Analiza podataka i obrada informacija - službeni šalabahter #2

Učitavanje podataka

```
data("mtcars") # učitavanje ugrađenih R podataka
cars <- mtcars

getwd() # vraća aktivni radni direktorij
podaci <- read.csv("data/primjer.csv") # učitavanje CSV datoteke

podaci <- read.table("data/primjer.txt", header = TRUE, sep = " ") # učitavanje TXT
datoteke

install.packages("readxl")
library(readxl)
podaci <- read_excel("data/primjer.xlsx", sheet = 1) # učitavanje XLSX datoteke
```

Kreiranje i korištenje slučajnog uzorka

```
SEED <- 1234567890
set.seed(SEED)

podaci <- c("Marko", "Ana", "Ivan")
imena <- sample(podaci, 2) # biranje 2 nasumična podatka

redci <- sample(nrow(podaci), 20) # biranje 20 nasumičnih redaka
stupci <- sample(ncol(podaci), 5)  # biranje 5 nasumičnih stupaca

# izdvajanje % redaka za train/test podatke
indeksi <- sample(1:nrow(podaci), 0.7 * nrow(podaci))
train <- podaci[indeksi, ]
test <- podaci[-indeksi, ]
```

Provjera i čišćenje podataka

```
is.na(podaci) # provjera nedostajućih vrijednosti
colSums(is.na(podaci)) # broj nedostajućih vrijednosti po stupcima

podaci_clean <- na.omit(podaci) # uklanjanje redaka s nedostajućim vrijednostima
podaci_clean2 <- podaci[complete.cases(podaci),] # uklanjanje redaka s nedostajućim
vrijednostima
```

Korelacija

```r
cor(x, y) # izračun korelacije (Pearson)

cor(x, y, method = "pearson")  # Pearsonov koeficijent korelacije
cor(x, y, method = "spearman") # Spearmanov koeficijent korelacije

cor(podaci) # korelacijska matrica
pairs(podaci) # vizualizacija korelacijske matrice

# ostale vizualizacije korelacijske matrice
library(corrplot) # install.packages("corrplot")

corrplot(cor_matrix) # method = "circle" - ako nije postavljeno
corrplot(cor_matrix, method = "color")
corrplot(cor_matrix, method = "number")
corrplot(cor_matrix, method = "square")
corrplot(cor_matrix, method = "ellipse")
corrplot(cor_matrix, method = "pie")

cov(x, y) # izračun kovarijance
```

Regresijska analiza

```r
# Linearna regresija
model <- lm(mpg ~ wt + hp, data = mtcars)
summary(model)
# *** → vrlo značajno (p < 0.001)
# ** → značajno (p < 0.01)
# * → umjereno značajno (p < 0.05)
# . → slabo značajno (p < 0.1)
# (prazno) → nije značajno (p ≥ 0.1)

# Logistička regresija
model <- glm(income_bin ~ age + education.num, data = adult, family = binomial)

# Provjera multikolearnosti
library(car) # install.packages("car")
vif(model)

sqrt(vif(model)) > 2 # Ako je sve false nema problema multikolinearnosti

# Izračunavanje vjerojatnosti
prob <- predict(model, type = "response")

pred_class <- ifelse(prob > 0.5, 1, 0)
table(Predicted = pred_class, Actual = adult$income_bin)
```

Vremenski nizovi

```r
library(tseries) # install.packages("tseries")
podaci <- c(23,45,67,34,56,78,89,45,67,89,43,56,
            34,67,89,54,76,88,91,52,71,85,49,63)

# stvaranje vremenskog niza
vremenski_niz <- ts(podaci, start = c(2020, 1), frequency = 12)

# čišćenje od nedostajućih vrijednosti i outliera
vremenski_niz <- tsclean(vremenski_niz)

# vizualizacija vremenskog niza
plot(vremenski_niz)

# Vizualizacija s pomoću moving average funkcije
vremenski_niz_ma3 <- ma(vremenski_niz, order = 3)
plot(vremenski_niz_ma3, main = "Pomični prosjek (red 3)")

# <===============| Dekompozicija vremenskog niza |===============>
dekompozicija <- decompose(vremenski_niz)
plot(dekompozicija)

# STL dekompozicija
stl_dekomp <- stl(vremenski_niz, s.window = "periodic")
plot(stl_dekomp, main = "STL dekompozicija")

# Aditivna dekompozicija
decomp_add <- decompose(vremenski_niz, type = "additive")
plot(decomp_add)

# Multiplikativna dekompozicija
decomp_mult <- decompose(vremenski_niz, type = "multiplicative")
plot(decomp_mult)

# <===============| Stacionarnosti vremenskog niza |===============>
library(tseries) # install.packages("tseries")

# Provjera stacionarnosti
adf_test <- adf.test(vremenski_niz)
adf_test # ako je p-vrijednost > 0.05 -> niz nije stacionaran

# Prilagodba niza za stacionarnost
diferencirani_niz <- diff(vremenski_niz, 1) # Diferenciranje
adf.test(diferencirani_niz)

ndiffs(vremenski_niz) # Vraća broj potrebnih diferencijacija za stacionarnost

# Autokorelacijske funkcije - ACF i PACF
acf(vremenski_niz, main = "ACF - vremenski_niz")
pacf(vremenski_niz, main = "PACF - vremenski_niz")
```

```r
# <===============| ARIMA model |===============>
ap_arima <- auto.arima(vremenski_niz)
summary(ap_arima)

# ARIMA predikcija
library(forecast) # install.packages("forecast")
ap_forecast_arima <- forecast(ap_arima, h = 36)

# Vizualizacija ARIMA predikcije
library(ggplot2)  # install.packages("ggplot2")
autoplot(ap_forecast_arima) + ggtitle("ARIMA predikcija")
```

Klasifikacija

```r
library(rpart) #install.packages("rpart")
library(rpart.plot) #install.packages("rpart.plot")

# Podjela podataka na trening i test skup
set.seed(123)
indeksi <- sample(1:nrow(iris), 0.7*nrow(iris))
train <- iris[indeksi,]
test <- iris[-indeksi,]

# Izgradnja stabla
stablo <- rpart(varijabla ~ ., data = train, method = "class")

# Vizualizacija
rpart.plot(stablo)

# najvažnije varijable
stablo$variable.importance

# tablica parametara rezanja stabla
stablo$cptable

# podrezivanje stabla
podreznao_stablo <- prune(stablo, cp=0.0100000)

# <===============| Predikcija i evaluacija |===============>
library(caret) #install.packages("caret")

predikcije <- predict(stablo, test, type = "class") # Predikcija

confusionMatrix(predikcije, test$varijabla) # Evaluacija
```

```r
# <===============| Random Forest |===============>
library(randomForest) # install.packages("randomForest")
library(caret) # install.packages("caret")

# Izgradnja stabla
forest <- randomForest(podaci ~ ., data = train)

forest$ntree       # Broj stabala
forest$mtry        # Broj varijabli (prediktora)
importance(forest) # Važnost varijabli

varImpPlot(forest) # grafički prikaz

# <===============| Predikcija i evaluacija |===============>
predikcije <- predict(forest, test)  # Predikcija

confusionMatrix(predikcije, test$varijabla) # Evaluacija
```