СЕМ Пр. 1 12.10.2022

sum(x>10) - сумира колко са числата >10

which(x>10) -

diff(x) - разликата на първия с втория ел. итн.

cumsum(x) - първият ел. е същия вторият е първият + вторият итн. последният е сумата на всички

sum(x) - сумира всички ел.

x^2 - прави всички ел. на квадрат

sort(x) - сортира в нарастващ ред

sort(x, decreasing = T) - в обратен ред

rm(x) - изтрива вектора

x[order(x, decreasing = TRUE)] - сортира == sort

x <- c(1,3,5,11,15) → numeric by default (реално число)

x <- as.integer(c(1,3,5,11,15)) → цяло число

class(x) → проверяване какъв тип са елементите

y <- vector("numeric", length = 5) // "logical", "character"

* Когато сумираме вектори с различни дължини, като свърши единият започва пак от начало да го чете

rep(c("a", "b"), times = 5) - докато не стигне дължина 5

rep(c("a", "b"), each = 3) - всеки по 3 пъти

10:1 → 10 9 8 7 6 5 4 3 2 1

редици seq(from = 1, to = 10, by = 2) → 1 3 5 7 9

seq(from = 0, to = 1, length.out = 11) → 0.0 0.1 0.2 ... 0.9 1.0

*Матрици

M <- rbind(c(5,3,5,6), c(8,3,7,4))

M

`M[2,3]` → втори ред, трети стълб

`M[ ,3]` → трети стълб

`M[2, ]` → втори ред

```
M <- cbind(c(5,3,5,6), c(8,3,7,4))
M
```

→

```
5 8
3 3
5 7
6 4
```

`t(M)` - транспонирана

`M[order(M[,1]), ]` - сортира матрицата по първа колона

`M <- matrix(c(1:12), nrow=3, ncol=4, byrow=TRUE)` → запълва по ред

```
head(M,2)
tail(M,2)
```

`sqrt(M)` - корен квадратен върху всички елементи

- data frame - колоните може да бъдат от различен тип

```
x <- c(5,8,11,3,2,9,4)
y <- c("Y","Y","N","Y","N","N","N")
df <- data.frame(x,y)
df
```

```
5  Y
8  Y
11 N
3  Y
2  N
9  N
4  N
```

`str(df)` - казва какви са видовете и колко ch. са и колко се повтарят

`df$x` - достъпва колоната x

`df$x[4]` - 4-ия елемент от x

`df[ ,1]  ==  df[ ,"x"]`

`df[5, ]`

`df$z <- seq(from=1, to=14, by=2)` - добавяме колона във вече съществуващ

`str(df)`

`sample(c(1:10), 2, replace=TRUE)` - 2 елемента от 1 до 10 на случаен принцип и винаги са различни

① Nrep <- 100000 → колко хвърляния да симулираме

dice <- sample(c(1:6), Nrep, replace =TRUE)

sum(dice==6) / Nrep

↳ приближава вероятност към 1/6

* Биномно разпределение $X \sim Bi(n,p)$

dbinom(k, n, p) → $\mathbb{P}(X=k)$

pbinom(k, n, p) → $\mathbb{P}(X \le k)$

rbinom(N, n, p) → генерира N случайни числа от биномно разпределение с параметри n и p.

* Геометрично разпределение $X \sim Ge(p)$

  - когато броим успеха → dgeom(k-1, p)
  - когато не броим успеха → dgeom(k, p)

* Отрицателно биномно разпределение → $X \sim NB(r,p)$

  - когато броим успеха → dnbinom(k-r, r, p)
  - когато не броим успеха → dnbinom(k, r, p)

* Ако се търси $\le k$ → p X ($\to$ bin, geom..)

Ако се търси $\ge k$ → $1 - \mathbb{P}(X \le k-1) = 1 -$ p X

* Поасоново разпределение

  → dpois(k, λ)

* Хипергеометрично разпределение

  dhyper(k, M, N-M, n)

* Равномерно разпределение $U(a,b)$

dunif (x,a,b) $= f(x)$ – графика

punif(q,a,b) $= \mathbb{P}(X \leq q) = F(q)$

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

runif (N,a,b) генерира N случайни числа от равномерно разпределение в интервала (a,b).

→ hist(x , probability= T)
curve (dunif (x,a,b), from= a, to=b, add=T , lwd= 2)
↳ графика на плътността

* Експоненциално разпределение $Exp(\lambda)$

dexp $(x, \lambda) = f(x)$

pexp $(q, \lambda) = F(q) = \mathbb{P}(X \leq q)$

qexp $(p, \lambda) = Q(p) = F^{-1}(p)$ → от редица лято

rexp (N, λ) генерира N случайни числа от експоненциално разпределение с параметър λ

* Нормално разпределение $N(\mu, \sigma^2)$

dnorm $(x, \mu, \sigma) = f(x)$

pnorm$(q, \mu, \sigma) = \mathbb{P}(X \leq q) = F(q)$

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

rnorm(N, μ, σ) - генерира N случайни числа от нормално разпределение с параметри μ, σ

→ hist (x, x , xlim = c(-3,5, 3,5)

curve (dnorm (x, 0,1), add= T, lwd= 2.5)

→ df <- read.table("text.txt")

brows <- df$V1 → записване прочетените данни във вектора brows

barplot → представлява съответната таблица във вид на графика, в която всяка категория е представена със стълб с височина равна на съответната стойност от тези данни.

pie → получаваме кръгова диаграма

wait.grp <- cut(wait, breaks= seq(0,12,2)) ← разбива интервала [0,12] на 6 подинтервала

вектор от данни

→ hist == barplot за числови данни
↳ може да зададем и ние разбиването на интервалите
(hist си ги прави самостоятелно)
↳ hist(wait, breaks= seq(0,12,3))

→ hist(..., probability= T) - дефинираме кривата графиката, така че сумата от лицата на всички правоъгълници е единица.

→ stripchart(wait, method="stack", pch= 20, cex= 1.5) ← всяко наблюдение на графиката е представено с кръгче

$\bar{x}$ = mean(x) – средна стойност

$\hat{Me}$ = median(x) – медиана

p-квантил = quantile(x, p)

Q_1 = quantile(x, 0.25)

Q_3 = quantile(x, 0.75)

IQR = IQR(x)

s = sd(x) → стандартно отклонение

→ boxplot(temp, horizontal = T) → кутия с мустаци
file

→ В данните airquality променливата ozone се достъпва по следния начин: airquality$ozone

→ fix(survey) – извежда данните като таблица
summary(survey) – числови характеристики за всяка променлива
survey[, 'Age'] – променливата Age ⇒ = survey$Age
survey[, 12] – 12 променлива
survey[5,] – 5-то наблюдение

→ Ако напишем attach(dataFrameName) – може да се обръщаме към променливите вместо с survey$Age само с Age
→ attach(survey)
summary(Age)

→ na.rm = T → да се игнорират липсващите стойности

4. Данни. Таблици и графики

* Типове данни:
  - числови (количествени)
  - категорни (неколичествени)

* barplot → представляваме съответната таблица във вид на графика, в която всяка категория е представена със стълб с височина равна на съответната стойност от таблицата
* pie → кръгова диаграма - всяка от категориите е представена като сектор от един кръг

→ barplot(table(brows))
barplot (table (brows) / length (brows))
barplot (sort (table(brows))/length(brows), decreasing =T))
pie (table (brows))

* dt <- read.table("browsers.txt")
brows <- dt $V1

* Когато пишем table(x), ни дава елементите в първия ред и честотата на срещане във втория.

→ wait <- c(2,3, 3,5 ...)
wait.grp <- cut (wait , breaks = seq(0, 12 ,2)) → разделяме интервала [0,12] на 6 подинтервала
table (wait.grp)

table (wait.grp) / length (wait) → гледаме по проценти

→ hist (wait) → хистограма, дава същия резултат като barplot, но я представяме чрез вектор.

→ hist сама определя какви да са подинтервалите

↳ но може да ги зададем и сами

hist(wait, breaks = seq(0, 12, 3))

↓ вектора с данните

↳ разделяме на 4 интервала

→ първият под-интервал е от вида [,], а останалите ( ].

→ hist(..., probability = T) – чертаe хистограма, така че сумата от лицата на всички стълбове (правоъгълници) е единица.

lines(x=c(min(h$breaks), h$mids, max(h$breaks)),
y= c(0, h$density, 0), type = "l", lwd = 2, col = "darkorange1")

↳ линия, т.нар. честотен полигон

* stripchart(wait, method = "stack", pch= 20, cex= 1.5)
↳ графика, където всяко наблюдение е представено с кръгче.

* Числовите данни могат да се представят и чрез диаграмата "клон с листа" (steam-and-leaf plot)
steam(wait)
steam(wait, scale= 2) → увеличаваме брой на подинтервалите
steam(wait, scale= 3) → всяко наблюдение в своя вътрешния интервал е представено с 0

5. Числови характеристики на данните

$\bar{x}$ = mean(x) - средна стойност (средно)

$\widehat{Me}$ = median(x) - медиана

p-квантил = quantile(x, p) - разделя данните на две части, съответно от 100p% и останалите 100(1-p)%.

* 0.5 квантил е медианата

Q_1 = quantile(x, 0.25)

Q_3 = quantile(x, 0.75)

IQR = IQR(x) - интерквартилен размах

s = sd(x) - стандартно отклонение

* функцията summary(temp) - ни дава всички разгледани числови характеристики

→ $[\bar{x} - ks; \bar{x} + ks]$ → неравенство на Чебишев

→ за k = 3 ⇒ $[\bar{x} - 3s, \bar{x} + 3s]$

mean(temp) - 3 * sd(temp) ; mean(temp) + 3 * sd(temp)

* boxplot - кутия с мустаци

→ boxplot(temp, horizontal = T)

mean(ozone, na.rm = T) → да не прави проблем за празни полета

i. Многомерни данни

- когато наблюдаваме(измерваме) повече от една променлива
- Многомерните данни се представят в R чрез обект наречен data frame.

- library(packageName) → за да използваме обекти от даден пакет

fix(survey) - извежда данните като таблица
summary(survey) - основни характеристики за всяка променлива
survey[, 'Age'] == survey$Age - променливата Age
survey[, 12] - дванадесетата променлива
survey[5,] - петото наблюдение

- summary(survey$Age) == attach(survey)
  summary(Age)

- с NA се означават липсващите наблюдения
  na.rm= T → игнорира липсващите

- prop.table(tab.smoke.hand, 1) → 2 → 1 - по ред
  ↳ проценти
  показва отделно
  besige=T
  2 - по колона

barplot(table(Smoke, W.Hnd), legend=T, args.legend=list(x="topright", insert=0.05))
кое от ляво, кое от дясно
легендата къде да се намира

- boxplot(Pulse ~ Smoke)

Pulse се разбива по категориите на Smoke

→ cor (x,y) $\overset{=r}{-}$ извадъчна корелация

→ Ако точките са близо до права линия, то r е близо до 1 или −1 и казваме, че променливите са положително или отрицателно ~~корелативни~~. корелирани

→ Ако r е близо до 0, казваме, че променливите са ~~некорелативни~~ некорелирани

→ За да се игнорират липсващите променливи, при пресмятане на корелация използваме cor (... , use = "complete.obs")

7. Доверителни интервали

- извадка

• Доверителен интервал за средно при известна дисперсия

```
z1.ci <- function( x.bar , sigma, n , alpha) {
    b1 <- x.bar - qnorm(1-alpha/2) * (sigma/sqrt(n))
    b2 <- x.bar + qnorm(1-alpha/2) * (sigma/sqrt(n))
    c(b1, b2)
}
```

• Доверителен интервал за средно при неизвестна дисперсия

```
t1.ci <- function (x.bar , s, n, alpha ) {
    b1 <- x.bar - qt(1-alpha/2, df=n-1) * (s/sqrt(n))
    b2 <- x.bar + qt(1-alpha/2, df=n-1) * (s/sqrt(n))
    c(b1, b2)
}
x <- c( ... )
```
→ данните дето са ни дадени

```
t1.ci (x.bar = mean(x), s=sd(x), n= length(x), alpha = 0.05)
```
↓ 95% доверителен интервал

```
t.test (x, conf.level = 0.95)$conf.int[1:2]
```

• доверителен интервал за пропорция (вероятност за успех)

8. Проверка на хипотези при една извадка

• z-тест за средно

z.obs <- (x.bar - mu) / (sigma / sqrt(n)) → колко са на брой изпробвани

x.bar → новата стойност
mu → старата
sigma → стандартното отклонение

p.value <- 1 - pnorm(z.obs) → $\mu > \mu_0$

→ Ако p.value ≤ 0.05 → отхвърляме нулевата хипотеза за в полза на алтернативната

• t-тест за средно

t.obs <- (mean(x) - mu) / (sd(x) / sqrt(n))
p.value <- 2 * (1 - pt(abs(t.obs), n-1))
$\mu \neq \mu_0$ t.test(x, mu = μ_0) → средно то → $\mu > \mu_0$ t.test(x, mu = μ_0, alternative = 'greater') вектор
↳ One Sample t-test

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

• z-тест за пропорция

→ За проверка на хипотези за пропорция може да се използва функцията prop.test, която прави подобен тест:

I
H1: $p \neq p_0$ → prop.test(x = x, n = n, p = p0, correct = F)
H1: $p > p_0$ → prop.test(x = x, n = n, p = p0, alternative = 'greater', correct = F)
H1: $p < p_0$ → prop.test(x = x, n = n, p = p0, alternative = 'less', correct = F)

↳ Може и така:

II
z.obs <- ((x/n) - p) / sqrt(p * (1-p) * (1/n))
p.value <- 2 * (1 - pnorm(abs(z.obs)))

↳ 1-sample proportions test without continuity correction
prob.test(...) # p.value → to get p.value

9. Проверка на хипотези при две извадки

• t-тест за разлика на средни

• t-тест за разлика на средни: независими извадки

↳ Ако във вектора x запишем наблюденията $x_1, x_2, \dots, x_n$ а във вектора y - наблюденията $y_1, y_2, \dots, y_m$, може да използваме функцията t.test:

$H_1: \mu_x \neq \mu_y$ → 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

• t-test при зависими извадки

→ измервания върху едни и същи обекти, но при различни условия → зависими

↳ данните в случая са paired data, paired samples

↳ За зависими извадки се използва функцията t.test, като трябва да се добави paired=T (във вектора x са наблюденията $x_1, x_2, \dots, x_n$, 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)

* Командата t.test(x, y, paired=T) дава същия резултат като t.test(x-y)

↳ Paired t-test

- **z-тест за разлика на пропорции**

↳ От n1 опита при дадено условие сме наблюдавали x1 пъти успеха и от n2 опита при друго условие сме наблюдавали x2 пъти успеха

↳ пр: - вероятността мъж да е пушач и вероятността жена да е пушач
- вероятността да се появи дефект в батериите произведени от завод 1 и завод 2

↳ Ако във вектора x запишем x1, x2, а във вектора n запишем n1, n2, може да използваме функцията prop.test:

H1: p1 ≠ p2 → prop.test(x, n, correct = F)

HA: p1 > p2 → prop.test(x, n, alternative='greater', correct = F)

H1: p1 < p2 → prop.test(x, n, alternative='less', correct = F)

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

10. Хи-квадрат тестове

- **Хи-квадрат тест за съгласуваност** – goodness-of-fit tests

↳ за да се провери доколко данните са съгласувани с даден вероятностен модел (дали този модел описва добре данните).

↳ Ако във вектора x запишем $x_1, \dots, x_k$, а във вектора probs запишем $p_1, \dots, p_k$, може да използваме функцията chisq.test(x, p=probs)

↳ Chi-squared test for given probabilities

- **Хи-квадрат тест за независимост**

↳ H_0: означава, че променливите A и B са независими, т.е. за всяка двойка (i, j) е вярно $\mathbb{P}(A_i, B_j) = \mathbb{P}(A_i)\,\mathbb{P}(B_j)$.
H_A: означава, че има някаква връзка между променливите A и B.
зависимост

↳ Ако запишем данните $\{x_{ij}\}$ в матрица или таблица x, може да използваме функцията chisq.test(x).

→
```
data(HairEyeColor)
tb <- HairEyeColor[,,1] + HairEyeColor[,,2]
chisq.test(tb)
```

↳ Pearson's Chi-squared test

11. Линейни модели

• Линеен модел с един предиктор

↳ Ако във вектора x са наблюденията $x_1, x_2, \dots, x_n$, а във вектора y са $y_1, y_2, \dots, y_n$ намираме оценения модел с lm(y~x)

• m1 <- lm(y~x)

summary(m1) - основна информация за оценения модел

summary(m1)$coefficients - таблица за оценените коефициенти

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

coef(m1) / coefficients(m1) - оценените коефициенти $\hat{\beta}_0, \hat{\beta}_1$

confint(m1) - доверителни интервали за β_0, β_1

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

fitted(m1) / fitted.values(m1) - $\hat{y}_i$

predict(m1, new, interval="prediction") - интервал за прогноза на y при $x = x^*$

predict(m1, new, interval="confidence") - доверителен интервал за $\mu_{y|x}$ при $x = x^*$

predict(m1, new, interval="none") - $\hat{y}$ за дадено $x = x^*$

$$\boxed{Y = \beta_0 + \beta_1 X + \varepsilon}$$

X - предиктор

Y - отклик

m1 <- lm(bac ~ beers, data=alco)

→ plot(x~y, data = z)

abline(coef(m1), lwd=2)

• линеен модел с няколко предиктора

модел: $\boxed{Y = \beta_0 + \beta_1 X_1 + \dots + \beta_n X_n + \varepsilon}$

$X_1, \dots, X_n$ – предиктори

Y – отклик

```
→ cher <- read.table("cherry.txt", header=T)
m1 <- lm(volume ~ diam, data=cher)
plot(volume ~ diam, data=cher)
abline(coef(m1), lwd=2)
```

Задача 7

```
→ exams <- read.table("exams.txt", header=T)
  attach(exams)
```

а) `fn[EA1 >= 4]`

б) `sum(spec=="KN" & DAA==5)`

`length(which(spec=="KN" & DAA=5))`

в) `mean(EA1[spec=="KN"])`

г) `median(DAA[EA1==5])`

д) `boxplot(DAA ~ spec)` или

`barplot(table(DAA,spec), beside=T, legend=T)`

Задача 8

```
sim.4dice <- function() {
    x <- sample(c(1:6), 4, replace=T)
    sum(x[1:2]) > sum(x[3:4])
}
rs <- replicate(100000, sim.4dice())
sum(rs)/length(rs)
```

8. Проверка на хипотези при една извадка

• z-тест за средно

↳ нова и стара стойност

z.obs = (x.bar − mu) / (sigma / sqrt(n))

отхвърляме H0 $p \le 0.05$

$\mu \ne \mu_0$ → 2 * (1 − pnorm(abs(z.obs)))

$\mu > \mu_0$ → 1 − pnorm(z.obs)

$\mu < \mu_0$ → pnorm(z.obs)

• t-test за средно → One Sample t-test

отхвърляме H0 $p \le 0.05$

↳ вектор и стандартното отклонение

t.test(x, mu = μ_0) → $\mu \ne \mu_0$

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

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

$p \le 0.05$

• z-тест за пропорция → 1-sample proportions test without ...

$p \ne p_0$ → prop.test(x=x, n=n, p=p0, correct=F)

$p > p_0$ → prop.test(x=x, n=n, p=p0, alternative='greater', correct=F)

$p < p_0$ → prop.test(x=x, n=n, p=p0, alternative='less', correct=F)

9. Проверки на хипотези при две извадки

• t-test за разлика на независими средни → Welch Two Sample t-test

↳ два вектора x и y

$\mu_x \ne \mu_y$ → t.test(x, y)

$p \le 0.05$

$\mu_x > \mu_y$ → t.test(x, y, alternative='greater')

$\mu_x < \mu_y$ → t.test(x, y, alternative='less')

• t-test за разлика на зависими средни → Paired t-test

$\mu_x \ne \mu_y$ → t.test(x, y, paired=T)

$p \le 0.05$

$\mu_x > \mu_y$ → t.test(x, y, alternative='greater', paired=T)

$\mu_x < \mu_y$ → t.test(x, y, alternative='less', paired=T)

• Z-тест за разлика на пропорции → 2 sample test for equality of proportions...

p1 ≠ p2 → prop.test(x, n, correct = F)

p1 > p2 → prop.test(x, n, alternative = 'greater', correct = F)

p1 < p2 → prop.test(x, n, alternative = 'less', correct = F)

$p \leq 0.05$

• Хи-квадрат тестове

• Хи-квадрат тестове за съгласуваност - chi-squared test for given probabilities

chisq.test(x, p=probs

$p \leq 0.05$

• Хи-квадрат тест за независимост → Pearson's chi-squared test

chisq.test(x)
с матрица

$p \leq 0.05$

H0 - независими

H1 - зависими

→ X-squared ⇒ chisq.test(x, p = probs)

Problems_past_exams 2022/3

Задача 1

1. boxplot (кутия с мустаци)
2. boxplot (x2)
3. min: 1, max: 36
4. 25%
5. 50%

Задача 2

* p1 = вероятността произволно избрана ядка да е фъстък
p2 = -ll- лешник
p3 = -ll- кашу
p4 = -ll- бадем

H0: (p1, p2, p3, p4) = (0.5, 0.2, 0.2, 0.1)

H1: (p1, p2, p3, p4) ≠ (0.5, 0.2, 0.2, 0.1)

→ # R code

```
x <- c(267, 91, 103, 39)
probs <- c(0.5, 0.2, 0.2, 0.1)
chisq.test(x, p=probs)
```

p.value = 0,2144 > 0.05
→ нямаме основание да отхвърлим хипотезата, че машинатa смесва ядките в желаното съотношение

Задача 3

p = вероятността роботът да сглоби дефектен елемент
H0: p = 0.035
H1: p < 0.035

→ prop.test(x=15, n=500, p=0.035, alternative="less", correct=T) ?
→ първият отговор → alternative hypothesis: true p is less than 0.0

p.value = 0.3132 → p.value > 0.05 ⇒ нямаме основание да твърдим, че процентът дефектни елементи при използване на робот е по-малък отколкото при сглобяване от човек.

Задача 4

mu-x = средното изминато разстояние до износване на гуми от марка 1

mu-y = марка 2

независими по 48 от марка

H0: mu-x = mu-y

HA: mu-x > mu-y

t.test(x, y, alternative = "greater")

→ Верен е втория отговор

alternative hypothesis: true difference in means is greater than 0

p.value = 0.04609

p.value < 0.05

→ Имаме основание да отхвърлим нулевата хипотеза и да твърдим, че средното изминато разстояние до износване е по-голямо при гумите от марка 1 в сравнение с марка 2.

Задача 5

mu-1 = средна оценка по Алгебра 1

mu_2 = средна оценка по Алгебра 2

зависими 80 студенти

H0: mu_1 = mu_2

H1: mu1 > mu_2

t.test(x, y, alternative = "greater", paired = T)

→ Paired t-test

p.value < 0.05 → отхвърляме нулевата хипотеза

→ Може да твърдим, че в средно оценките по Алгебра 1 са по-високи от тези по Алгебра 2

задача 6

а) модел: $grade = \beta_0 + \beta_1 * study.hours + \varepsilon$

grade = beta0 + beta1 * study.hours + eps

оцененото регресивно уравнение:

1.01617 + 0.15806 * study.hours

grade е отклик

study.hours е предиктор

б) При един час повече подготовка за изпита оценката е по-висока средно с 0.15806

в) H0: beta1 = 0

H1 beta1 != 0

p.value 2.9e-12 < 0.05 → отхвърляме H0

Може да се твърди, че има линейна връзка между оценката и часовете отделени за подготовка

г) Модел: $grade = \beta_0 + \beta_1 * study.hours + \beta_2 * lectures + \varepsilon$

оценено регресивно уравнение:

grade = 0.54255 + 0.14767 * study.hours + 0.09432 * lectures

→ При един час повече подготовка за изпита и фиксиран брой посетени лекции оценката е по-висока средно с 0.14767

→ При още една посетена лекция и фиксирани часове за подготовка за изпита оценката е по-висока средно с 0.09432

past_exams 2023/4

(1.)

```
sim.4dice <- function() {
    x <- sample(c(1:6), 4, replace = T)
    sum(x[1:2]) > sum(x[3:4])
}
rs <- replicate(100000, sim.4dice())
sum(rs)/length(rs)
```

(2.) 500 елемента, сглобени от робот, 15 дефектни
3.5% вероятност за дефект, при сглобяване от човек

↳ p = вероятността робот да сглоби дефектен елемент

Ho: $p = 0.035$

H1: $p < 0.035$

```
prop.test(x=15, n=500, p=0.035, alternative="less", correct=T)
```

↳ вярно е първото

p.value = 0.3132 > 0.05 → не отхвърляме нулевата хипотеза

→ Нямаме основание да твърдим, че процентът дефектни елементи при използване на робот е по-малък отколкото при сглобяване от човек.

(3.) a1 <- оценките по Алгебра 1 → зависими са
a2 <- оценките по Алгебра 2

mu_1 → средна оценка по Алгебра 1
mu_2 → средна оценка по Алгебра 2

```
t.test(a1,a2, alternative="greater", paired=T)
```

↳ Paired t-test

p.value = 1.422e-05 < 0.05 → отхвърляме нулевата хипотеза
~~[illegible]~~ имаме основание да твърдим, че средно оценките по Алгебра 1 са по-високи от тези по Алгебра 2.

7)
```
students <- read.table("students.txt", header=T)
attach(students)
```

а) модел: grade = β0 + β1 * study.hours + eps

оценено регресионно уравнение:

grade = 1.01617 + 0.15806 * study.hours

б) При 1 час повече учене, оценката е по-висока средно с 0.15806

в) H0: beta1 = 0

H1: beta1 != 0

p-value = 2.9e-12 < 0.05 → отхвърляме нулевата хипотеза

Имаме основание да твърдим, че има линейна връзка между оценката и часовете отделени за подготовка.

г) модел: grade = β0 + β1 * study.hours + β2 * lectures + eps

оценено регресионно уравнение:

grade = 0.54255 + 0.14767 * study.hours + 0.09432 * lectures

→ При един час повече учене, оценката е по-висока средно с 0.14767 (и фиксиран брой посетени лекции)

→ При една повече посетена лекция оценката е по-висока средно с 0.09432 (и фиксиран брой часове за подготовка)

(1) 200 шофьори
25 са говорили по време на шофиране } 2008 юни

180 шофьори
22 са говорили по ~~[illegible]~~ телефон } 2008 октомври

? през октомври е по-голям отколкото през юни ?

→ x <- (25, 22)
n <- (200, 180)

prop.test(x, n, correct = F)

p1 = % говорили по тел от юни
p2 = % говорили от октомври

H0: p1 = p2
H0 = p1 < p2

ср