2. laboratorijska vježba

Multivarijatna analiza podataka

ak. god. 2021/2022

1. Uvod i upute za predaju

Cilj ove laboratorijske vježbe je primijeniti osnovne koncepte multivarijatne analize podataka, istražiti podatke te ispitati hipoteze. Preduvjet za rješavanje vježbe je osnovno znanje programskog jezika *R* i rad s *R Markdown* dokumentima. Sama vježba je koncipirana kao projekt u kojem istražujete i eksperimentirate koristeći dane podatke - ne postoji nužno samo jedan točan način rješavanja svakog podzadatka.

Rješavanje vježbe svodi se na čitanje uputa u tekstu ovog dokumenta, nadopunjavanje blokova kôda (možete dodavati i dodatne blokove kôda ukoliko je potrebno) i ispisivanje rezultata (u vidu ispisa iz funkcija, tablica i grafova). Vježbu radite samostalno, a svoje rješenje branite na terminima koji su vam dodijeljeni u kalendaru. Pritom morate razumjeti teorijske osnove u okviru onoga što je obrađeno na predavanjima i morate pokazati da razumijete sav kôd koji ste napisali.

Vaše rješenje potrebno je predati u sustav *Moodle* u obliku dvije datoteke:

1. Ovaj .Rmd dokument s Vašim rješenjem (naziva IME_PREZIME_JMBAG.rmd),

2. PDF ili HTML dokument kao izvještaj generiran iz vašeg .Rmd rješenja (također naziva IME_PREZIME_JMBAG).

Rok za predaju je **15. svibnja 2022. u 23:59h**. **Jedan od uvjeta za prolaz predmeta je minimalno ostvarenih 50% bodova na svim laboratorijskim vježbama. Nadoknade laboratorijskih vježbi neće biti organizirane.** Za sva dodatna pitanja svakako se javite na email adresu predmeta: *map@fer.hr*.

2. Podatkovni skup

U laboratorijskoj vježbi razmatra se dinamika cijena vrijednosnica na financijskim tržištima. Dane su povijesne tjedne cijene ETF-ova (eng. exchange traded fund) koji prate određene dioničke, obvezničke ili druge indekse. Konkretno, radi se o sljedećim fondovima:

- `AGG` (iShares Core U.S. Aggregate Bond ETF) - obveznice s američkog tržišta,
- `IEF` (iShares 7-10 Year Treasury Bond ETF) - srednjeročne državne obveznice,
- `LQD` (iShares iBoxx $ Investment Grade Corporate Bond ETF) - korporativne obveznice,
- `SHY` (iShares 1-3 Year Treasury Bond ETF) - kratkoročne državne obveznice,
- `TIP` (iShares TIPS Bond ETF) - državne obveznice zaštićene od inflacije,
- `TLT` (iShares 20+ Year Treasury Bond ETF) - dugoročne državne obveznice,
- `DBC` (Invesco DB Commodity Index Tracking Fund) - sirovine i roba,
- `GLD` (SPDR Gold Trust) - zlato,
- `USO` (United States Oil Fund) - nafta,
- `IJH` (iShares Core S&P Mid-Cap ETF) - dionice tvrtki s američkog tržišta,
- `IWM` (iShares Russell 2000 ETF) - dionice američkih tvrtki s malim kapitalom,
- `SPY` (SPDR S&P 500 ETF Trust) - dionice tvrtki s američkog tržišta,
- `VTV` (Vanguard Value ETF) - dionice tvrtki s američkog tržišta,
- `XLB` (Materials Select Sector SPDR Fund) - dionice tvrtki za materijale,
- `XLE` (Energy Select Sector SPDR Fund) - dionice tvrtki energetskog sektora,
- `XLF` (Financial Select Sector SPDR Fund) - dionice tvrtki financijskog sektora,

- XLI (Industrial Select Sector SPDR Fund) - dionice tvrtki industrijskog sektora,
- XLK (Technology Select Sector SPDR Fund) - dionice tvrtki iz tehnološkog sektora,
- XLP (Consumer Staples Select Sector SPDR Fund) - dionice tvrki za necikličku potrošačku robu,
- XLU (Utilities Select Sector SPDR Fund) - dionice tvrtki komunalnih djelatnosti,
- XLV (Health Care Select Sector SPDR Fund) - dionice tvrtki iz zdravstvenog sektora,
- XLY (Consumer Discretionary Select Sector SPDR Fund) - dionice tvrtki za cikličku potrošačku robu,
- IYR (iShares U.S. Real Estate ETF) - dionice tvrtki iz područja nekretnina,
- VNQ (Vanguard Real Estate Index Fund) - dionice tvrtki iz područja nekretnina.

Pri modeliranju zajedničkog kretanja i rizika vrijednosnica, najčešće se koriste povrati: $R(t) = \frac{S(t)-S(t-1)}{S(t-1)}$, gdje je $S(t)$ cijena vrijednosnice u tjednu t.

2.1. Učitavanje podataka i korelacijska analiza

Podaci se nalaze u datoteci "ETFprices.csv". Učitajte ih, provjerite ispravnost, izračunajte tjedne povrate te vizualizirajte matricu korelacije povrata - razmislite o grupama i korelacijskim strukturama koje u njoj vidite. U ostatku laboratorijske vježbe također koristite povrate, a ne cijene.

```
df <- read.csv(file = 'ETFprices.csv')

print(nrow(df))
```

```
## [1] 667
```

```
print(sapply(df, typeof))
```

```
##        Time         AGG         IEF         LQD         SHY         TIP 
## "character"    "double"    "double"    "double"    "double"    "double" 
##         TLT         DBC         GLD         USO         IJH         IWM 
##    "double"    "double"    "double"    "double"    "double"    "double" 
##         SPY         VTV         XLB         XLE         XLF         XLI 
##    "double"    "double"    "double"    "double"    "double"    "double" 
##         XLK         XLP         XLU         XLV         XLY         IYR 
##    "double"    "double"    "double"    "double"    "double"    "double" 
##         VNQ 
##    "double"
```

```
df$Time <- as.Date(df$Time , format = "%d-%b-%Y 00:00:00")

print(sapply(df, typeof))
```

```
##     Time      AGG      IEF      LQD      SHY      TIP      TLT      DBC 
## "double" "double" "double" "double" "double" "double" "double" "double" 
##      GLD      USO      IJH      IWM      SPY      VTV      XLB      XLE 
## "double" "double" "double" "double" "double" "double" "double" "double" 
##      XLF      XLI      XLK      XLP      XLU      XLV      XLY      IYR 
## "double" "double" "double" "double" "double" "double" "double" "double" 
##      VNQ 
## "double"
```

```
for (i in colnames(df)) {
  num_na = sum(is.na(df$i))
  num_inf = sum(is.infinite(df$i))

  if (num_na != 0){
    print(i)
    print(num_na)
```

```
  }

  if (num_inf != 0) {
    print(i)
    print(num_inf)
  }
}
```

```
# check if its already in order
print(all(df == df[order(df$Time),]))
```

Tjedni povrati

```
## [1] TRUE
```

```
head(df, 5)
```

```
##         Time      AGG      IEF      LQD      SHY      TIP      TLT      DBC
## 1 2006-04-09 65.09827 56.76505 60.70945 65.21442 68.62030 55.09219 23.18837
## 2 2006-04-16 65.11148 57.07544 60.98924 65.30439 69.29211 55.32708 24.19736
## 3 2006-04-23 65.32363 57.01193 61.00089 65.36169 69.18821 54.90950 23.71601
## 4 2006-04-30 65.17391 56.81649 60.83233 65.35593 69.04548 54.75883 23.79006
## 5 2006-05-07 64.93439 56.54039 60.71520 65.35593 69.11513 54.13662 24.85459
##     GLD   USO      IJH      IWM       SPY      VTV      XLB      XLE      XLF
## 1 59.50 68.82 65.63398 62.55068  99.18915 42.67178 24.63047 42.91956 15.42653
## 2 63.20 72.81 67.69395 64.32922 101.06951 43.65933 25.71436 45.94183 15.73049
## 3 65.09 69.62 67.35899 63.92653 101.31612 43.97420 25.04160 43.94236 16.12946
## 4 67.99 68.00 68.38899 65.03390 102.12528 44.59678 25.89375 45.26510 16.22445
## 5 71.12 69.11 66.19505 61.66143  99.59763 43.43749 25.25837 43.63475 15.76849
##        XLI      XLK      XLP      XLU      XLV      XLY      IYR      VNQ
## 1 25.99025 18.21236 16.62211 18.97177 24.89937 27.91007 40.64932 36.12251
## 2 26.78085 18.17940 16.73646 19.60541 25.01174 27.96821 42.06361 37.39742
## 3 26.50452 18.00634 17.02945 19.64933 24.87528 28.31710 41.89862 37.17645
## 4 27.38724 18.06403 17.19381 20.18887 24.70672 28.74904 42.19325 37.41442
## 5 26.81155 17.28938 16.91511 19.61169 24.39367 28.31710 40.82611 36.26982
```

```
n = nrow(df)
p = ncol(df)

ETF_returns = ((data.matrix(df[2:n,2:p]) - data.matrix(df[1:(n-1),2:p]))/data.matrix(df[1:(n-1),2:p]))

ETF_returns = cbind(df$Time[2:n],as.data.frame(ETF_returns))
head(ETF_returns, 5)
```

```
##   df$Time[2:n]           AGG          IEF           LQD           SHY
## 2   2006-04-16  0.0002029854  0.005468119  0.0046086562  1.379541e-03
## 3   2006-04-23  0.0032582579 -0.001112755  0.0001910173  8.773836e-04
## 4   2006-04-30 -0.0022919577 -0.003428160 -0.0027633038 -8.812502e-05
## 5   2006-05-07 -0.0036751668 -0.004859505 -0.0019255222  0.000000e+00
## 6   2006-05-14  0.0066600305  0.009517515  0.0072324729  1.632430e-03
##            TIP          TLT         DBC        GLD         USO          IJH
## 2  0.009790208  0.004263435  0.04351289  0.06218489  0.05797730  0.031385785
## 3 -0.001499406 -0.007547480 -0.01989275  0.02990498 -0.04381260 -0.004948152
## 4 -0.002062938 -0.002743879  0.00312249  0.04455373 -0.02326922  0.015291069
## 5  0.001008842 -0.011362752  0.04474696  0.04603626  0.01632354 -0.032080195
```

```
## 6  0.005337094  0.021536346 -0.05325876 -0.07789652 -0.05122270 -0.031752524
##             IWM          SPY          VTV         XLB         XLE          XLF
## 2  0.028433553  0.018957346  0.023142976  0.04400611  0.07041725  0.019704242
## 3 -0.006259722  0.002439954  0.007212043 -0.02616282 -0.04352184  0.025362649
## 4  0.017322447  0.007986479  0.014157800  0.03402970  0.03010175  0.005889347
## 5 -0.051857111 -0.024750474 -0.025994947 -0.02453823 -0.03601792 -0.028103259
## 6 -0.021769201 -0.016558939 -0.017792487 -0.04557559 -0.04793810 -0.018072367
##            XLI          XLK          XLP          XLU          XLV          XLY
## 2  0.03041930 -0.001809924  0.006879332  0.033399474  0.004513127  0.002083227
## 3 -0.01031827 -0.009519512  0.017506275  0.002239841 -0.005455758  0.012474556
## 4  0.03330440  0.003203872  0.009651340  0.027458601 -0.006776284  0.015253609
## 5 -0.02102023 -0.042883509 -0.016209326 -0.028589021 -0.012670681 -0.015024433
## 6 -0.02547943 -0.018112389 -0.002112136 -0.005438440 -0.003619504 -0.012027361
##             IYR          VNQ
## 2  0.034792391  0.035294060
## 3 -0.003922369 -0.005908803
## 4  0.007032022  0.006401258
## 5 -0.032401913 -0.030592375
## 6 -0.014578809 -0.014372443
```

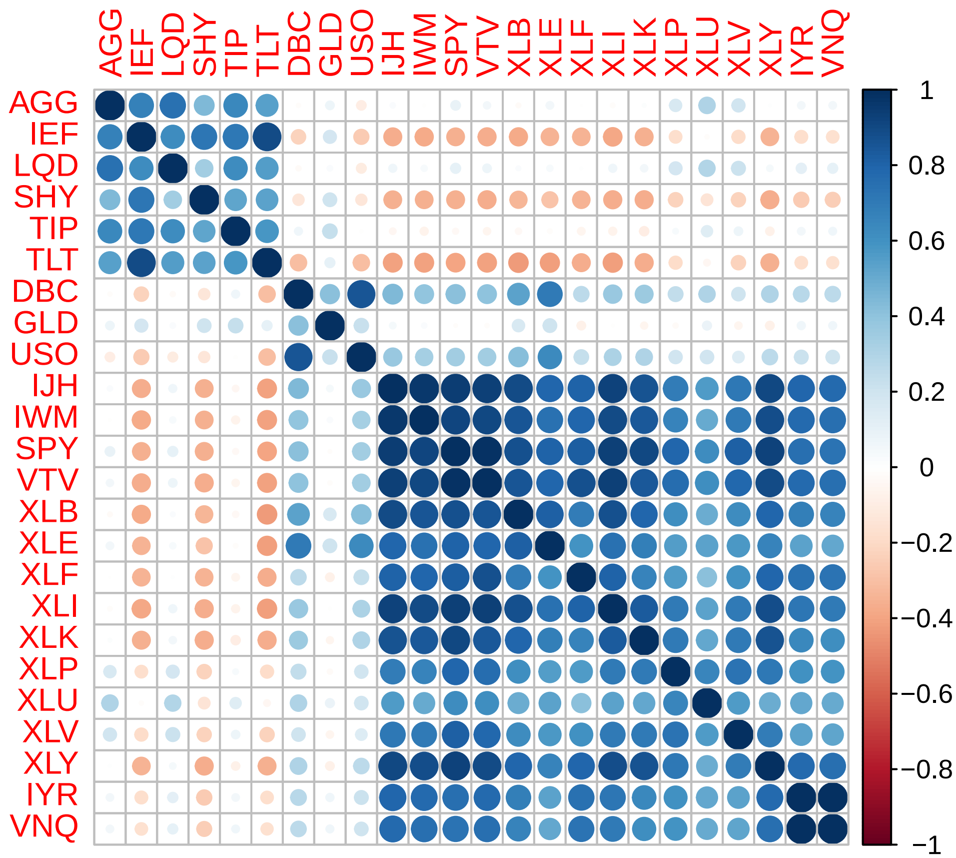
```
ETF_returns = ETF_returns[,2:p]
```

```
library(corrplot)
```

Korelacijska matrica

```
## corrplot 0.92 loaded
```

```
corr_mat = cor(ETF_returns, method='pearson')
corrplot(
  corr_mat,
  method='circle'
)
```



3. Analiza glavnih komponenti

Cilj ovog zadatka je analizirati kretanje danih ETF-ova i izračunati glavne komponente koje objašnjavaju njihovu dinamiku.

3.1. Glavne komponente

Izračunajte glavne komponente matrice korelacije i izračunajte koliki udio varijance objašnjavaju. Odredite broj glavnih komponenti koje ćete zadržati u analizi. Grafički prikažite i usporedite koeficijente prvih nekoliko komponenti.

```
pca <- prcomp(ETF_returns, center=TRUE, scale=TRUE)

print(round(pca$rotation,digits=3))
```

```
##        PC1    PC2    PC3    PC4    PC5    PC6    PC7    PC8    PC9   PC10
## AGG  0.002 -0.424  0.010  0.231 -0.071  0.034 -0.173  0.341 -0.164  0.272
## IEF  0.123 -0.422 -0.014 -0.099 -0.042 -0.030  0.100 -0.177 -0.108  0.149
## LQD -0.009 -0.411  0.051  0.217 -0.108  0.150 -0.419  0.012 -0.135 -0.151
## SHY  0.121 -0.297 -0.115 -0.122 -0.150 -0.461  0.629  0.221 -0.248 -0.107
## TIP  0.025 -0.410 -0.117 -0.070 -0.185 -0.077 -0.092  0.009  0.593 -0.495
## TLT  0.131 -0.368  0.059 -0.129  0.014  0.184  0.000 -0.479 -0.096  0.369
## DBC -0.139  0.015 -0.559  0.079 -0.093  0.176  0.007 -0.045  0.044 -0.002
## GLD -0.009 -0.102 -0.445 -0.386  0.600 -0.302 -0.272 -0.001  0.049  0.173
## USO -0.120  0.058 -0.525  0.136 -0.310  0.265  0.240 -0.161  0.101  0.158
## IJH -0.277 -0.024  0.008 -0.063 -0.067 -0.088 -0.060 -0.023 -0.140 -0.074
## IWM -0.269 -0.009  0.031 -0.087 -0.088 -0.129 -0.067 -0.044 -0.131 -0.058
```

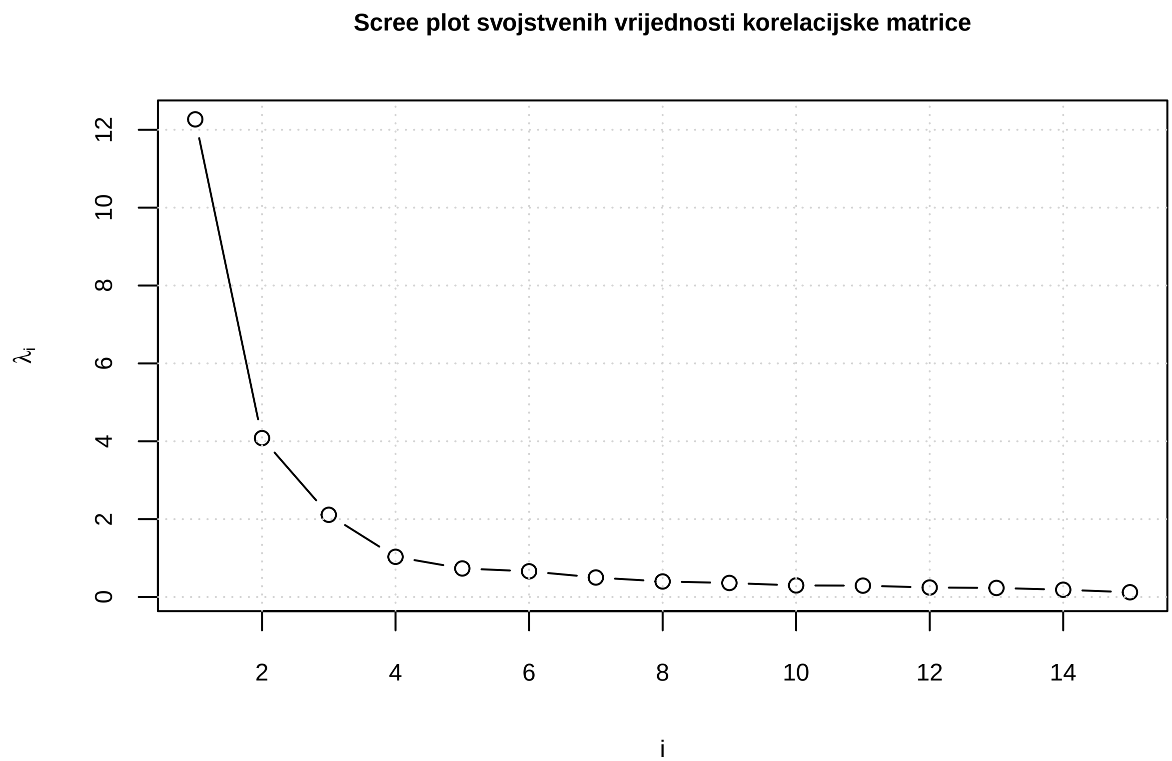
```
## SPY -0.280 -0.042  0.046  0.090 -0.010 -0.102  0.001 -0.014 -0.010  0.082
## VTV -0.278 -0.029  0.052  0.034 -0.021 -0.056  0.031  0.119  0.055  0.172
## XLB -0.258  0.000 -0.112 -0.048 -0.012 -0.175 -0.159 -0.031 -0.188 -0.211
## XLE -0.236 -0.007 -0.276  0.161 -0.036 -0.036 -0.065  0.141 -0.143  0.067
## XLF -0.239  0.000  0.120 -0.175 -0.217 -0.009 -0.029  0.380  0.211  0.461
## XLI -0.268 -0.006  0.049 -0.003 -0.057 -0.148 -0.081 -0.027 -0.086 -0.083
## XLK -0.252 -0.006  0.060  0.123 -0.022 -0.157 -0.020 -0.405 -0.208 -0.133
## XLP -0.219 -0.103  0.116  0.200  0.311  0.042  0.350 -0.263  0.220  0.076
## XLU -0.176 -0.170  0.002  0.242  0.511  0.391  0.245  0.284 -0.179 -0.235
## XLV -0.218 -0.103  0.139  0.268  0.152 -0.247  0.055 -0.038  0.477  0.160
## XLY -0.263 -0.014  0.118 -0.065 -0.113 -0.074  0.024 -0.209 -0.081  0.016
## IYR -0.231 -0.085  0.097 -0.445 -0.013  0.300  0.079  0.043  0.021 -0.090
## VNQ -0.226 -0.086  0.098 -0.460 -0.016  0.320  0.098  0.053  0.028 -0.093
##       PC11   PC12   PC13   PC14   PC15   PC16   PC17   PC18   PC19   PC20
## AGG -0.188  0.627 -0.079  0.033 -0.081  0.141 -0.136  0.061 -0.133 -0.084
## IEF  0.185 -0.048 -0.039  0.077 -0.045  0.014 -0.006 -0.013  0.110  0.811
## LQD -0.423 -0.512  0.183 -0.143  0.083 -0.088  0.023  0.023  0.129 -0.043
## SHY -0.162 -0.167  0.001 -0.080 -0.029 -0.060  0.043 -0.029 -0.017 -0.220
## TIP  0.304  0.192  0.113  0.003  0.115 -0.042  0.014  0.011 -0.001 -0.101
## TLT  0.324 -0.140 -0.102  0.126 -0.058  0.036  0.088 -0.041 -0.115 -0.490
## DBC -0.100  0.025  0.016 -0.136 -0.420  0.119  0.442 -0.429 -0.123  0.052
## GLD -0.099  0.007  0.086 -0.170  0.128 -0.026 -0.130  0.058  0.019 -0.030
## USO -0.062 -0.074 -0.034 -0.112  0.192  0.097 -0.432  0.380  0.047 -0.006
## IJH  0.059 -0.002 -0.081  0.018  0.173  0.317  0.126  0.029  0.052  0.030
## IWM  0.024 -0.006 -0.148 -0.047  0.408  0.553  0.326  0.094 -0.009 -0.003
## SPY  0.032  0.053  0.008 -0.052 -0.007 -0.150 -0.004 -0.011 -0.033 -0.016
## VTV  0.101 -0.099  0.089  0.032  0.013 -0.128 -0.010 -0.012 -0.065  0.044
## XLB  0.108 -0.047  0.011  0.414 -0.572  0.102 -0.110  0.349  0.319 -0.093
## XLE  0.140 -0.025 -0.139  0.496  0.399 -0.479  0.123 -0.197  0.058 -0.020
## XLF  0.243 -0.154  0.312 -0.240 -0.113 -0.067  0.196  0.162  0.143 -0.018
## XLI  0.120 -0.188  0.232  0.067 -0.079  0.108 -0.385 -0.170 -0.721  0.085
## XLK  0.032  0.259 -0.056 -0.456 -0.052 -0.436  0.195  0.301 -0.120  0.036
## XLP -0.311  0.128  0.526  0.305  0.068  0.093  0.169  0.080  0.033  0.013
## XLU  0.388 -0.104 -0.059 -0.233 -0.013  0.075 -0.072  0.008  0.071 -0.030
## XLV -0.168 -0.224 -0.608 -0.028 -0.158  0.044 -0.074 -0.051  0.003  0.020
## XLY  0.008  0.168  0.113 -0.203  0.034  0.043 -0.398 -0.579  0.487 -0.090
## IYR -0.224  0.040 -0.149  0.058 -0.029 -0.127 -0.032  0.021 -0.064  0.028
## VNQ -0.240  0.052 -0.178  0.069 -0.026 -0.126 -0.004  0.013 -0.082  0.009
##       PC21   PC22   PC23   PC24
## AGG  0.016 -0.017 -0.049 -0.045
## IEF -0.010 -0.039  0.031  0.017
## LQD -0.009  0.003  0.011 -0.003
## SHY  0.010  0.003 -0.008  0.002
## TIP -0.024  0.022  0.000  0.002
## TLT -0.003  0.030 -0.003  0.008
## DBC -0.026  0.010 -0.015  0.002
## GLD -0.002  0.013  0.006 -0.008
## USO -0.002  0.002  0.013  0.003
## IJH  0.525  0.661  0.033 -0.035
## IWM -0.322 -0.388 -0.035  0.014
## SPY -0.243  0.158  0.591  0.655
## VTV -0.644  0.489 -0.277 -0.290
## XLB -0.061 -0.101 -0.011 -0.022
## XLE  0.158 -0.175 -0.045 -0.046
```

```
## XLF  0.255 -0.209 -0.001 -0.020
## XLI  0.138 -0.168 -0.014 -0.033
## XLK  0.088 -0.081 -0.122 -0.145
## XLP  0.089 -0.050 -0.019 -0.025
## XLU  0.021 -0.061 -0.002 -0.003
## XLV  0.115 -0.082 -0.049 -0.050
## XLY  0.000 -0.105 -0.051 -0.073
## IYR  0.037  0.000 -0.534  0.487
## VNQ -0.032 -0.049  0.509 -0.462
```

```
summary(pca)
```

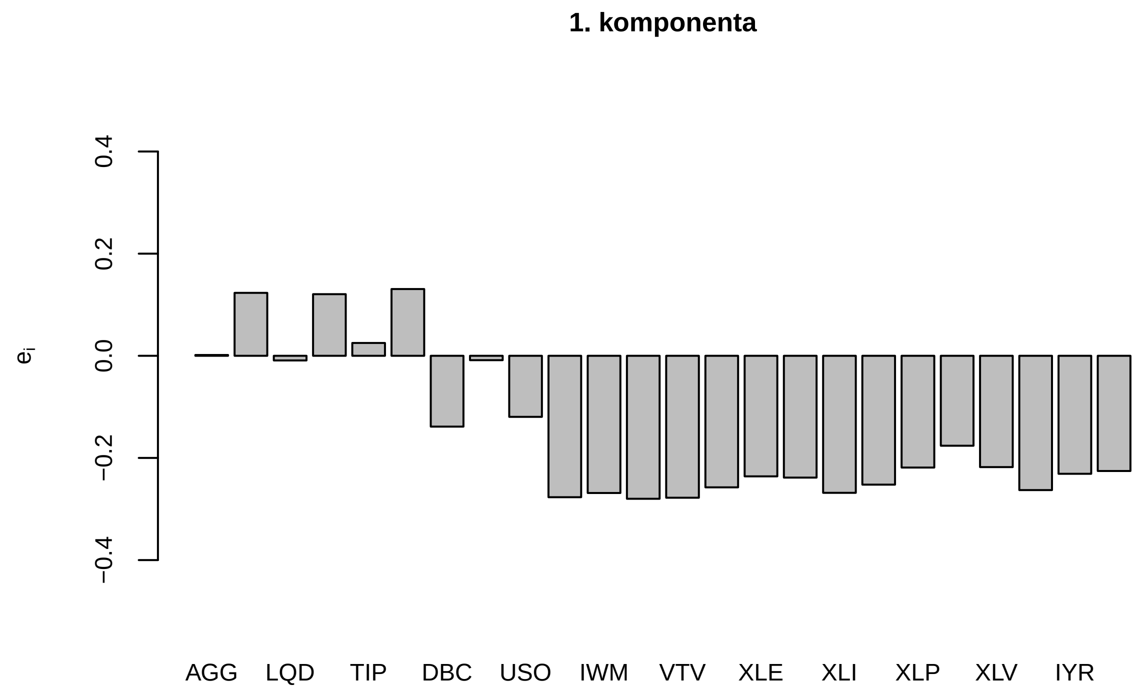
```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5    PC6     PC7
## Standard deviation     3.5026 2.0201 1.45277 1.01573 0.85603 0.8124 0.70740
## Proportion of Variance 0.5112 0.1700 0.08794 0.04299 0.03053 0.0275 0.02085
## Cumulative Proportion  0.5112 0.6812 0.76914 0.81213 0.84266 0.8702 0.89101
##                           PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     0.6313 0.59983 0.54444 0.53993 0.49353 0.48162 0.43129
## Proportion of Variance 0.0166 0.01499 0.01235 0.01215 0.01015 0.00966 0.00775
## Cumulative Proportion  0.9076 0.92260 0.93495 0.94710 0.95725 0.96691 0.97466
##                           PC15    PC16    PC17    PC18    PC19    PC20    PC21
## Standard deviation     0.34917 0.33995 0.31615 0.30848 0.27417 0.22747 0.14648
## Proportion of Variance 0.00508 0.00482 0.00416 0.00396 0.00313 0.00216 0.00089
## Cumulative Proportion  0.97974 0.98456 0.98872 0.99269 0.99582 0.99797 0.99887
##                           PC22    PC23    PC24
## Standard deviation     0.13519 0.07108 0.06184
## Proportion of Variance 0.00076 0.00021 0.00016
## Cumulative Proportion  0.99963 0.99984 1.00000
```

```
plot(pca$sdev[1:15]^2,
  type = "b",
  cex.lab=0.75,
  cex.main=0.75,
  cex.axis=0.75,
  xlab="i",
  ylab=expression(lambda["i"]),
  main='Scree plot svojstvenih vrijednosti korelacijske matrice'
)
grid()
```

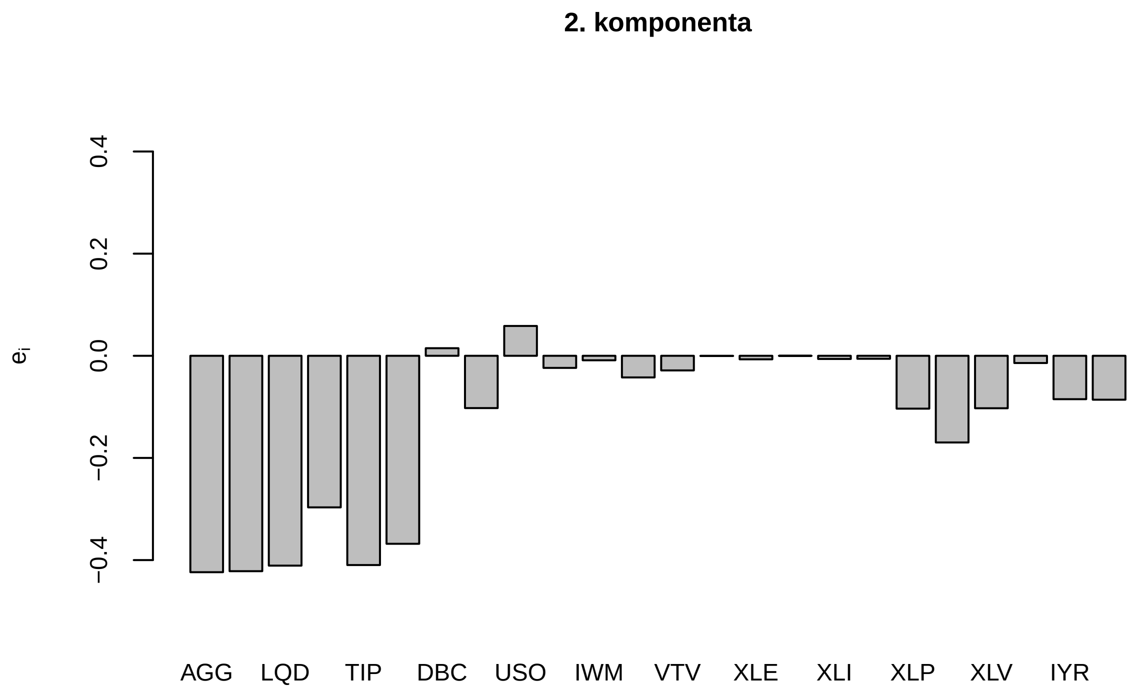


Otprilike se vidi pregib na 3. komponenti pa bih za broj komponenti odabrao 2.

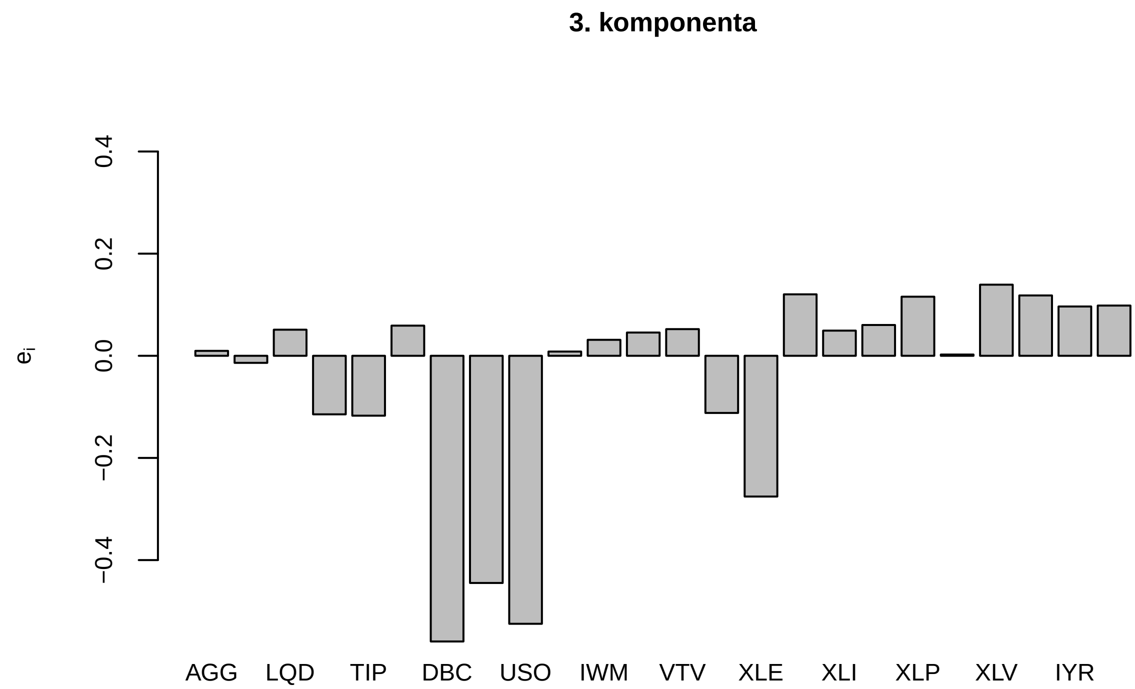
```
barplot(
  pca$rotation[,1],
  main="1. komponenta",
  ylab=expression("e"["i"]), cex.axis = 0.75,
  cex.names = 0.75,
  cex.main=0.85,
  cex.lab=0.75,
  ylim=c(-0.5, 0.5)
)
```



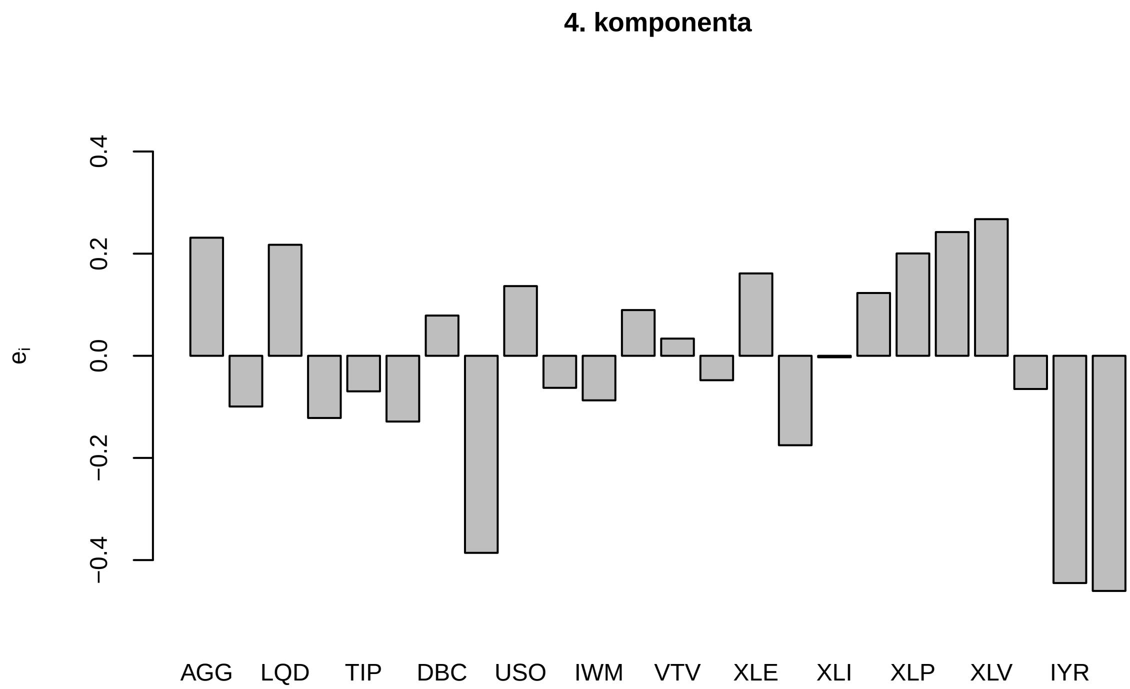

```
barplot(
  pca$rotation[,2],
  main="2. komponenta",
  ylab=expression("e"["i"]), cex.axis = 0.75,
  cex.names = 0.75,
  cex.main=0.85,
  cex.lab=0.75,
  ylim=c(-0.5, 0.5)
)
```




```
barplot(
  pca$rotation[,3],
  main="3. komponenta",
  ylab=expression("e"["i"]), cex.axis = 0.75,
  cex.names = 0.75,
  cex.main=0.85,
  cex.lab=0.75,
  ylim=c(-0.5, 0.5)
)
```




```
barplot(
  pca$rotation[,4],
  main="4. komponenta",
  ylab=expression("e"["i"]), cex.axis = 0.75,
  cex.names = 0.75,
  cex.main=0.85,
  cex.lab=0.75,
  ylim=c(-0.5, 0.5)
)
```



Prikažite graf raspršenja prve dvije glavne komponente i proučite možete li primijetiti neke grupe fondova.

```
library(devtools)
```

```
## Loading required package: usethis
```

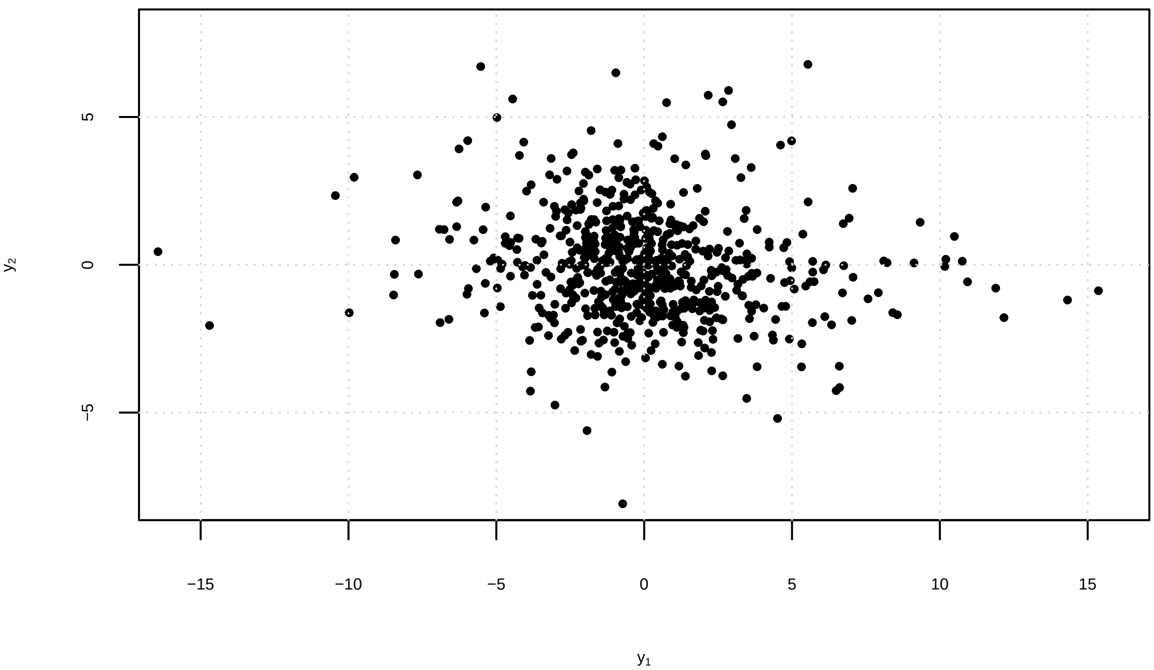
```
library(ggbiplot)
```

```
## Loading required package: ggplot2
## Loading required package: plyr
## Loading required package: scales
## Loading required package: grid
```

```
library(usethis)
library(ggplot2)
library(plyr)
library(scales)
library(grid)


Y = as.matrix(scale(ETF_returns))%*%pca$rotation

plot(Y[,1], Y[,2], pch = 20, cex=0.7, cex.lab=0.5, cex.axis=0.5, xlab=expression("y"["1"]), ylab=express
grid()
```



Ovdje se ne mogu primijetiti neke grupe fondova.

3.2. Svojstveni portfelji

U primjeni PCA i svojstvenoj dekompoziciji kovarijance u financijama, svojstveni vektori se često zovu i tzv. svojstveni portfelji. Općenito, portfelj je vektor $w = [w_1, ..., w_N]$ u kojem svaki element predstavlja težinu ili udio kapitala u određenoj vrijednosnici. Često je dobro pomnožiti njihove težine s predznakom njihove sume - na taj način zapravo samo "okrećemo" predznak svojstvenog vektora tako da mu je suma pozitivna (konačni PCA rastav je i dalje isti ako svojstveni vektor pomnožimo s -1). Također, dobro je i skalirati svojstvene portfelje sa sumom njihovih apsolutnih vrijednosti: $\tilde{w}_i = \frac{w_i}{\sum_j^N |w_j|}$. Na taj način se osigurava da visoke magnitude pojedinih elemenata ne uzrokuju velike razlike u volatilnostima svojstvenih portfelja. Ukoliko znamo povrate $R \in \mathbb{R}^{T \times N}$ (gdje je $R_i \in \mathbb{R}^T$ vektor povrata za vrijednosnicu i) za N vrijednosnica u nekom vremenskom periodu od T dana, povrate portfelja w u tom istom periodu možemo izračunati kao: $R_p = \sum R_i w_i = R \cdot w$. Izračunajte skalirane svojstvene portfelje $\tilde{w}$ koji proizlaze iz prve dvije glavne komponente. Za ta dva svojstvena portfelja izračunajte povijesne povrate kroz razmatrani period. Grafički prikažite vremensko kretanje njihovih vrijednosti tako da njihove povrate "vratite" natrag u cijene, s tim da početna cijena bude jednaka za oba portfelja, npr. $V_0 = 100$. Vrijednost portfelja u trenutku t možemo izračunati po formuli: $V_t = V_{t-1} \cdot (1 + R_t)$.

```
first = pca$rotation[1:nrow(pca$rotation), 1]
second = pca$rotation[1:nrow(pca$rotation), 2]

print('Before multiplying with -1:')
```

```
## [1] "Before multiplying with -1:"
```

```r
print(sum(first))
```

```
## [1] -3.563283
```

```r
print(sum(second))
```

```
## [1] -3.043826
```

```r
first = -first
second = -second

print('After multiplying:')
```

```
## [1] "After multiplying:"
```

```r
print(sum(first))
```

```
## [1] 3.563283
```

```r
print(sum(second))
```

```
## [1] 3.043826
```

```r
print('Before scaling')
```

```
## [1] "Before scaling"
```

```r
print(first[1:5])
```

```
##          AGG          IEF          LQD          SHY          TIP
## -0.001579180 -0.123201737  0.009073106 -0.120679855 -0.025089910
```

```r
print(second[1:5])
```

```
##       AGG       IEF       LQD       SHY       TIP
## 0.4237079 0.4217203 0.4108685 0.2967833 0.4097050
```

```r
first = first/(sum(abs(first)))
second = second/(sum(abs(second)))

print('After scaling')
```

```
## [1] "After scaling"
```

```r
print(first[1:5])
```

```
##           AGG           IEF           LQD           SHY           TIP
## -0.0003617219 -0.0282201925  0.0020782564 -0.0276425384 -0.0057470139
```

```r
print(second[1:5])
```

```
##        AGG        IEF        LQD        SHY        TIP
## 0.13280551 0.13218251 0.12878118 0.09302272 0.12841648
```

```r
r1 <- as.matrix(ETF_returns) %*% first
r2 <- as.matrix(ETF_returns) %*% second

head(r1, 5)
```

```
##           [,1]
## 2  0.024107611
## 3 -0.004377362
```

```
## 4  0.012250805
## 5 -0.021539673
## 6 -0.021397245
```

```
head(r2, 5)
```

```
##           [,1]
## 2  0.008968167
## 3  0.001395104
## 4  0.002655574
## 5 -0.006941796
## 6  0.002911501
```

```
v1 <- c(100)
v2 <- c(100)


for (i in 1:length(r1)){
  v1 <- append(v1, v1[i] * (1 + r1[i]))
  v2 <- append(v2, v2[i] * (1 + r2[i]))
}

print(v1[1:5])
```

```
## [1] 100.0000 102.4108 101.9625 103.2116 100.9885
```

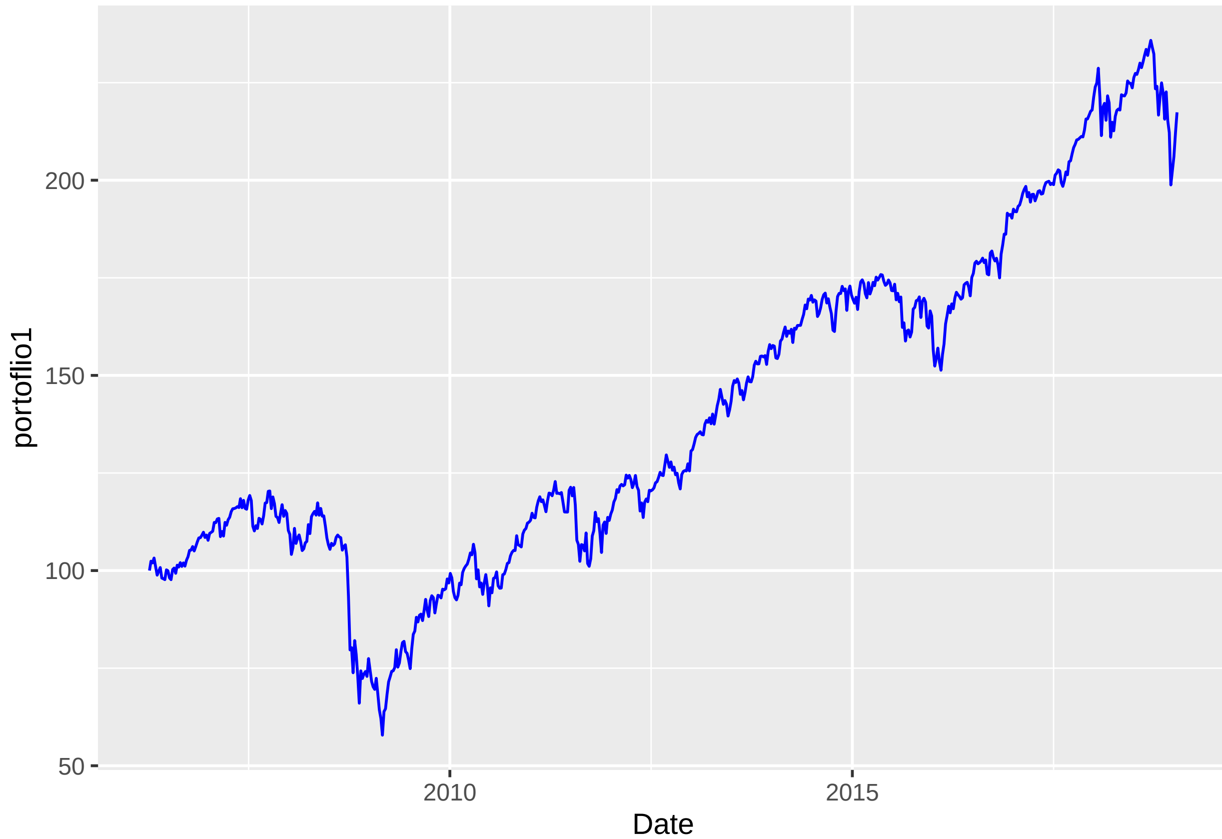
```
print(v2[1:5])
```

```
## [1] 100.0000 100.8968 101.0376 101.3059 100.6026
```

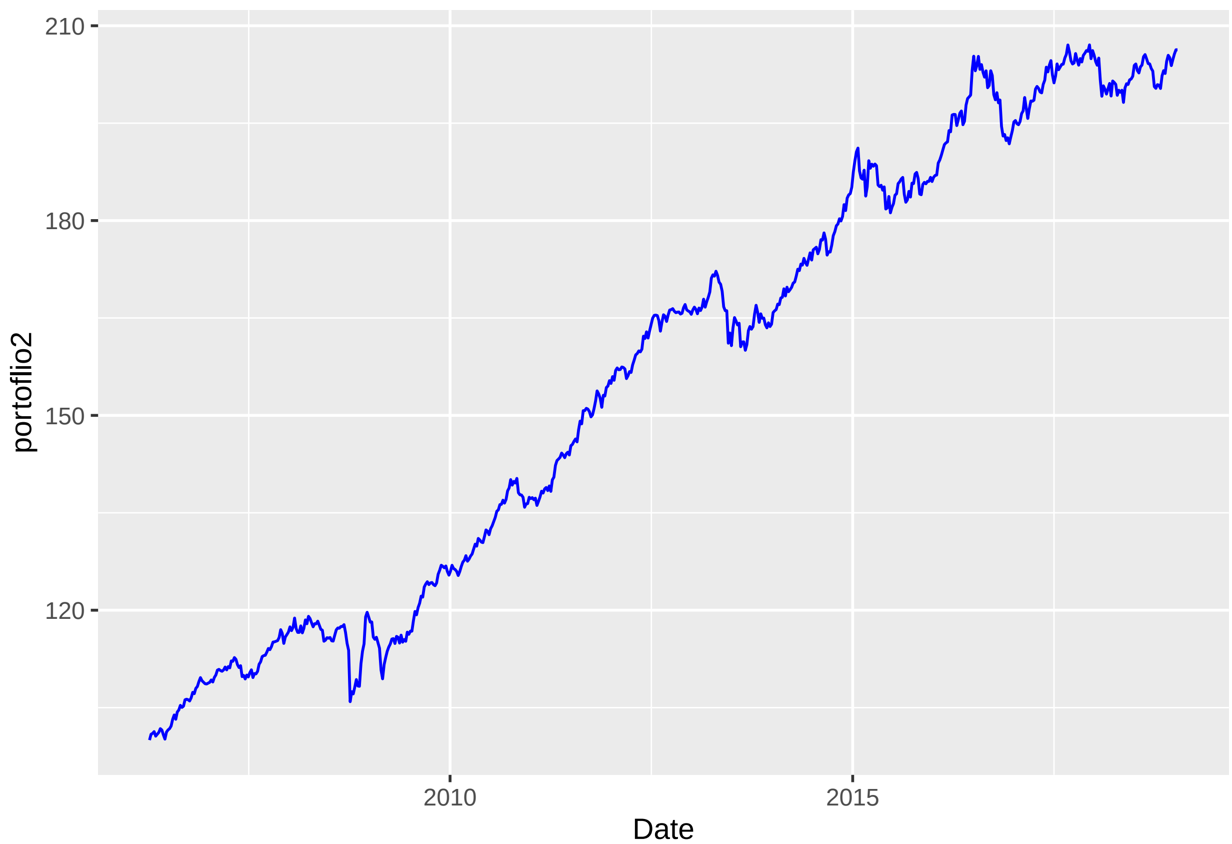
```
c_ret1 <- as.data.frame(cbind(df$Time, v1))
c_ret2 <- as.data.frame(cbind(df$Time, v2))
dates <- as.Date(c_ret1[,1],origin = "1970-01-01")
c_ret1[,1] <- dates
c_ret2[,1] <- dates
colnames(c_ret1) <- c('Date', 'portoflio1')
colnames(c_ret2) <- c('Date', 'portoflio2')

library("reshape2")
library("ggplot2")

ggplot(c_ret1, aes(x = Date, y = portoflio1)) + geom_line(color='blue')
```




```r
ggplot(c_ret2, aes(x = Date, y = portoflio2)) + geom_line(color='blue')
```



4. Faktorska analiza

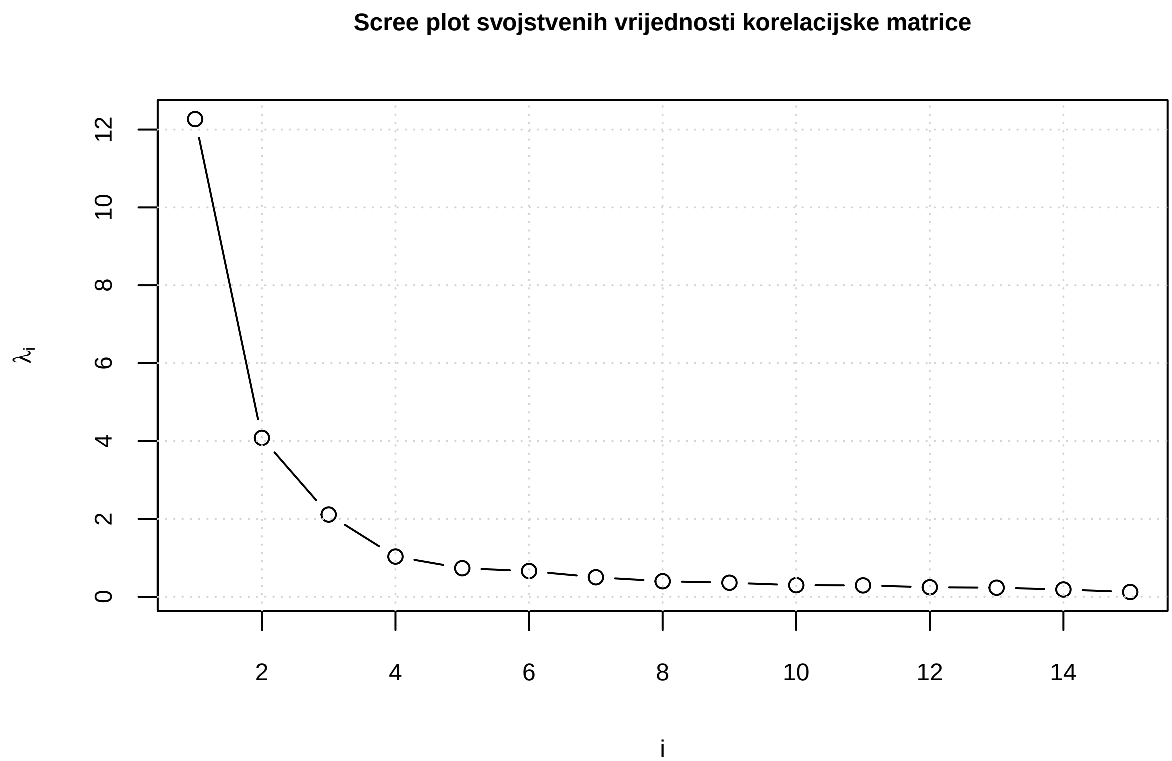
4.1. Metode procjena koeficijenata modela

Na danim podacima odredite broj faktora te procijenite faktorski model pomoću metode glavnih komponenti i metode najveće izglednosti. Usporedite procjene ove dvije metode. Koja Vam se čini bolja? Što možete zaključiti iz vrijednosti faktora? Pronađite procjenu vrijednosti faktora koja daje najbolju interpretabilnost.

```
ev = eigen(corr_mat)

lambda = ev$values
e = ev$vectors

plot(lambda[1:15],
  type = "b",
  cex.lab=0.75,
  cex.main=0.75,
  cex.axis=0.75,
  xlab="i",
  ylab=expression(lambda["i"]),
  main='Scree plot svojstvenih vrijednosti korelacijske matrice'
)
grid()
```



Teško je procijeniti pošto ne postoji očiti pregib, ovdje bih se odlučio za 2 ili 3 komponente. Daljnju analizu ću nastaviti s 2 komponente.

```
# procjena koristenjem metode glavnih komponenata
L = cbind(sqrt(lambda[1]) * e[,1], sqrt(lambda[2])*e[,2])
h = rowSums(L^2)
psi = 1-h
print(L)
```

```
##               [,1]          [,2]
##  [1,] -0.005531183 -0.8559314269
##  [2,] -0.431522255 -0.8519162491
##  [3,]  0.031779155 -0.8299946576
##  [4,] -0.422689195 -0.5995314187
##  [5,] -0.087879074 -0.8276441361
##  [6,] -0.457689589 -0.7436582124
##  [7,]  0.485698786  0.0301004272
##  [8,]  0.029982575 -0.2068726109
##  [9,]  0.418825918  0.1179865619
## [10,]  0.969807725 -0.0478616829
## [11,]  0.940870002 -0.0177036971
## [12,]  0.980678431 -0.0855238359
## [13,]  0.973462337 -0.0577645923
## [14,]  0.902105962 -0.0008876998
## [15,]  0.826887806 -0.0138717603
## [16,]  0.835386919 -0.0001495148
## [17,]  0.939862497 -0.0125122788
```

```
## [18,]  0.883757522 -0.0118619536
## [19,]  0.766541827 -0.2090250761
## [20,]  0.616949639 -0.3428207058
## [21,]  0.763266785 -0.2074221818
## [22,]  0.920928763 -0.0286089746
## [23,]  0.809191381 -0.1716330762
## [24,]  0.789963020 -0.1737168153
```

```
residual = corr_mat - L %*% t(L) - diag(psi)
print(round(residual, 3))
```

```
##        AGG    IEF    LQD    SHY    TIP    TLT    DBC    GLD    USO    IJH
## AGG  0.000 -0.057  0.033 -0.070 -0.068 -0.089  0.014 -0.101  0.007 -0.010
## IEF -0.057  0.000 -0.069  0.033 -0.024  0.062  0.006  0.022  0.027  0.016
## LQD  0.033 -0.069  0.000 -0.144 -0.064 -0.051 -0.015 -0.144 -0.018 -0.006
## SHY -0.070  0.033 -0.144  0.000 -0.006 -0.106  0.086  0.093  0.114  0.028
## TIP -0.068 -0.024 -0.064 -0.006  0.000 -0.071  0.132  0.065  0.128  0.004
## TLT -0.089  0.062 -0.051 -0.106 -0.071  0.000 -0.058 -0.035 -0.022  0.007
## DBC  0.014  0.006 -0.015  0.086  0.132 -0.058  0.000  0.410  0.648 -0.026
## GLD -0.101  0.022 -0.144  0.093  0.065 -0.035  0.410  0.000  0.241  0.004
## USO  0.007  0.027 -0.018  0.114  0.128 -0.022  0.648  0.241  0.000 -0.030
## IJH -0.010  0.016 -0.006  0.028  0.004  0.007 -0.026  0.004 -0.030  0.000
## IWM -0.014  0.016 -0.007  0.036  0.002  0.012 -0.057 -0.005 -0.055  0.056
## SPY  0.029 -0.004  0.005  0.005 -0.019 -0.007 -0.058 -0.057 -0.052 -0.006
## VTV  0.009  0.002 -0.003  0.011 -0.016  0.000 -0.071 -0.057 -0.055 -0.011
## XLB -0.016  0.018 -0.009  0.050  0.040 -0.017  0.100  0.140  0.052  0.025
## XLE  0.052  0.002  0.000  0.054  0.042 -0.049  0.300  0.178  0.291 -0.007
## XLF  0.004  0.014 -0.022  0.010  0.014  0.021 -0.142 -0.102 -0.113 -0.002
## XLI -0.024  0.007  0.026  0.022  0.006  0.003 -0.076 -0.035 -0.078  0.013
## XLK  0.006  0.012  0.016 -0.001 -0.023  0.026 -0.068 -0.079 -0.062  0.007
## XLP -0.011 -0.022 -0.015 -0.031 -0.075  0.008 -0.122 -0.091 -0.102 -0.056
## XLU  0.013 -0.041 -0.010 -0.092 -0.094 -0.017  0.012 -0.005 -0.018 -0.051
## XLV  0.017 -0.037  0.018 -0.026 -0.030 -0.033 -0.164 -0.121 -0.150 -0.031
## XLY -0.013  0.024 -0.012  0.006 -0.016  0.042 -0.137 -0.112 -0.114  0.015
## IYR -0.085  0.023 -0.057 -0.019 -0.013  0.070 -0.112  0.008 -0.108  0.005
## VNQ -0.086  0.023 -0.063 -0.017 -0.013  0.072 -0.111  0.004 -0.108  0.004
##        IWM    SPY    VTV    XLB    XLE    XLF    XLI    XLK    XLP    XLU
## AGG -0.014  0.029  0.009 -0.016  0.052  0.004 -0.024  0.006 -0.011  0.013
## IEF  0.016 -0.004  0.002  0.018  0.002  0.014  0.007  0.012 -0.022 -0.041
## LQD -0.007  0.005 -0.003 -0.009  0.000 -0.022  0.026  0.016 -0.015 -0.010
## SHY  0.036  0.005  0.011  0.050  0.054  0.010  0.022 -0.001 -0.031 -0.092
## TIP  0.002 -0.019 -0.016  0.040  0.042  0.014  0.006 -0.023 -0.075 -0.094
## TLT  0.012 -0.007  0.000 -0.017 -0.049  0.021  0.003  0.026  0.008 -0.017
## DBC -0.057 -0.058 -0.071  0.100  0.300 -0.142 -0.076 -0.068 -0.122  0.012
## GLD -0.005 -0.057 -0.057  0.140  0.178 -0.102 -0.035 -0.079 -0.091 -0.005
## USO -0.055 -0.052 -0.055  0.052  0.291 -0.113 -0.078 -0.062 -0.102 -0.018
## IJH  0.056 -0.006 -0.011  0.025 -0.007 -0.002  0.013  0.007 -0.056 -0.051
## IWM  0.000 -0.006 -0.012  0.006 -0.029  0.006  0.011  0.012 -0.061 -0.079
## SPY -0.006  0.000  0.020 -0.014 -0.005  0.008  0.011  0.042  0.021 -0.011
## VTV -0.012  0.020  0.000 -0.023 -0.008  0.066  0.016 -0.018  0.011 -0.003
## XLB  0.006 -0.014 -0.023  0.000  0.073 -0.057  0.023 -0.007 -0.078 -0.058
## XLE -0.029 -0.005 -0.008  0.073  0.000 -0.094 -0.029 -0.046 -0.082  0.020
## XLF  0.006  0.008  0.066 -0.057 -0.094  0.000  0.015 -0.077 -0.073 -0.096
## XLI  0.011  0.011  0.016  0.023 -0.029  0.015  0.000  0.007 -0.019 -0.049
## XLK  0.012  0.042 -0.018 -0.007 -0.046 -0.077  0.007  0.000  0.023 -0.032
```

```
## XLP -0.061  0.021  0.011 -0.078 -0.082 -0.073 -0.019  0.023  0.000  0.113
## XLU -0.079 -0.011 -0.003 -0.058  0.020 -0.096 -0.049 -0.032  0.113  0.000
## XLV -0.018  0.049  0.033 -0.061 -0.058 -0.030 -0.011  0.029  0.102  0.019
## XLY  0.018  0.016 -0.001 -0.031 -0.098  0.024  0.022  0.051  0.005 -0.084
## IYR  0.011 -0.055 -0.025 -0.042 -0.134  0.070 -0.033 -0.076 -0.050 -0.047
## VNQ  0.013 -0.057 -0.026 -0.046 -0.137  0.070 -0.039 -0.082 -0.050 -0.046
##        XLV    XLY    IYR    VNQ
## AGG  0.017 -0.013 -0.085 -0.086
## IEF -0.037  0.024  0.023  0.023
## LQD  0.018 -0.012 -0.057 -0.063
## SHY -0.026  0.006 -0.019 -0.017
## TIP -0.030 -0.016 -0.013 -0.013
## TLT -0.033  0.042  0.070  0.072
## DBC -0.164 -0.137 -0.112 -0.111
## GLD -0.121 -0.112  0.008  0.004
## USO -0.150 -0.114 -0.108 -0.108
## IJH -0.031  0.015  0.005  0.004
## IWM -0.018  0.018  0.011  0.013
## SPY  0.049  0.016 -0.055 -0.057
## VTV  0.033 -0.001 -0.025 -0.026
## XLB -0.061 -0.031 -0.042 -0.046
## XLE -0.058 -0.098 -0.134 -0.137
## XLF -0.030  0.024  0.070  0.070
## XLI -0.011  0.022 -0.033 -0.039
## XLK  0.029  0.051 -0.076 -0.082
## XLP  0.102  0.005 -0.050 -0.050
## XLU  0.019 -0.084 -0.047 -0.046
## XLV  0.000 -0.012 -0.114 -0.116
## XLY -0.012  0.000  0.028  0.027
## IYR -0.114  0.028  0.000  0.326
## VNQ -0.116  0.027  0.326  0.000
```

```
r_ = corr_mat
diag(r_) = (1 - 1 / diag(solve(corr_mat)))

eigen_r_ = eigen(corr_mat)

L_ = as.matrix(eigen_r_$vectors[,1:2]) %*% diag(sqrt(eigen_r_$values[1:2]))

h_ = rowSums(L_^2)

L_
```

```
##               [,1]          [,2]
##  [1,] -0.005531183 -0.8559314269
##  [2,] -0.431522255 -0.8519162491
##  [3,]  0.031779155 -0.8299946576
##  [4,] -0.422689195 -0.5995314187
##  [5,] -0.087879074 -0.8276441361
##  [6,] -0.457689589 -0.7436582124
##  [7,]  0.485698786  0.0301004272
##  [8,]  0.029982575 -0.2068726109
##  [9,]  0.418825918  0.1179865619
## [10,]  0.969807725 -0.0478616829
## [11,]  0.940870002 -0.0177036971
```

```
## [12,]  0.980678431 -0.0855238359
## [13,]  0.973462337 -0.0577645923
## [14,]  0.902105962 -0.0008876998
## [15,]  0.826887806 -0.0138717603
## [16,]  0.835386919 -0.0001495148
## [17,]  0.939862497 -0.0125122788
## [18,]  0.883757522 -0.0118619536
## [19,]  0.766541827 -0.2090250761
## [20,]  0.616949639 -0.3428207058
## [21,]  0.763266785 -0.2074221818
## [22,]  0.920928763 -0.0286089746
## [23,]  0.809191381 -0.1716330762
## [24,]  0.789963020 -0.1737168153
```

```r
fa = factanal(
  factors=2,
  covmat=corr_mat,
  rotation="none",
  method="mle",
  lower=0.0121
)

fa$loadings
```

```
##
## Loadings:
##     Factor1 Factor2
## AGG -0.154   0.748
## IEF -0.616   0.780
## LQD -0.119   0.715
## SHY -0.533   0.499
## TIP -0.258   0.719
## TLT -0.616   0.655
## DBC  0.428
## GLD          0.186
## USO  0.380
## IJH  0.926   0.265
## IWM  0.906   0.233
## SPY  0.948   0.296
## VTV  0.944   0.273
## XLB  0.862   0.203
## XLE  0.787   0.181
## XLF  0.818   0.202
## XLI  0.918   0.227
## XLK  0.868   0.224
## XLP  0.709   0.341
## XLU  0.521   0.399
## XLV  0.728   0.338
## XLY  0.891   0.254
## IYR  0.715   0.334
## VNQ  0.694   0.331
##
##                Factor1 Factor2
## SS loadings     11.665   4.089
## Proportion Var   0.486   0.170
```

```
## Cumulative Var   0.486   0.656
```

```r
fa = factanal(
  factors=2,
  covmat=corr_mat,
  rotation="varimax",
  method="mle",
  lower=0.0121
)

fa$loadings
```

```
##
## Loadings:
##     Factor1 Factor2
## AGG  0.127   0.754
## IEF -0.293   0.950
## LQD  0.148   0.710
## SHY -0.316   0.658
## TIP          0.763
## TLT -0.337   0.833
## DBC  0.415  -0.115
## GLD          0.194
## USO  0.344  -0.163
## IJH  0.959
## IWM  0.929  -0.110
## SPY  0.991
## VTV  0.979
## XLB  0.877  -0.123
## XLE  0.800  -0.116
## XLF  0.836  -0.108
## XLI  0.937  -0.120
## XLK  0.890  -0.105
## XLP  0.784
## XLU  0.630   0.183
## XLV  0.801
## XLY  0.923
## IYR  0.787
## VNQ  0.767
##
##                Factor1 Factor2
## SS loadings     11.843   3.911
## Proportion Var   0.493   0.163
## Cumulative Var   0.493   0.656
```

4.2. Specifične varijance faktora

Izračunajte specifične varijance faktora za model s dva faktora i model s tri faktora. Pomoću stupčastog dijagrama prikažite i usporedite dobivene vrijednosti.

```r
efa2 = factanal(
  factors=2,
  covmat=corr_mat,
  lower = 0.0121,
  rotation="none"
```

```r
)

efa3 = factanal(
  factors=3,
  covmat=corr_mat,
  rotation="none"
)

spec2 = efa2$uniquenesses
spec3 = efa3$uniquenesses

print(sum(spec2))
```

```
## [1] 8.245452
```

```r
print(sum(spec3))
```

```
## [1] 7.269145
```

```r
print(as.matrix(spec2))
```

```
##           [,1]
## AGG 0.41622330
## IEF 0.01210000
## LQD 0.47473593
## SHY 0.46741978
## TIP 0.41712554
## TLT 0.19247895
## DBC 0.81463324
## GLD 0.96209250
## USO 0.85492017
## IJH 0.07185763
## IWM 0.12464654
## SPY 0.01308330
## VTV 0.03439314
## XLB 0.21603419
## XLE 0.34729002
## XLF 0.29018870
## XLI 0.10662123
## XLK 0.19654223
## XLP 0.38131350
## XLU 0.56863933
## XLV 0.35548893
## XLY 0.14157102
## IYR 0.37766120
## VNQ 0.40839138
```

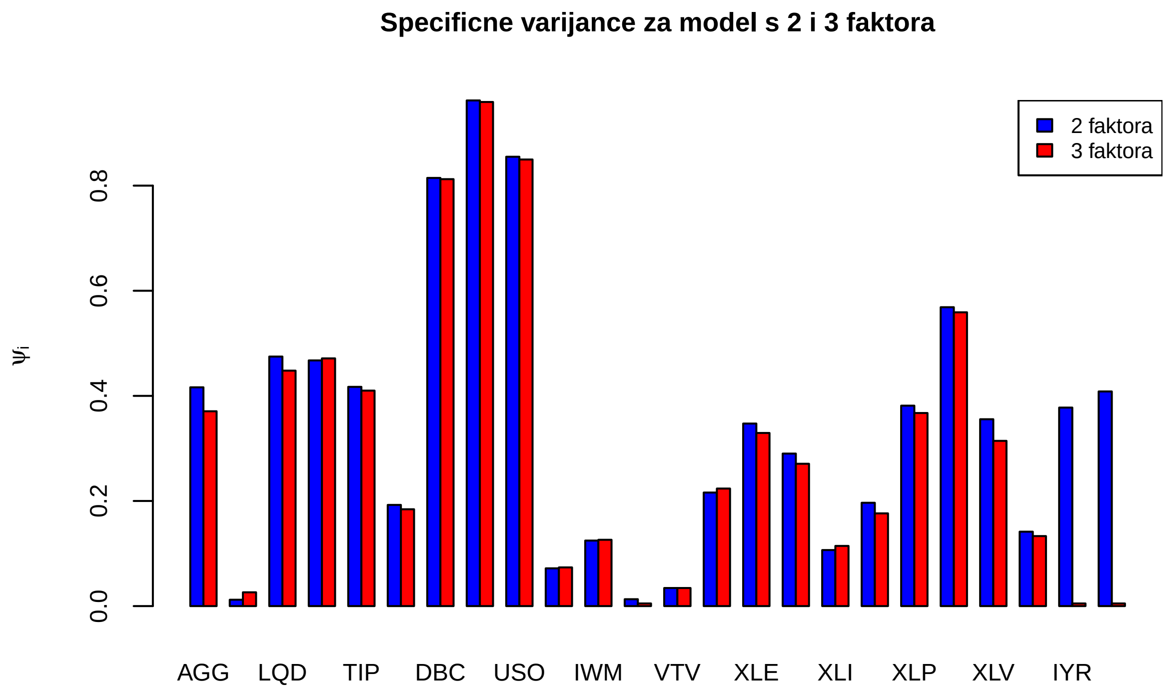
```r
print(as.matrix(spec3))
```

```
##           [,1]
## AGG 0.37070543
## IEF 0.02621197
## LQD 0.44793206
## SHY 0.47128939
## TIP 0.40993210
## TLT 0.18424117
```

```
## DBC 0.81230607
## GLD 0.95911061
## USO 0.84969459
## IJH 0.07361819
## IWM 0.12613619
## SPY 0.00500000
## VTV 0.03442122
## XLB 0.22365841
## XLE 0.32937108
## XLF 0.27079465
## XLI 0.11447156
## XLK 0.17641811
## XLP 0.36732958
## XLU 0.55895240
## XLV 0.31437596
## XLY 0.13317385
## IYR 0.00500000
## VNQ 0.00500000
```

```
barplot(
  rbind(spec2, spec3),
  beside=TRUE,
  col=c("blue", "red"),
  ylab=expression(psi["i"]), cex.axis = 0.75,
  cex.names = 0.75,
  cex.main=0.85,
  cex.lab=0.75,
  main="Specifične varijance za model s 2 i 3 faktora"
)

legend("topright",
       legend = c("2 faktora","3 faktora"),
       fill = c("blue", "red"),
       cex = 0.65)
```



5. Diskriminantna analiza

Financijska tržišta su od listopada 2007. do srpnja 2009. godine bila u krizi. U datoteci "crisis.csv" za svaki tjedan iz prethodno učitanih povijesnih tjednih cijena možete pronaći je li tržište tada bilo u krizi ili ne - 1 predstavlja krizu, 0 predstavlja period bez krize. Učitajte nove podatke te ih spojite s tablicom povrata.

```
df_crisis <- read.csv('crisis.csv')
df_crisis$Time <- as.Date(df$Time , format = "%d.%m.%Y 00:00")
head(df_crisis)
```

```
##         Time Crisis
## 1 2006-04-09      0
## 2 2006-04-16      0
## 3 2006-04-23      0
## 4 2006-04-30      0
## 5 2006-05-07      0
## 6 2006-05-14      0
```

```
print(all(df_crisis == df_crisis[order(df_crisis$Time),]))
```

```
## [1] TRUE
```

```
df_final <- ETF_returns
df_final['crisis'] <- df_crisis[2:nrow(df_crisis),]$Crisis
# df_final['time'] <- df_crisis[2:nrow(df_crisis),]$Time

print(nrow(ETF_returns))
```

```
## [1] 666
```

```
print(nrow(df_final))
```

```
## [1] 666
```

```
head(df_final)
```

```
##             AGG          IEF           LQD           SHY          TIP
## 2  0.0002029854  0.005468119  0.0046086562  1.379541e-03  0.009790208
## 3  0.0032582579 -0.001112755  0.0001910173  8.773836e-04 -0.001499406
## 4 -0.0022919577 -0.003428160 -0.0027633038 -8.812502e-05 -0.002062938
## 5 -0.0036751668 -0.004859505 -0.0019255222  0.000000e+00  0.001008842
## 6  0.0066600305  0.009517515  0.0072324729  1.632430e-03  0.005337094
## 7  0.0018327944  0.001736643 -0.0002868154  1.629770e-03  0.002705270
##            TLT         DBC          GLD         USO          IJH          IWM
## 2  0.004263435  0.04351289  0.062184891  0.05797730  0.031385785  0.028433553
## 3 -0.007547480 -0.01989275  0.029904984 -0.04381260 -0.004948152 -0.006259722
## 4 -0.002743879  0.00312249  0.044553728 -0.02326922  0.015291069  0.017322447
## 5 -0.011362752  0.04474696  0.046036257  0.01632354 -0.032080195 -0.051857111
## 6  0.021536346 -0.05325876 -0.077896524 -0.05122270 -0.031752524 -0.021769201
## 7 -0.002605885  0.02478361 -0.007319365  0.03324691  0.001959927  0.010292443
##            SPY          VTV          XLB         XLE          XLF         XLI
## 2  0.018957346  0.023142976  0.044006110  0.07041725  0.019704242  0.03041930
## 3  0.002439954  0.007212043 -0.026162816 -0.04352184  0.025362649 -0.01031827
## 4  0.007986479  0.014157800  0.034029699  0.03010175  0.005889347  0.03330440
## 5 -0.024750474 -0.025994947 -0.024538233 -0.03601792 -0.028103259 -0.02102023
## 6 -0.016558939 -0.017792487 -0.045575589 -0.04793810 -0.018072367 -0.02547943
## 7  0.010071201  0.010566879  0.009612687  0.01999286  0.010429283  0.00293776
##            XLK          XLP          XLU          XLV           XLY
## 2 -0.001809924  0.006879332  0.033399474  0.004513127  0.0020832268
## 3 -0.009519512  0.017506275  0.002239841 -0.005455758  0.0124745559
## 4  0.003203872  0.009651340  0.027458601 -0.006776284  0.0152536095
## 5 -0.042883509 -0.016209326 -0.028589021 -0.012670681 -0.0150244326
## 6 -0.018112389 -0.002112136 -0.005438440 -0.003619504 -0.0120273611
## 7  0.007767273  0.019898066  0.017369210  0.008916627 -0.0002970705
##            IYR          VNQ crisis
## 2  0.034792391  0.035294060      0
## 3 -0.003922369 -0.005908803      0
## 4  0.007032022  0.006401258      0
## 5 -0.032401913 -0.030592375      0
## 6 -0.014578809 -0.014372443      0
## 7  0.016552173  0.018544674      0
```

```
tail(df_final)
```

```
##              AGG          IEF           LQD           SHY          TIP
## 662  0.000758370 -0.002528685  0.0075074372  0.0004803300 -0.007011445
## 663  0.004173237  0.008523725 -0.0004304921  0.0023907603  0.002789412
## 664  0.004064656  0.005715917  0.0016024214  0.0023983332  0.002381625
## 665  0.004424788  0.005683499  0.0056883832  0.0003589784  0.006031216
## 666 -0.001312204 -0.002203103  0.0053026778 -0.0001196125  0.001635035
## 667 -0.001877100 -0.007103754  0.0036922901 -0.0008372204 -0.004806375
##               TLT          DBC          GLD          USO         IJH
## 662  0.0005912056 -0.015923603 -0.008722144 -0.026102610 -0.02574031
## 663  0.0213229333 -0.045954680  0.014180788 -0.115526802 -0.07042255
```

```
## 664  0.0027336150 -0.006941104  0.019710217 -0.004179728  0.02240054
## 665  0.0087566954  0.027681661  0.003138972  0.068205666  0.02379650
## 666 -0.0096634263  0.031649832  0.002964435  0.070726916  0.04660562
## 667 -0.0113288845  0.015665796 -0.006403990  0.037614679  0.03061456
##             IWM         SPY         VTV         XLB         XLE          XLF
## 662 -0.02367723 -0.01176158 -0.01543068 -0.01136579 -0.03090370 -0.034647550
## 663 -0.08399405 -0.07050491 -0.06239585 -0.04684085 -0.08785867 -0.053956248
## 664  0.03497707  0.02928958  0.02081839  0.03086420  0.01675277  0.035103070
## 665  0.03259070  0.01872855  0.02031138  0.02355292  0.04943033  0.028401865
## 666  0.04730659  0.02611043  0.02061440  0.01930579  0.03574410  0.009892828
## 667  0.02540374  0.02888246  0.02930692  0.02372294  0.02951140  0.061224490
##             XLI           XLK          XLP          XLU          XLV
## 662 -0.01440543 -0.0004643753 -0.004020485  0.006376194 -0.017667872
## 663 -0.06585238 -0.0791861623 -0.070840574 -0.044930677 -0.066374969
## 664  0.02441767  0.0371621784  0.007772021 -0.018212209  0.031083696
## 665  0.02164030  0.0016286319  0.013644414 -0.001135756  0.008328434
## 666  0.04174982  0.0331707480  0.007218163  0.008717074  0.021521627
## 667  0.03418298  0.0284859454  0.015882239 -0.001690795  0.025851316
##             XLY           IYR           VNQ crisis
## 662 -0.01006089 -0.0173699647 -0.0184884365      0
## 663 -0.08015757 -0.0647148673 -0.0660389563      0
## 664  0.04479519  0.0002677333 -0.0002687996      0
## 665  0.02929770 -0.0017400214 -0.0006723544      0
## 666  0.03719131  0.0449181806  0.0461517761      0
## 667  0.02132335  0.0221994611  0.0216077170      0
```

5.1. Diskriminantna analiza pomoću povrata

Provedite diskriminantnu analizu koja tjedne odvaja na krizne i one bez krize pomoću povrata fondova. Pomoću stupčastog dijagrama prikažite vektore srednjih vrijednosti u krizi i izvan nje. Također, na isti način prikažite korelaciju fonda AGG (Aggregate Bond ETF-a) s ostalim fondovima u krizi i izvan krize. Usporedite rezultate linearne diskriminantne analize (funkcija u R-u: `lda`) i kvadratne diskriminantne analize (funkcija u R-u: `qda`) pomoću tablica konfuzije i mjere APER (eng. apparent error rate). Razmislite o tome koji je razlog razlike u rezultatima ove dvije metode.

```r
df_not_crisis <- df_final[df_final$crisis==0,]
df_crisis <- df_final[df_final$crisis==1,]

head(df_not_crisis, 5)
```

```
##             AGG          IEF           LQD           SHY          TIP
## 2  0.0002029854  0.005468119  0.0046086562  1.379541e-03  0.009790208
## 3  0.0032582579 -0.001112755  0.0001910173  8.773836e-04 -0.001499406
## 4 -0.0022919577 -0.003428160 -0.0027633038 -8.812502e-05 -0.002062938
## 5 -0.0036751668 -0.004859505 -0.0019255222  0.000000e+00  0.001008842
## 6  0.0066600305  0.009517515  0.0072324729  1.632430e-03  0.005337094
##            TLT         DBC         GLD         USO          IJH          IWM
## 2  0.004263435  0.04351289  0.06218489  0.05797730  0.031385785  0.028433553
## 3 -0.007547480 -0.01989275  0.02990498 -0.04381260 -0.004948152 -0.006259722
## 4 -0.002743879  0.00312249  0.04455373 -0.02326922  0.015291069  0.017322447
## 5 -0.011362752  0.04474696  0.04603626  0.01632354 -0.032080195 -0.051857111
## 6  0.021536346 -0.05325876 -0.07789652 -0.05122270 -0.031752524 -0.021769201
##            SPY          VTV         XLB         XLE          XLF         XLI
## 2  0.018957346  0.023142976  0.04400611  0.07041725  0.019704242  0.03041930
## 3  0.002439954  0.007212043 -0.02616282 -0.04352184  0.025362649 -0.01031827
```

```
## 4  0.007986479  0.014157800  0.03402970  0.03010175  0.005889347  0.03330440
## 5 -0.024750474 -0.025994947 -0.02453823 -0.03601792 -0.028103259 -0.02102023
## 6 -0.016558939 -0.017792487 -0.04557559 -0.04793810 -0.018072367 -0.02547943
##            XLK          XLP          XLU          XLV          XLY          IYR
## 2 -0.001809924  0.006879332  0.033399474  0.004513127  0.002083227  0.034792391
## 3 -0.009519512  0.017506275  0.002239841 -0.005455758  0.012474556 -0.003922369
## 4  0.003203872  0.009651340  0.027458601 -0.006776284  0.015253609  0.007032022
## 5 -0.042883509 -0.016209326 -0.028589021 -0.012670681 -0.015024433 -0.032401913
## 6 -0.018112389 -0.002112136 -0.005438440 -0.003619504 -0.012027361 -0.014578809
##            VNQ crisis
## 2  0.035294060      0
## 3 -0.005908803      0
## 4  0.006401258      0
## 5 -0.030592375      0
## 6 -0.014372443      0
```

```
head(df_crisis, 5)
```

```
##            AGG           IEF          LQD           SHY          TIP
## 79 0.005866296 -0.0018027594 -0.000760888 -0.0018524394 -0.003948308
## 80 0.012871331  0.0223961136  0.016282876  0.0075478449  0.020514790
## 81 0.000296943 -0.0003539275 -0.003747835  0.0007365983  0.003690381
## 82 0.002069395  0.0060115127 -0.001536703  0.0027535363  0.009431489
## 83 0.002486547  0.0089317229  0.001703947  0.0040534073  0.012981087
##              TLT        DBC        GLD         USO          IJH          IWM
## 79 -0.0073635678 0.01333337 0.01621245 0.027617229 -0.002526461 -0.003685481
## 80  0.0322955388 0.02809385 0.01488137 0.049062484 -0.042281180 -0.056430203
## 81  0.0004415043 0.03251472 0.02628805 0.060023817  0.024143592  0.036920024
## 82  0.0103912384 0.03048573 0.02754537 0.044681749 -0.010215811 -0.030240205
## 83  0.0031852769 0.01137841 0.02943753 0.005648944 -0.027333472 -0.033697953
##             SPY          VTV          XLB         XLE         XLF          XLI
## 79  0.003080024 -0.002598532  0.009312523  0.02742622 -0.00975210 -0.006253286
## 80 -0.042601797 -0.045934303 -0.037599877 -0.03589506 -0.08272396 -0.034365485
## 81  0.026390756  0.022994957  0.037391402  0.03427491  0.03374248  0.010024797
## 82 -0.015752804 -0.026973376 -0.012939059 -0.01429517 -0.05519279 -0.001984946
## 83 -0.040079970 -0.032197534 -0.027855449 -0.01845761 -0.05339242 -0.032322123
##            XLK          XLP           XLU          XLV         XLY         IYR
## 79  0.01131353  0.008214139  0.0095358071  0.000000000 -0.00865036 -0.02066841
## 80 -0.02020910 -0.017711574 -0.0375393345 -0.028965204 -0.04865115 -0.06811620
## 81  0.02909746  0.021997897  0.0452942113  0.013636358  0.01584164  0.02929248
## 82  0.00751634 -0.010938968  0.0091479627 -0.007567441 -0.02325576 -0.04044775
## 83 -0.07744223 -0.005351473  0.0002383058 -0.014120032 -0.05013986 -0.03487607
##            VNQ crisis
## 79 -0.02803474      1
## 80 -0.07311535      1
## 81  0.03734257      1
## 82 -0.04339339      1
## 83 -0.03806862      1
```

```
print(nrow(df_not_crisis))
```

```
## [1] 574
```

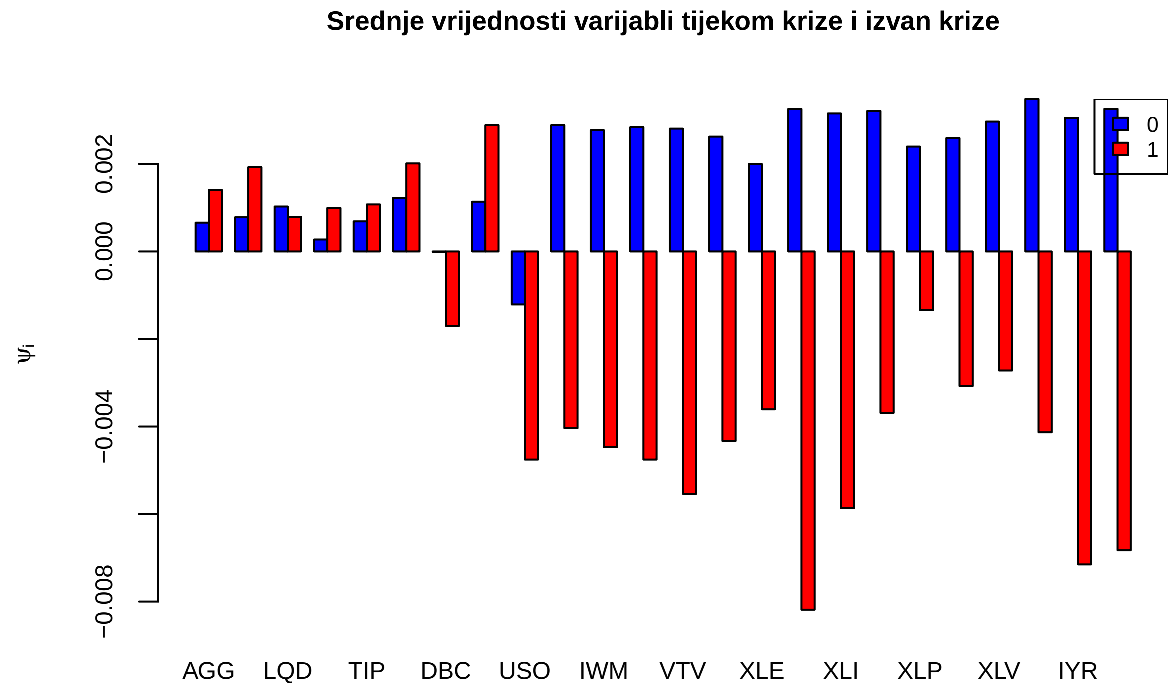
```
print(nrow(df_crisis))
```

```
## [1] 92
```

```r
means_not_crisis = colMeans(df_not_crisis[,1:ncol(df_not_crisis)-1])
means_crisis = colMeans(df_crisis[,1:ncol(df_not_crisis)-1])

barplot(
  rbind(means_not_crisis, means_crisis),
  beside=TRUE,
  col=c("blue", "red"),
  ylab=expression(psi["i"]), cex.axis = 0.75,
  cex.names = 0.75,
  cex.main=0.85,
  cex.lab=0.75,
  main="Srednje vrijednosti varijabli tijekom krize i izvan krize"
)

legend("topright",
       legend = c("0","1"),
       fill = c("blue", "red"),
       cex = 0.65)
```



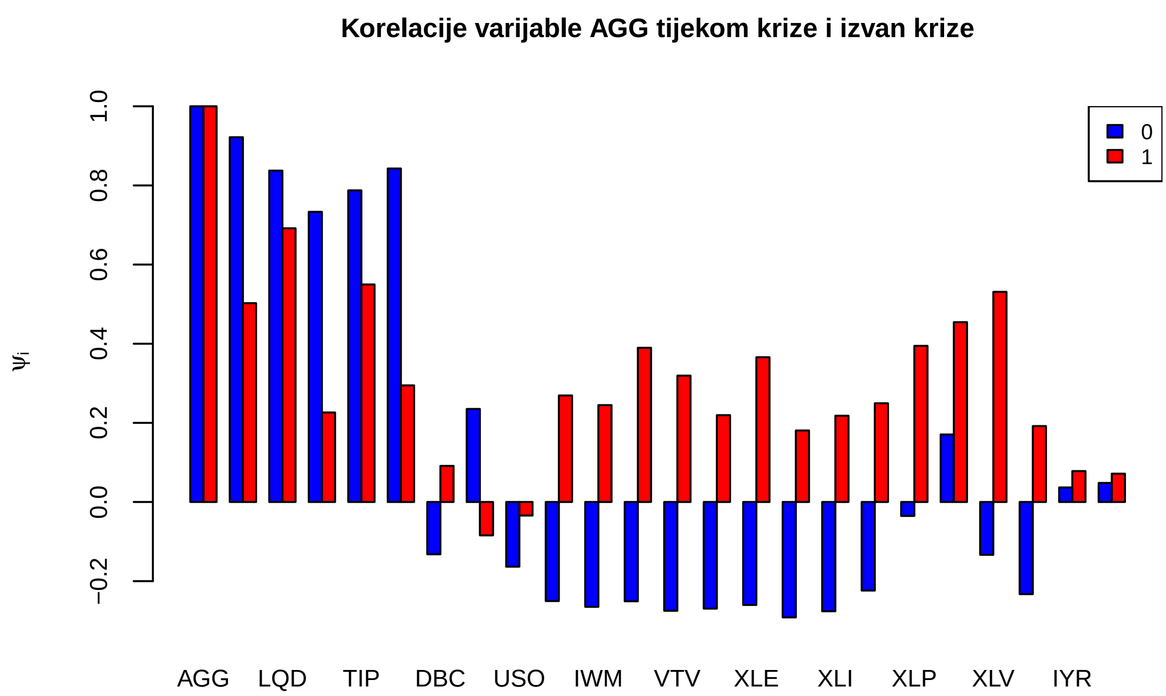

```r
corr_agg_not_crisis = cor(df_not_crisis[,1:ncol(df_not_crisis)-1], method='pearson')
corr_agg_crisis = cor(df_crisis[,1:ncol(df_not_crisis)-1], method='pearson')

corr_agg_not_crisis = corr_agg_not_crisis[1, 1:ncol(corr_agg_not_crisis)]
corr_agg_crisis = corr_agg_crisis[1, 1:ncol(corr_agg_crisis)]

barplot(
```

```
  rbind(corr_agg_not_crisis, corr_agg_crisis),
  beside=TRUE,
  col=c("blue", "red"),
  ylab=expression(psi["i"]), cex.axis = 0.75,
  cex.names = 0.75,
  cex.main=0.85,
  cex.lab=0.75,
  main="Korelacije varijable AGG tijekom krize i izvan krize"
)

legend("topright",
       legend = c("0","1"),
       fill = c("blue", "red"),
       cex = 0.65)
```




```
library(MASS)
lda <- lda(crisis ~ ., data = df_final)
lda
```

```
## Call:
## lda(crisis ~ ., data = df_final)
##
## Prior probabilities of groups:
##         0         1
## 0.8618619 0.1381381
##
```

```
## Group means:
##             AGG          IEF          LQD          SHY          TIP         TLT
## 0 0.0006585002 0.0007814796 0.0010273553 0.0002737896 0.0006877832 0.001227078
## 1 0.0014035647 0.0019259795 0.0007907013 0.0009937794 0.0010748529 0.002012661
##              DBC         GLD          USO          IJH          IWM          SPY
## 0 -9.140811e-06 0.001137580 -0.001210776  0.002886411  0.002771468  0.002839549
## 1 -1.699944e-03 0.002886794 -0.004755440 -0.004038833 -0.004468004 -0.004755435
##            VTV          XLB          XLE          XLF          XLI          XLK
## 0  0.002807967  0.002624798  0.001995896  0.003258310  0.003153840  0.003211696
## 1 -0.005538801 -0.004331118 -0.003607383 -0.008185944 -0.005865831 -0.003688247
##            XLP          XLU          XLV          XLY          IYR          VNQ
## 0  0.002397484  0.002591474  0.002968770  0.003484401  0.003050848  0.003260062
## 1 -0.001336865 -0.003076004 -0.002720177 -0.004132541 -0.007151279 -0.006828675
##
## Coefficients of linear discriminants:
##              LD1
## AGG   94.4759503
## IEF -108.4554216
## LQD    2.5452465
## SHY  593.4382982
## TIP  -17.5557319
## TLT   -2.7609786
## DBC    0.3654499
## GLD    0.3048000
## USO   -6.3658965
## IJH   26.8197254
## IWM    3.7078582
## SPY -182.2000073
## VTV  -36.9014610
## XLB    8.8045718
## XLE   31.4339760
## XLF   23.0391953
## XLI   -2.7914377
## XLK   35.3926434
## XLP   38.9220649
## XLU   -0.6212128
## XLV   18.8076922
## XLY   29.1550464
## IYR  -48.5644652
## VNQ   40.2462783
```

```
qda <- qda(crisis ~ ., data = df_final)
qda
```

```
## Call:
## qda(crisis ~ ., data = df_final)
##
## Prior probabilities of groups:
##         0         1
## 0.8618619 0.1381381
##
## Group means:
##             AGG          IEF          LQD          SHY          TIP         TLT
## 0 0.0006585002 0.0007814796 0.0010273553 0.0002737896 0.0006877832 0.001227078
## 1 0.0014035647 0.0019259795 0.0007907013 0.0009937794 0.0010748529 0.002012661
```

```
##              DBC         GLD          USO          IJH          IWM          SPY
## 0 -9.140811e-06 0.001137580 -0.001210776  0.002886411  0.002771468  0.002839549
## 1 -1.699944e-03 0.002886794 -0.004755440 -0.004038833 -0.004468004 -0.004755435
##             VTV          XLB          XLE          XLF          XLI          XLK
## 0  0.002807967  0.002624798  0.001995896  0.003258310  0.003153840  0.003211696
## 1 -0.005538801 -0.004331118 -0.003607383 -0.008185944 -0.005865831 -0.003688247
##             XLP          XLU          XLV          XLY          IYR          VNQ
## 0  0.002397484  0.002591474  0.002968770  0.003484401  0.003050848  0.003260062
## 1 -0.001336865 -0.003076004 -0.002720177 -0.004132541 -0.007151279 -0.006828675
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
library(lattice)
lda_pred <- predict(lda, df_final)$class
qda_pred <- predict(qda, df_final)$class

c1 = confusionMatrix(lda_pred, as.factor(df_final$crisis))
c2 = confusionMatrix(qda_pred, as.factor(df_final$crisis))

print('Confusion matrix for the LDA model:')
```

```
## [1] "Confusion matrix for the LDA model:"
```

```
c1$table
```

```
##           Reference
## Prediction   0   1
##          0 573  81
##          1   1  11
```

```
print('Confusion matrix for the QDA model:')
```

```
## [1] "Confusion matrix for the QDA model:"
```

```
c2$table
```

```
##           Reference
## Prediction   0   1
##          0 559  16
##          1  15  76
```

```
# 1 - TN,  2 - FP, 3 - FN, 4 - TP
#aper1 = (c1$table[2] + c1$table[3])/sum(c1$table)
#aper2 = (c2$table[2] + c2$table[3])/sum(c2$table)

#print('APER for LDA model:')
#print(aper1)
#print('APER for QDA model:')
#print(aper2)

# print(as.vector(1-c1$overall[1]))
# print(as.vector(1-c2$overall[1]))
```

```
mistakes <- 0
for (i in 1:nrow(df_final)) {
  holdout <- df_final[i, ]
  tmp <- df_final[-i, ]
```

```
  lda.fit <- lda(crisis ~ ., data = tmp)
  if (predict(lda.fit, holdout)$class != holdout$crisis) mistakes <- mistakes + 1
}

library(stringr)
str_c("APER: ",mistakes / nrow(df_final) * 100, "%")
```

```
## [1] "APER: 13.2132132132132%"
```

```
mistakes <- 0
for (i in 1:nrow(df_final)) {
  holdout <- df_final[i, ]
  tmp <- df_final[-i, ]
  qda.fit <- qda(crisis ~ ., data = tmp)
  if (predict(qda.fit, holdout)$class != holdout$crisis) mistakes <- mistakes + 1
}

library(stringr)
str_c("APER: ",mistakes / nrow(df_final) * 100, "%")
```

```
## [1] "APER: 6.15615615615616%"
```

###5.2. Diskriminantna analiza pomoću glavnih komponenti

Provedite diskriminantnu analizu kao u prošlom podzadatku, no ovaj put koristeći glavne komponente izračunate u 3. zadatku kao varijable. Provjerite i usporedite uspješnost klasifikacije koristeći tablice konfuzije i APER za različit broj komponenti.

2 komponente

```
pcdata = data.frame(pca$x[,1:2], crisis=df_final$crisis)
lda <- lda(crisis ~ ., data=pcdata)
qda <- qda(crisis ~ ., data=pcdata)
```

```
lda_pred <- predict(lda, pcdata)$class
qda_pred <- predict(qda, pcdata)$class

c1 = confusionMatrix(lda_pred, as.factor(pcdata$crisis))
c2 = confusionMatrix(qda_pred, as.factor(pcdata$crisis))

print('Confusion matrix for the LDA model:')
```

```
## [1] "Confusion matrix for the LDA model:"
```

```
c1$table
```

```
##           Reference
## Prediction   0   1
##          0 574  92
##          1   0   0
```

```
print('Confusion matrix for the QDA model:')
```

```
## [1] "Confusion matrix for the QDA model:"
```

```
c2$table
```

```
##           Reference
```

```
## Prediction   0   1
##          0 553  67
##          1  21  25
```

```
#print('APER for LDA model:')
#print(as.vector(1-c1$overall[1]))
#print('APER for QDA model:')
#print(as.vector(1-c2$overall[1]))
```

```
mistakes <- 0
for (i in 1:nrow(pcdata)) {
  holdout <- pcdata[i, ]
  tmp <- pcdata[-i, ]
  lda.fit <- lda(crisis ~ ., data = tmp)
  if (predict(lda.fit, holdout)$class != holdout$crisis) mistakes <- mistakes + 1
}
str_c("APER: ",mistakes / nrow(pcdata) * 100, "%")
```

```
## [1] "APER: 13.8138138138138%"
```

```
mistakes <- 0
for (i in 1:nrow(pcdata)) {
  holdout <- pcdata[i, ]
  tmp <- pcdata[-i, ]
  qda.fit <- qda(crisis ~ ., data = tmp)
  if (predict(qda.fit, holdout)$class != holdout$crisis) mistakes <- mistakes + 1
}
str_c("APER: ",mistakes / nrow(pcdata) * 100, "%")
```

```
## [1] "APER: 13.2132132132132%"
```

3 komponente

```
pcdata = data.frame(pca$x[,1:3], crisis=df_final$crisis)
lda <- lda(crisis ~ ., data=pcdata)
qda <- qda(crisis ~ ., data=pcdata)
```

```
lda_pred <- predict(lda, pcdata)$class
qda_pred <- predict(qda, pcdata)$class

c1 = confusionMatrix(lda_pred, as.factor(pcdata$crisis))
c2 = confusionMatrix(qda_pred, as.factor(pcdata$crisis))

print('Confusion matrix for the LDA model:')
```

```
## [1] "Confusion matrix for the LDA model:"
```

```
c1$table
```

```
##           Reference
## Prediction   0   1
##          0 574  92
##          1   0   0
```

```
print('Confusion matrix for the QDA model:')
```

```
## [1] "Confusion matrix for the QDA model:"
```

```r
c2$table
```

```
##           Reference
## Prediction   0   1
##          0 557  55
##          1  17  37
```

```r
#print('APER for LDA model:')
#print(as.vector(1-c1$overall[1]))
#print('APER for QDA model:')
#print(as.vector(1-c2$overall[1]))
```

```r
mistakes <- 0
for (i in 1:nrow(pcdata)) {
  holdout <- pcdata[i, ]
  tmp <- pcdata[-i, ]
  lda.fit <- lda(crisis ~ ., data = tmp)
  if (predict(lda.fit, holdout)$class != holdout$crisis) mistakes <- mistakes + 1
}
str_c("APER: ",mistakes / nrow(pcdata) * 100, "%")
```

```
## [1] "APER: 13.8138138138138%"
```

```r
mistakes <- 0
for (i in 1:nrow(pcdata)) {
  holdout <- pcdata[i, ]
  tmp <- pcdata[-i, ]
  qda.fit <- qda(crisis ~ ., data = tmp)
  if (predict(qda.fit, holdout)$class != holdout$crisis) mistakes <- mistakes + 1
}
str_c("APER: ",mistakes / nrow(pcdata) * 100, "%")
```

```
## [1] "APER: 10.960960960961%"
```