = 0.5)
```



- *Избор на регресионен модел*

Ако точките се групират около права линия, избираме регресионен модел

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i,$$

където β_0 и β_1 са неизвестни параметри, а ε_i са стохастичните грешки, които трябва да са некорелирани, с $E\varepsilon_i = 0$ и $D\varepsilon_i = \sigma^2$.

Да отбележим, че линията на регресия е

$$\hat{Y}_i = \beta_0 + \beta_1 X_i.$$

Забележка: Във втория пример, от графиката е ясно, че условието за равенство на дисперсиите на остатъците не е изпълнено, защото дисперсията на цената нараства с нарастването на диаметъра. Нататък алгоритъмът дава добри резултати само ако предните условия са удовлетворени. Т.е. за да имаме добри резултати трябва да трансформираме по подходящ начин данните. Това, обаче ще направим на по-късен етап от нашето обучение. Т.е. за тези данни простата линейна регресия не е подходящ модел.

- *Определяне на коефициентите в линията на регресия*

Най-добрите оценки на β_0 и β_1 се получават по метода на най-малките квадрати. Те са такива, че минимизират сумата от квадратите на отклоненията

$$\sum_{i=1}^{n} \varepsilon_i^{\,2} = \sum_{i=1}^{n} (Y_i - \beta_0 - \beta_1 X_i)^2 .$$

Ако оценъчните стойности са $\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X$, то те трябва да са такива, че минимумът на горната сума да се достига при тях, т.е. той трябва да бъде

$$\sum_{i=1}^{n} \hat{\varepsilon}_i^{\,2} = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 = \sum_{i=1}^{n} (Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i)^2 .$$

Да означим тези оценки с $\hat{\beta}_0$ и $\hat{\beta}_1$. Намираме ги от системата нормални уравнения, която в случая има вида

$$\begin{cases} \sum_{i=1}^{n} Y_i = \hat{\beta}_0 n + \hat{\beta}_1 \sum_{i=1}^{n} X_i \\ \sum_{i=1}^{n} Y_i X_i = \hat{\beta}_0 \sum_{i=1}^{n} X_i + \hat{\beta}_1 \sum_{i=1}^{n} X_i^2 \end{cases}.$$

Решението на тази система е:

$$\begin{cases} \hat{\beta}_0 = \overline{Y}_n - \hat{\beta}_1 \overline{X}_n \\ \hat{\beta}_1 = \dfrac{\sum_{i=1}^{n} (X_i - \overline{X}_n)^2 Y_i}{\sum_{i=1}^{n} (X_i - \overline{X}_n)^2} = \dfrac{\sum_{i=1}^{n} (X_i - \overline{X}_n)(Y_i - \overline{Y}_n)}{\sum_{i=1}^{n} (X_i - \overline{X}_n)^2} = \dfrac{S_{XY}}{S_X^2} \end{cases}$$

Ако гледаме на тези оценки като на случайни величини, те са неизместени[1] и при определени условия имат минимална дисперсия, в сравнение с всички неизместени оценки, линейно зависещи от $Y_1, Y_2, \ldots, Y_n$.

Оценката на уравнението на регресия е

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X.$$

Тя минава през точката с координати $(\overline{X}_n, \overline{Y_n})$.

Коефициентът $\hat{\beta}_1$ в това уравнение показва, с колко единици средно, в приетата за резултативния признак Y мярка, би се изменил той, ако изменим фактор-признака X с една единица, в приетата за X мярка. Когато зависимостта на резултативния признак от фактор-признака е правопропорционална, коефициентът $\hat{\beta}_1$ е положителен. Обратно, ако тази зависимост е обратнопропорционална, този коефициент е отрицателен. Коефициентът $\hat{\beta}_0$ е равен на ординатата на точката, в която линията на регресия пресича ординатната ос. Линията на регресия ще е успоредна на абсцисната ос ако значенията на резултативния признак не се влияят от тези на фактор-признака.

R изчертава тези графики по следния начин. Първо, с помощта на функцията ***plot*** изчертаваме точките от корелационното поле както по-горе. После чрез функцията ***lm*** намираме стойностите на коефициентите $\hat{\beta}_0$ и $\hat{\beta}_1$. Тази функция дава възможност, с помощта на оператора "~" да се зададе регресионния модел. От ляво на този оператор стои зависимата променлива (в случая Height или X9), а отдясно независимата променлива (в случая това е Weight или X4). По-подробна информация за оценките на коефициентите можем да получим чрез функцията ***summary***. След това се добавя линията на регресия към графиката с корелационното поле на данните. Последното може да стане с помощта на функциите ***abline*** или ***lines***.

Нека, по данните за студентите, да моделираме формата на зависимостта между ръста и теглото с права линия.

```
> plot(Height, Weight, xlab = " Височина ", ylab = " Тегло ", col = "blue", cex = 0.5)
> Model = lm(Height ~ Weight)
> summary(Model)
```

```
Call:
lm(formula = Height ~ Weight)

Residuals:
```

[1] Една оценка е неизместена ако математическото очакване на оценката е равно на оценяваната величина.

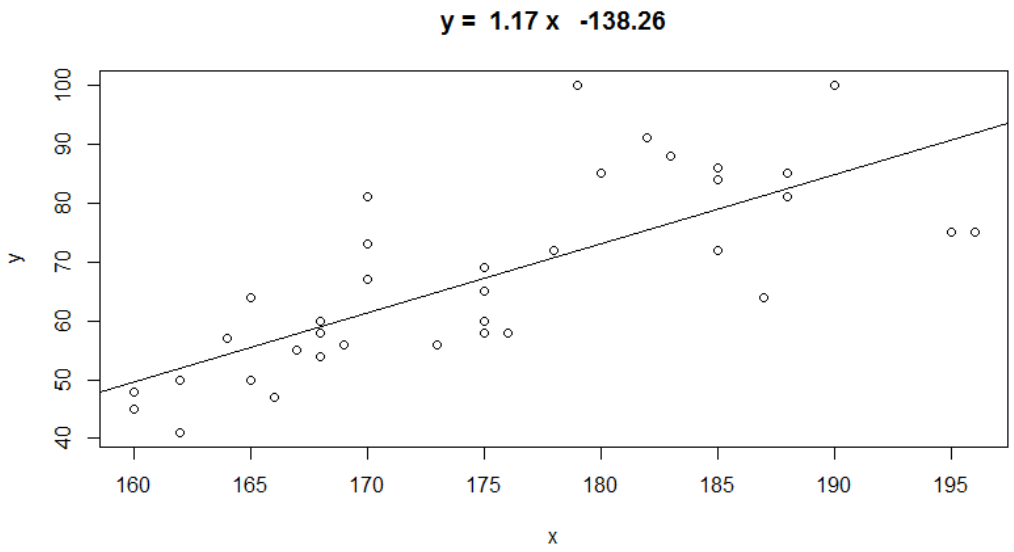
```
    Min      1Q  Median      3Q     Max            # това са минимум, I квартил, медиана, III
-11.7453 -3.7861 -0.7243  2.7911 17.3684     # квартил, максимум на остатъците епсиолон

Coefficients:
             Estimate   Std. Error  t value   Pr(>|t|)
(Intercept) 142.29046    4.75514   29.924    < 2e-16 ***   # това са оценките β̂0 стандартната
Weight        0.48455    0.06933    6.989    2.93e-08 ***  # грешка и p-value, т.е. лицето под
---                                                        # нормалната крива и извън ± t value.
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1          # това са кодовете, с които се четат
                                                           # резултатите от проверките за
                                                           #статистическа значимост на кефициентите.
Residual standard error: 6.51 on 37 degrees of freedom   # това е стандартната грешка на
                                                           # остатъците
Multiple R-squared:  0.569,      Adjusted R-squared:  0.5574  # това е коефициентът на
                                                              # детерминация
F-statistic: 48.85 on 1 and 37 DF,  p-value: 2.928e-08        # това е резултатът от критерия
                                                              # на  Фишър
>  abline(lm(Height ~ Weight), col = "red")  # Добавя червената линия на регресия на
                                                              # корелационното поле
```

Като алтернатива на съвкупността от горни функции може да бъде използвана функцията

simple.lm

от библиотеката ***UsingR.*** По подразбиране тя връща оценките на коефициентите и горната графика в черно.

```
> library(UsingR)
> simple.lm(Height, Weight)
Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
   -138.256        1.174
```



За да извлечем например коефициентите в уравнението на регресия първо трябва да присвоим резултата от тази функция на променлива и после да вземем само компонентата, която съдържа коефициентите. Това можем да направим с $coefficients или с помощта на функцията ***coef***.

Всички компоненти на резултата могат да бъдат разгледани с функцията ***ls.***

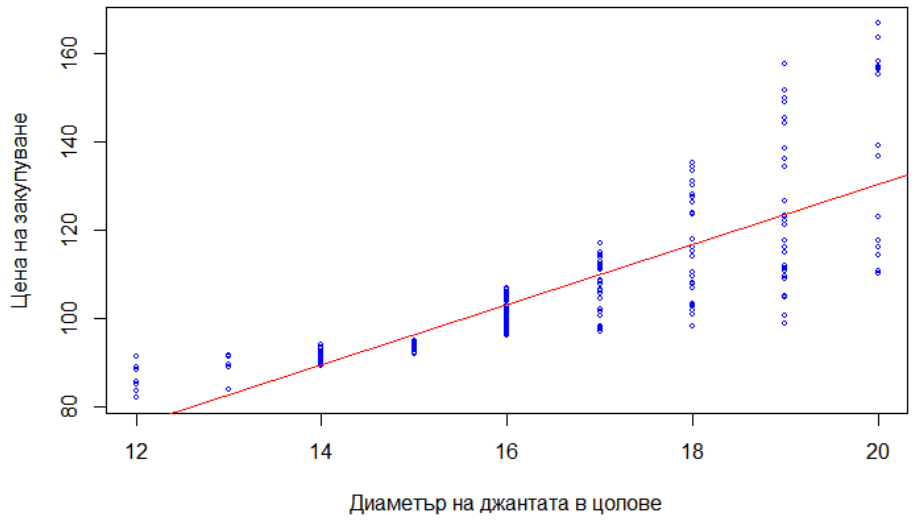
```
> Myresult = simple.lm(Height, Weight)
> coef(Myresult)              # извеждаме коефициентите от уравнението на регресия
          (Intercept)          x
          -138.256417    1.174347
> ls(Myresult) # разглеждаме компонентите на резултата от регр. анализ, направен с simple.lm
        [1] "assign"        "call"          "coefficients"  "df.residual"
        [5] "effects"       "fitted.values" "model"         "qr"
        [9] "rank"          "residuals"     "terms"         "xlevels"
> Myresult$coefficients       # извеждаме коефициентите от уравнението на регресия
            (Intercept)          x
           -138.256417    1.174347
 > Myresult$coefficients[1] # извеждаме първия от коефициентите от уравнението на регресия
         (Intercept)
        -138.256417
> Myresult$coefficients[2] # извеждаме втория от коефициентите от уравнението на регресия
              x
       1.174347
```

Като заместим измерените значения на фактор-признака X, в уравнението на регресия, намираме съответните оценки $\hat{Y}$ за значенията на резултативния признак Y. Сумата и съответно средната аритметична на тези оценки е равна на съответната характеристика на изходните данни. Например, според построения регресионен модел, при ръст 170 см., очакваното тегло е 60.64 кг.

```
> Y170 = -138.26 + 1.17*170
> Y170
       [1] 60.64
```

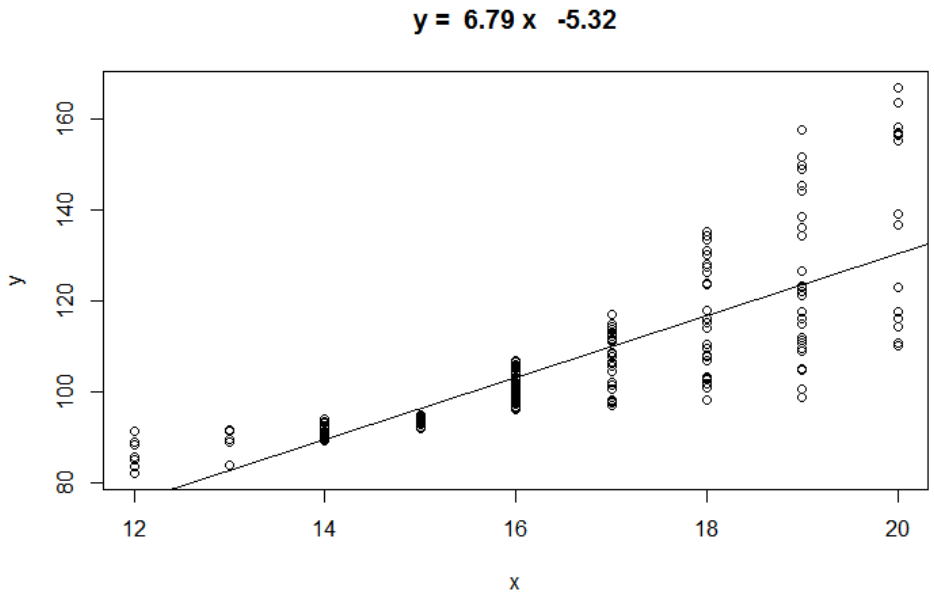
Нека, по данните от файла tires, да моделираме формата на зависимостта на цена на закупуване от диаметъра на джантата в цолове, с права линия.

```
> plot(X9, X4, xlab = " Диаметър на джантата в цолове ", ylab = " Цена на закупуване ", col = "blue", cex = 0.5)
> My_model = lm(X4 ~ X9)
> summary(My_model)
        Call:
        lm(formula = X4 ~ X9)
        Residuals:
           Min       1Q     Median     3Q     Max      # това са мин., I квартил, медиана, III квартил,
                                                        # максимум на остатъците епсилон
        -24.660  -4.166  -1.341   2.591  36.530
        Coefficients:
                        Estimate     Std. Error      t value     Pr(>|t|)
         (Intercept)    -5.3225      5.2314          -1.017       0.31        #това са оценката β̂0 , стандартната
                                                                 #й грешка (обяснена е по-долу),  и p-value то, т.е.
                                                                 # лицето под съответната нормалната крива
                                                                          # и извън ± t value - то.

        X9              6.7896      0.3171           21.415    <2e-16 ***#това са оценката β̂1 , стандартната
                                                                 #й грешка(обяснена е по-долу), и p-value - то, т.е.
                                                                 # лицето под съответната нормална крива
                                                                          # и извън ± t value -то.
        ---
        Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1      # това са кодовете, с които се четат
                                                                             # резултатите от проверките за ста-
                                                                             # тистическите значимости на
                                                                             # коефициентите по-горе.
        Residual standard error: 9.624 on 248 degrees of freedom      #Това е стандартната грешка на
                                                                      # остатъците, т.е. S на епсилоните
                                                                      # (обяснена е по-долу)
        Multiple R-squared:  0.649,      Adjusted R-squared:  0.6476  #Това е коефициентът на детерми-
                                                                      # нация (определеност), т.е. квад-
                                                                   # ратът на корелационния коефициент между X4 и X9,
                                                                   # обяснен по-долу и показваш каква част от вариацията на
                                                                   # зависимата променлива X4 се определя от изменения в
                                                                   #  независимата променлива X9.
        F-statistic: 458.6 on 1 and 248 DF,  p-value: < 2.2e-16          #Това е резултатът от критерия на
                                                                        # Фишер, който също е обяснен по-долу.
>  abline(lm(X4 ~ X9), col = "red")  # Добавя червената линия на регресия на корелационното поле
```



Като алтернатива на съвкупността от горни функции може да бъде използвана функцията ***simple.lm*** от библиотеката ***UsingR.*** По подразбиране тя връща оценките на коефициентите и горната графика в черно.

```
> library(UsingR)
> simple.lm(X9,X4)
Call:
lm(formula = y ~ x)
Coefficients:
(Intercept)           x
     -5.323        6.790
```



За да извлечем например коефициентите в уравнението на регресия първо трябва да присвоим резултата от тази функция на променлива и после да вземем само компонентата, която съдържа коефициентите. Това можем да направим с $coefficients или с помощта на функцията

coef.

Всички компоненти на резултата могат да бъдат разгледани с функцията ***ls.***

```
> Myresult = simple.lm(X9,X4)
> coef(Myresult)          # извеждаме коефициентите от уравнението на регресия
      (Intercept)           x
       -5.322501     6.789611
> ls(Myresult) # разглеждаме компонентите на резултата от регр. анализ, направен с simple.lm
       [1] "assign"       "call"         "coefficients"  "df.residual"
       [5] "effects"      "fitted.values" "model"        "qr"
       [9] "rank"         "residuals"    "terms"         "xlevels"
> Myresult$coefficients # извеждаме коефициентите от уравнението на регресия
      (Intercept)           x
       -5.322501     6.789611
> Myresult$coefficients[1] # извеждаме първия от коефициентите от уравнението на регресия
      (Intercept)
       -5.322501
> Myresult$coefficients[2] # извеждаме втория от коефициентите от уравнението на регресия
            x
     6.789611
```

Като заместим измерените значения на фактор-признака *X*, в уравнението на регресия, намираме съответните оценки $\hat{Y}$ за значенията на резултативния признак *Y*. Сумата и съответно средната аритметична на тези оценки е равна на съответната характеристика на изходните данни от наблюдения върху Y.

Например, според построения регресионен модел, при диаметър 16, очакваната цена е 103.32 лв.

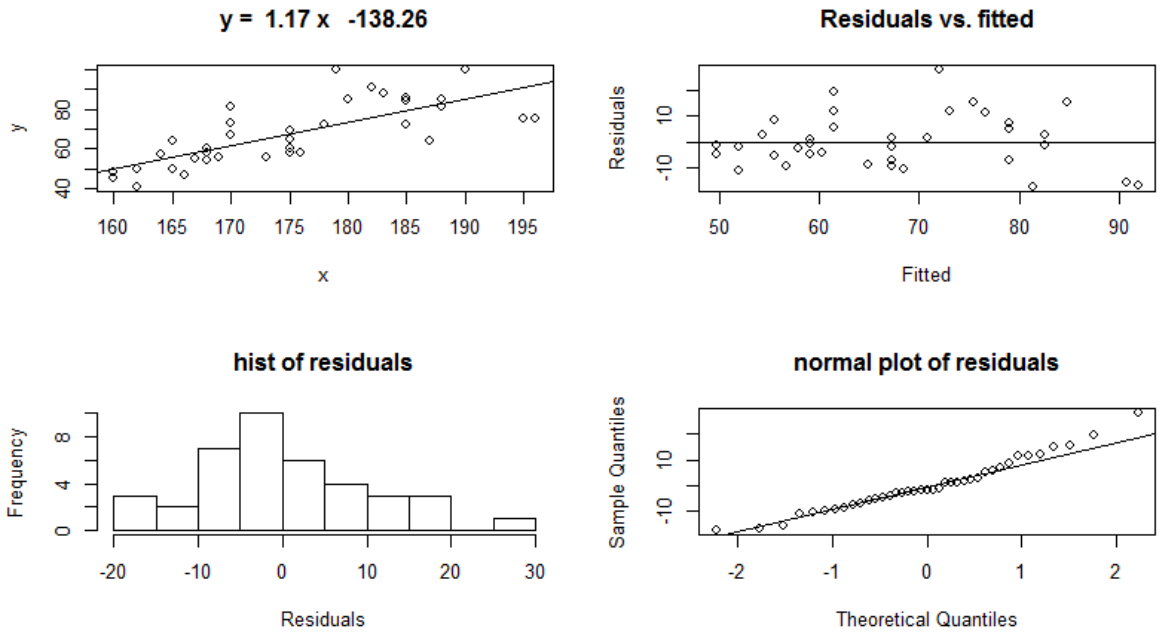
```
> Y16 = -5.32 + 6.79*16
> Y16
      [1] 103.32
```

- *Графика и анализ на остатъците*

Една твърде полезна графика, по която можем да преценим дали един регресионен модел е подходящ за данните е графиката на остатъците. Тя, заедно с корелационното поле на данните, хистограмата на остатъците и normal plot на остатъците може да бъде получена, когато във функцията ***simple.lm*** зададем параметърът ***show.residuals*** да бъде TRUE.

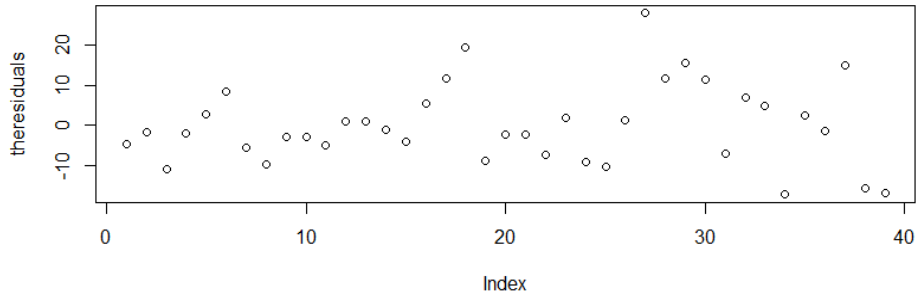
В първия пример за студентите.

```
> Myresult = simple.lm(Height, Weight, show.residuals = TRUE)
```

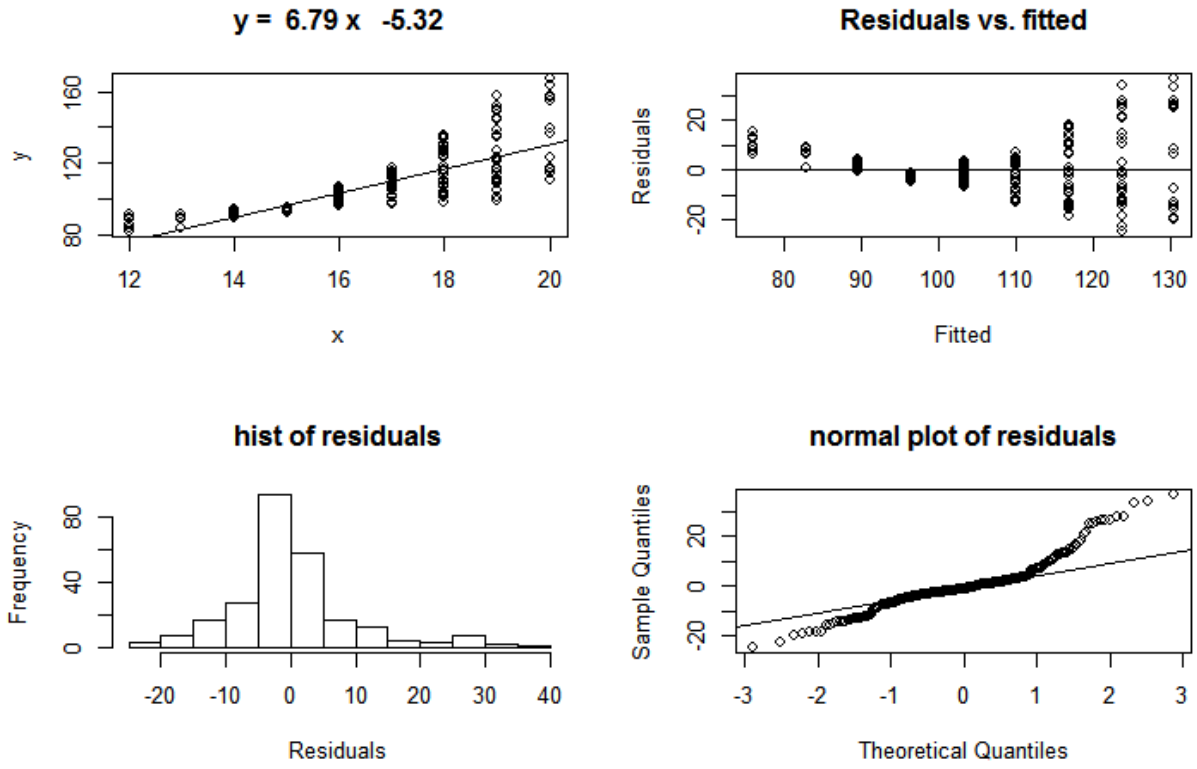


Достъпът до остатъците на регресионния модел и тяхната графика могат да бъдат постигнати и с последователно използване на функциите ***resid*** и ***plot***.

```
> Myresult = simple.lm(Height, Weight)
> theresiduals = resid(Myresult) # достига до остатъците
> plot(theresiduals)
```



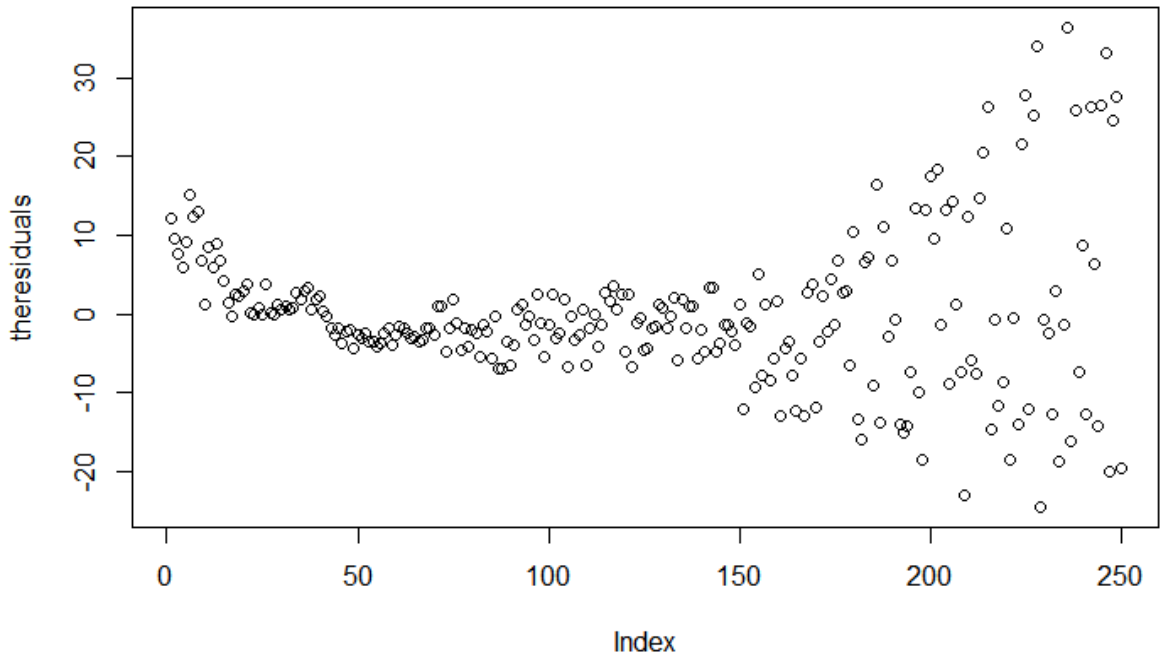

```
> Myresult = simple.lm(X9, X4, show.residuals = TRUE)
```



Първата графика е корелационното поле на данните заедно с линията на регресия. Втората графика на първия ред е графиката на остатъците. В случая наблюдаваме, че те не са независими и нямат постоянна дисперсия, т.е. би трябвало да трансформираме данните по подходящ начин и тогава да приложим простата линейна регресия или изцяло да сменим модела. Първата графика на втория ред е хистограмата на остатъците. В идеалния случай тя би била близка до съответната нормална крива. В случая имаме малко по-голяма изостреност, която води до по-тежки опашки на разпределението. Това още веднъж показва, че преди да приложим простата линейна регресия е добре да трансформираме данните или да сменим простата линейна регресия с друг модел. Втората графика на втория ред е normal plot на остатъците. Ако остатъците са нормално разпределени точките от тази графика лежат точно върху ъглополовящата на първи квадрант. Непрекъснатата линия е регресията за normal plot на остатъците. В случая тя не съвпада с ъглополовящата на първи квадрант. Това още веднъж показва, че разглежданият регресионен модел не е добър за данните. Отклоненията в началото и в края показват, че разпределението на остатъците има опашки, които са по-тежки от нормалните. Простият линеен регресионен модел не е най-добрият възможен за нашите данни.

Достъпът до остатъците на регресионния модел и тяхната графика могат да бъдат постигнати и с последователно използване на функциите ***resid*** и ***plot***.

```
> Myresult = simple.lm(X9,X4)
> theresiduals = resid(Myresult) # достига до остатъците
> plot(theresiduals)
```



Отново наблюдаваме изменяща се дисперсия и зависими данни, които говорят, че може да се построи по-добър модел. При нарастваща дисперсия е добре да се логаритмува зависимата променлива (в случая цената) преди да се приложи простият линеен регресионен анализ.

- *Обща стандартна грешка на модела*

За да можем да направим статистически изводи за β_0, β_1 и Y първо трябва да оценим дисперсията σ^2 на грешката и после да опишем разпределението на грешката. От теорията на общите линейни модели, най-добра неизместена оценка за σ^2 е

$$S_\varepsilon^2 = \frac{\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2}{n-r} = \frac{\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2}{n-2}.$$

Тук r е броят на неизвестните параметри в модела. В случая те са два β_0 и β_1, т.е. r = 2.
Тази величина се нарича ***среден квадрат на грешката***.
Величината

$$S_\varepsilon = \sqrt{\frac{\sum_{i=1}^{n}\varepsilon_i^2}{n-r}} = \sqrt{\frac{\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2}{n-r}} = \sqrt{\frac{\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2}{n-2}},$$

се нарича ***обща стандартна грешка*** на модела.

Ако $S_\varepsilon = 0$, значи имаме пълно съвпадение на изходните данни с техните оценки.

В нашия пример по данните от файла tires, с помощта на функцията ***summary*** получихме, че:

Residual standard error: 9.624 on 248 degrees of freedom #Това е стандартната грешка на оста-
тъците, т.е. S на епсилоните.

Мярката на грешката е същата както на зависимата променлива, в случая лева. Тук стандартната грешка на остатъците е 9.624 лв.

- Проверка за хипотезата за адекватност на модела[2]

Често пъти резултатите от изследването се оформят в таблица от вида:

Източник на дисперсията	*Сума от квадратите (Девиация)*	*Степени на свобода*	*Дисперсия*	*F – критерий F- statistic F- емпирично*
Регресия	$SS_D = \sum_{i=1}^{n} (\hat{Y}_i - \bar{Y}_n)^2$	*1*	$S_D^2 = \frac{SS_D}{1}$	$F_{емп} = \frac{S_D^2}{S_\varepsilon^2}$
Отклонение от регресията	$SS_\varepsilon = \sum_{i=1}^{n} (\hat{Y}_i - Y_i)^2$	*n - 2*	$S_\varepsilon^2 = \frac{SS_\varepsilon}{n-2}$	
Общо:	$SS_Y = \sum_{i=1}^{n} (Y_i - \bar{Y}_n)^2$	*n - 1*	$S_Y^2 = \frac{SS_Y}{n-1}$	

За сумата от квадратите, обусловена от регресията е вярно следното съотношение

$$SS_D = \beta_1^2 \sum_{i=1}^{n} (X_i - \bar{X}_n)^2 .$$

В случая на нормално разпределени грешки можем да направим проверка на хипотезата за адекватност на тествания модел. Един от начините е да проверим хипотезата

H_0: $\beta_1 = 0$

срещу алтернативата

H_1: $\beta_1 \neq 0$.

Избираме риска за грешка от първи род $\alpha \in (0, 1)$. [3] Обикновено $\alpha \in [0.01, 0.05]$.

Критичната област за нулевата хипотеза[4] има вида

$$W_\alpha = \left\{ (x_1, \ldots, x_n) \in \aleph : \frac{S_D^2}{S_\varepsilon^2} \geq C_\alpha \right\},$$

където C_α е $1- \alpha$ квантилът на $F(1, n - 2)$. [5]

Когато пресметнатото от данните отношение $\frac{S_D^2}{S_\varepsilon^2}$ е по-голямо от теоретичния квантил C_α това означава, че по-горното неравенство е удовлетворено, т.е. извадката е в критичната област за нулевата хипотеза, т.е. отхвърляме нулевата хипотеза. Това означава, че приемаме алтелнативната $\beta_1 \neq 0$ и моделът е адекватен.

В нашия пример с колите, с помощта на функцията ***summary*** получихме, че

F-statistic: 458.6 on 1 and 248 DF, p-value: < 2.2e-16

---

[2] Това е допълнително четиво в тази тема. Повече разбиране по тази точка ще има след разглеждане на темата „Проверка на хипотези“.

[3] Това е вероятността да се отхвърли нулевата хипотеза, когато тя е вярна. Тази грешка не се избира 0, т.к. при намаляване на грешката от първи род α се увеличава грешката от втори род, т.е. вероятността да се отхвърли алтернативната хипотеза, когато тя е вярна.

[4] Подмножеството на n-мерното пространство, в което ако попадне извадката отхвърляме нулевата хипотеза.

[5] Това е съкращение за разпределение на Fisher с 1 степен на свобода на числителя и n – 2 степени на свобода на знаменателя.

Т.е. стайността на p-value, която съответства на F-statistic е много малка и е по-малка от риска за грешка от първи род α, т.е. трябва да отхвърлим нулевата хипотеза и моделът е адекватен в смисъл, че β_1 е статистически значимо различен от 0, т.е. зависимостта на зависимата променлива (Цена) от независимата променлива (размер в цолове) е статистически значима.

В нашия пример с колите, с помощта на функцията ***summary*** получихме, че

F-statistic: 458.6 on 1 and 248 DF, p-value: < 2.2e-16

Т.е. стайността на p-value, която съответства на F-statistic е много малка и е по-малка от риска за грешка от първи род α, т.е. трябва да отхвърлим нулевата хипотеза и моделът е адекватен в смисъл, че коефициентът β_1 е статистически значимо различен от 0, т.е. зависимостта на зависимата променлива (Цена) от независимата променлива (Размер в цолове) е статистически значима.

- *Построяване на доверителни интервали на коефициентите и на оценките на зависимата променлива.*[6]

Вече можем да построим и доверителни интервали на β_0 и β_1 и Y.

Величина	*Стандартна грешка* Std. Error	*Степени на свобода*	*Граници на доверителния интервал*
β_0	$S_{\beta_0} = \sqrt{\frac{SS_\varepsilon^2 \sum_{i=1}^{n} X_i^{\,2}}{n\sum_{i=1}^{n}(X_i - X_n)^2}}$	$n - 2$	$\hat{\beta}_0 \pm S_{\beta_0} t_{1-\alpha/2}(n-2)$
β_1	$S_{\beta_1} = \sqrt{\frac{SS_\varepsilon^2}{\sum_{i=1}^{n}(X_i - \overline{X}_n)^2}}$	$n - 2$	$\hat{\beta}_1 \pm S_{\beta_1} t_{1-\alpha/2}(n-2)$
Средното значение на Y (т.е. ако независимите променливи не са случайни)	$S_{EY_i} = S_\varepsilon \sqrt{1 + \frac{1}{n} + \frac{(X_i - \overline{X}_n)^2}{\sum_{i=1}^{n}(X_i - \overline{X}_n)^2}}$	$n - 2$	$\hat{Y}_i \pm S_{EY_i} t_{1-\alpha/2}(n-2)$
Средното значение на Y (т.е. ако независимите променливи са случайни)	$S_{EY/X_i=x_i} = S_\varepsilon \sqrt{\frac{1}{n} + \frac{(X_i - \overline{X}_n)^2}{\sum_{i=1}^{n}(X_i - \overline{X}_n)^2}}$	$n - 2$	$\hat{Y}_i \pm S_{EY/X_i=x_i} t_{1-\alpha/2}(n-2)$

В нашия пример с колите, с помощта на функцията ***summary*** получихме, че

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	-5.3225	5.2314	-1.017	0.31	#това са оценката $\hat{\beta}_0$, стандартната #й грешка (обяснена е по-долу), и # p-value то, т.е. лицето под съответната нормална # крива и извън ± t value -то.

[6] Това е допълнително четиво в тази тема. Повече разбиране по тази точка ще има след разглеждане на темата „Доверителни интервали“.

X9 6.7896 0.3171 21.415 <2e-16 ***#това са оценката $\hat{\beta}_1$, стандартната
#й грешка(обяснена е по-долу), и
p-value то, т.е. лицето под съответната нормалната крива
и извън ± t value -то.

Заместваики оценките и техните стандартни грешки лесно можем да получим границите на доверителните интервали за коефициентите.

При 5% риск за грешка доверителният интервал за β_0 *е*

-5.3225 ± 1.96*5.2314

-5.3225 ± 10.25354

(-15.57604; 4.93104)

Т.е. този доверителен интервал е доста широк и е удачно да се провери хипотезата дали не можем да приемем, че този коефициент е нула.

При 5% риск за грешка доверителният интервал за β_1 *е*

6.7896 ± 1.96*0.3171

6.7896 ± 0.621516

(6.168084; 7.411116)

Т.е. можем да очакваме, че с нарастване на диаметъра на гумите с 1 цол, тяхната цена ще нарасне от 6.168084 лв. до 7.411116 лв.

- *Корелационен коефициент*[7]

При достатъчно голям брой опити $S_Y^2 \approx S_\varepsilon^2$. Ето защо S_ε често се приема за стандартна грешка на оценката $\hat{Y}$. Ако отнесем тази грешка S_ε към стандартното отклонение S_Y на данните от извадката, отнасящи се за резултативния признак *Y*, ще получим величина, която е *0* при пълно съвпадение, т.е. при функционална зависимост между *X* и *Y* и е *1* ако оценките на *Y* не се влияят от *X*. В последния случай всички оценки $\hat{Y}$ на резултативния признак *Y* ще са равни помежду си и по тази причина ще са равни на своята средна аритметична и на средната аритметична на изходните данни за *Y*. На основата на тези разсъждения е образуван корелационния коефициент на Пирсън

$$r = \sqrt{1 - \frac{S_\varepsilon^2}{S_Y^2}} = \frac{S_D}{S_Y}$$

Той се изменя от *0* до *1*.

За посоката на зависимостта между *X* и *Y* се съди по знака на регресионния коефициент $\hat{\beta}_1$.

В нашия пример с колите, с помощта на функцията ***summary*** получихме, че

Multiple R-squared: 0.649, Adjusted R-squared: 0.6476

Т.е. 64,76% от вариацията на зависимата променлива Y – Цена е определена от вариацията на независимата променлива X – Размер на гумите. Т.е. изменението на размера на гумите статистически значимо влияе на изменението на цената им. Това е в синхрон с извода, направен при проверката за адекватност.

```
> cor(X4,X9)
[1] 0.8056144
> cor(X4,X9)^2
```

[7] Това е допълнително четиво в тази тема. Повече разбиране по тази точка ще има след разглеждане на темата „Регресионен анализ“.

[1] 0.6490146

Въпроси:

1. По какво се различават фактор-признака и резултативния признак? Ще се промени ли извода от регресионния анализ ако сменим местата им? Винаги ли можем да сменим местата им?
2. Какъв е смисълът на коефициента $\hat{\beta}_1$ в уравнението на изглаждащата права и как се намира самия коефициент?
3. С какво се различава изглаждащата права от всички останали прави, които можем да прекараме между точките от корелационното поле?
4. Кои са логическите обосновки, които ни дават основание да използваме корелационния коефициент на Пирсън за измерител на силата на зависимостта между наблюдаваните признаци?

5. Рангов корелационен коефициент на Спирмън[8]

Да предположим, че над единиците от съвкупността са извършени наблюдения, върху два признака, измерени на рангова скала. Т.е. статистическите единици са подредени по големина и на най-голямото наблюдение е даден ранг 1, на следващото ранг 2 и т.н. Например:

```
> rank(c(6,5,7,2,3))
[1] 4 3 5 1 2
```

Ако имаме две равни по големина наблюдения те имат равни рангове и стойността им е средното аритметично на ранговете, които биха имали ако наблюденията бяха различни по стойност.

```
> rank(c(6,6,5,7,2,3))
[1] 4.5 4.5 3.0 6.0 1.0 2.0
```

Спирмън *използва като измерител на силата на зависимостта между наблюдаваните признаци близостта на ранговете, и по-точно сумата от квадратите на разликите им.*

Ако съществува силна положителна зависимост между ранговете на единиците, те би трябвало да съвпадат и сумата от квадратите на разликите им би била нула.

Ако зависимостта е силна отрицателна, ранговете ще са подредени в обратен ред. Разликите им в този случай, ако *n* е четно, ще образуват редица само от нечетните числа от – *(n-1)* до *(n-1)* или ако *n* е нечетно, само от четните числа в този интервал. Тогава сумата от квадратите им ще е

$$\frac{n(n^2-1)}{3}.$$

При липсата на каквато и да е зависимост можем да приемем, че тази сума ще е средното аритметично на двете крайни възможности, т.е. $\frac{0+\frac{n(n^2-1)}{3}}{2}=\frac{n(n^2-1)}{6}$.

Като отнесем тази величина към действителната сума от квадратите на разликите, т.е. $\sum_{i=1}^{n} d_i^2$, получаваме измерител на зависимостта, който би бил нула при силна правопропорционална зависимост между ранговете. Ето защо ***ранговият коефициент на корелация на Спирмън*** се пресмята по формулата

[8] Допълнителен материал

$$r_{Cn} = 1 - \frac{\sum_{i=1}^{n} d_i^2}{\frac{n(n^2-1)}{6}} = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)}.$$

Той може да се определи с помощта на функциите ***cor*** и ***rank*** в R.
За примера с колите това е:

```
> cor(rank(X4), rank(X9))
[1] 0.9004327
```