R-тестове

1. x = c(267, 91, 103, 39)
probs = c(0.5, 0.2, 0.2, 0.1)
chisq.test(x, p=probs) - Хи-квадрат тест за съгласуваност

2. За проверка на хипотези за пропорция може да се използва функцията prop.test, която прави подробен тест

$H_1: p \neq p_0$ prop.test(x=x, n=N, p=p0, correct=F)

$H_1: p > p_0$ prop.test(x=x, n=N, p=p0, alternative="greater", correct=F)

$H_2: p < p_0$ prop.test(x=x, n=N, p=p0, alternative="less", correct=F)

1-simple proportions test ... data: 2700 out of 2500

3. Ако във вектора x запишем x_1, x_2, а във вектора n запишем n_1, n_2, може да използваме функцията

$H_1: p_1 \neq p_2$ 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 ... data x out of n

4 t-test за средно

- във вектора x са записани наблюденията $x_1, x_2, \dots, x_n$

$H_1: \mu \neq \mu_0$ t.test(x, mu=μ_0)

$H_1: \mu > \mu_0$ t.test(x, mu=μ_0, alternative="greater")

$H_1: \mu < \mu_0$ t.test(x, mu=μ_0, alternative="less")

t.test(x, mu=5.2)$p.value

// One sample t-test

5 t-test за разлика на средни. Независими извадки

Ако във вектора x запишем наблюденията $x_1, x_2, \dots, x_n$, а във вектора y наблюденията $y_1, y_2, \dots, y_m$

$H_1: \mu_x \neq \mu_y$ 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

6 t.test за разлика на средни: зависими извадки
За зависими извадки отново се ползва функцията t.test, като трябва да се добави paired = T
(във вектора x са наблюдения $x_1, x_2, \dots, x_n$, а във вектора y са $y_1, y_2, \dots, y_n$):
$H_1: \mu_x \neq \mu_y$ t.test(x, y, paired = T)
$H_1: \mu_x > \mu_y$ t.test(x, y, alternative = "greater", paired = T)
$H_1: \mu_x < \mu_y$ t.test(x, y, alternative = "less", paired = T)
t.test(x, y, paired = T) дава същия резултат като
t.test(x - y) // Paired t-test

7. Линеен модел с един предикатор
Предполагаме, че случайните величини Y, X, ε са свързани по следния начин:
$Y = \beta_0 + \beta_1 X + \varepsilon$, β_0, β_1 са константи
X - предикатор, Y - отклик
Параметрите β_0, β_1 са неизвестни и въз основа на наблюдения намираме оценките $\hat{\beta}_0, \hat{\beta}_1$
$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$ - оценено регресионно уравнение, оценен модел
Проверка на хипотези за β_1
$H_0: \beta_1 = 0$
$H_1: \beta_1 \neq 0$
Ако във вектора x са наблюденията $x_1, x_2, \dots x_n$, а във вектора y наблюденията $y_1, y_2, \dots y_n$
lm(y ~ x) - оцененият модел

R - подготовка

1. Хи-квадрат тест за съгласуваност

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)

Разглеждаме експеримент (опит) с k възможни изхода $A_1, A_2, \dots, A_k$ (пълна група събития). Означаваме вероятностите $P(A_1) = p_1, \dots, P(A_k) = p_k$. В поредица от n независими опита X_i е броя сбъдвания на събитието A_i. Случайният вектор $(X_1, X_2, \dots, X_k)$ има полиномно разпределение (multinomial distribution) с параметри $n, p_1, \dots, p_k$

Искаме да проверим хипотезата

$H_0: (p_1, p_2, \dots, p_k) = (p_1^0, p_2^0, \dots, p_k^0)$

$H_1: (p_1, p_2, \dots, p_k) \neq (p_1^0, p_2^0, \dots, p_k^0)$, където

$p_1^0, \dots, p_k^0$ са предварително зададени стойности.

Ако p-value $\leq \alpha$, отхвърляме H_0 в полза на H_1

Ако p-value $> \alpha$, нямаме достатъчно основания да отхвърлим H_0

```
chisq.test(x, p=probs)
```

8. Линеен модел с няколко предикатора

Предполагаме, че случайните величини $Y, X_1, \dots, X_k$ са свързани по следния начин:

$$Y = \beta_0 + \beta_1 X_1 + \dots + \beta_k X_k + \varepsilon, \quad \beta_0, \beta_1, \dots, \beta_k \text{ са const}$$

$X_1, \dots, X_k$ - предикатори, Y - отклик

```
lm(y ~ x1 + x2)
```

9.
```
* fn[EAI >= 4]
* sum(spec=="KN" & DAA==5)
* ~~sum(EAI[spec=="KN"]) / length(EAI[spec=="KN"])~~
* mean(EAI[spec=="KN"])   -средна оценка
* median(DAA[EAI==5])
* boxplot(DAA ~ spec)
  boxplot(table(DAA, spec), beside=T, legend=T)
```

10.
```
sim.dice = function() {
  x = sample(c(1:6), 4, replace=T)
  sum(x[1:2]) == sum(x[3:4])
}
sim.prob = function(Nrep) {
  res = replicate(Nrep, sim.dice())
  sum(res) / length(res)
}
sim.prob(100000)
```