Projektna naloga iz statistike

Gal Anton Gorše

4. oktober 2023

Projektna naloga je bila rešena v programskem jeziku Python s pomočjo knjižnice Pandas.

Kazalo

Prva naloga

Zanima nas število otrok v družinah. Imamo populacijo m družin, katerih povprečje je

$$\mu = \frac{x_1 + x_2 + \cdots + x_m}{m},$$

varianca pa

$$\sigma^2 = \frac{(x_1 - \mu)^2 + (x_2 - \mu)^2 + \cdots + (x_m - \mu)^2}{m},$$

kjer smo z x_i označili število otrok i-te družine.

```
kibergrad = pd.read_csv("Kibergrad.csv")
m = len(kibergrad.index)
otroci = kibergrad["'OTROK'"]
globalno_povprecje = otroci.mean()
print(globalno_povprecje)
```

```
0.9479332816843641
```

(a)

Vzamemo vzorec 200 družin

$$(X_1, X_2, \ldots, X_{200}).$$

Vemo, da je povprečje njihovih števil otrok nepristranska cenilka za μ:

$$\overline{X} = \frac{X_1 + X_2 + \cdots + X_{200}}{200}.$$

Če to izračunamo:

```
kibergrad = pd.read_csv("Kibergrad.csv")
otroci = kibergrad["'OTROK'"]
m = len(kibergrad.index)
```

dobimo rezultat

```
0.975
```

(b)

Vemo, da je kvadrat standardne napake $\hat{\mu}$ pri enostavnem slučajnem vzorčenju z n elementi enak

$$\mathrm{SE}^2_{\overline{X}} = \frac{m-n}{m-1}\frac{\sigma^2}{n}.$$

Če uporabimo nepristransko cenilko za varianco

$$\widehat{\sigma^2} = \frac{m-1}{m}\frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2,$$

dobimo nepristransko cenilko

$$\widehat{\mathrm{SE}^2}_{\overline{X}} = \frac{m-n}{m} \frac{1}{n(n-1)} \sum_{i=1}^{n} (X_i - \overline{X})^2.$$

Tako dobimo tudi cenilko

$$\widehat{\mathrm{SE}}_{\overline{X}} = \sqrt{\frac{m-n}{mn}} \sqrt{\frac{1}{(n-1)} \sum_{i=1}^{n} (X_i - \overline{X})^2},$$

ki sicer ni več nepristranska, je pa še zmerom dovolj dobra. Iz knjige [1] sledi, da če je n velik, ampak še zmerom majhen glede na m, potem je $\overline{X}$ porazdeljen približno normalno. Od tod dobimo aproksimativni interval zaupanja:

$$\overline{X} - \mathrm{SE}_{\overline{X}} \cdot \Phi^{-1}\left(1 - \frac{\alpha}{2}\right) \leq \mu \leq \overline{X} + \mathrm{SE}_{\overline{X}} \cdot \Phi^{-1}\left(1 - \frac{\alpha}{2}\right).$$

Ponavadi ne poznamo dejanske $\mathrm{SE}_{\overline{X}}$, zato uporabimo cenilko $\widehat{\mathrm{SE}}_{\overline{X}}$. Tako dobimo interval zaupanja

$$\overline{X} - \widehat{\mathrm{SE}}_{\overline{X}} \cdot \Phi^{-1}\left(1 - \frac{\alpha}{2}\right) \leq \mu \leq \overline{X} + \widehat{\mathrm{SE}}_{\overline{X}} \cdot \Phi^{-1}\left(1 - \frac{\alpha}{2}\right).$$

```
cenilka_se = math.sqrt((m - 200)/(200*m))*vzorec.std()
c = norm.ppf(0.975)
delta = c*cenilka_se
print(povprecje - delta, povprecje + delta)
```

```
0.8179002058696819 1.132099794130318
```

Tako dobimo 95% interval zaupanja:

$$0.82 \leq \mu \leq 1.13.$$

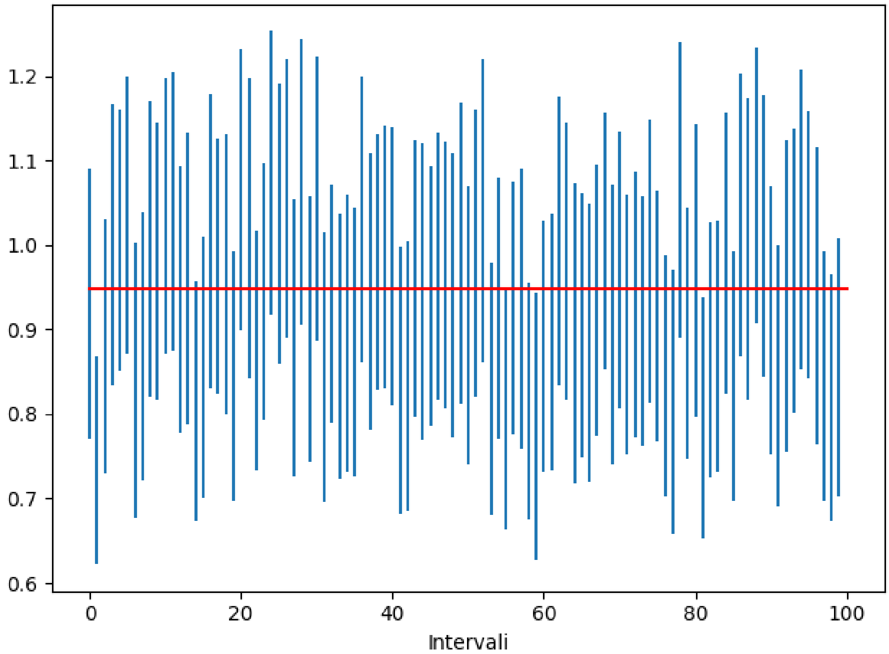
(c) in (d)

Zgornji potek konstrukcije 95% intervala zaupanja ponovimo na 100 vzorcih.

```
x = np.arange(100)
pomozni_x = np.linspace(0, 100, 1000)
podatki = np.empty(100)
intervali = np.empty(100)
stevec = 0

for i in range(100):
    zacasni_vzorec = otroci.sample(200)
    zacasno_povprecje = zacasni_vzorec.mean()
    zacasna_se = math.sqrt((m - 200)/(200*m))*zacasni_vzorec.
                                  std()
    podatki[i] = zacasno_povprecje
    intervali[i] = zacasna_se
    if abs(zacasno_povprecje - globalno_povprecje) <=
                                  zacasna_se*c:
        stevec += 1

fig, ax = plt.subplots()
```

```
    plt.errorbar(x, podatki, yerr=intervali*c, fmt="none")
    plt.plot(pomozni_x, globalno_povprecje*np.ones(1000), "r")
    ax.grid(False)
    ax.set_xlabel("Intervali")
    plt.tight_layout()
    plt.show()
    print(stevec)
```

```
96
```



Slika 1: Intervali zaupanja za μ za vzorce velikosti $n = 200$

Vidimo, da je povprečje populacije μ vsebovano v 96 intervalih zaupanja.

(e)

Standardna napaka cenilke $\overline{X}$ za vzorce velikosti 200 je

$$\mathrm{SE}_{\overline{X}} = \sqrt{\frac{m-200}{m-1}}\frac{\sigma}{\sqrt{200}}$$

(glej točko (b)). Po definiciji je $\mathrm{SE}_{\overline{X}} = \sqrt{\mathrm{var}(\overline{X})} = \sigma_{\overline{X}}$, kar pa je natanko standardna deviacija slučajne spremenljivke $\overline{X}$. Označimo povprečje i-tega vzorca iz točke (d) z $\overline{X}_i$. Očitno so $\overline{X}_1, \overline{X}_2, \ldots, \overline{X}_k$ neodvisne enako porazdeljene slučajne spremenljivke. Empirična standardna deviacija vzorčnih povprečij je enaka

$$\widehat{\sigma}_{\overline{X}} = \sqrt{\frac{1}{k}\sum_{i=1}^{k}\left(\overline{X}_i - \frac{1}{k}\sum_{j=1}^{k}\overline{X}_j\right)^2}.$$

Iz predavanj vemo, da je $\widehat{\sigma}_{\overline{X}}$ dosledna cenilka za standardno deviacijo $\sigma_{\overline{X}}$, torej gre

$$\widehat{\sigma}_{\overline{X}} \xrightarrow[k\to\infty]{d} \sigma_{\overline{X}} = \mathrm{SE}_{\overline{X}}\,.$$

Pričakujemo lahko torej, da bosta za $k = 100$ izračunani vrednosti $\mathrm{SE}_{\overline{X}}$ ter $\widehat{\sigma}_{\overline{X}}$ dovolj blizu. Pa jih izračunajmo.

```
dejanska_se = math.sqrt((m - 200)/(200*(m - 1)))*otroci.std(
                                  ddof=0)
sd_povprecij = math.sqrt(sum((podatki - podatki.mean()*np.ones(
                                  100))*(podatki - podatki.mean
                                  ()*np.ones(100)))*(1/100))
print(dejanska_se, sd_povprecij)
```

```
0.0816404987959038 0.07567725880342124
```

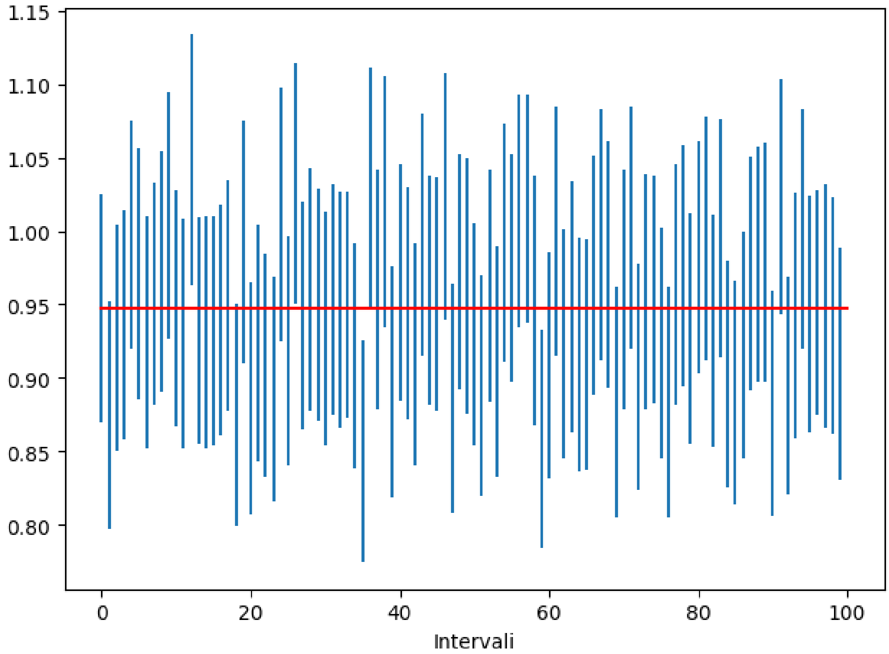
Tako dobimo $\mathrm{SE}_{\overline{X}} = 0.082$ in $\widehat{\sigma}_{\overline{X}} = 0.076$.

(f)

Ponovimo prejšnji dve točki še za vzorce velikosti $n = 800$.

```
podatki_alt = np.empty(100)
intervali_alt = np.empty(100)
stevec_alt = 0

for i in range(100):
    zacasni_vzorec = otroci.sample(800)
    zacasno_povprecje = zacasni_vzorec.mean()
    zacasna_se = math.sqrt((m - 800)/(800*m))*zacasni_vzorec.
                                      std()
    podatki_alt[i] = zacasno_povprecje
    intervali_alt[i] = zacasna_se
    if abs(zacasno_povprecje - globalno_povprecje) <=
                                      zacasna_se*c:
        stevec_alt += 1

fig, ax = plt.subplots()
plt.errorbar(x, podatki_alt, yerr=intervali_alt*c, fmt="none")
plt.plot(pomozni_x, globalno_povprecje*np.ones(1000), "r")
ax.grid(False)
ax.set_xlabel("Intervali")
plt.tight_layout()
plt.show()
print(stevec_alt)
```

```
96
```



Slika 2: Intervali zaupanja za μ za vzorce velikosti $n = 800$

Izračunajmo še $\mathrm{SE}_{\overline{X}}$ ter $\sigma_{\overline{X}}$.

```
dejanska_se_alt = math.sqrt((m - 800)/(800*(m - 1)))*otroci.std
                                    (ddof=0)
sd_povprecij_alt = math.sqrt(sum((podatki_alt - podatki_alt.
                                    mean()*np.ones(100))*(
                                    podatki_alt - podatki_alt.
                                    mean()*np.ones(100)))*(1/99))
print(dejanska_se_alt, sd_povprecij_alt)
```

```
0.040538959874385 0.042084231538569404
```

Dobimo torej, da je $\mathrm{SE}_{\overline{X}} = 0.041$ in $\sigma_{\overline{X}} = 0.042$. Takoj opazimo, da je standardna napaka pri $n = 800$ manjša kot pri $n = 200$ in so intervali zaupanja na grafu krajši. To smo pričakovali, saj je jasno, da z večjimi vzorci dobimo boljšo napoved. Sledi pa tudi iz formule za standardno napako: če sta $n_1 < n_2$ naravni števili, potem je

$$\begin{aligned}\mathrm{SE}_{\overline{X}}^{(n=n_1)} &= \sqrt{\frac{m-n_1}{m-1}}\frac{\sigma}{\sqrt{n_1}} \\ &\geq \sqrt{\frac{m-n_2}{m-1}}\frac{\sigma}{\sqrt{n_2}} \\ &= \mathrm{SE}_{\overline{X}}^{(n=n_2)}.\end{aligned}$$

Opazimo tudi, da je standardna napaka $\mathrm{SE}_{\overline{X}}^{(n=200)}$ približno dvakrat večja od

$\mathrm{SE}_{\overline{X}}^{(n=800)}$. To lahko utemeljimo tudi računsko.

$$\begin{aligned}\frac{\mathrm{SE}_{\overline{X}}^{(n=200)}}{\mathrm{SE}_{\overline{X}}^{(n=800)}} &= \frac{\sqrt{\frac{m-200}{m-1}}\frac{\sigma}{\sqrt{200}}}{\sqrt{\frac{m-800}{m-1}}\frac{\sigma}{\sqrt{800}}} \\ &= \sqrt{\frac{m-200}{m-800}}\sqrt{\frac{800}{200}} \\ &\approx 2.\end{aligned}$$

V splošnem velja, da če je $m >> n$, potem pri množenju n s faktorjem k standardna napaka vzorčnega povprečja zmanjša za približno faktor $\sqrt{k}$.

Druga naloga

Podanih imamo n meritev $x_1, x_2, \ldots, x_n$. Označimo vrstilno statistiko

$$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}.$$

```
mangan = pd.read_csv("Mangan.csv")
mangan = pd.DataFrame(
    mangan[["'ODLITEK1\'", "'ODLITEK2\'",
    "'ODLITEK3\'", "'ODLITEK4\'", "'ODLITEK5\'"]]
        .unstack().reset_index(drop=True),
        columns=["ODLITEK"])

odlitek = mangan["ODLITEK"]
n = len(odlitek)
```

(a)

Najprej izračunamo interkvartilni razmik

$$\mathrm{IQR} = x^+_{\frac{3}{4}} - x^-_{\frac{1}{4}} = 0.18$$

s pomočjo katerega dobimo širino za modificirano Freedman-Diaconisovo pravilo:

$$\text{širina} = \frac{2.6 \cdot \mathrm{IQR}}{\sqrt[3]{n}} = 0.09.$$

Od tod dobimo število stolpcev v histogramu:

$$\left\lfloor \frac{x_{(n)} - x_{(1)}}{\text{širina}} \right\rfloor + 1 = 9.$$

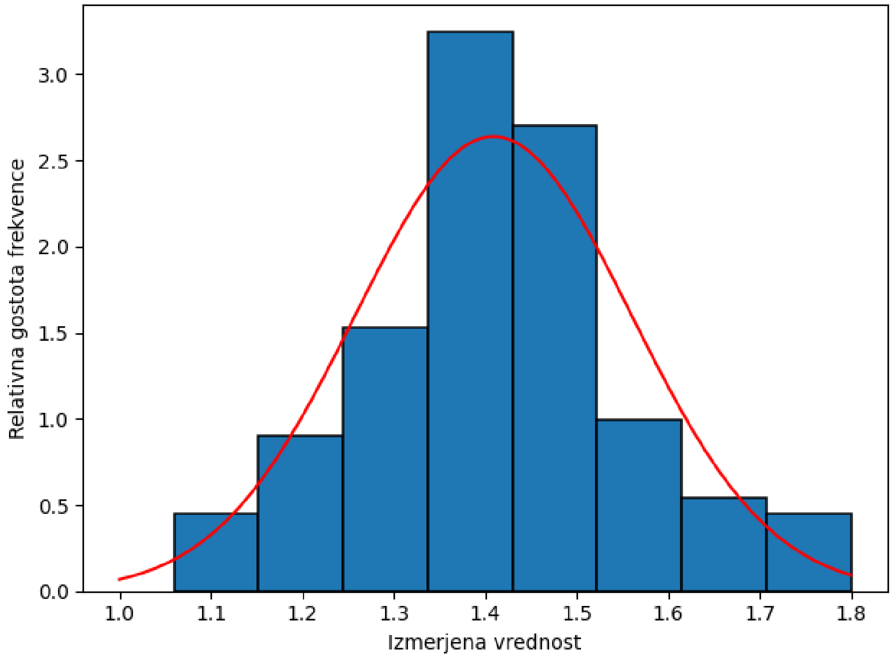
Nadalje izračunamo povprečje meritev

$$\widehat{\mu} = \frac{x_1 + x_2 + \cdots + x_n}{n} = 1.41$$

in popravljeni vzorčni standardni odklon

$$\widehat{\sigma}_+ = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \widehat{\mu})^2} = 0.15.$$

Ti statistiki bomo potrbovali za primerjavo empirične porazdelitve s teoretično (normalno) porazdelitvijo.



Slika 3: Histogram po modificiranem Freedman-Diaconisovem pravilu

Poglejmo si kodo.

```
Q1, Q3 = odlitek.quantile([.25, .75])
IQR = Q3 - Q1
sirina = 2.6 * IQR / pow(120, 1/3)
stevilo = int((odlitek.max() - odlitek.min())/sirina + 1)
mi, sigma = odlitek.mean(), odlitek.std()

fig, ax = plt.subplots(nrows=1, ncols=1)
ax.hist(odlitek, stevilo, density=True, edgecolor="black",
                                linewidth=1.1)
ax.set_xlabel("Izmerjena vrednost")
ax.set_ylabel("Relativna gostota frekvence")
ax.grid(False)
x = np.linspace(1.0, 1.8, 100)
plt.plot(x, stats.norm.pdf(x, mi, sigma), "r")
plt.tight_layout()
plt.show()
print(IQR, sirina, mi, sigma)
```

```
0.17999999999999994 0.09488235113094502 1.409 0.151182454186263
```

(b)

Empirično porazdelitev želimo podrobneje primerjati z normalno. Denimo, da razdelimo vrednosti na k intervalov, pri čemer ima i-ti levo krajišče y_{i-1} in desno

y_i. Označimo z n_i frekvenco ponovitev v i-tem intervalu ter definiramo relativno frekvenco $p_i = \frac{n_i}{n}$. Ker empirično porazdelitev primerjamo z normalno, definiramo še teoretično relativno frekvenco

$$\widehat{p_i} = \Phi\left(\frac{y_i - \widehat{\mu}}{\widehat{\sigma}_+}\right) - \Phi\left(\frac{y_{i-1} - \widehat{\mu}}{\widehat{\sigma}_+}\right)$$

in teoretično frekvenco $\widehat{n_i} = n\widehat{p_i}$. Ker je n_i empirična slučajna spremenljivka, velja

$$\begin{aligned} \mathrm{var}(n_i - \widehat{n_i}) &= \mathrm{var}(n_i) \\ &= np_i(1 - p_i) \\ &= np_i - np_i^2 \\ &\approx np_i, \end{aligned}$$

pri čemer smo predpostavili, da so relativne gostote dovolj majhne. To nam pove, da je variabilnost $n_i - \widehat{n_i}$ večja v tistih intervalih, kjer je p_i večja. Denimo, da bi narisali histogram, kjer ima i-ti interval stolpec z višino $n_i - \widehat{n_i}$. Ker je variabilnost razlikuje od intervala do intervala, je težko oceniti, kdaj gre za dejansko odstopanje od modela.

Rešitev je naslednja. Denimo, da ima naključna spremenljivka X povprečje μ in varianco $\sigma^2(\mu)$, ki je odvisna le od μ. Če je $Y = f(X)$, potem iz [1] sledi

$$\mathrm{var}(Y) \approx \sigma^2(\mu)(f'(\mu))^2.$$

V našem primeru je

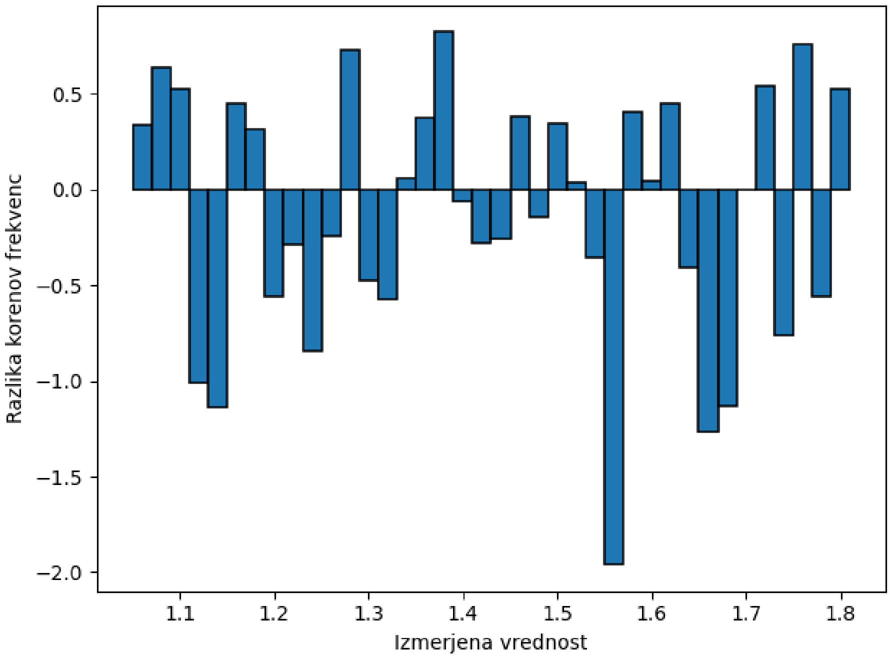
$$\mu = \mathbb{E}(n_i) = np_i \approx \mathrm{var}(n_i) = \sigma^2(\mu).$$

Najti želimo tako funkcijo f, da je $\mu(f'(\mu))^2$ konstanta. Taka funkcija je recimo $f(x) = \sqrt{x}$. Zato narišemo histogram, katerega stolpci so višine $\sqrt{n_i} - \sqrt{\widehat{n_i}}$.

Najprej naredimo primer, ko za vsako vrednost narišemo samostojen stolpec.

```
x = np.linspace(1.05, 1.81, 39)
x_mid = (x[1:] + x[:-1])/2
frekvence = mangan.value_counts()
koreni_razlik = np.empty(38)
stevec = 0
for i in x_mid:
    pricakovana_frekvenca = (stats.norm.cdf((i+ 0.01 - mi)/
                                    sigma) - stats.norm.cdf((
                                    i- 0.01 - mi)/sigma))*n
    if stevec == 0:
        dejanska_frekvenca = len(odlitek[odlitek.between(x[
                                        stevec], x[stevec + 1
                                        ])])
    else:
        dejanska_frekvenca = len(odlitek[odlitek.between(x[
                                        stevec], x[stevec + 1
                                        ], inclusive="right")
                                        ])
    koreni_razlik[stevec] = math.sqrt(dejanska_frekvenca) -
                                    math.sqrt(
                                    pricakovana_frekvenca)
    stevec += 1
```

```
fig, ax = plt.subplots(nrows=1, ncols=1)
ax.bar(x_mid, koreni_razlik, width=0.02, bottom=None, align='
                                        center', edgecolor="black",
                                        linewidth=1.1)
ax.grid(False)
ax.set_xlabel("Izmerjena vrednost")
ax.set_ylabel("Razlika korenov frekvenc")
plt.tight_layout()
plt.show()
```

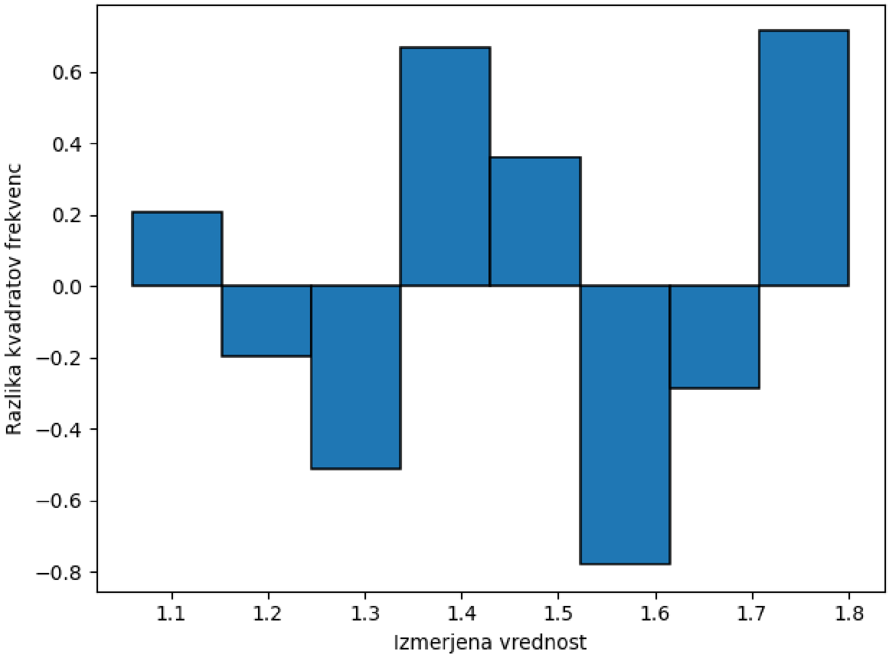


Slika 4: Viseči histogram razlik korenov frekvenc

Sedaj si pa še poglejmo primer, ko izberemo širino intervalov po modificiranem Freedman-Diaconisovem pravilu kot v prejšnji točki.

```
x_alt = np.linspace(1.06, 1.80, 9)
x_alt_mid = (x_alt[1:] + x_alt[:-1]) / 2
koreni_razlik_alt = np.empty(8)
stevec_alt = 0
for i in x_alt_mid:
    pricakovana_frekvenca = (stats.norm.cdf((i + (0.74/8)/2 -
                                    mi)/sigma) - stats.norm.
                                    cdf((i - (0.74/8)/2 - mi)
                                    /sigma))*n
    if stevec_alt == 0:
        dejanska_frekvenca = len(odlitek[odlitek.between(x_alt[
                                        stevec_alt], x_alt[
                                        stevec_alt + 1])])
    else:
        dejanska_frekvenca = len(odlitek[odlitek.between(x_alt[
                                        stevec_alt], x_alt[
                                        stevec_alt + 1],
                                        inclusive="right")])
```

```
        koreni_razlik_alt[stevec_alt] = math.sqrt(
                                          dejanska_frekvenca) -
                                          math.sqrt(
                                          pricakovana_frekvenca)
        stevec_alt += 1

    fig, ax = plt.subplots(nrows=1, ncols=1)
    ax.bar(x_alt_mid, koreni_razlik_alt, 0.74/8, bottom=None, align
                                        ='center', edgecolor="black",
                                         linewidth=1.1)
    ax.set_xlabel("Izmerjena vrednost")
    ax.set_ylabel("Razlika kvadratov frekvenc")
    ax.grid(False)
    plt.tight_layout()
    plt.show()
```

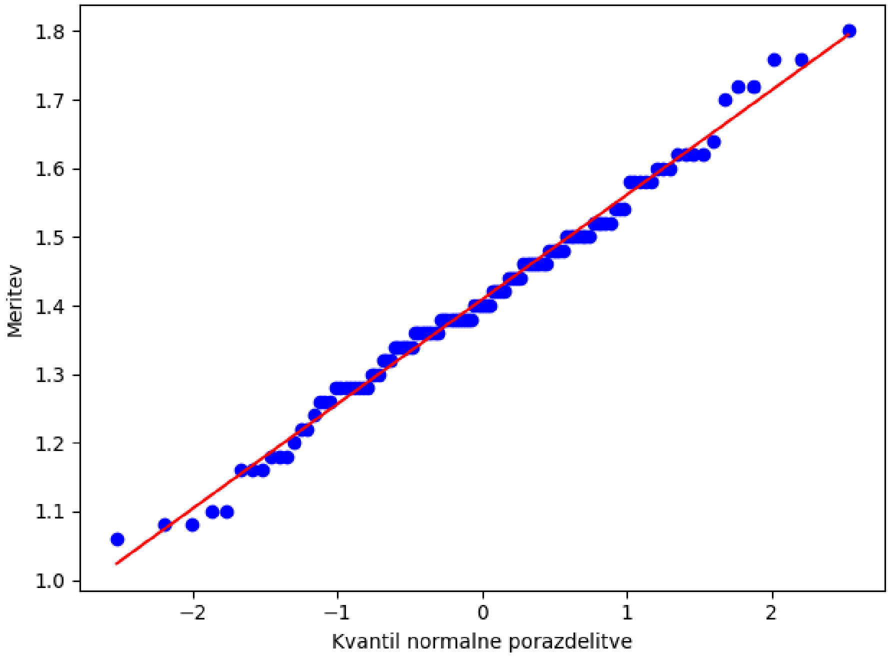


Slika 5: Viseči histogram razlik kvadratov frekvenc pri MFD

(c)

V Q-Q grafikonu prikažemo urejena opažanja proti kvantilom teoretične (v našem primeru seveda normalne) porazdelitve za $\frac{1}{n+1}, \dots, \frac{n}{n+1}$.

```
    fig, ax = plt.subplots(nrows=1, ncols=1)
    stats.probplot(odlitek, dist="norm", plot=plt)
    ax.set_title(None)
    ax.set_xlabel("Kvantil normalne porazdelitve")
    ax.set_ylabel("Meritev")
    ax.grid(False)
    plt.tight_layout()
    plt.show()
```



Slika 6: Q-Q grafikon za normalno porazdelitev

Tretja naloga

V tej nalogi imamo podanih n meritev dolžin odontoblastov morskih prašičkov in količino C vitamina, ki so jo prejeli (v obliki VC ali OJ).

```
zobje = pd.read_csv("Zobje.csv")
zobje_vc = (zobje[zobje["NACIN"]=="VC"])[["DOLZINA", "KOLICINA"
                                          ]]
zobje_oj = (zobje[zobje["NACIN"]=="OJ"])[["DOLZINA", "KOLICINA"
                                          ]].reset_index(drop=True)

def nacin(x):
    if x == "VC":
        return 0.0
    else:
        return 1.0

zobje["NACIN"] = zobje["NACIN"].apply(lambda x: nacin(x))
zobje["NACIN*KOLICINA"] = zobje["NACIN"]*zobje["KOLICINA"]
n = length(zobje)
```

(a)

V prvem delu naloge privzamemo linearni model

$$\text{dolžina} = \beta_0 + \beta_1 \cdot \text{količina} + \varepsilon.$$

Denimo, da je x_i količina odmerka vitamina C (bodisi v obliki VC bodisi OJ) na i-temu morskemu prašičku in Y_i njegova izmerjena dolžina odontoblastov.

Imamo torej

$$\underbrace{\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_{60} \end{bmatrix}}_{\underline{Y}} = \underbrace{\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_{60} \end{bmatrix}}_{\underline{X}} \cdot \underbrace{\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}}_{\underline{\beta}} + \underbrace{\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_{60} \end{bmatrix}}_{\underline{\varepsilon}},$$

kjer za šum velja $\mathbb{E}(\underline{\varepsilon}) = 0$ in $\underline{\varepsilon} \sim N(0, \sigma^2 \underline{I}_{60})$.

Iz predavanj vemo, da je cenilka za $\underline{\beta}$

$$\widehat{\underline{\beta}} = (\underline{X}^\top \underline{X})^{-1} \underline{X}^\top \underline{Y}.$$

Označimo

$$\mathrm{RSS} = \|\underline{Y} - \underline{X}\widehat{\underline{\beta}}\|^2.$$

Na predavanjih smo pokazali, da je

$$\widehat{\sigma_+^2} = \frac{\mathrm{RSS}}{n-2}$$

nepristranska cenilka za σ^2 in za poljuben vektor $\underline{c} \in \mathbb{R}^2$ velja

$$\frac{\underline{c}^\top \widehat{\underline{\beta}} - \underline{c}^\top \underline{\beta}}{\widehat{\mathrm{SE}}_+} \sim \mathrm{Student}(n-2),$$

kjer je

$$\widehat{\mathrm{SE}}_+ = \widehat{\sigma}_+ \sqrt{\underline{c}^\top (\underline{X}^\top \underline{X})^{-1} \underline{c}}. \tag{1}$$

Testiramo, če je $\beta_1 = 0$; naša ničelna hipoteza je torej

$$H_0 : \ \beta_1 = 0,$$

ki jo moramo zavrniti z $100(1-\alpha)\%$ verjetnostjo. Alternativna hipoteza je zato seveda

$$H_1 : \ \beta_1 \neq 0.$$

V enačbi (1) torej vzamemo

$$\underline{c} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

in dobimo

$$\frac{\widehat{\beta_1} - \beta_1}{\widehat{\mathrm{SE}}_+} \sim \mathrm{Student}(n-2).$$

Ničelne hipoteze torej ne zavrnemo natanko tedaj, ko je

$$-F^{-1}_{\mathrm{Student}(n-2)}\left(1 - \frac{\alpha}{2}\right) \leq \frac{\widehat{\beta_1}}{\widehat{\mathrm{SE}}_+} \leq F^{-1}_{\mathrm{Student}(n-2)}\left(1 - \frac{\alpha}{2}\right)$$

oziroma

$$\left|\frac{\widehat{\beta_1}}{\widehat{\mathrm{SE}}_+}\right| \leq F^{-1}_{\mathrm{Student}(n-2)}\left(1 - \frac{\alpha}{2}\right).$$

Ker je funkcija gostote $F_{\text{Student}(n-2)}$ naraščajoča, lahko to zapišemo tudi kot

$$F_{\text{Student}(n-2)}\left(\left|\frac{\widehat{\beta_1}}{\widehat{\text{SE}}_+}\right|\right) \leq 1 - \frac{\alpha}{2}.$$

Zaradi simetričnosti velja $F_{\text{Student}(n-2)}(x) = 1 - F_{\text{Student}(n-2)}(-x)$, zato je zgornja neenakost ekvivalentna

$$F_{\text{Student}(n-2)}\left(\left|\frac{\widehat{\beta_1}}{\widehat{\text{SE}}_+}\right|\right) - F_{\text{Student}(n-2)}\left(-\left|\frac{\widehat{\beta_1}}{\widehat{\text{SE}}_+}\right|\right) \leq 1 - \alpha.$$

To pa pomeni, da je

$$\mathbb{P}\left(-\left|\frac{\widehat{\beta_1}}{\widehat{\text{SE}}_+}\right| \leq X \leq \left|\frac{\widehat{\beta_1}}{\widehat{\text{SE}}_+}\right|\right) \leq 1 - \alpha, \quad X \sim \text{Student}(n-2)$$

oziroma

$$\alpha \leq 1 - \mathbb{P}\left(-\left|\frac{\widehat{\beta_1}}{\widehat{\text{SE}}_+}\right| \leq X \leq \left|\frac{\widehat{\beta_1}}{\widehat{\text{SE}}_+}\right|\right), \quad X \sim \text{Student}(n-2).$$

Izrazu na desni strani neenakosti pravimo p-vrednost. Ničelne hipoteze ne zavrnemo, če je $\alpha \leq p$, sicer pa jo. Sedaj izvedemo test.

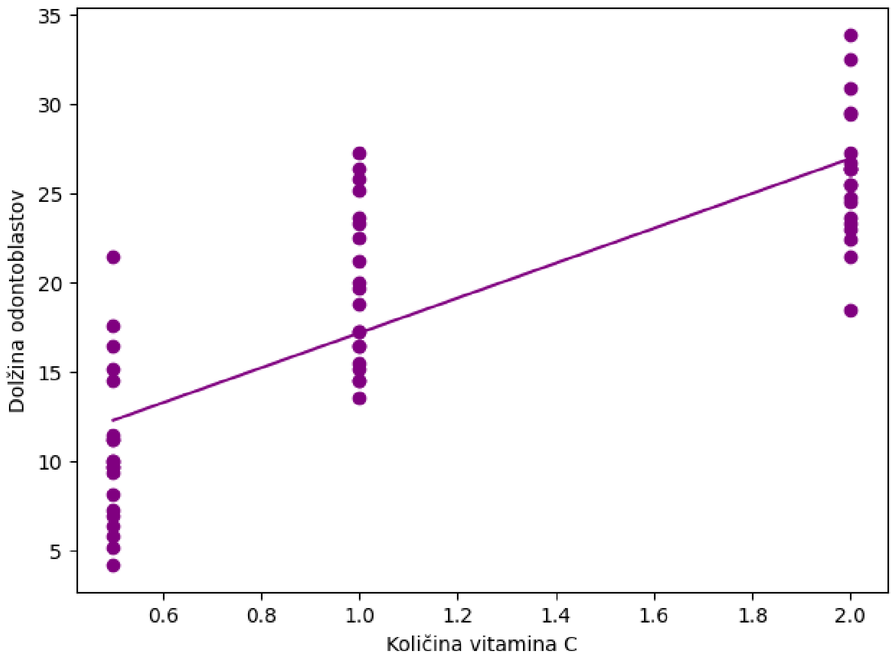
```
neodvisna = zobje["KOLICINA"]
odvisna = zobje["DOLZINA"]
neodvisna = sm.add_constant(neodvisna)
reg = sm.OLS(odvisna, neodvisna).fit()
koef = reg.params
print(reg.summary())
c0, c1 = koef["const"], koef["KOLICINA"]

x = np.linspace(0.5, 2.0, 100)
fig, ax = plt.subplots()
ax.set_facecolor('white')
plt.scatter(zobje["KOLICINA"], zobje["DOLZINA"], c="purple")
plt.plot(x, c0 * np.ones(100) + c1 * x, "purple")
ax.set_xlabel("Kolicina vitamina C")
ax.set_ylabel("Dolzina odontoblastov")
ax.grid(False)
plt.tight_layout()
plt.show()
```

Dobimo rezultat:

	coef	std err	t	$P > \|t\|$	0.025	0.975
const	7.4225	1.260	5.890	0.000	4.900	9.945
KOLICINA	9.7636	0.953	10.250	0.000	7.857	11.670

Slika 7: Tabela koeficientov v regresijskem modelu iz točke (a)

Ker je p-vrednost v stolpcu KOLICINA zelo majhna, zavrnemo ničelno hipotezo tako v primeru $\alpha = 0.05$ kot tudi v primeru $\alpha = 0.01$ in sprejmemo alternativno hipotezo H_1. Prišli smo do zaključka, da dodajanje vitamina C vpliva na dolžino odontoblastov.



Slika 8: Regresijski model iz točke (a)

(b)

Sedaj definiramo indikatorsko spremenljivko način, ki je 0, če je morski prašiček dobil vitamin C v obliki VC, sicer pa je 1. Postavimo model

$$\text{dolžina} = \beta_0 + \beta_1 \cdot \text{količina} + \beta_2 \cdot \text{način} + \beta_3 \cdot \text{količina} \cdot \text{način} + \varepsilon.$$

Od tod sledi, da je za morske prašičke, ki so dobili vitamin C v obliki VC, model enak

$$\text{dolžina} = \beta_0 + \beta_1 \cdot \text{količina},$$

za preostale pa

$$\text{dolžina} = (\beta_0 + \beta_2) + (\beta_1 + \beta_3) \cdot \text{količina}.$$

Razlika v učinkovitosti VC in OJ je torej ravno koeficient β_3.

Sedaj pa nadaljujemo popolnoma enako kot pri prejšnji točki: testiramo ničelno hipotezo

$$H_0 : \ \beta_3 = 0$$

proti alternativni hipotezi

$$H_1 : \ \beta_3 \neq 0.$$

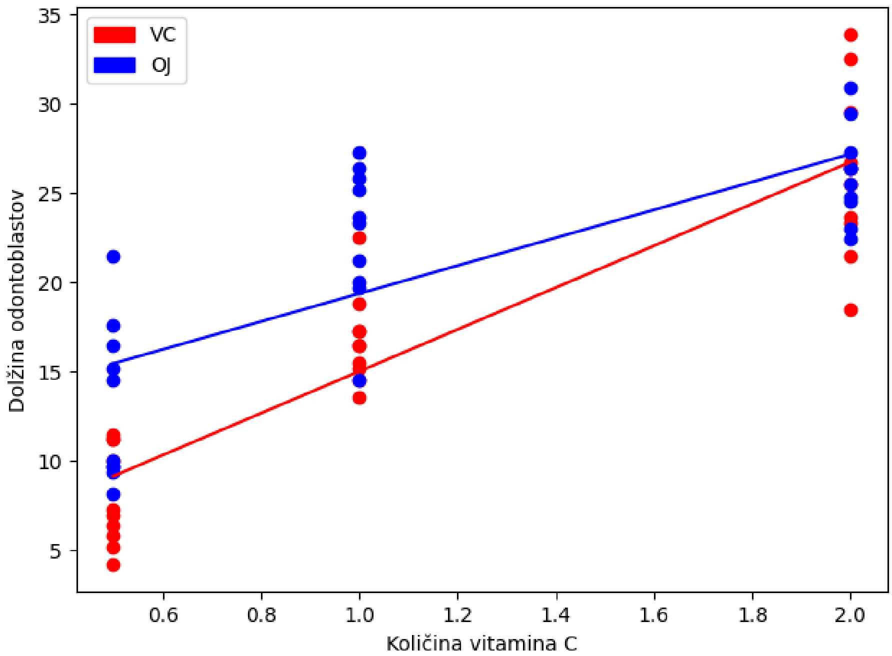
Pa si poglejmo rezultat.

```
neodvisna_alt = zobje[["NACIN", "KOLICINA", "NACIN*KOLICINA"]]
odvisna_alt = zobje["DOLZINA"]
neodvisna_alt = sm.add_constant(neodvisna_alt)
reg_alt = sm.OLS(odvisna_alt, neodvisna_alt).fit()
koef_alt = reg_alt.params
print(reg_alt.summary())
c0_alt, c1_alt, c2_alt, c3_alt = koef_alt["const"], koef_alt["
                                  NACIN"], koef_alt["KOLICINA"]
                                  , koef_alt["NACIN*KOLICINA"]

y_vc = zobje_vc["DOLZINA"]
x_vc = zobje_vc["KOLICINA"]
y_oj = zobje_oj["DOLZINA"]
x_oj = zobje_oj["KOLICINA"]

x = np.linspace(0.5, 2.0, 100)
fig, ax = plt.subplots()
ax.set_facecolor('white')
plt.scatter(x_vc, y_vc, c="red")
plt.scatter(x_oj, y_oj, c="blue")
rdeca = mpatches.Patch(color="red", label="VC")
modra = mpatches.Patch(color="blue", label="OJ")
plt.plot(x, c0_alt * np.ones(100) + c2_alt * x, "r")
plt.plot(x, (c0_alt + c1_alt) * np.ones(100) + (c2_alt + c3_alt
                                  ) * x, "b")
ax.set_xlabel("Kolicina vitamina C")
ax.set_ylabel("Dolzina odontoblastov")
ax.legend(handles=[rdeca, modra])
ax.grid(False)
plt.tight_layout()
plt.show()
```

	coef	std err	t	$P > \|t\|$	0.025	0.975
const	3.2950	1.581	2.084	0.042	0.127	6.463
NACIN	8.2550	2.236	3.691	0.001	3.775	12.735
KOLICINA	11.7157	1.195	9.800	0.000	9.321	14.110
NACIN*KOLICINA	−3.9043	1.691	−2.309	0.025	−7.291	−0.518

Slika 9: Tabela koeficientov v regresijskem modelu iz točke (b)

Iz tabele preberemo, da je p-vrednost koeficienta pri NAČIN*KOLICINA enaka 0.025. Če je $\alpha = 0.05$, potem ničelno hipotezo zavrnemo in zaključimo, da je način doziranja VC učinkovitejši od OJ. Če pa je $\alpha = 0.01$, potem ničelne hipoteze ne ovržemo in zaključimo, da ni dovolj dokazov, da je kateri izmed načinov učinkovitejši. Razlika v naklonih torej ni statistično značilna.



Slika 10: Regresijski model iz točke (b)

Literatura

[1] John Rice. *Mathematical Statistics & Data Analysis*. Duxburry, 2007.