

R FUNKCIJE ZA EKONOMISTE

Tabela s podatki v programu R se imenuje »data frame«. Takšno tabelo lahko poljubno manipuliramo in analiziramo z R funkcijami in R ukazi. Primer takšne tabele, ki jo poimenujemo podatki, je podana spodaj:

	ID	Starost	Višina	Spol	Kraj
1	1	30	180	1	LJ
2	2	25	170	0	LJ
3	3	50	175	1	MB
4	4	40	170	0	LJ
5	5	16	190	1	KP
6	6	22	178	1	NM
7	7	29	177	0	GO

Tabelo manipuliramo z ukazom `podatki[,]`. Zapis pred vejico se nanaša na vrstice, za vejico pa na stolpce. Primeri:

```
podatki[c(-4, -5), ]
podatki[ , c(1, 2, 3)] ali podatki[ , c(1:3)]
podatki[4, 3] <- 172
```

V prvem primeru iz tabele s podatki izbrišemo 4. in 5. vrstico, v drugem primeru izberemo 1., 2. in 3. spremenljivko, v tretjem primeru pa element v 4. vrstici in 3. stolpcu (tj. 170) nadomestimo z vrednostjo 172. Do posamezne spremenljivke dostopamo z znakom `$`. Primer: `podatki$Starost`.

Tabelo s podatki lahko uredimo po velikosti po izbrani spremenljivki z ukazom `order`:

```
podatki[order(podatki$Starost), ]
podatki[order(-podatki$Višina), ]
```

V prvem primeru tabelo s podatki uredimo v rastočem zaporedju glede na starost oseb, v drugem primeru pa v padajočem zaporedju glede na višino oseb.

Znak `<-` (assign left) pripiše funkcijo oz. ukaz, ki je zapisan desno, izbranemu objektu, ki je zapisan levo.

Znak `%>%` (pipe) pomeni »potem«. Za njegovo uporabo moramo namestiti knjižnico `dplyr`.

R funkcije in ukaze pišemo v R Markdown, vsak tekst pa mora biti naveden v narekovajih. Program R je občutljiv na velike/male črke, na ločila, na tip oklepaja ipd., ni pa občutljiv na število presledkov.

Širok format podatkov

	ID	Zaupanje_A	Zaupanje_B
1	1	8	3
2	2	5	5
3	3	4	6
4	4	7	2
5	5	2	9

Dolg format podatkov

	ID	Zaupanje	Faktor
1	1	8	A
2	2	5	A
3	3	4	A
4	4	7	A
5	5	2	A
6	1	3	B
7	2	5	B
8	3	6	B
9	4	2	B
10	5	9	B

Spreminjanje razvrščenih podatkov v nerazvrščene podatke s funkcijo `rep` (glej razlago funkcije v nadaljevanju):

Razvrščeni podatki

	Člani	Frekvenca
1	1	2
2	2	2
3	3	3
4	4	1

Nerazvrščeni podatki

	Člani	Frekvenca
1	1	2
1.1	1	2
2	2	2
2.1	2	2
3	3	3
3.1	3	3
3.2	3	3
4	4	1

R FUNKCIJE ZA EKONOMISTE

OSNOVNE R FUNKCIJE in UKAZI	
install.packages () library ()	Namestitev in aktiviranje nove knjižnice.
<pre>install.packages ("psych") library (psych)</pre> <p>Osnovni funkciji za nameščanje in aktiviranje knjižnic, ki niso še nameščene v programu R. Knjižnico se namesti samo enkrat, nato se jo samo še aktivira. Osnovne knjižnice (npr. base, stats) so nameščene in aktivirane avtomatično.</p>	
help ()	Pomoč pri uporabi funkcij in ukazov.
<pre>help (read.table)</pre> <p>Razlaga, kako deluje posamezna funkcija oz. R ukaz in kako mora biti zapisana. Do pomoči lahko dostopamo tudi s klikom na F1, ko z miško v R Markdowdnu izberemo ime zapisane funkcije.</p>	
read.table ()	Branje .csv datoteke s podatki.
<pre>podatki <- read.table ("./podatki.csv", header = TRUE, sep = ";", dec = ",")</pre> <p>Branje datoteke .csv, ki ima v prvi vrstici imena spremenljivk. Določimo, kako so ločene spremenljivke (sep) in tip decimalnega ločila (dec).</p>	
read_xlsx ()	Branje Excel datoteke s podatki.
<pre>library (readxl) podatki <- read_xlsx ("./podatki.xlsx") podatki <- as.data.frame (podatki)</pre> <p>Branje datoteke shranjene v Excel formatu. Za uporabo funkcije moramo namestiti in aktivirati knjižnico readxl. Ko uvozimo podatke, uporabimo še funkcijo as.data.frame, da objekt (tabelo s podatki) spremenimo v obliko data.frame.</p>	
data.frame ()	Ročen vnos podatkov, ki ustvari tabelo s podatki.
<pre>podatki <- data.frame ("Starost" = c (30, 25, 50), "Višina" = c (180, 170, 175))</pre> <p>Ustvarili smo tabelo s podatki s tremi enotami in dvema spremenljivkama.</p>	

c ()	Ustvarjanje vektorja elementov.
<pre>c (4, 5, 10) c ("M", "M", "Ž")</pre> <p>Kadar navedemo več kot en element, pred oklepaj zapišemo c. Črke/besedila morajo biti vedno navedene v narekovajih.</p>	
sort ()	Razvrščanje vrednosti po velikosti.
<pre>sort (podatki\$Starost, decreasing = FALSE)</pre> <p>Razvrstimo vrednosti spremenljivke v rastočem oz. padajočem zaporedju.</p>	
factor ()	Ustvarjanje faktorja.
<pre>factor (podatki\$Spol, levels = c (1, 0), labels = c ("M", "Ž"))</pre> <p>Kategorialne spremenljivke, katerih vrednosti so običajno kodirane s številskimi kodami, spremenimo v faktor. Šele takrat jih program R ustrezno obravnava kot kategorialne spremenljivke. V argument levels zapišemo kode, s katerimi je spremenljivka vnesena v tabelo s podatki, v argument labels pa zapišemo, kaj te kode pomenijo.</p> <pre>podatki\$SpolF <- factor (podatki\$Spol, levels = c (1, 0), labels = c ("M", "Ž"))</pre> <p>Ustvarili smo novo spremenljivko SpolF, ki smo jo shranili v obstoječo tabelo s podatki.</p>	
head () print () tail ()	Prikaz tabele s podatki.
<pre>head (podatki) print (podatki) tail (podatki)</pre> <p>Prikažemo prvih 6 vrstic tabele s podatki (head), celotno tabelo (print) ali zadnjih 6 vrstic tabele s podatki (tail).</p>	

R FUNKCIJE ZA EKONOMISTE

round() floor() ceiling()	Zaokroževanje vrednosti.
<pre>round(podatki, 2)</pre> <p>Vrednosti v tabeli s podatki zaokrožimo na dve decimalni mesti.</p> <pre>floor(214.8)</pre> <p>Zaokroženje vrednosti navzdol na prvo celo število. Rezultat funkcije je v konkretnem primeru 214.</p> <pre>ceiling(214.8)</pre> <p>Zaokroženje vrednosti navzgor na prvo celo število. Rezultat funkcije je v konkretnem primeru 215.</p>	
ifelse()	Pogojni ukaz.
<pre>ifelse(podatki\$Starost > 22, yes = 1, no = 0)</pre> <p>Funkcija se uporablja za izvedbo pogojnega ukaza. V konkretnem primeru je rezultat funkcije vektor vrednosti, kjer imajo enote, starejše od 22 let, vrednost 1, drugače 0.</p> <pre>ifelse(podatki\$Starost >= 22, yes = 1, no = 0)</pre> <p>V konkretnem primeru je rezultat funkcije vektor vrednosti, kjer imajo enote, stare 22 let ali več, vrednost 1, drugače 0.</p>	
seq()	Zaporedje vrednosti.
<pre>seq(from = 0, to = 100, by = 20)</pre> <p>Funkcija se uporablja za ustvarjanje vektorja vrednosti s konstantno razliko med njimi. V konkretnem primeru je rezultat funkcije <code>c(0, 20, 40, 60, 80, 100)</code>.</p>	
log() exp()	Izračun logaritma in eksponentne funkcije (antilogaritmiranje).
<pre>podatki\$lnVišina <- log(podatki\$Višina) podatki\$lnVišina <- log(podatki\$Višina, base = 10)</pre> <p>Izračun logaritma. V prvem primeru izračunamo naravni logaritem, v drugem primeru desetiški. V konkretnem primeru novo spremenljivko pripišemo obstoječi tabeli s podatki.</p>	

scale()	Standardizacija.
<pre>podatki\$Višina_z <- scale(podatki\$Višina)</pre> <p>Standardizacija posamezne spremenljivke. V konkretnem primeru novo spremenljivko pripišemo obstoječi tabeli s podatki.</p> <pre>podatki_std <- scale(podatki)</pre> <p>Standardizacija tabele s podatki.</p>	
sqrt()	Kvadratni koren.
<pre>sqrt(podatki\$Višina)</pre> <p>Izračun korenjenih vrednosti.</p>	
aggregate()	Agregacija.
<pre>aggregate(podatki\$Višina, by = list(podatki\$Spol), FUN = sum)</pre> <p>Agregacija spremenljivke glede na kategorialno spremenljivko. V konkretnem primeru seštejemo vse višine oseb za vsak spol posebej. V argument FUN vpišemo poljubno funkcijo (sum, mean itd.)</p>	
recode()	Prekodiranje vrednosti.
<pre>library(car) recode(podatki\$Višina, "150:160 = 155; 160:170 = 165; 170:180 = 175; 180:190 = 185")</pre> <p>Prekodiranje razpona vrednosti v novo vrednost. V konkretnem primeru vse vrednosti med 150 in 160 cm prekodiramo v 155 cm itd. (mejne vrednosti so prekodirane v vrednost, v katero so prvič zajete glede na zapisano kodo; v konkretnem primeru se vrednost 160 prekodira v vrednost 155 in ne v 165. Če bi zamenjal vrstni red zapisa kode kot "160:170 = 165; 150:160 = 155 ...", bi se vrednost 160 prekodirala v vrednost 165). Za uporabo funkcije moramo namestiti in aktivirati knjižnico <code>car</code>.</p>	
rep()	Reproduciranje vrednosti.
<pre>podatki[rep(1:nrow(podatki), times = podatki\$Frekvenca),]</pre> <p>S funkcijo reproduciramo vsako vrstico v tabeli s podatki tolikokrat, kolikor znaša frekvenca (argument <code>times</code>). Funkcijo se npr. uporablja takrat, ko imamo podatke v razvrščeni obliki in jih moramo spremeniti v nerazvrščeno obliko.</p>	

R FUNKCIJE ZA EKONOMISTE

replace_with_na()	Zamenjava vrednosti z NA.
<pre>library(dplyr) library(naniar) podatki <- podatki %>% replace_with_na(replace = list(parlament = c(77, 88, 99), policija = c(77, 88, 99), politika = c(77, 88, 99)))</pre>	
Zamenjava številskih kod, ki predstavljajo manjkajoče vrednosti, z vrednostjo NA. Za uporabo funkcije moramo namestiti in aktivirati knjižnico dplyr in naniar.	
drop_na()	Odstranjevanje enot z manjkajočimi vrednostmi.
<pre>library(tidyr) podatki <- drop_na(podatki)</pre>	
Odstranjevanje enot, ki imajo manjkajočo vrednost pri kateri izmed spremenljivk. Za uporabo funkcije moramo namestiti in aktivirati knjižnico tidyr.	
order()	Razvrščanje enot po vrednosti spremenljivke.
<pre>print(podatki[order(-podatki\$Tocke, podatki\$Cas),])</pre>	
Prikažemo tabelo s podatki, razvrščeno v padajočem vrstnem redu po številu doseženih točk, enote z enakim številom doseženih točk pa so razvrščene v rastočem zaporedju po času pisanja.	

OPISNA STATISTIKA

mean() median() sd() var() range() min() max() sum()	Ocena osnovnih parametrov: povprečje, mediana, standardni odklon, varianca, variacijski razmik, minimum, maksimum, vsota.
<pre>mean(podatki\$Starost) median(podatki)</pre>	
Ocena posameznega parametra za izbrano spremenljivko ali celotno tabelo.	
quantile()	Ocena kvantila.
<pre>quantile(podatki\$Starost, p = 0.35)</pre>	
S pomočjo argumenta p, s katerim določimo kvantilni rang, ocenimo poljubni kvantil. V konkretnem primeru ocenimo 35. centil.	
summary()	Ocene nabora parametrov.
<pre>summary(podatki)</pre>	
Funkcija poda oceno minimuma, prvega kvartila, povprečja, mediane, tretjega kvartila in maksimuma. V primeru spremenljivk, definiranih kot faktor, funkcija poda frekvenco posamezne kategorije.	
stat.desc()	Ocene nabora parametrov.
<pre>library(pastecs) stat.desc(podatki) round(stat.desc(podatki), 1)</pre>	
Funkcija poda širok nabor ocen parametrov. Vrednosti so podane s 7 decimalnimi mesti, zato je funkcijo smiselno kombinirati s funkcijo round. Za uporabo funkcije moramo namestiti in aktivirati knjižnico pastecs.	
describe()	Ocene nabora parametrov.
<pre>library(psych) describe(podatki)</pre>	
Funkcija poda širok nabor ocen parametrov. Za uporabo funkcije moramo namestiti in aktivirati knjižnico psych.	

R FUNKCIJE ZA EKONOMISTE

describeBy ()	Ocene nabora parametrov, ločene po kategorijah.
<pre>library(psych) describeBy(podatki\$Starost, group = podatki\$Spol)</pre> <p>Funkcija poda širok nabor ocen parametrov, ločenih po kategorijah (group). Za uporabo funkcije moramo namestiti in aktivirati knjižnico psych.</p>	
sapply ()	Ocene izbranega parametra.
<pre>sapply(podatki, FUN = mean) sapply(podatki, FUN = quantile, probs = c(0.01, 0.99))</pre> <p>Funkcija poda ocene izbranega parametra za vse spremenljivke, ki se nahajajo v tabeli s podatki.</p>	
table ()	Število ponovitev posamezne vrednosti.
<pre>table(podatki\$Spol)</pre> <p>Funkcija poda število ponovitev posamezne vrednosti za izbrano spremenljivko.</p>	
nrow () length ()	Število vrstic. Dolžina vektorja.
<pre>nrow(podatki) length(podatki\$Starost)</pre> <p>Število vrstic v tabeli s podatki oz. dolžina vektorja (spremenljivke).</p>	
frq ()	Frekvenčna tabela.
<pre>library(sjmisc) table(podatki\$Višina)</pre> <p>Frekvenčna tabela za izbrano številsko spremenljivko. Za uporabo funkcije moramo namestiti in aktivirati knjižnico sjmisc.</p>	

STATISTIČNE PORAZDELITEVE

pnorm ()	Izračun ploščine pod krivuljo normalne porazdelitve.
<pre>pnorm(q = 5, mean = 3, sd = 2, lower.tail = TRUE)</pre> <p>Funkcija poda ploščino pod krivuljo gostote verjetnosti za normalno porazdelitev levo od izbrane vrednosti spremenljivke (lower.tail = TRUE) ali desno od izbrane vrednosti (lower.tail = FALSE).</p>	
qnorm ()	Izračun vrednosti spremenljivke za izbrano ploščino za normalno porazdelitev.
<pre>qnorm(p = 0.30, mean = 3, sd = 2, lower.tail = TRUE)</pre> <p>Funkcija poda vrednost spremenljivke pri izbrani ploščini pod krivuljo gostote verjetnosti za normalno porazdelitev. Izbrana ploščina se lahko nahaja levo (lower.tail = TRUE) ali desno od izračunane mejne vrednosti (lower.tail = FALSE).</p>	
rnorm ()	Generiranje slučajnih števil iz določene normalne porazdelitve.
<pre>rnorm(n = 20, mean = 3, sd = 2)</pre> <p>Iz določene normalne porazdelitve generiramo izbrano število (n) slučajnih števil.</p>	
dnorm ()	Izračun gostote verjetnosti za normalno porazdelitev pri izbrani vrednosti spremenljivke.
<pre>dnorm(x = 2.5, mean = 3, sd = 2)</pre> <p>Izračunamo gostoto verjetnosti za normalno porazdelitev pri izbrani vrednosti spremenljivke.</p>	
qt ()	Izračun vrednosti spremenljivke za izbrano ploščino za t-porazdelitev.
<pre>qt(p = 0.05, df = n-1)</pre> <p>Funkcija poda vrednost spremenljivke pri izbrani ploščini pod krivuljo gostote verjetnosti za t-porazdelitev pri izbranih stopinjah prostosti (df). Ploščina se nanaša na območje med $-\infty$ in izračunano mejno vrednostjo.</p>	

VZORČENJE**choose ()**

Število koimbinacij brez ponavljanja.

```
library(combinat)
choose(6, 4)
```

Iz populacije velikosti 6 izberemo vse možno vzorce po 4 enote brez ponavljanja. V konkretnem primeru je to $\binom{6}{4} = 15$. Za uporabo funkcije moramo namestiti in aktivirati knjižnico `combinat`.

combn ()

Izpis vseh možnih vzorcev.

```
library(combinat)
combn(podatki, 4)
```

Izpis vrednosti spremenljivke za vse možne vzorce. Za uporabo funkcije moramo namestiti in aktivirati knjižnico `combinat`.

```
combn(podatki, 4, mean)
```

Izpis ocen povprečja za vse možne vzorce.

VIZUALIZACIJA PODATKOV**hist ()**

Histogram.

```
hist(podatki$Starost,
     main = "Porazdelitev starosti",
     ylab = "Frekvenca",
     xlab = "Starost",
     breaks = seq(from = 20, to = 50, by = 5))
```

Podatke prikažemo s pomočjo histograma. Argument `main` določi naslov grafikona, `ylab` in `xlab` pa imena osi, širino stolpcev pa določimo z argumentom `breaks`.

boxplot ()

Grafikon kvantilov.

```
boxplot(podatki$Starost)
```

Podatke prikažemo s pomočjo grafikona kvantilov. Ta je sestavljen iz pravokotnika, ki določa prvi, drugi in tretji kvartil, ter navpične črte, ki določa minimum in maksimum proučevane spremenljivke. S krogi so označeni potencialni osamelci.

ggplot ()

Orodje za vizualizacijo podatkov.

```
library(ggplot2)
ggplot(podatki, aes(x = Starost)) +
  geom_histogram(binwidth = 5, colour = "gray") +
  ylab("Frekvenca")
```

```
library(ggplot2)
ggplot(podatki, aes(y = Starost)) +
  geom_boxplot()
```

```
library(ggplot2)
ggplot(podatki, aes(y = Višina, x = Starost)) +
  geom_point()
```

Izberemo tabelo s podatki, z argumentom `aes` pa določimo, katero spremenljivko rišemo na posamezni osi. Nato sestavljamo grafikon s pomočjo znakov `+`. Z argumentom `geom_` določimo tip grafikona. Za uporabo funkcije moramo namestiti in aktivirati knjižnico `ggplot2`. Za podrobnejšo razlago glej dokument `ggplot2.pdf`.

scatterplot ()

Razsevni grafikon.

```
library(car)
scatterplot(podatki$Višina ~ podatki$Starost,
            smooth = FALSE,
            boxplots = FALSE,
            regLine = FALSE,
            ylab = "Višina v cm", xlab = "Starost v letih")
```

Razsevni grafikon za par številskih spremenljivk. Prva navedena spremenljivka je na y-osi, druga na x-osi. Z argumentov `smooth = FALSE` izključimo glajenje, z argumentom `boxplots = FALSE` izključimo prikaz grafikona kvantilov na oseh, z argumentom `regLine = FALSE` izključimo linearno regresijsko funkcijo, z argumentoma `ylab` in `xlab` pa poimenujemo osi. Za uporabo funkcije moramo namestiti in aktivirati knjižnico `car`.

```
library(car)
scatterplot(podatki$Višina ~ podatki$Starost | podatki$Spol,
            smooth = FALSE,
            boxplots = FALSE,
            ylab = "Višina v cm", xlab = "Starost v letih")
```

Razsevni grafikon, ločen po kategorialni spremenljivki.

R FUNKCIJE ZA EKONOMISTE

scatterplotMatrix()

Matrika razsevnih grafikonov.

```
library(car)
scatterplotMatrix(podatki, smooth = FALSE)
```

Matrika razsevnih grafikonov za vsak par številskih spremenljivk. Neštevilске spremenljivke moramo iz analize izključiti. Za uporabo funkcije moramo namestiti in aktivirati knjižnico car.

STATISTIČNO PREVERJANJE DOMNEV

t.test()

Domneva o vrednosti aritmetične sredine.
Domneva o razliki dveh aritmetičnih sredin.

```
t.test(podatki$Starost,
      mu = 25,
      alternative = "two.sided")
```

Preverjanje domneve o vrednosti aritmetične sredine. Vrednost v ničelni domnevi vnesemo v argument mu, z argumentom alternative pa določimo ali gre za dvostransko domnevo ("two.sided"), domnevo usmerjeno v levo ("less") ali v desno ("greater").

```
t.test(podatki$Teža1, podatki$Teža2,
      paired = TRUE,
      alternative = "less")
```

Preverjanje domneve o razliki dveh aritmetičnih sredin za odvisna vzorca (paired = TRUE).

```
t.test(podatki$Višina ~ podatki$Spol,
      paired = FALSE,
      var.equal = FALSE,
      alternative = "greater")
```

Preverjanje domneve o razliki dveh aritmetičnih sredin za neodvisna vzorca (paired = FALSE). Z argumentom var.equal = FALSE izberemo Welchov popravek za potencialno razliko v variancah skupin.

anova_test()

Analiza variance za odvisne vzorce, rANOVA.

```
library(rstatix)
anova_test(dv = Zaupanje,
           wid = ID,
           within = Sistem,
           data = podatki_dolgi_format)
```

Določimo odvisno spremenljivko (dv), določimo identifikator enot (wid), določimo faktor, ki pove, kaj meri odvisna spremenljivka (within), podatki pa morajo biti v dolgem formatu. Za uporabo funkcije moramo namestiti in aktivirati knjižnico rstatix.

aov()

Analiza variance za neodvisne vzorce, ANOVA.

```
aov(Plača ~ Država,
    data = podatki)
```

Določimo odvisno spremenljivko in faktor, ki ločuje enote v skupine.

prop.test()

Domneva o vrednosti deleža. Domneva o enakosti dveh deležev.

```
prop.test(x = 150, n = 250,
         p = 0.50,
         correct = FALSE,
         alternative = "two.sided")
```

Določimo, kolikokrat se je zgodil dogodek (x) in število vseh poskusov (n). Z argumentom p določimo ničelno domnevo, popravek izključimo, ter določimo alternativno domnevo.

```
prop.test(c(195, 45), c(234, 89),
         correct = TRUE,
         alternative = "two.sided")
```

Preverjanje domneve o enakosti dveh deležev. V konkretnem primeru preverjamo, če se delež 195/234 statistično razlikuje od deleža 45/89.

shapiro.test()

Domneva, če se spremenljivka porazdeljuje normalno.

```
shapiro.test(podatki$Diferenca)
```

S Shapiro-Wilkovim preizkusom preverimo, če je spremenljivka porazdeljena normalno.

R FUNKCIJE ZA EKONOMISTE

<p>pairwise.t.test()</p> <pre>pairwise.t.test(podatki\$Plača, g = podatki\$Država, paired = FALSE, p.adj = "bonf")</pre> <p>Preverjanje domnev o razliki dveh aritmetičnih sredin za odvisne vzorce (<code>paired = TRUE</code>) ali neodvisne vzorce (<code>paired = FALSE</code>) za vsak par spremenljivk. Pogosto se uporabi Bonferronijev popravek pri izračunu p-vrednosti.</p>	<p>Primerjava vseh parov aritmetičnih sredin s t-preizkusi.</p>
<p>cohens_d() interpret_cohens_d()</p> <pre>library(effectsize) cohens_d(podatki\$Starost, mu = 25) interpret_cohens_d(0.52, rules = "sawilowsky2009")</pre> <p>Določimo velikost učinka, nato pa ga interpretiramo na podlagi različnih pravil. Za uporabo funkcije moramo namestiti in aktivirati knjižnico <code>effectsize</code>.</p>	<p>Velikost učinka, določen na podlagi Cohenove D statistike.</p>
<p>wilcox_test()</p> <pre>wilcox_test(podatki\$Starost, mu = 23, correct = FALSE)</pre> <p>Wilcoxonov preizkus o vrednosti mediane (neparametrični preizkus) izračunamo tako, da določimo vrednost mediane (μ), popravek pa izključimo.</p> <pre>wilcox_text(podatki\$Težal, podatki\$Teža2, paired = TRUE, correct = FALSE, exact = FALSE, alternative = "two.sided")</pre> <p>Wilcoxonov preizkus o enakosti dveh lokacij porazdelitev za odvisne vzorce (Wilcoxonov preizkus predznačenih rangov – neparametrični preizkus) izračunamo tako, da navedemo obe spremenljivki ter določimo argument <code>paired = TRUE</code>. Določimo alternativno domnevo. Popravek in izračun natančne p-vrednosti izključimo.</p> <pre>wilcox_text(podatki\$Višina ~ podatki\$Spol, paired = FALSE, correct = FALSE, exact = FALSE, alternative = "less")</pre>	<p>Wilcoxonov preizkus o vrednosti mediane. Wilcoxonov preizkus o enakosti lokacij dveh porazdelitev za odvisna in neodvisna vzorca.</p>

<p>Wilcoxonov preizkus o enakosti dveh lokacij porazdelitev za neodvisne vzorce (Wilcoxonov preizkus vsot rangov – neparametrični preizkus) izračunamo tako, da navedemo odvisno spremenljivko ter faktor, ki določa, na katero skupino se nanaša opazovanje. Določimo argument <code>paired = FALSE</code> in alternativno domnevo. Popravek in izračun natančne p-vrednosti izključimo.</p>	
<p>binom.test()</p> <pre>binom.test(x = 150, n = 250, p = 0.70, alternative = "two.sided")</pre> <p>Določimo, kolikokrat se je zgodil dogodek (x) in število vseh poskusov (n). Z argumentom <code>p</code> določimo ničelno domnevo, ter določimo alternativno domnevo. V kolikor je <code>p = 0.50</code>, izvedemo preizkus s predznaki.</p>	<p>Binomski preizkus (preizkus o deležu). Preizkus s predznaki.</p>
<p>friedman_test()</p> <pre>library(rstatix) friedman_test(Zaupanje ~ Sistem ID, data = podatki_dolgi_format)</pre> <p>Določimo odvisno spremenljivko in faktor, ki pove, kaj meri odvisna spremenljivka, nato pa še identifikator, ki meritev pripiše posamezni enoti. Podatki pa morajo biti v dolgem formatu. Gre za neparametrični preizkus. Za uporabo funkcije moramo namestiti in aktivirati knjižnico <code>rstatix</code>.</p>	<p>Friedmanova analiza variance.</p>
<p>kruskal.test()</p> <pre>kruskal.test(Plača ~ Država, data = podatki)</pre> <p>Določimo odvisno spremenljivko in faktor, ki ločuje enote v skupine. Gre za neparametrični preizkus.</p>	<p>Kruskal-Wallisov preizkus.</p>

R FUNKCIJE ZA EKONOMISTE

<p>summarySE ()</p> <pre>library(Rmisc) summarySE(podatki, measurevar = "Višina", groupvars = "Spol", conf.interval = 0.95)</pre> <p>Določimo podatke, ki vsebujejo številsko spremenljivko, za katero računamo interval zaupanja za aritmetično sredino (<code>measurevar</code>), in kategorialno spremenljivko, ki vrednosti spremenljivke ločuje po skupinah (<code>groupvars</code>). Za uporabo funkcije moramo namestiti in aktivirati knjižnico <code>Rmisc</code>.</p>	<p>Izračun parametrov intervalov zaupanja za aritmetično sredino po skupinah.</p>
<p>VarTest ()</p> <pre>library(DescTools) VarTest(podatki\$Višina, sigma.squared = 5, alternative = "two.sided")</pre> <p>Preverjanje domneve o vrednosti variance. Z argumentom <code>sigma.squared</code> določimo vrednost v ničelni domnevi, z argumentom <code>alternative</code> pa določimo ali gre za dvostransko domnevo ("two.sided"), domnevo usmerjeno v levo ("less") ali v desno ("greater"). Za uporabo funkcije moramo namestiti in aktivirati knjižnico <code>DescTools</code>.</p> <pre>library(DescTools) VarTest(podatki\$Višina ~ podatki\$Spol, alternative = "two.sided")</pre> <p>Preverjanje domneve o enakosti dveh varianc (ali je njuno razmerje enako 1). V konkretnem primeru preverjamo, če se varianci višine med spoloma razlikujeta, z argumentom <code>alternative</code> pa določimo ali gre za dvostransko domnevo ("two.sided"), domnevo usmerjeno v levo ("less") ali v desno ("greater"). Za uporabo funkcije moramo namestiti in aktivirati knjižnico <code>DescTools</code>.</p>	<p>Domneva o vrednosti variance. Domneva o enakosti dveh varianc.</p>

ANALIZA KATEGORIALNIH SPREMENLJIVK

<p>chisq.test ()</p> <pre>chisq.test(podatki\$Spol, podatki\$Kraj, correct = FALSE)</pre> <p>S preizkusom preverimo, če obstaja povezanost med dvema kategorialnima spremenljivkama. V kolikor imate obe spremenljivki natanko dve kategoriji, uporabimo argument <code>correct = TRUE</code>.</p> <pre>chisq.test(x = c(150, 200, 350), p = c(0.30, 0.30., 0.40), correct = FALSE)</pre> <p>S preizkusom preverimo skladnost dveh porazdelitev. V argument <code>p</code> zapišemo pričakovane verjetnosti, na podlagi katerih so določene pričakovane frekvence.</p>	<p>Pearsonov χ^2-preizkus.</p>
<p>fisher.test ()</p> <pre>fisher.test(podatki\$Spol, podatki\$Kraj)</pre> <p>S preizkusom preverimo, če obstaja povezanost med dvema kategorialnima spremenljivkama.</p>	<p>Fisherjev natančni preizkus.</p>

KORELACIJSKA IN REGRESIJSKA ANALIZA

cor()
cor.test() | Ocena korelacijskega koeficienta. Preizkus domneve o vrednosti korelacijskega koeficienta.

```
cor(podatki$Starost, podatki$Višina,
    method = c("pearson", "spearman"))
```

Ocena korelacijskega koeficienta med dvema številskima spremenljivkama. Izbiramo lahko med Pearsonovim ali Spearmanovim koeficientom korelacije.

```
cor(podatki)
```

Ocena korelacijske matrike. V tabeli s podatki morajo biti samo številске spremenljivke.

```
cor.test(podatki$Starost, podatki$Višina,
         method = c("pearson", "spearman"))
```

Domneva o vrednosti korelacijskega koeficienta.

rcorr() | Korelacijska matrika.

```
library(Hmisc)
rcorr(as.matrix(podatki), type = c("pearson", "spearman"))
```

Ocena korelacijske matrike, ki vključuje ali Pearsonove ali Spearmanove korelacijske koeficiente za vsak par številskih spremenljivk, pod matriko pa so podane tudi p -vrednosti za preizkus domneve o vrednosti korelacijskega koeficienta. Za uporabo funkcije moramo namestiti in aktivirati knjižnico `Hmisc`.

pcor() | Ocena parcialnega korelacijskega koeficienta.

```
library(ppcor)
pcor(podatki)
```

Ocena parcialnega korelacijskega koeficienta za vsak par številskih spremenljivk v tabeli s podatki. Za uporabo funkcije moramo namestiti in aktivirati knjižnico `ppcor`.

lm() | Ocena linearnega regresijskega modela (OLS metoda).

```
lm(Višina ~ Starost + Spol + Starost:Spol,
    data = podatki)
```

Ocena linearne regresijske funkcije po metodi najmanjših kvadratov. Navedemo odvisno spremenljivko, znak tilda (~), pojasnjevalne spremenljivke pa ločujemo z znakom +. Interakcije vključimo tako, da med spremenljivki navedemo znak :

```
fit <- lm(Višina ~ Starost + Spol,
          data = podatki)
summary(fit)
```

Prikaz rezultatov ocenjene regresije.

glm() | Ocena linearnega regresijskega modela (ML metoda).

```
glm(Višina ~ Starost + Spol,
    data = podatki)
```

Ocena linearne regresijske funkcije po metodi največjega verjetja. Navedemo odvisno spremenljivko, znak tilda (~), pojasnjevalne spremenljivke pa ločujemo z znakom +. Interakcije vključimo tako, da med spremenljivki navedemo znak :

```
glm(Kadilec ~ Starost + Spol,
    family = binomial,
    data = podatki)
```

Ocena binarne logistične regresijske funkcije.

```
fit <- glm(Kadilec ~ Starost + Spol,
          family = binomial,
          data = podatki)
summary(fit)
```

Prikaz rezultatov ocenjene regresijske funkcije.

ols_test_breusch_pagan() | Breusch-Paganov preizkus heteroskedastičnosti.

```
library(olsrr)
ols_test_breusch_pagan(fit)
```

Preizkus za prisotnost heteroskedastičnosti. Za uporabo funkcije moramo namestiti in aktivirati knjižnico `olsrr`.

R FUNKCIJE ZA EKONOMISTE

shapiro.test()	Domneva, če se spremenljivka porazdeljuje normalno.
<pre>shapiro.test(podatki\$StdOstanki)</pre> <p>S Shapiro-Wilkovim preizkusom preverimo, če je spremenljivka porazdeljena normalno.</p>	
vif()	Preverba multikolinearnosti.
<pre>vif(fit)</pre> <p>Preverba stopnje povezanosti med pojasnjevalnimi spremenljivkami.</p>	
lm.beta()	Ocena standardiziranih parcialnih regresijskih koeficientov.
<pre>library(lm.beta)</pre> <pre>lm.beta(fit)</pre> <p>Ocena standardiziranih parcialnih regresijskih koeficientov regresijskega modela. Za uporabo funkcije moramo namestiti in aktivirati knjižnico <code>lm.beta</code>.</p>	
anova()	Primerjava dveh regresijskih modelov glede na prileganje podatkom.
<pre>anova(fit1, fit2)</pre> <p>Statistična primerjava dveh regresijskih modelov, ocenjenih po metodi najmanjših kvadratov.</p> <pre>anova(fit1, fit2, test = "Chi")</pre> <p>Statistična primerjava dveh regresijskih modelov, ocenjenih po metodi največjega verjetja.</p>	
lm_robust()	Ocena linearnega regresijskega modela z robustnimi standardnimi napakami.
<pre>library(estimatr)</pre> <pre>lm_robust(Višina ~ Starost + Spol,</pre> <pre> se_type = "HC1",</pre> <pre> data = podatki)</pre> <p>Ocena regresijske funkcije z Whitovimi robustnimi standardnimi napakami. Popravek uporabimo takrat, ko je kršena predpostavka homoskedastičnosti. Za uporabo funkcije moramo namestiti in aktivirati knjižnico <code>estimatr</code>.</p>	

lme()	Ocena večnivojskega linearnega regresijskega modela (ML metoda).
<pre>library(nlme)</pre> <pre>fit <- lme(Stres ~ Starost + SpolF + VelikostF,</pre> <pre> random = ~ 1 ID_bolniscnica / ID_oddelek,</pre> <pre> method = "ML",</pre> <pre> data = podatki)</pre> <p>Ocena večnivojske (hierarhične) regresije po metodi največjega verjetja. Specificiramo regresijski model, z argumentom <code>random</code> pa določimo slučajno regresijsko konstanto. V konkretnem primeru je ta določena na nivoju bolnišnice in oddelka. Za uporabo funkcije moramo namestiti in aktivirati knjižnico <code>nlme</code>.</p>	

METODA GLAVNIH KOMPONENT IN FAKTORSKA ANALIZA

cortest.bartlett()	Bartlettov preizkus sferičnosti.
<pre>library(psych)</pre> <pre>cortest.bartlett(R, n = nrow(podatki))</pre> <p>Bartlettov preizkus sferičnosti korelacijske matrike. Vstopni element je ocena korelacijske matrike, pripravljena s funkcijo <code>cor()</code>, določiti pa moramo tudi število enot v vzorcu. Za uporabo funkcije moramo namestiti in aktivirati knjižnico <code>psych</code>.</p>	
det()	Determinanta korelacijske matrike.
<pre>det(R)</pre> <p>Determinanta korelacijske matrike, ki jo uporabljamo za preverbo stopnje povezanosti med spremenljivkami.</p>	
KMO()	Kaiser-Meyer-Olkinova mera vzorčne ustreznosti.
<pre>library(psych)</pre> <pre>KMO(R)</pre> <p>Izračun KMO statistike in posameznih MSA statistik za preverbo ustreznosti posamezne spremenljivke. Za uporabo funkcije moramo namestiti in aktivirati knjižnico <code>psych</code>.</p>	

PCA ()	Izvedba metode glavnih komponent.
<pre>library(FactoMineR) library(factoextra) PCA(podatki, scale.unit = TRUE, ncp = 3, graph = FALSE)</pre>	
<p>Izvedba metode glavnih komponent na izbranih številskih spremenljivkah. Z argumentom <code>scale.unit = TRUE</code> izvedemo standardizacijo spremenljivk, z argumentom <code>ncp</code> določimo število komponent, grafikon pa izključimo. Za uporabo funkcije moramo namestiti in aktivirati knjižnici <code>FactoMineR</code> in <code>factoextra</code>.</p>	
<pre>mgk <- PCA(podatki, scale.unit = TRUE, graph = FALSE)</pre>	
<p><code>get_eigenvalue(mgk)</code></p> <p>Prikažemo lastne vrednosti glavnih komponent.</p>	
<pre>fviz_eig(mgk, choice = "eigenvalue", addlabels = TRUE)</pre>	
<p>Grafikon lastnih vrednosti.</p>	
<pre>fviz_pca_var(mgk, repel = TRUE)</pre>	
<p>Grafičen prikaz uteži glavnih komponent.</p>	
<pre>fviz_pca_biplot(mgk)</pre>	
<p>Grafičen prikaz položaja enot glede na vrednosti prvih dveh glavnih komponent.</p>	
fa.parallel ()	Paralelna analiza.
<pre>library(psych) fa.parallel(podatki, sim = FALSE, fa = c("pc", "fa"))</pre>	
<p>Paralelna analiza za določanje števila glavnih komponent oz. faktorjev. V tabelo s podatki vključimo samo številске spremenljivke, izključimo simulacijo, z argumentom <code>fa</code> pa določimo, ali izvajamo metodo glavnih komponent (<code>pc</code>) ali faktorsko analizo (<code>fa</code>). Za uporabo funkcije moramo namestiti in aktivirati knjižnico <code>psych</code>.</p>	

corPlot ()	Grafičen prikaz korelacijske matrike.
<pre>library(psych) corPlot(R)</pre>	
<p>Grafičen prikaz korelacijske matrike. Vstopni element je ocena korelacijske matrike, pripravljena s funkcijo <code>cor ()</code>. Barve določajo moč povezanosti za vsak par spremenljivk. Za uporabo funkcije moramo namestiti in aktivirati knjižnico <code>psych</code>.</p>	
fa ()	Izvedba faktorske analize.
<pre>library(psych) library(GPARotation) fa(podatki, covar = FALSE, nfactors = 3, fm = "minres", rotate = "oblimin", inpute = "mean")</pre>	
<p>Izvedba faktorske analize na izbranih številskih spremenljivkah. Z argumentom <code>covar</code> določimo, ali izvajamo analiza na kovariančni (<code>covar = TRUE</code>) ali na korelacijski matriki (<code>covar = FALSE</code>), z argumentom <code>nfactors</code> določimo število faktorjev, z argumentom <code>fm</code> določimo faktorsko metodo (priporočena uporaba <code>minres</code> – minimum residual method), z argumentom <code>rotate</code> določimo tip faktorske rotacije (poševna <code>oblimin</code> ali pravokotna <code>varimax</code>). Manjkajoče vrednosti lahko nadomestimo s povprečji po spremenljivkah (<code>inpute = "mean"</code>). Za uporabo funkcije moramo namestiti in aktivirati knjižnici <code>psych</code> in <code>GPARotation</code>.</p>	
<pre>faktorska <- fa(podatki, covar = FALSE, nfactors = 3, fm = "minres", rotate = "oblimin", inpute = "mean")</pre>	
<pre>print.psych(faktorska, cut = 0.3, sort = TRUE)</pre>	
<p>Prikažemo rezultate faktorske analize. Nizke faktorske uteži niso prikazane (<code>cut</code>), prikazane faktorske uteži pa so razporejene po velikosti (<code>sort</code>).</p>	
<pre>fa.diagram(faktorska)</pre>	
<p>Grafičen prikaz faktorskega modela.</p>	

R FUNKCIJE ZA EKONOMISTE

alpha () | Cronbachova alfa.

```
library(psych)
alpha(podatki, check.keys = TRUE)
```

Izračun Cronbachove alfe za izbrane številске spremenljivke. Z argumentom `check.keys = TRUE` zagotovimo, da so vsi indikatorji obrnjeni v isto smer. Za uporabo funkcije moramo namestiti in aktivirati knjižnico `psych`.

RAZVRŠČANJE V SKUPINE

get_dist () | Izračun razdalj med enotami.

```
library(factoextra)
get_dist(podatki, method = "euclidean")
```

Izračun evklidskih razdalj med enotami. Tip razdalje določimo z argumentom `method`. Pogosto se uporablja še razdalje "manhattan" in "minkowski". Za uporabo funkcije moramo namestiti in aktivirati knjižnico `factoextra`.

```
razdalje <- get_dist(podatki, method = "euclidean")
fviz_dist(razdalje)
```

Grafičen prikaz matrike razdalj.

get_clust_tendency () | Izračun Hopkinsove statistike.

```
library(factoextra)
get_clust_tendency(podatki,
  n = nrow(podatki)-1,
  graph = FALSE)
```

Izračun Hopkinsove statistike, ki preveri, če so podatki primerni za izvedbo razvrščanja.

hclust () | Hierarhično razvrščanje.

```
library(dplyr)
library(factoextra)
WARD <- podatki %>%
  get_dist(method = "euclidean") %>%
  hclust(method = "ward.D2")
```

Izvedba hierarhičnega razvrščanja na podlagi izbranih številskih spremenljivk. V prvem koraku izberemo podatke, potem (`%>%`) izračunamo razdalje med enotami (`get_dist`), nato pa na njihovi podlagi izvedemo hierarhično razvrščanje (`hclust`) po izbranem algoritmu (`method`). Priporočena je uporaba Wardovega algoritma, ki ga kombiniramo s kvadratno evklidsko razdaljo (`ward.D2`). Za uporabo funkcije moramo namestiti in aktivirati knjižnici `dplyr` in `factoextra`.

```
fviz_dend(WARD)
```

Prikaz dendrograma (drevo razvrščanja).

```
cutree(WARD, k = 3)
```

Razvrstitev v izbrano število skupin.

hkmeans () | Razvrščanje po metodi voditeljev.

```
library(factoextra)
hkmeans(podatki,
  k = 3,
  hc.metric = "euclidean",
  hc.method = "ward.D2")
```

Izvedba razvrščanja po metodi voditeljev, pri tem pa za začetne voditeljev izberemo rešitev, dobljeno po hierarhičnem pristopu. Določiti moramo število skupin (`k`), v katere želimo enote združiti, argumenta `hc.metric` in `hc.method` pa se naj ujemata z izbranim algoritmom pri hierarhičnem razvrščanju, na podlagi katerega smo se odločili o številu skupin. Za uporabo funkcije moramo namestiti in aktivirati knjižnico `factoextra`.

```
metodavod <- hkmeans(podatki,
  k = 3,
  hc.metric = "euclidean",
  hc.method = "ward.D2")
```

```
fviz_cluster(metodavod)
```

Grafičen prikaz razvrstitve v skupine, pri tem pa sta na oseh prikazani prva in druga glavna komponenta.

ČASOVNE VRSTE	
lm()	Izračun linearnega ali eksponentnega trenda (OLS metoda).
<pre>lm(Nocitve ~ t, data = podatki)</pre> <p>Linearna funkcija trenda. Spremenljivka t predstavlja čas, ki teče 1, 2, 3, ..., N.</p> <pre>lm(logNocitve ~ t, data = podatki)</pre> <p>Eksponentna funkcija trenda. Spremenljivka t predstavlja čas, ki teče 1, 2, 3, ..., N.</p>	
ts()	Določitev časovne vrste.
<pre>ts(podatki\$Nocitve, start = c(2020, 1), end = c(2022, 12), frequency = 12)</pre> <p>Spremenljivko opredelimo kot časovno vrsto. Z argumentom start določimo prvo opazovanje (v konkretnem primeru je to 2020, prvi mesec), z argumentom end določimo zadnje opazovanje (v konkretnem primeru je to 2022, zadnji mesec), z argumentom frequency pa določimo, koliko časovnih enot je znotraj ene periode. V konkretnem primeru so podatki podani na mesečni ravni, skupaj 36 opazovanj.</p> <pre>prenocitve <- ts(podatki\$Nocitve, start = c(2020, 1), end = c(2022, 12), frequency = 12)</pre> <pre>plot(prenocitve, ylab = "Mesecno stevilo nocitev")</pre> <p>Grafičen prikaz časovne vrste.</p>	

decompose()	Razčlenitev časovne vrste.
<pre>decompose(prenocitve, type = "multiplicative")</pre> <p>Razčlenitev časovne vrste na osnovne elemente (trend, ciklična komponenta, perioda in iregularna komponenta). Spremenljivka, ki jo razčlenjujemo, mora biti rezultat funkcije ts(). Pri tipu razčlenitve izberemo multiplikativni pristop.</p> <pre>prenocitve <- ts(podatki\$Nocitve, start = c(2020, 1), end = c(2022, 12), frequency = 12)</pre> <pre>razclenitev <- decompose(prenocitve, type = "multiplicative")</pre> <pre>plot(razclenitev)</pre> <p>Grafičen prikaz razčlenitve časovne vrste.</p>	