which(...) – позицията на елемент който отговаря на изискване

any(...) – наличие на елемент който отговаря на изискване

all(...) – дали всички елементи отговарят на изискване

unique(...) – връща уникалните стойности

duplicated(...) – връща дали стойността се среща повече от веднъж

is.element(x, y) – дали елементите на x се с срещат сред елементите на y

x %in% y – дали съответния елемент от x се среща в y

substr(x, start, stop) – под низ с начало start и край end

length(x) – дължина на x

sum(x) – сумира елементите на x

abs(x) – връща абсолютни стойности

sqrt(x) – връща корен квадратен на стойностите

mean(x) – средно аритметично

median(x) - медианата

quantile(x, p) - p-квантил

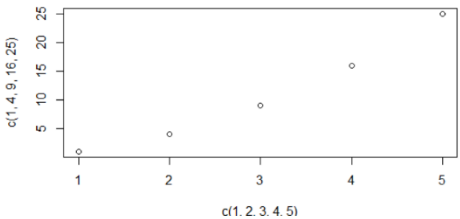
sd(x) - Стандартно отклонение s

var(x) – Дисперсия D(x)

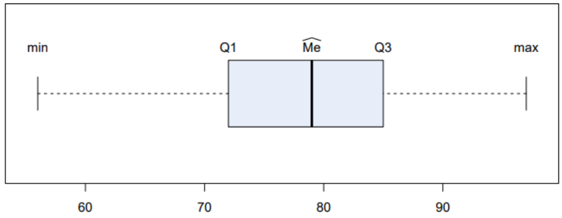
table(x) – създава таблица с данните

table(x, y) – създава „двойна“ таблица

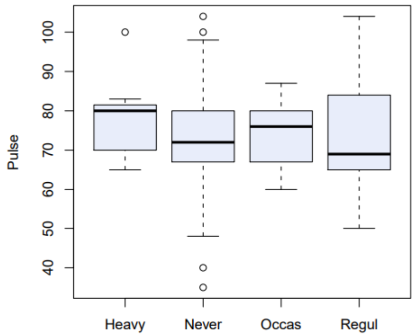
plot(x, y) – нормално чертае точка по точка



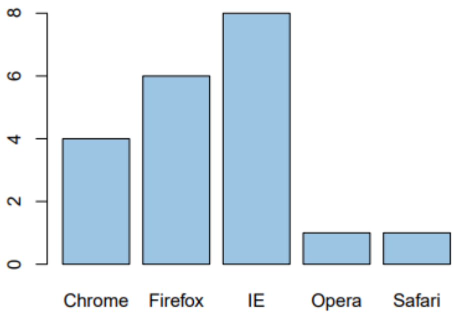
plot(y~x, data) - променливата x се разбива по категориите на y boxplot(x) - Кутия с мустаци

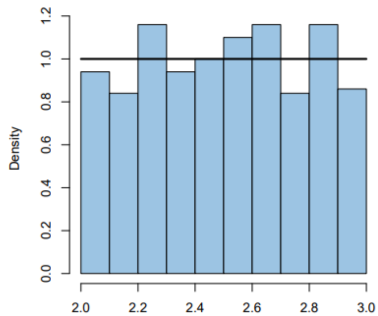


boxplot(y~x, data) - променливата x се разбива по категориите на y и за всяка категория се рисува кутия с мустаци



barplot(table(x), beside=F, legend=F) - всяка категория е представена със стълб с височина равна на съответната стойност от таблицата





hist(x, probability=F, breaks=) – хистограма

sample(x, size, replace=F) - „тегли“ size на брой стойности от x (replace дали с повторения)

dbinom(k, n, p) - Bi(n, p) биномно разпределение $P(X = k) = \binom{n}{k} p^{k} q^{n-k}$

pbinom(k, n, p) - P($X \leq k$)

dgeom(k, p) - геометрично разпределена(X=брой неуспехи преди първия успех) P($X = k$) = $q^{k} p$.

pgeom(k, p) - P($X \leq k$)

dnbinom(k, r, p) - отрицателно биномно разпределена (Y= брой неуспехи преди r-тия успех) P($Y = k$) =$\binom{r+k-1}{r-1} p^{r} q^{k}$

pnbinom(k, r, p) - P($Y \leq k$).

dhyper(k, M, $N - M$, n) - хипергеометрично разпределена (В кутия има M бели и $N - M$ черни топки. Вадим n топки без да ги връщаме.) P($X = k$) = $\frac{\binom{M}{k}\binom{N-M}{n-k}}{\binom{N}{n}}$

phyper(k, M, $N - M$, n) - P($X \leq k$)

dpois(k, λ) - Поасоново разпределение P($X = k$) = $e^{-\lambda} (\lambda^{k} / k!)$.

ppois(k, λ) - P($X \leq k$)

punif(q, a, b) - равномерно разпределение P($X \leq q$) = $F(q)$

pexp(q, lambda) - експоненциално разпределение P($X \leq q$) = $F(q)$.

pnorm(q, mu, sigma) - нормално разпределение P($X \leq q$) = $F(q)$

pt(q, df) - the area to the left of a given value x in the Student t distribution (t-тест за средно)

pchisq(q, df) - P-value = $P_0(\chi^{2} > \chi^{2}_{obs})$ = 1 - pchisq(chi2.obs, df = k-1) Хи квадрат тест за съгласуваност (дали)

qunif(p, a, b) = $Q(p) = F^{-1}(p)$

qexp(p, lambda) = $Q(p) = F^{-1}(p)$

qnorm(p, mu, sigma) = $Q(p) = F^{-1}(p)$

qt(p, df) - what the t-score is of the pth quantile of the Student t distribution

qchisq(p, df) - specify a desired area in a tail and the number of degrees of freedom и изчислява x-стойноста

t.test(x, mu, alternative=c("two.sided", "less", "greater"), conf.level=0.95) - t-тест за средно : $H_0 : \mu = \mu_0$

- $H1 : \mu \neq \mu_0$ t.test(x, mu = μ_0)

- $H1 : \mu > \mu_0$ t.test(x, mu = μ_0, alternative='greater')

- $H1 : \mu < \mu_0$ t.test(x, mu = μ_0, alternative='less')

```
One Sample t-test

data:  x
t = 1.7556, df = 11, p-value = 0.1069
alternative hypothesis: true mean is not equal to 5.2
95 percent confidence interval:
 5.081616 6.251717
sample estimates:
mean of x
 5.666667
```

t.test(x, y, alternative=c("two.sided", "less", "greater"), paired=F) - t-тест за разлика на средни: независими извадки: $H_0 : \mu_X = \mu_Y$

- $H_1 : \mu_X \neq \mu_Y$ t.test(x, y)

- $H_1 : \mu_X > \mu_Y$ t.test(x, y, alternative='greater')

- $H_1 : \mu_X < \mu_Y$ t.test(x, y, alternative='less')

```
Welch Two Sample t-test

data:  x and y
t = -2.1264, df = 15.78, p-value = 0.02481
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
      -Inf -0.4099189
sample estimates:
mean of x mean of y
      7.1       9.4
```

t-тест за разлика на средни: зависими извадки : paired=T

```
Paired t-test

data:  x and y
t = 5.4659, df = 14, p-value = 4.158e-05
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 18.20922      Inf
sample estimates:
mean of the differences
               26.86667
```

t.test(...)$p.value; - взимаме p-стойността която е резултат от теста

t.test(...)$conf.int – взимаме confidence interval (доверителен интервал 95%)

prop.test(x, n, p, alternative=c("two.sided", "less", "greater"), correct=T) - z-тест за разлика на пропорции (Искаме да сравним вероятностите p_1 и p_2 . Пример: вероятността да се появи дефект в батериите произведени от завод 1 и завод 2):

$H_0 : p_1 = p_2$

$H_1 : p_1 \neq p_2$ prop.test(x, n, correct=F)

$H_1 : p_1 > p_2$ prop.test(x, n, alternative='greater', correct=F)

$H_1 : p_1 < p_2$ prop.test(x, n, alternative='less', correct=F)

```
2-sample test for equality of proportions without continuity correction

data:  x out of n
X-squared = 1.108, df = 1, p-value = 0.2925
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.03991226  0.13298585
sample estimates:
   prop 1    prop 2
0.3227273 0.2761905
```

chisq.test(x, p) – Тестовете за съгласуваност (goodness-of-fit tests) се използват за да се провери доколко данните са съгласувани с даден вероятностен модел (дали този модел описва добре данните): При p-стойността<0,05 отхвърляме $H_0 : (p_1, p_2, . . . , p_k) = (p^o{}_1 , p^o{}_2 , . . . , p^o{}_k)$ хипотезата (H_1: () ≠())

```
Chi-squared test for given probabilities

data:  x1
X-squared = 160.36, df = 10, p-value < 2.2e-16
```

chisq.test(x) - Хи-квадрат тест за независимост - Разглеждаме експеримент, чиито изходи могат да бъдат класифицирани по два критерия на $A1, A2, . . . , Ar$ или $B1, B2, . . . , Bc$, т.е. изходите могат да бъдат представени като двойки (Ai , Bj).:): При p-стойността<0,05 отхвърляме $H_0 : p_{ij} = p_i{\cdot}p{\cdot}_j$ за всяка двойка (i, j) (H_1: $p_{ij} \neq p_i{\cdot}p{\cdot}_j$ за всяка двойка (i, j))

```
Pearson's Chi-squared test

data:  tb
X-squared = 138.29, df = 9, p-value < 2.2e-16
```

m1 <- lm(y ~ x1 + x2, data) - намираме оценения модел($y = \beta_0 + \beta_1 x1 + \beta_2 x2 + \varepsilon$)

```
Call:
lm(formula = volume ~ diam + height, data = cher)

Coefficients:
(Intercept)         diam       height
   -57.9877       4.7082       0.3393
```

$y =$ обем (volume)
$x_1 =$ диаметър (diam)
$x_2 =$ височина (height)
Модел: $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon.$
Оценено регресионно уравнение: $\widehat{y} = -57.9877 + 4.7082x_1 + 0.3393x_2.$

```
Call:
lm(formula = volume ~ diam, data = cher)

Coefficients:
(Intercept)         diam
    -36.943        5.066
```

$y =$ обем (volume)
$x =$ диаметър (diam)
Модел: $y = \beta_0 + \beta_1 x + \varepsilon.$
Оценено регресионно уравнение: $\widehat{y} = -36.943 + 5.066x.$

summary(m1) - основна информация за оценения модел

```
> summary(m1)$coefficients
             Estimate Std. Error   t value     Pr(>|t|)
(Intercept) -36.943459   3.365145 -10.97827 7.621449e-12
diam          5.065856   0.247377  20.47829 8.644334e-19
```

summary(m1)$coefficients - таблица за оценените коефициенти

summary(m1)$r.squared - Коефициент на детерминация (R^2)

summary(m1)$adj.r.squared - коригиран R^2

coef(m1) и coefficients(m1) - оценените коефициенти $\widehat{\beta}_0, \ldots, \widehat{\beta}_k$

confint(m1) - доверителни интервали за $\beta_0, \ldots, \beta_k$

resid(m1) и residuals(m1) - остатъците $e_i = y_i - \widehat{y}_i$

fitted(m1) и fitted.values(m1) - $\widehat{y}_i$

predict(m1, newdata, interval=c("none", "confidence", "prediction"), level=0.95)

`predict(m1, new, interval="confidence")` доверителен интервал за $\mu_{y|x_1 \ldots x_k}$ при $(x_1, \ldots, x_k) = (x_1^\star, \ldots, x_k^\star)$

`predict(m1, new, interval="prediction")` интервал за прогноза на y при $(x_1, \ldots, x_k) = (x_1^\star, \ldots, x_k^\star)$

`predict(m1, new, interval="none")` $\widehat{y}$ за дадено $(x_1, \ldots, x_k) = (x_1^\star, \ldots, x_k^\star)$

```
> ci.b <- predict( m2, data.frame( diam=14, height=70 ), interval="confidence" )
> ci.b
       fit      lwr      upr
1 31.67417 29.34183 34.00652
```