



**UNIVERZA V LJUBLJANI**  
University of Ljubljana

# Programsko orodje R v vodarstvu

**REŠITVE NALOG**

Nejc Bezak

Ljubljana, 2024

## **Programsko orodje R v vodarstvu**

*Rešitve nalog*

**Avtor:** *Nejc Bezak*

**Recenzenta:** *Mojca Šraj, Lovrenc Pavlin*

**Jezikovni pregled:** *Mojca Vilfan*

**Oblikovalec naslovnice:** *Gašper Mrak*

**Bolivian in prelom:** *Nejc Bezak*

**Bibliografsko-informacijska podpora:** *Elizabeta Kralj, Matevž Rudolf*

**Založnik:** *Založba Univerze v Ljubljani*

**Za založbo:** *Gregor Majdič, rektor Univerze v Ljubljani*

**Izdajatelj:** *Univerza v Ljubljani, Fakulteta za gradbeništvo in geodezijo*

**Za izdajatelja:** *Violeta Bokan Bosiljkov, dekanja Fakultete za gradbeništvo in geodezijo*

Ljubljana, 2024

Prva e-izdaja.

Publikacija je brezplačna.

Publikacija je v digitalni obliki prosto dostopna na: <https://ebooks.uni-lj.si>

DOI: 10.15292/9789612974459



To delo je ponujeno pod licenco Creative Commons Priznanje avtorstva-Nekomercialno-Brez predelav 4.0 Mednarodna licenca. / This work is licenced under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International licence.

Katalogni zapis o publikaciji (CIP) pripravili v  
Narodni in univerzitetni knjižnici v Ljubljani

COBISS.SI-ID 213713667

ISBN 978-961-297-445-9 (PDF)

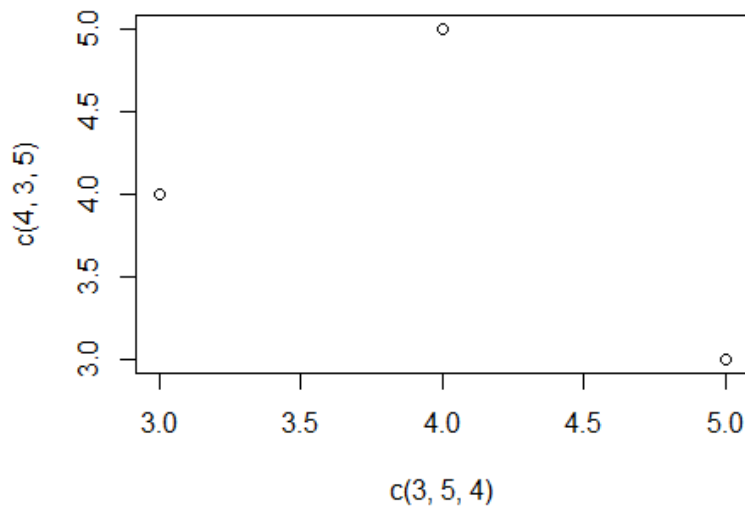
## Rešitve podanih praktičnih nalog

**Naloga 1: Poiščite funkcijo, s katero lahko izračunate standardno deviacijo v programskem orodju R, in funkcijo uporabite.**

```
?sd  
sd(c(3,4,2,54))  
## [1] 25.51307
```

**Naloga 2: Poiščite funkcijo, s katero izrišite graf v programskem orodju R (kakršen koli graf), in funkcijo uporabite.**

```
?plot  
plot(c(3,5,4),c(4,3,5))
```



**Naloga 3: Poiščite funkcijo, s katero lahko generirate zaporedje števil, recimo 2, 4, 6, 8, 10 itd., in to funkcijo tudi uporabite za izračun zaporedja, ki naj ima 50 elementov, začetna vrednost je 2 in korak zaporedja prav tako 2.**

```
?seq  
seq(from=2, by=2, length.out=50)  
## [1] 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34  
36 38  
## [20] 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68 70 72  
74 76  
## [39] 78 80 82 84 86 88 90 92 94 96 98 100
```

**Naloga 4: Poiščite funkcijo za zaokroževanje števil in zaokrožite število 2,464646 na 2 decimalni mesti.**

```
?round
```

```
round(x=2.464646, digits = 2)
```

```
## [1] 2.46
```

**Naloga 5: Recimo, da izvajamo meritve padavin vsak dan v tednu (od ponedeljka do nedelje) na dveh padavinskih postajah. Na prvi postaji smo izmerili 10, 0, 0, 20, 15, 10, 5 mm padavin, na drugi postaji pa 5, 0, 0, 25, 10, 5, 10 mm padavin. Podatke shranite v dva ločena objekta. Nato z uporabo objektov izračunajte vrednosti padavin v vseh sedmih dneh: izračunajte povprečne padavine z upoštevanjem obeh postaj (postaja1 in postaja2), izračunajte povprečne padavine v celem tednu (za vse dni skupaj), izračunajte največje dnevne in najmanjše dnevne padavine glede na meritve z obeh postaj, izračunajte razpon izmerjenih vrednosti (range), mediano in standardno deviacijo za vsako padavinsko postajo.**

```
pos1 <- c(10,0,0,20,15,10,5)
```

```
pos2 <- c(5,0,0,25,10,5,10)
```

```
(pos1+pos2)/2
```

```
## [1] 7.5 0.0 0.0 22.5 12.5 7.5 7.5
```

```
mean(pos1)
```

```
## [1] 8.571429
```

```
mean(pos2)
```

```
## [1] 7.857143
```

```
mean((pos1+pos2)/2)
```

```
## [1] 8.214286
```

```
summary(c(pos1,pos2))
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.000  1.250  7.500  8.214 10.000 25.000
```

```
range(c(pos1,pos2))
```

```
## [1] 0 25
```

```
median(pos1)
```

```
## [1] 10
```

```
sd(pos1)
```

```
## [1] 7.480132
```

```
median(pos2)
```

```
## [1] 5
```

```
sd(pos2)
```

```
## [1] 8.591247
```

**Naloga 6: Poiščite funkcijo za generiranje naključnih števil (na podlagi enakomerne porazdelitve) in generirajte 10 naključnih števil v razponu med 1 in 35. Rezultate shranite v nov objekt in zaokrožite na eno decimalno mesto. Izračunajte vsoto vseh generiranih števil, produkt oziroma zmnožek vseh generiranih vrednosti in razvrstite generirane vrednosti od najmanjše do največje.**

```
gen <- runif(n=10,min=1,max=35)
round(gen,digits = 1)

## [1] 4.9 8.9 15.9 3.9 15.2 12.9 5.0 33.2 6.4 18.3

prod(gen)

## [1] 10298673695

sort(gen)

## [1] 3.883061 4.879640 5.017991 6.439330 8.893927 12.874112 15.187857
## [8] 15.905254 18.320724 33.193895
```

**Naloga 7: Izračunajte seštevek vseh celih števil med 1 in 10.000.**

```
sum(1:10000)

## [1] 50005000
```

**Naloga 8: Na primeru s padavinami iz naloge 5 preverite, v katerih dneh na postaji 1 je povprečno padlo več kot 10 mm padavin in manj kot 20 mm padavin.**

```
which(pos1>10 & pos1<20)

## [1] 5
```

**Naloga 9: Naknadno je bilo ugotovljeno, da je bila izmerjena količina padavin za postajo 1 v četrtek napačna in bi morala znašati 30 mm. Preverite, ali ta sprememba vpliva na rezultat naloge 8.**

```
pos1[4] <- 30
which(pos1>10 & pos1<20)

## [1] 5
```

**Naloga 10: Podatke o padavinah z dveh postaj, ki so bili podani v sklopu naloge 5, združite v matriko. Dodajte tretji stolpec, kjer so prikazane povprečne vrednosti padavin na obeh postajah (po dnevih). Dodajte oznake vrstic (imena dni v tednu) in stolpcev (postaja 1, postaja 2, povprečje) z uporabo funkcij colnames in rownames.**

```
pos1 <- c(10,0,0,20,15,10,5)
pos2 <- c(5,0,0,25,10,5,10)
m1 <- cbind(pos1,pos2)
m1 <- cbind(m1,(pos1+pos2)/2)
```

```
rownames(m1) <- c("P","T","S","Č","P","S","N")
colnames(m1) <- c("Postaja1", "Postaja 2", "Povprečje")
m1
##   Postaja1 Postaja 2 Povprečje
## P      10      5      7.5
## T       0       0      0.0
## S       0       0      0.0
## Č      20      25     22.5
## P      15      10     12.5
## S      10       5      7.5
## N       5      10      7.5
```

**Naloga 11:** Poskusite uporabiti funkcijo *apply* na prej definirani matriki (naloga 10) za izračun povprečnih vrednosti na postaji 1, postaji 2 in povprečnih vrednosti.

```
apply(m1,2,mean)
##   Postaja1 Postaja 2 Povprečje
## 8.571429  7.857143  8.214286
```

**Naloga 12:** Na podlagi objekta *airquality* (za uporabo podatkov potrebujete funkcijo *data(airquality)*) preverite, katere dni je bila hitrost vetra večja od 10 mph (enota, v kateri je podana hitrost vetra), preverite, katere dni je bila temperatura zraka med 60 in 70 F (enota, v kateri je podana temperatura zraka) in kateri dan je bila izmerjena največja ter najmanjša koncentracija ozona.

```
data(airquality)
which(airquality$Wind>10)
## [1]  3  4  5  6  8  9 14 15 16 17 18 19 22 24 25 26 28
## [20] 37 40 41 42 45 46 47 48 50 51 58 59 60 65 67 73 74
## [39] 78 81 84 88 94 100 103 104 105 107 108 111 112 113 114 115 129 1
## [58] 132 134 135 137 138 140 141 142 144 146 147 148 150 151 153
which(airquality$Temp>60 & airquality$Temp<70)
## [1]  1  4  6  7  9 10 12 13 14 16 17 19 20 23 24 28 34
## [20] 142 144 147 148 153
which.max(airquality$Ozone)
## [1] 117
which.min(airquality$Ozone)
## [1] 21
```

**Naloga 13: Razvrstite podatke airquality glede na izmerjeno temperaturo zraka. Preverite delovanje funkcij order in sort.**

```
head(airquality[order(airquality$Temp),])

##      Ozone Solar.R Wind Temp Month Day
## 5      NA      NA 14.3   56     5   5
## 18     6      78 18.4   57     5  18
## 25     NA      66 16.6   57     5  25
## 27     NA      NA  8.0   57     5  27
## 15    18      65 13.2   58     5  15
## 26     NA     266 14.9   58     5  26

sort(airquality$Temp)

## [1] 56 57 57 57 58 58 59 59 61 61 61 62 62 63 64 64 65 65 66 66 66 67 67
## [26] 68 68 68 68 69 69 69 70 71 71 71 72 72 72 73 73 73 73 73 74 74 74 74
## [51] 75 75 76 76 76 76 76 76 76 76 76 77 77 77 77 77 77 77 78 78 78 78 78
## [76] 79 79 79 79 79 80 80 80 80 80 81 81 81 81 81 81 81 81 81 81 81 82 82
## [101] 82 82 82 82 82 83 83 83 83 84 84 84 84 84 85 85 85 85 85 86 86 86 86
## [126] 86 87 87 87 87 87 88 88 88 89 89 90 90 90 91 91 92 92 92 92 92 93 93
## [151] 94 96 97

order(airquality$Temp)

## [1] 5 18 25 27 15 26 8 21 9 23 24 4 20 148 16 144 7
## [19] 6 13 17 1 28 34 140 14 19 142 153 10 12 147 149 137 138
## [37] 2 48 114 22 50 58 73 133 3 11 33 82 56 115 132 151 31
## [55] 53 54 55 110 135 141 152 47 52 60 108 113 136 150 32 57 111
## [73] 131 139 30 37 46 107 109 116 45 59 76 106 130 29 64 74 77
## [91] 92 93 94 117 134 146 38 44 72 78 84 87 95 105 143 61 66
## [109] 91 35 62 65 79 129 36 63 81 86 97 85 88 90 96 103 104
## [127] 39 41 80 98 128 68 89 119 71 99 40 100 101 75 124 43 69
## [145] 102 125 42 126 127 121 123 122 120
```

**Naloga 14: Z uporabo funkcije apply izračunajte povprečje vseh stolpcev v objektu airquality.**

```
apply(airquality,2,mean)
```

```
##      Ozone      Solar.R      Wind      Temp      Month      Day
##      NA          NA  9.957516  77.882353  6.993464  15.803922
```

```
apply(airquality,2,mean,na.rm=T)
```

```
##      Ozone      Solar.R      Wind      Temp      Month      Day
##  42.129310 185.931507  9.957516  77.882353  6.993464  15.803922
```

***Naloga 15: Definirajte nov objekt v obliki seznama (list), kjer združite dva poljubna stolpca objekta airquality in povprečne padavine, ki ste jih obravnavali v okviru naloge 5.***

```
obj <- list(temp=airquality$Temp,veter=airquality$Wind,pad=pos1)
head(obj)
```

```
## $temp
## [1] 67 72 74 62 56 66 65 59 61 69 74 69 66 68 58 64 66 57 68 62 59 73 61
61 57
## [26] 58 57 67 81 79 76 78 74 67 84 85 79 82 87 90 87 93 92 82 80 79 77 72
65 73
## [51] 76 77 76 76 76 75 78 73 80 77 83 84 85 81 84 83 83 88 92 92 89 82 73
81 91
## [76] 80 81 82 84 87 85 74 81 82 86 85 82 86 88 86 83 81 81 81 82 86 85 87
89 90
## [101] 90 92 86 86 82 80 79 77 79 76 78 78 77 72 75 79 81 86 88 97 94 96 94
91 92
## [126] 93 93 87 84 80 78 75 73 81 76 77 71 71 78 67 76 68 82 64 71 81 69 63
70 77
## [151] 75 76 68
##
## $veter
## [1] 7.4 8.0 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 6.9 9.7 9.2 10.
9 13.2
## [16] 11.5 12.0 18.4 11.5 9.7 9.7 16.6 9.7 12.0 16.6 14.9 8.0 12.0 14.
9 5.7
## [31] 7.4 8.6 9.7 16.1 9.2 8.6 14.3 9.7 6.9 13.8 11.5 10.9 9.2 8.
0 13.8
## [46] 11.5 14.9 20.7 9.2 11.5 10.3 6.3 1.7 4.6 6.3 8.0 8.0 10.3 11.
5 14.9
## [61] 8.0 4.1 9.2 9.2 10.9 4.6 10.9 5.1 6.3 5.7 7.4 8.6 14.3 14.
9 14.9
## [76] 14.3 6.9 10.3 6.3 5.1 11.5 6.9 9.7 11.5 8.6 8.0 8.6 12.0 7.
4 7.4
## [91] 7.4 9.2 6.9 13.8 7.4 6.9 7.4 4.6 4.0 10.3 8.0 8.6 11.5 11.
5 11.5
## [106] 9.7 11.5 10.3 6.3 7.4 10.9 10.3 15.5 14.3 12.6 9.7 3.4 8.0 5.
7 9.7
## [121] 2.3 6.3 6.3 6.9 5.1 2.8 4.6 7.4 15.5 10.9 10.3 10.9 9.7 14.
9 15.5
```



```
## [136] 6.3 10.9 11.5 6.9 13.8 10.3 10.3 8.0 12.6 9.2 10.3 10.3 16.6 6.
9 13.2
## [151] 14.3 8.0 11.5
##
## $pad
## [1] 10 0 0 20 15 10 5
```

**Naloga 16: Izračunajte povprečne vrednosti vseh petih spremenljivk, ki so v datoteki s podatki vodomerne postaje Veliko Širje na reki Savinji (za leto 2005).**

```
podatki <- read.table(file="C:/Users/nbezak/OneDrive - Univerza v Ljubljani/U
cbenik/Savinja-Veliko SirjeI-2005.txt",header=TRUE,sep=";",dec=".")
apply(podatki[,2:6],2,mean) # drugi argument 2 označuje stolpce, 1 bi pomenil
o vrstice
```

```
##                vodostaj.cm.
##                229.70685
##                pretok.m3.s.
##                42.50639
##                temp.vode.C.
##                10.57425
## transport_suspendiranega_materiala.kg.s.
##                10.68967
## vsebnost_suspendiranega_materiala.g.m3.
##                57.06575
```

**Naloga 17: V programsko okolje R uvozite poljubne podatke, ki ste jih že kdaj uporabili.**

```
podatki <- read.table(file="C:/Users/nbezak/OneDrive - Univerza v Ljubljani/U
cbenik/Savinja-Veliko SirjeI-2005.txt",header=TRUE,sep=";",dec=".")
```

**Naloga 18: Shranite poljubni objekt v format .Rdata in ga nato naložite nazaj v R z uporabo funkcije load.**

```
save(pos1,file="C:/Users/nbezak/OneDrive - Univerza v Ljubljani/Ucbenik/Nalog
e-gradivo/padavine.Rdata")
# objekt pos1 mora obstajati
load("padavine.Rdata")
```

**Naloga 19: Najdite paket za uporabo in izračun momentov L, paket namestite in ga uporabite, izračunajte L-momente podatkov o pretoku, ki ste jih obravnavali v okviru Naloga 16 (L-momenti so nekaj podobnega kot navadni statistični momenti, povprečje, varianca, asimetrija (angl. skewness), sploščenost (angl. kurtosis)).**

```
# install.packages("lmomco")
library(lmomco)
lmoms(podatki$pretok.m3.s.)

## $lambdas
## [1] 42.506386 20.381628 10.237436 6.157273 4.216836
##
## $ratios
```

```
## [1] NA 0.4794957 0.5022874 0.3020992 0.2068940
##
## $trim
## [1] 0
##
## $lefttrim
## NULL
##
## $righttrim
## NULL
##
## $source
## [1] "lmoms"
```

**Naloga 20:** Za podatke, vključene v paket *airGR*, z uporabo funkcije *apply* izračunajte osnovno statistiko, z uporabo funkcije *summary* pa osnovne statistike za stolpce 2, 3 in 4 ter ugotovite, za katere hidrološke spremenljivke gre ter kakšne so njihove enote.

```
library(airGR) # aktivacija paketa z imenom airGR
data(L0123001)
apply(BasinObs[,c(2,3,4)],2,summary)

##           P           T           E
## Min.      0.000000 -18.700000 0.000000
## 1st Qu.    0.000000  4.100000 0.600000
## Median    0.300000  9.100000 1.400000
## Mean      2.914595  9.147088 1.764099
## 3rd Qu.   3.600000 14.500000 2.900000
## Max.     66.800000 28.400000 5.500000

# 2 stolpec-padavine (mm), 3 stolpec-temperatura zraka (C)
# 4 stolpec-evapotranspiracija (mm)
```

**Naloga 21:** Izračunajte največje, najmanjše in povprečne vrednosti pretokov v posameznih obdobjih (januar-marec, april-junij itd., torej za četrtine leta). Namig: iščete podobno funkcijo kot je *as.yearmon*.

```
library(zoo)

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##      as.Date, as.Date.numeric

podatki[,1] <- as.POSIXct(strptime(podatki[,1],format="%d.%m.%Y"))
Qzoo <- zoo(podatki[,3],podatki[,1]) # definiramo objekt zoo
cet <- as.yearqtr(time(Qzoo)+3600)
aggregate(Qzoo, cet, max)
```

```
## 2005 Q1 2005 Q2 2005 Q3 2005 Q4
## 79.226 179.064 369.615 382.100

aggregate(Qzoo, cet, min)

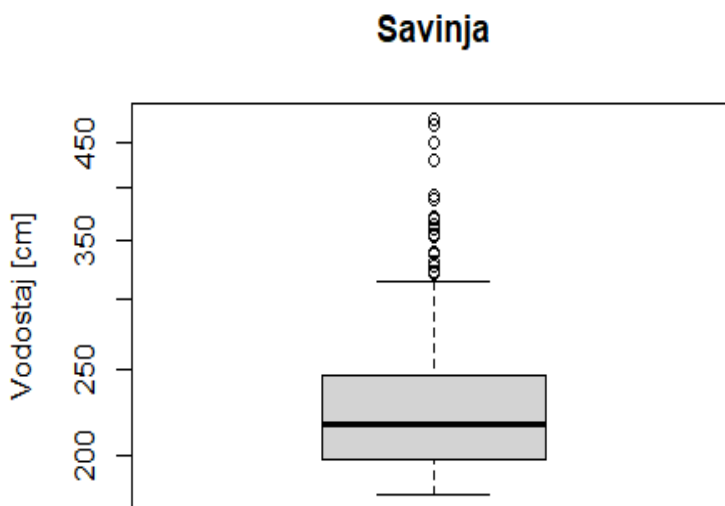
## 2005 Q1 2005 Q2 2005 Q3 2005 Q4
## 9.309 9.568 15.205 10.400

aggregate(Qzoo, cet, mean)

## 2005 Q1 2005 Q2 2005 Q3 2005 Q4
## 21.01474 34.82178 65.38862 48.54896
```

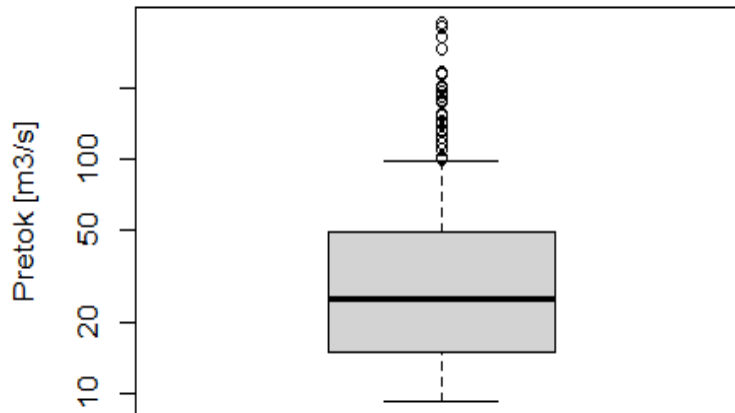
**Naloga 22: Za podatke z vodotoka Savinja (vodostaj, pretok in transport suspendiranih snovi) izrišite okvir z ročaji (boxplot) in graf ustrezno opremite z naslovom, oznakami osi itd. Y-os na grafu naj bo izrisana v log merilu.**

```
podatki[,1] <- as.POSIXct(strptime(podatki[,1],format="%d.%m.%Y"))
boxplot(podatki$vodostaj.cm.,log="y",ylab="Vodostaj [cm]",main="Savinja")
```



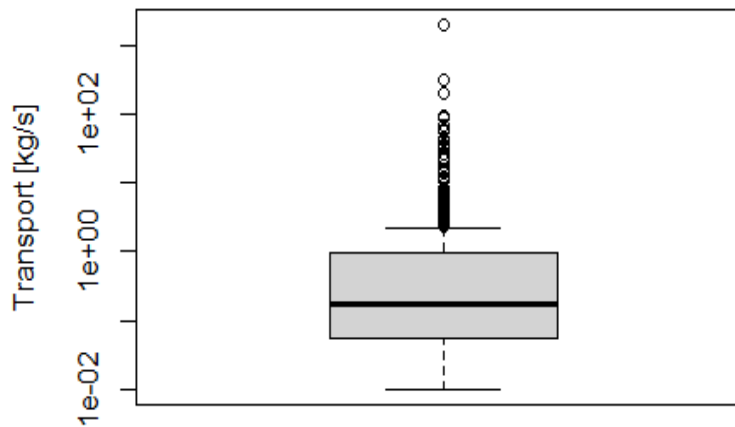
```
boxplot(podatki[,3],log="y",ylab="Pretok [m3/s]",main="Savinja")
```

## Savinja



```
boxplot(podatki[,5],log="y",ylab="Transport [kg/s]",main="Savinja")
```

## Savinja



**Naloga 23: Iz nabora bolj naprednih grafov (dober pregled podaja spletna stran R graph gallery<sup>1</sup>) izberite en tip grafa, ki ga tudi izrišite in ustrezno opremito.**

```
# graf, ki prikazuje povezanost pretokov z vodostajem ter raztros vrednosti  
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

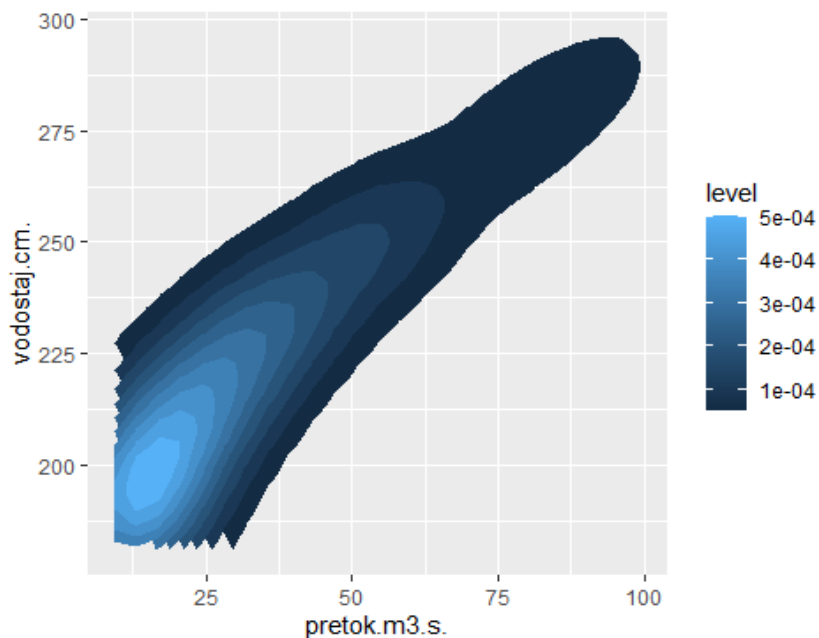
```
ggplot(podatki, aes(x=pretok.m3.s., y=vodostaj.cm.) ) +  
  stat_density_2d(aes(fill = ..level..), geom = "polygon")
```

```
## Warning: The dot-dot notation (`..level..`) was deprecated in ggplot2 3.4.  
0.
```

---

<sup>1</sup> <https://r-graph-gallery.com/>.

```
## i Please use `after_stat(level)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



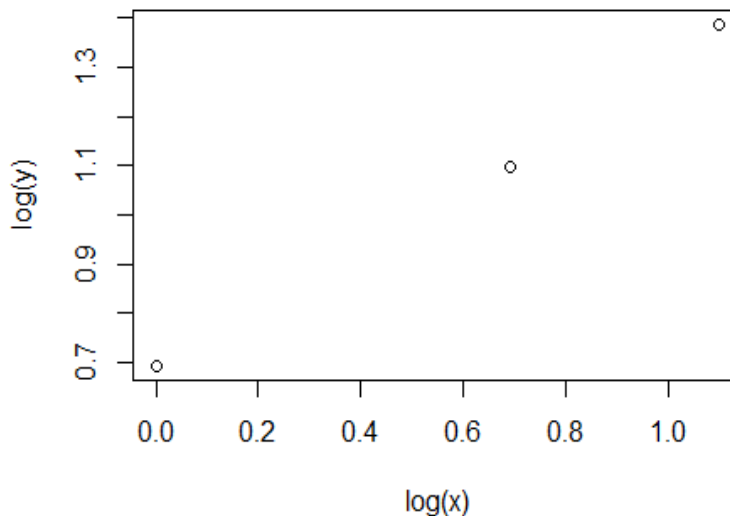
**Naloga 24: Zapišite funkcijo, ki vam omogoča normalizacijo podatkov.**

```
norm <- function(x){
k1 <- x-min(x)
k2 <- max(x)-min(x)
k3 <- k1/k2
return(k3)}
norm(c(4,3,5,3.3,4.5))

## [1] 0.50 0.00 1.00 0.15 0.75
```

**Naloga 25: Definirajte funkcijo, ki ima dva argumenta (vektor  $x$  in  $y$ ) in ki vam izriše razsevni (scatter) graf teh dveh vektorjev, kjer sta  $x$  in  $y$  os v log merilu.**

```
graf <- function(x,y){
plot(log(x),log(y))
}
graf(c(1:3),c(2:4))
```



**Naloga 26: Definirajte in uporabite funkcijo, ki vam izračuna povprečje in varianco vašega vzorca.**

```
povvar <- function(x){
  povp <- mean(x)
  varianca <- var(x)
  return(c(povp,varianca))
}
povvar(3:50)
## [1] 26.5 196.0
```

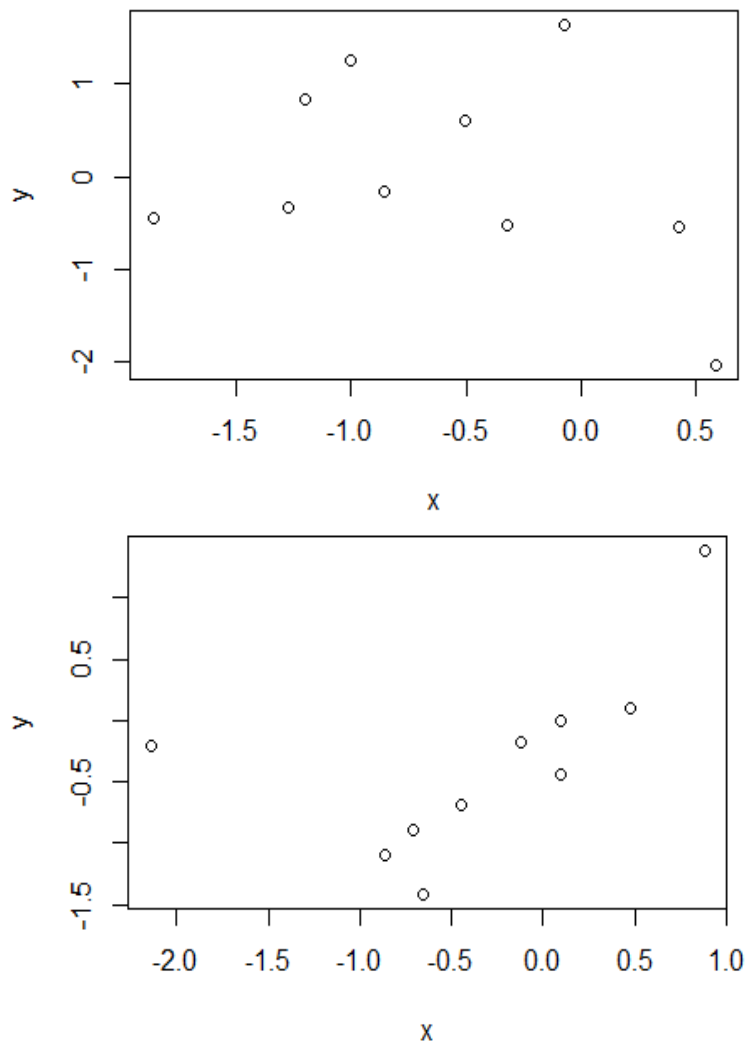
**Naloga 27: Z uporabo zanke if preverite, ali so na voljo vsi dnevni podatki o pretokih z vodotoka Savinja za leto 2005. Rezultat zanke naj bo opisne oblike (character) in naj pove, ali je podatkov dovolj ali kakšen manjka.**

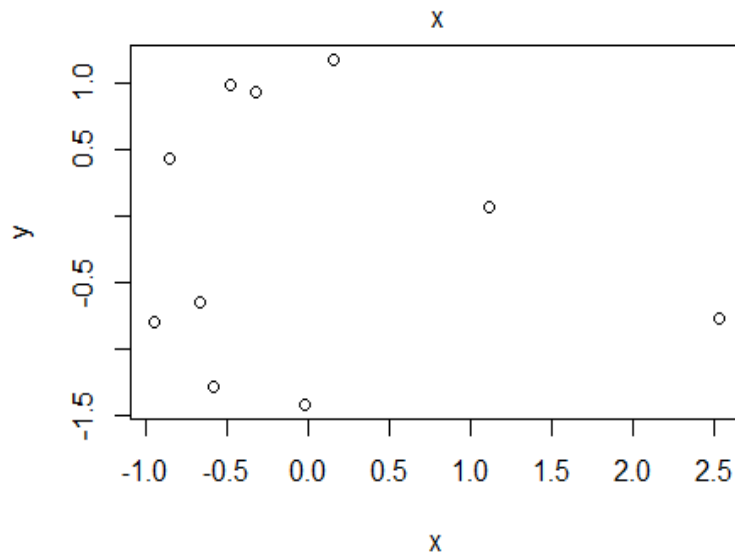
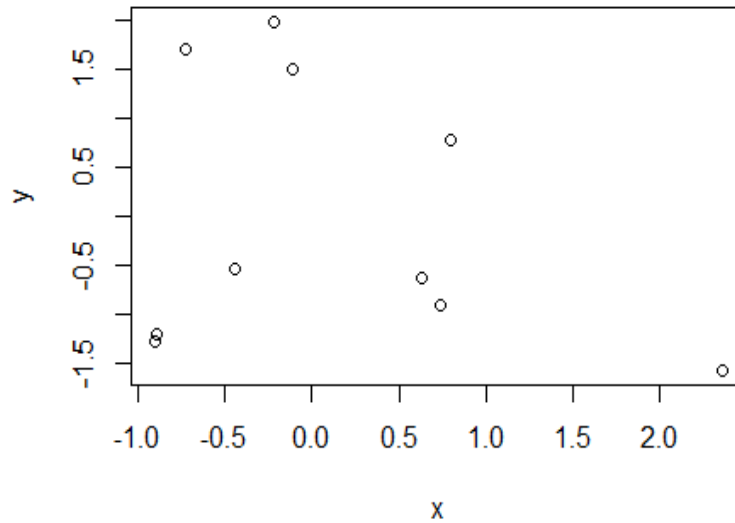
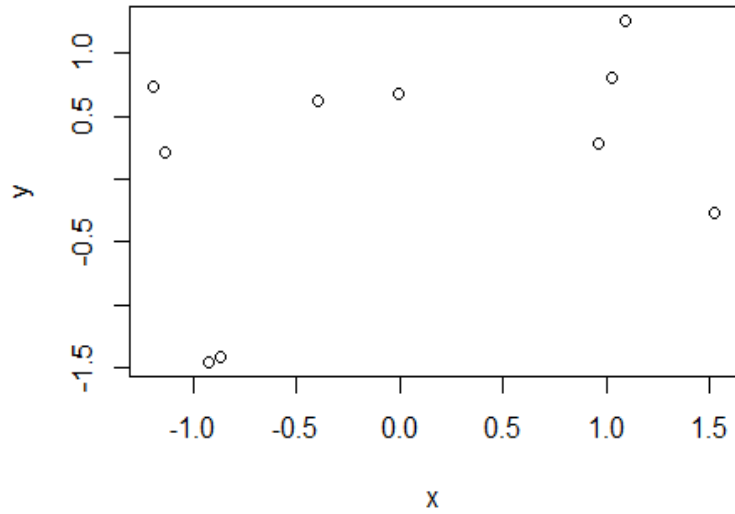
```
# Glede na dolžino
if(length(podatki$vodostaj.cm.)==365) {
  "Na voljo so vsi podatki"
} else {
  "Del podatkov manjka"
}
## [1] "Na voljo so vsi podatki"

# z uporabo funkcije NA
if(any(is.na(podatki$vodostaj.cm.))) {
  "Vsaj en podatek manjka"
} else {
  "Na voljo so vsi numerični podatki"
}
## [1] "Na voljo so vsi numerični podatki"
```

**Naloga 28:** Z uporabo zanke `for` 5-krat naključno generirajte vektor  $x$  in vektor  $y$  (oba vektorja naj vsebujeta 10 elementov, glede na normalno porazdelitev, povprečje 0 in standardno deviacijo 1) ter oba vektorja prikažite na razsevnem diagramu z uporabo funkcije `plot`.

```
for(i in 1:5){  
  x <- rnorm(10)  
  y <- rnorm(10)  
  plot(x,y)  
}
```



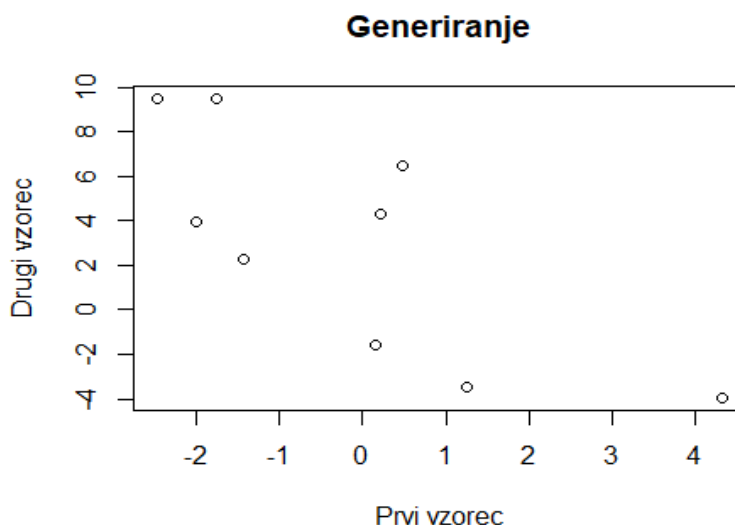


**Naloga 29: Izrišite graf dveh naključno generiranih vzorcev. Oba vzorca generirajte z uporabo normalne porazdelitve: enkrat povprečje 0 in standardna deviacija 2, drugič**



**povprečje 3 in standardna deviacija 5. Na grafu naj bo 9 točk. Graf ustrezno opremite z oznakami, osmi, legendo itd.**

```
x <- rnorm(9,mean=0,sd=2)
y <- rnorm(9,mean=3,sd=5)
plot(x,y,xlab="Prvi vzorec",ylab="Drugi vzorec",main="Generiranje")
```



**Naloga 30: Poiščite ustrezen test, ki ga lahko uporabite za izračun, ali je trend v vzorcu statistično značilen ali ne. Naložite ustrezen paket in uporabite test z enim naključno generiranim vzorcem na podlagi normalne porazdelitve s srednjo vrednostjo 10 in standardno deviacijo 5 (generirajte 20 števil). Postopek ponovite za zaporedje števil od 1:20.**

```
library(Kendall)
## Warning: package 'Kendall' was built under R version 4.1.3
MannKendall(rnorm(20,mean=10,sd=5))
## tau = -0.0632, 2-sided pvalue =0.72118
MannKendall(1:20)
## tau = 1, 2-sided pvalue =< 2.22e-16
```

**Naloga 31: Najdite ustrezno funkcijo oziroma model, ki opiše povezavo med pretokom in vsebnostjo suspendiranih snovi tako, da je  $r^2$  večji od 0,3. Uporabite podatke z vodomerne postaje Veliko Širje na Savinji.**

```
mod <- nls(vsebnost_suspendiranega_materiala.g.m3. ~ a*(pretok.m3.s.)^b, data
= podatki, start = list(a=10,b=3))
RSS.p <- sum(residuals(mod)^2)
TSS <- sum((podatki$vsebnost_suspendiranega_materiala.g.m3. - mean(podatki$vsebnost_suspendiranega_materiala.g.m3.))^2)
```

```
r2 <- 1 - (RSS.p/TSS)
r2 # r2 za uporabljen model

## [1] 0.3419598
```

**Naloga 32: S spletne strani Agencija RS za okolje prenesite dnevne podatke o pretokih z vodomerne postaje Sava Šentjakob za obdobje 1990–2021. Z uporabo paketa lfstat analizirajte največje sušne dogodke (najnižji letni pretoki) v posameznem letu. Izvedite tako verjetnostne analize nizkih pretokov (Pearsonova porazdelitev tipa 3) kot analize trajanja teh sušnih dogodkov.**

```
Sava <- read.delim("C:/Users/nbezak/OneDrive - Univerza v Ljubljani/Ucbenik/Naloga-gradivo/hid_3570_1990_2021.txt")
Sava[,1] <- as.POSIXct(strptime(Sava[,1],format="%d.%m.%Y"))
library(lfstat);library(zoo)

## Warning: package 'lfstat' was built under R version 4.1.3

## Loading required package: xts

## Loading required package: lmom

##
## Attaching package: 'lmom'

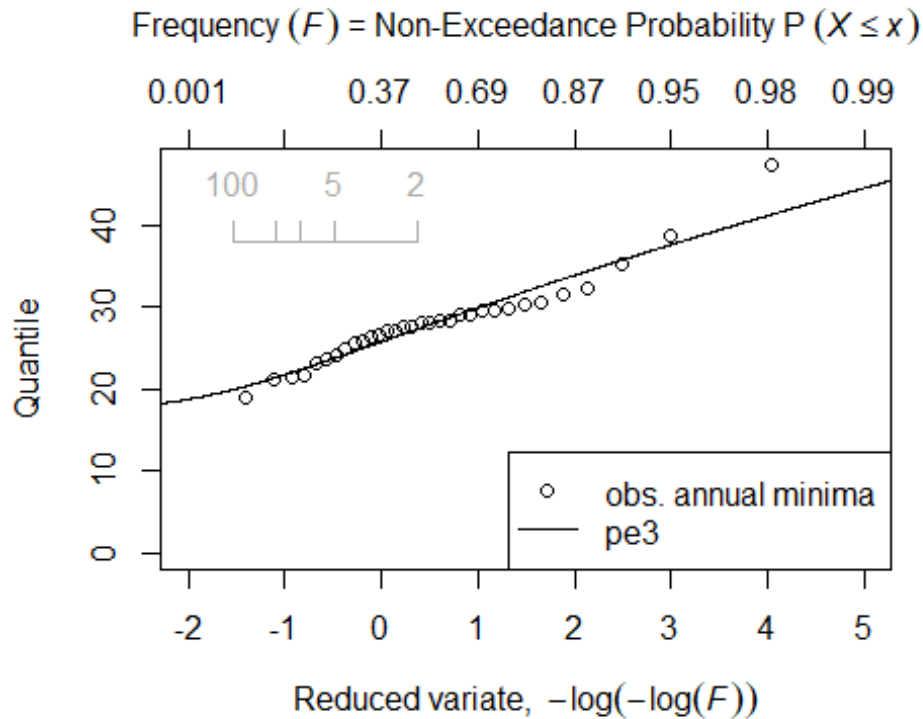
## The following objects are masked from 'package:lmomco':
##
##      cdfexp, cdfgam, cdfgev, cdfglo, cdfgno, cdfgpa, cdfgum, cdfkap,
##      cdfln3, cdfnor, cdfpe3, cdfwak, cdfwei, quaexp, quagam, quagev,
##      quaglo, quagno, quagpa, quagum, quakap, qualn3, quanor, quape3,
##      quawak, quawei

## Loading required package: lattice

# definiramo objekt zoo
QzooSava <- zoo(Sava[,3],Sava[,1])
# objekt lfobj, argument hyearstart določa začetek hidrološkega leta
nizkiQSava <- createlfobj(ts(QzooSava), startdate = "01/01/1990", hyearstart
= 1)
setlfunit("m^3/s") # definiramo enote naših podatkov
nletniSava <- MAM(nizkiQSava,n=1,yearly=TRUE)
rezSava <- tyears(nizkiQSava, dist = "pe3", event = 100, plot = TRUE)

## Warning in as.xts.lfobj(x): No unit found in attributes, assuming 'm3/s'.
## Use flowunit(x) <- "l/s" to define the flow unit. See help(flowunit).

## Warning: For fitting minima, a Weibull distribution with parameter 'zeta =
0'
## may be best.
```



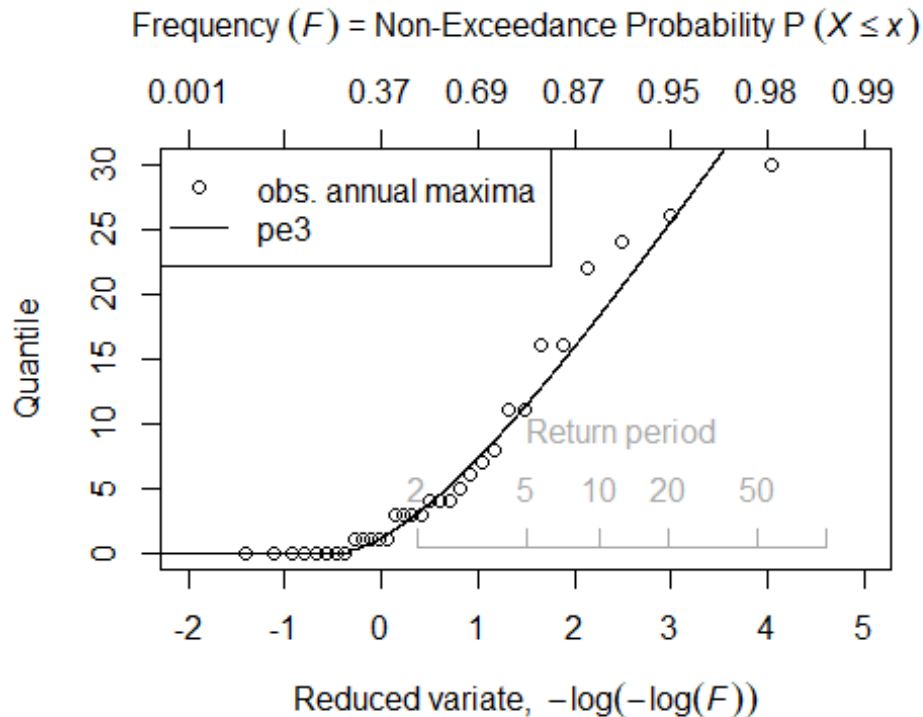
```

maksTrajanjeSava <- tyearsS(nizkiQSava, dist = "pe3", variable = "d", aggr =
"max", event = 100, hyearstart = 1, pooling = "IC")

## Warning in as.xts.lfobj(x): No unit found in attributes, assuming 'm3/s'.
## Use flowunit(x) <- "l/s" to define the flow unit. See help(flowunit).

## Warning in evfit(x = ag, distribution = dist, zeta = zeta, check = check,
:
## There were 8 years with zero flow extremes. Therefore a mixed distribution
with
## p_0 = 0.25 and zeta = '0' was fitted. L-moments and parameters are only va
lid
## for the censored time series. See ?tyears for details.

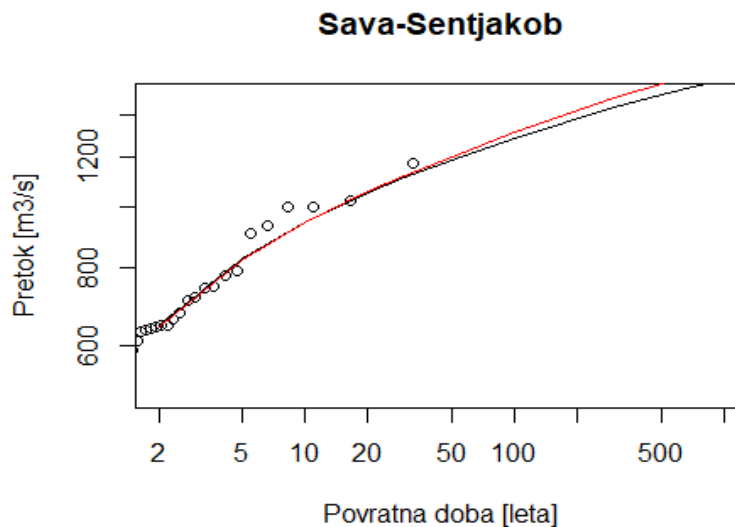
```



**Naloga 33:** S spletne strani Agencije RS za okolje prenesite podatke o največjih letnih konicah z vodomerne postaje Sava Šentjakob za obdobje 1990–2021 (lahko uporabite tudi podatke iz naloge 32). Z uporabo paketov *lmom* in *lmomco* analizirajte največje konice pretokov. Izvedite verjetnostne analize konic pretokov (GEV in Gumbelova porazdelitev).

```
# prikazan bo postopek z uporabo dnevni podatkov iz naloge 32
Sava <- read.delim("C:/Users/nbezak/OneDrive - Univerza v Ljubljani/Ucbenik/Naloga-gradivo/hid_3570_1990_2021.txt")
Sava[,1] <- as.POSIXct(strptime(Sava[,1],format="%d.%m.%Y"))
library(zoo)
QzooSava <- zoo(Sava[,3],Sava[,1])
leto1Sava <- as.numeric(floor(as.yearmon(time(QzooSava)+3600)))
letQkonSava <- aggregate(QzooSava,leto1Sava,max,na.rm=TRUE)
library(lmomco);detach(package:lfstat); detach(package:lmom)
lmomentiSava <- lmoms(as.numeric(letQkonSava))
gevparSava <- lmom2par(lmomentiSava,type="gev")
gumparSava <- lmom2par(lmomentiSava,type="gum")
Pdobe <- c(2,5,10,20,30,50,100,300,500,1000)
verj <- 1-1/Pdobe
rezgumSava <- quagum(verj,gumparSava)
rezgevSava <- quagev(verj,gevparSava)
weibullSava <- pp(as.numeric(letQkonSava),a=0)
Pdobe1Sava <- 1/(1-weibullSava)
plot(Pdobe,rezgevSava,type="l",log="xy",xlab="Povratna doba [leta]",
ylab="Pretok [m3/s]",main="Sava-Sentjakob",ylim=c(500,1500))
```

```
points(Pdobe1Sava, sort(as.numeric(1etQkonSava)))
lines(Pdobe, rezgumSava, col="red")
```



**Naloga 34: S spletne strani Agencije RS za okolje prenesite podatke o največjih letnih konicah z vodomernih postaj na reki Savi: Radovljica I, Šentjakob, Litija (in Litija I), Hrastnik, Čatež I za obdobje 2000–2021. Izvedite analize trendov za posamezne vodomerne postaje (izberite poljuben test) in tudi regionalni Mann-Kendallov test.**

```
SavaPostaje <- read.delim("C:/Users/nbezak/OneDrive - Univerza v Ljubljani/Ub  
benik/Naloge-gradivo/SavaPostaje.txt")
library(trend)
apply(SavaPostaje, 2, mk.test)
```

```
## $Leto
##
## Mann-Kendall trend test
##
## data: newX[, i]
## z = 6.4855, n = 22, p-value = 8.842e-11
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##      S      varS      tau
## 231.000 1257.667   1.000
##
##
## $Radovljica
##
## Mann-Kendall trend test
##
## data: newX[, i]
## z = 0.45117, n = 22, p-value = 0.6519
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##      S      varS      tau
```

```

## 1.700000e+01 1.257667e+03 7.359307e-02
##
##
## $Sentjakob
##
## Mann-Kendall trend test
##
## data: newX[, i]
## z = 0.22558, n = 22, p-value = 0.8215
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##           S           varS           tau
## 9.000000e+00 1.257667e+03 3.896104e-02
##
##
## $Litija
##
## Mann-Kendall trend test
##
## data: newX[, i]
## z = 0.73315, n = 22, p-value = 0.4635
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##           S           varS           tau
## 27.00000000 1257.6666667 0.1168831
##
##
## $Hrastnik
##
## Mann-Kendall trend test
##
## data: newX[, i]
## z = 0.3103, n = 22, p-value = 0.7563
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##           S           varS           tau
## 1.200000e+01 1.256667e+03 5.206086e-02
##
##
## $Catez
##
## Mann-Kendall trend test
##
## data: newX[, i]
## z = -0.16919, n = 22, p-value = 0.8656
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##           S           varS           tau
## -7.00000000 1257.6666667 -0.03030303

```

```

library(rkt)
rkt(date=rep(SavaPostaje[,1],5),y=c(SavaPostaje[,2],SavaPostaje[,3],SavaPostaje[,4],SavaPostaje[,5],SavaPostaje[,6]),block=c(rep(1,length(SavaPostaje[,1])),rep(2,length(SavaPostaje[,1])),rep(3,length(SavaPostaje[,1])),rep(4,length(SavaPostaje[,1])),rep(5,length(SavaPostaje[,1]))))

##
## Standard model
## Tau = 0.05021645
## Score = 58
## var(Score) = 6287.333
## 2-sided p-value = 0.4722299
## Theil-Sen's (MK) or seasonal/regional Kendall (SKT/RKT) slope= 2.57325

```

***Naloga 35: Na podlagi kopule Joe z vrednostjo parametra 2 generirajte bivariatni vzorec s 40 elementi. Na podlagi generiranega vzorca ocenite parametre kopule Frank in preverite njeno ustreznost za opis tega vzorca. Dodatno grafično primerjajte generirani vzorec na podlagi kopule Frank (N = 500) in kopule Joe.***

```

library(copula)

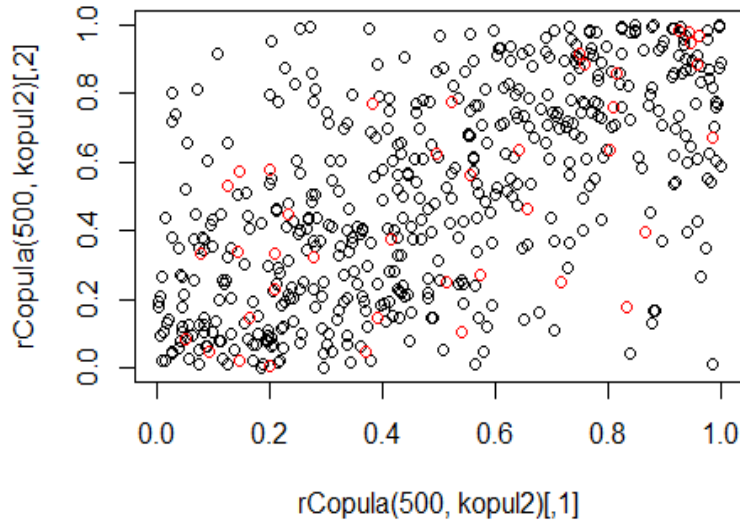
## Warning: package 'copula' was built under R version 4.1.3

gen <- rCopula(40,joeCopula(2))
gofCopula(copula=frankCopula(),gen,optim.method = "L-BFGS-B",N=500,estim.method="mpl",simulation="mult",method="Sn")

##
## Multiplier bootstrap-based goodness-of-fit test of Frank copula, dim.
## d = 2, with 'method'="Sn", 'estim.method'="mpl":
##
## data: x
## statistic = 0.033922, parameter = 4.925, p-value = 0.02894

param <- fitCopula(copula=frankCopula(),data=gen,method="mpl")
kopul2 <- frankCopula(param@estimate, dim=2)
plot(rCopula(500,kopul2))
points(gen,col="red")

```



**Naloga 36: Za podatke z ene izmed postaj ARSO izvedite analize sezonskosti: (i) izračunajte koeficient sezonskosti in ii) pripravite krožni graf, kjer je prikazan čas nastopa izbranih dogodkov. Analize ponovite za vzorec največjih letnih konic (AM) in za vzorec POT, kjer povprečno izberite tri dogodke nad izbranim pragom na leto (POT3).**

```
library(zoo)
Sava <- read.delim("C:/Users/nbezak/OneDrive - Univerza v Ljubljani/Ucbenik/Naloga-gradivo/hid_3570_1990_2021.txt")
Sava[,1] <- as.POSIXct(strptime(Sava[,1],format="%d.%m.%Y"))
QzooSava <- zoo(Sava[,3],Sava[,1])
leto1Sava <- as.numeric(floor(as.yearmon(time(QzooSava))+3600))
letQkonSava <- aggregate(QzooSava,leto1Sava,which.max)
dan <- letQkonSava
kot <- (dan-0.5)*(2*pi/365)
xpov <- mean(cos(kot));x <- cos(kot)
ypov <- mean(sin(kot));y <- sin(kot)
r <- sqrt(xpov*xpov+ypov*ypov)
if(xpov>=0 && ypov>=0) dan.rez <- (atan(ypov/xpov))*(365/(2*pi)) else
  if(xpov<0) dan.rez <- (atan(ypov/xpov) + pi)*(365/(2*pi)) else
    if (ypov<0 && xpov >= 0) dan.rez <- (atan(ypov/xpov) + 2*pi)*(365/(2*pi))
else
  end
dan.rez

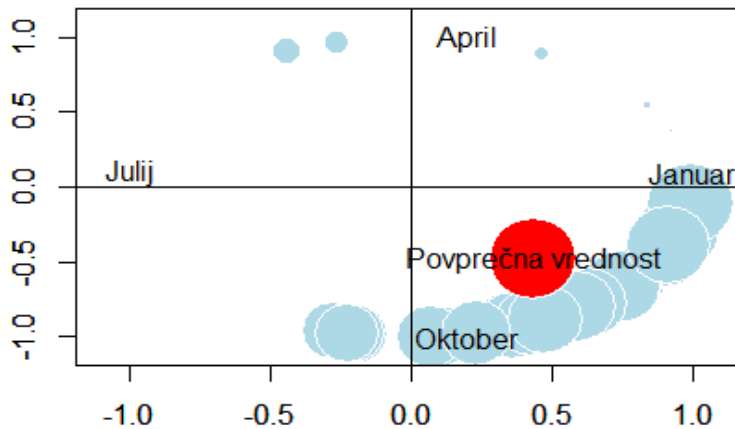
## [1] 316.1657

symbols(x,y,letQkonSava, inches=0.25,bg="lightblue",xlab="",ylab="",
xlim=c(-1.1,1.1),ylim=c(-1.1,1.1),fg="white",
main="Sezonski prikaz vseh vQvk");abline(0,0);abline(v=0)
par(new=TRUE)
symbols(xpov,ypov,mean(letQkonSava), inches=0.25,bg="red",
xlab="",ylab="",xlim=c(-1.1,1.1),ylim=c(-1.1,1.1),fg="white")
text(xpov,ypov,"Povprečna vrednost")
text(1,0.1,labels="Januar")
```



```
text(0.2,1,labels="April")
text(-1,0.1,labels="Julij")
text(0.2,-1,labels="Oktober")
```

### Sezonski prikaz vseh vQvk



**Naloga 37:** S spletne strani ARSO ([meteo.si](http://meteo.si), arhiv meritev) pridobite mesečne podatke o padavinah za obdobje 1961–2020 za postajo Murska Sobota ter izračunajte standardizirane padavinske indekse (SPI-1, SPI-3, SPI-6 in SPI-12) in jih prikažite tudi grafično.

```
murska <- read.delim("C:/Users/nbezak/OneDrive - Univerza v Ljubljani/Ucbenik
/Naloge-gradivo/MurskaS.txt",header=FALSE)
library(SPEI)

## Warning: package 'SPEI' was built under R version 4.1.3

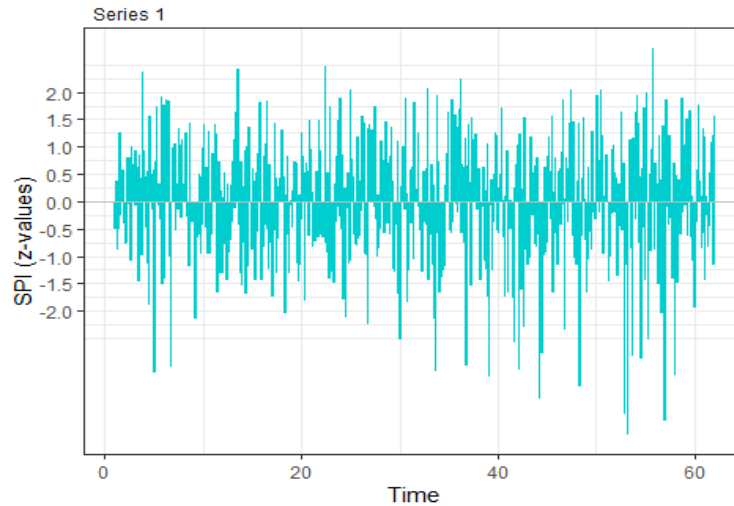
## # Package SPEI (1.8.1) loaded [try SPEINews()].

spiM1 <- spi(data=murska[,2], scale=1, distribution = "Gamma")

## [1] "Calculating the Standardized Precipitation Evapotranspiration Index (
SPEI) at a time scale of 1. Using kernel type 'rectangular', with 0 shift. Fi
tting the data to a Gamma distribution. Using the ub-pwm parameter fitting me
thod. Checking for missing values (`NA`): all the data must be complete. Usin
g the whole time series as reference period. Input type is vector. No time in
formation provided, assuming a monthly time series."

plot(spiM1)

## Warning: Removed 732 rows containing missing values (`geom_point()`).
```

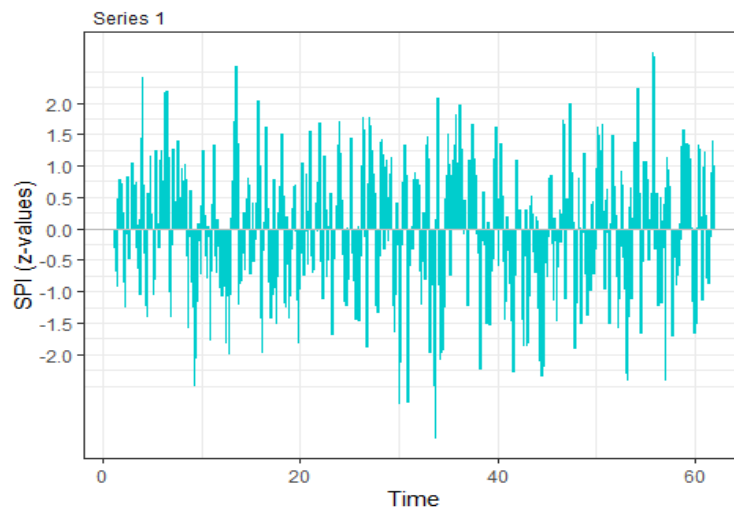


```
spiM3 <- spi(data=murska[,2], scale=3, distribution = "Gamma")
```

```
## [1] "Calculating the Standardized Precipitation Evapotranspiration Index (SPEI) at a time scale of 3. Using kernel type 'rectangular', with 0 shift. Fitting the data to a Gamma distribution. Using the ub-pwm parameter fitting method. Checking for missing values (`NA`): all the data must be complete. Using the whole time series as reference period. Input type is vector. No time information provided, assuming a monthly time series."
```

```
plot(spiM3)
```

```
## Warning: Removed 730 rows containing missing values (`geom_point()`).
```



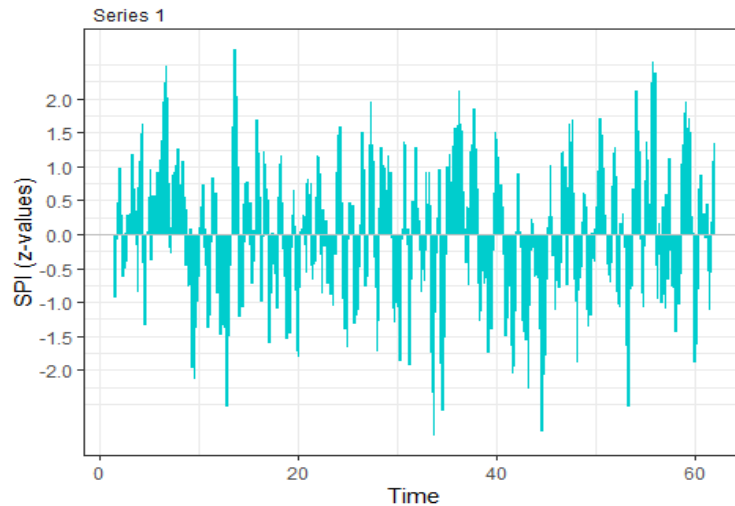
```
spiM6 <- spi(data=murska[,2], scale=6, distribution = "Gamma")
```

```
## [1] "Calculating the Standardized Precipitation Evapotranspiration Index (SPEI) at a time scale of 6. Using kernel type 'rectangular', with 0 shift. Fitting the data to a Gamma distribution. Using the ub-pwm parameter fitting method. Checking for missing values (`NA`): all the data must be complete. Using the whole time series as reference period. Input type is vector. No time information provided, assuming a monthly time series."
```

g the whole time series as reference period. Input type is vector. No time information provided, assuming a monthly time series."

```
plot(spiM6)
```

```
## Warning: Removed 727 rows containing missing values (`geom_point()`).
```

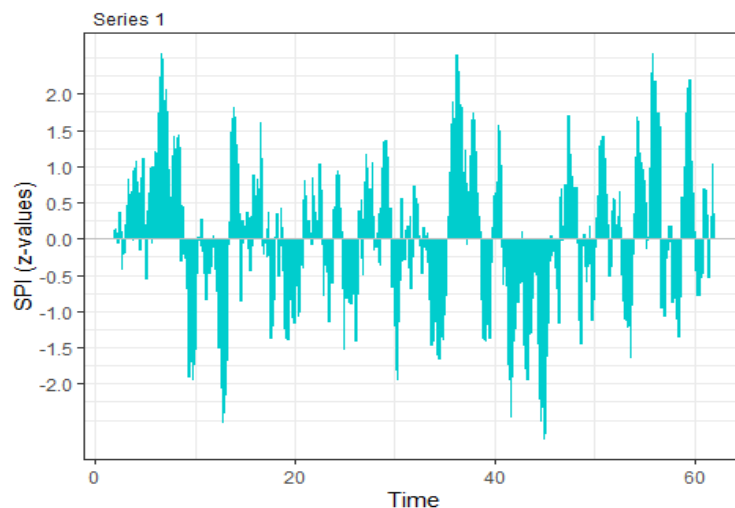


```
spiM12 <- spi(data=murska[,2], scale=12, distribution = "Gamma")
```

```
## [1] "Calculating the Standardized Precipitation Evapotranspiration Index (SPEI) at a time scale of 12. Using kernel type 'rectangular', with 0 shift. Fitting the data to a Gamma distribution. Using the ub-pwm parameter fitting method. Checking for missing values (`NA`): all the data must be complete. Using the whole time series as reference period. Input type is vector. No time information provided, assuming a monthly time series."
```

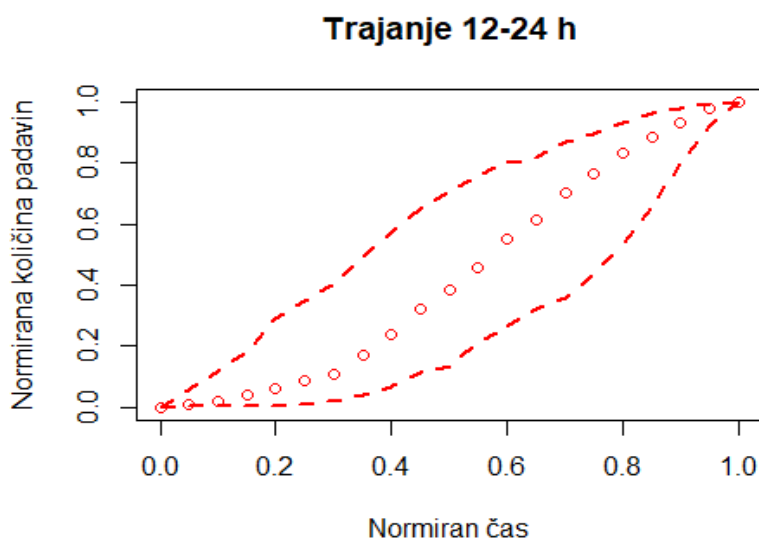
```
plot(spiM12)
```

```
## Warning: Removed 721 rows containing missing values (`geom_point()`).
```



**Naloga 38: Izdelajte Huffove krivulje za padavinske dogodke s trajanjem med 12 in 24 h ter določite tako mediano Huffovih krivulj kot intervale zaupanja (10 % in 90 % percentilni krivulji).**

```
data <- read.table("C:/Users/nbezak/OneDrive - Univerza v Ljubljani/Ucbenik/Ljubljana-padavine.txt",header=FALSE, sep=" ")
dataP <- data.frame(Datum=as.POSIXct(paste0(data[,1]," ",data[,2]),format="%Y-%m-%d %H:%M:%S"),P=((data[,3])))
load("C:/Users/nbezak/OneDrive - Univerza v Ljubljani/Ucbenik/Naloge-gradivo/erozivni.Rdata")
izbira <- which(dogodki$D_h<24 & dogodki$D_h>12)
zaporedje <- seq(from=0,to=1,by=0.05)
huffvmes <- matrix(NA,length(izbira),length(zaporedje))
for(i in 1:length(izbira)){
  padizb <- dataP[(dogodki$endIndex[izbira[i]]-2*dogodki$D_h[izbira[i]]):dogodki$endIndex[izbira[i]],2]
  vsotapad <- sum(padizb)
  for(k in 0:(length(zaporedje))){
    huffvmes[i,k] <- approx(seq(0,1,length=length(cumsum(padizb))),cumsum(padizb)/vsotapad,xout=zaporedje[k])$y
  }
}
huffvmes[,1] <- 0
huffvmes[,21] <- 1
spmeja <- apply(huffvmes,2,quantile,probs=0.1)
med <- apply(huffvmes,2,quantile,probs=0.5)
zgmeja <- apply(huffvmes,2,quantile,probs=0.9)
plot(zaporedje,med,col="red",lty=1,lwd=1,xlab="Normiran čas",ylab="Normirana količina padavin",main="Trajanje 12-24 h")
lines(zaporedje,spmeja,col="red",lty=2,lwd=2)
lines(zaporedje,zgmeja,col="red",lty=2,lwd=2)
```



**Naloga 39: S spletne strani meteo.si prenesite podatke o dnevnih padavinah za pet postaj na območju vzhodne Slovenije (za obdobje 2020–2023), definirajte stohastični**

**padavinski model in generirajte 20 realizacij padavinskih podatkov v trajanju treh let za vsako postajo.**

```
padavine <- read.table("C:/Users/nbezak/OneDrive - Univerza v Ljubljani/Ucbenik/Naloge-gradivo/padavine.txt",header=TRUE)
library(GWEX)
skupajpad <- as.matrix(padavine[,4:6])
myObsPrecpad <- GwexObs(variable='Prec',date=as.Date(padavine[,1],format="%d.%m.%Y"),obs=skupajpad)
myparPrecpad <- fitGwexModel(myObsPrecpad, listOption=list(th=0.5,nChainFit=1000))

## [1] "Fit generator"

rez <- simGwexModel(myparPrecpad,nb.rep=20,d.start = as.Date("01012025", "%d%m%Y"),
  d.end = as.Date("31122027", "%d%m%Y"))

## [1] "Generate scenarios"
```

**Naloga 40: Na podatkih z vodomerne postaje Veliko Širje (Savinja, leto 2005) ocenite uspešnost algoritma regresijskih dreves za napovedovanje koncentracije suspendiranih snovi (na podlagi podatkov o pretokih in temperaturi vode), pri tem del podatkov (okoli 80 %) uporabite za umerjanje modela, preostanek pa za preverjanje.**

```
podatki <- read.table(file="C:/Users/nbezak/OneDrive - Univerza v Ljubljani/Ucbenik/Savinja-Veliko SirjeI-2005.txt",header=TRUE,sep=";",dec=".")
set.seed(1)
izbirased <- sample.int(365,290)
kalibsed <- podatki[izbirased,]
validsed <- podatki[-izbirased,]
library(rpart)
library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 4.1.3

drevokalibsed <- rpart(vsebnost_suspendiranega_materiala.g.m3. ~ pretok.m3.s. + temp.vode.C.,data=kalibsed)
napovedsed <- predict(drevokalibsed, validsed, type = 'vector')
cor(validsed$vsebnost_suspendiranega_materiala.g.m3.,napovedsed)

## [1] 0.6130527

# vidimo, da izbrani model relativno slabo napove vrednosti koncentracije
# suspendiranih snovi na postaji Veliko Širje (Savinja)
```

**Naloga 41: Na podlagi podatkov o digitalnem modelu višin določite karto naklona z uporabo funkcije terrain v paketu terra. Izračunajte osnovno statistiko naklonov.**

```
library(terra)

## Warning: package 'terra' was built under R version 4.1.3
```

```
## terra 1.7.23

##
## Attaching package: 'terra'

## The following object is masked from 'package:zoo':
##
##      time<-

teren <- rast("C:/Users/nbezak/OneDrive - Univerza v Ljubljani/Ucbenik/GIS/So
raSuha.sdat")
naklon <- terrain(teren,v="slope") # v stopinjah
summary(naklon)

##      slope
## Min.   : 0.02
## 1st Qu.:11.10
## Median :16.39
## Mean   :16.64
## 3rd Qu.:21.74
## Max.   :47.13
## NA's   :42619
```

***Naloga 42: Izvedite umerjanje modela GR5J za obdobje 2005–2011 (z uporabo podatkov za porečje Selške Sore) in validacijo modela z uporabo umerjenih parametrov za leti 2012 in 2013. Prikažite rezultate validacije modela.***

```
library(airGR)
data <- read.table("C:/Users/nbezak/OneDrive - Univerza v Ljubljani/Ucbenik/a
irGR/Data-Example.txt",header=T)
potET <- PE_Oudin(JD = as.POSIXlt(strptime(data[,1], "%d.%m.%Y"))$yday, Temp
= data[,5], Lat = 46.2, LatUnit = "deg")
InputsModel <- CreateInputsModel(FUN_MOD = RunModel_GR5J, DatesR = as.POSIXct
(strptime(data[,1], "%d.%m.%Y")), Precip = data[,4], PotEvap = potET)
konec <- which(format(data[,1], format = "%d.%m.%Y")==="31.12.2011")
Ind_Run <- seq(366, konec)
RunOptions <- CreateRunOptions(FUN_MOD = RunModel_GR5J, InputsModel = InputsM
odel, IndPeriod_Run = Ind_Run, IndPeriod_WarmUp = 1:365)
InputsCrit <- CreateInputsCrit(FUN_CRIT = ErrorCrit_NSE, InputsModel = Inputs
Model, RunOptions = RunOptions, Obs = data[366:konec,3])
CalibOptions <- CreateCalibOptions(FUN_MOD = RunModel_GR5J, FUN_CALIB = Calib
ration_Michel)
OutputsCalib <- Calibration_Michel(InputsModel = InputsModel, RunOptions = Ru
nOptions, InputsCrit = InputsCrit, CalibOptions = CalibOptions, FUN_MOD = Run
Model_GR5J)

## Grid-Screening in progress (0% 20% 40% 60% 80% 100%)
##   Screening completed (243 runs)
##       Param = 445.858,   -1.386,   83.931,   1.105,   0.467
##       Crit. NSE[Q]      = 0.4626
## Steepest-descent local search in progress
```

```

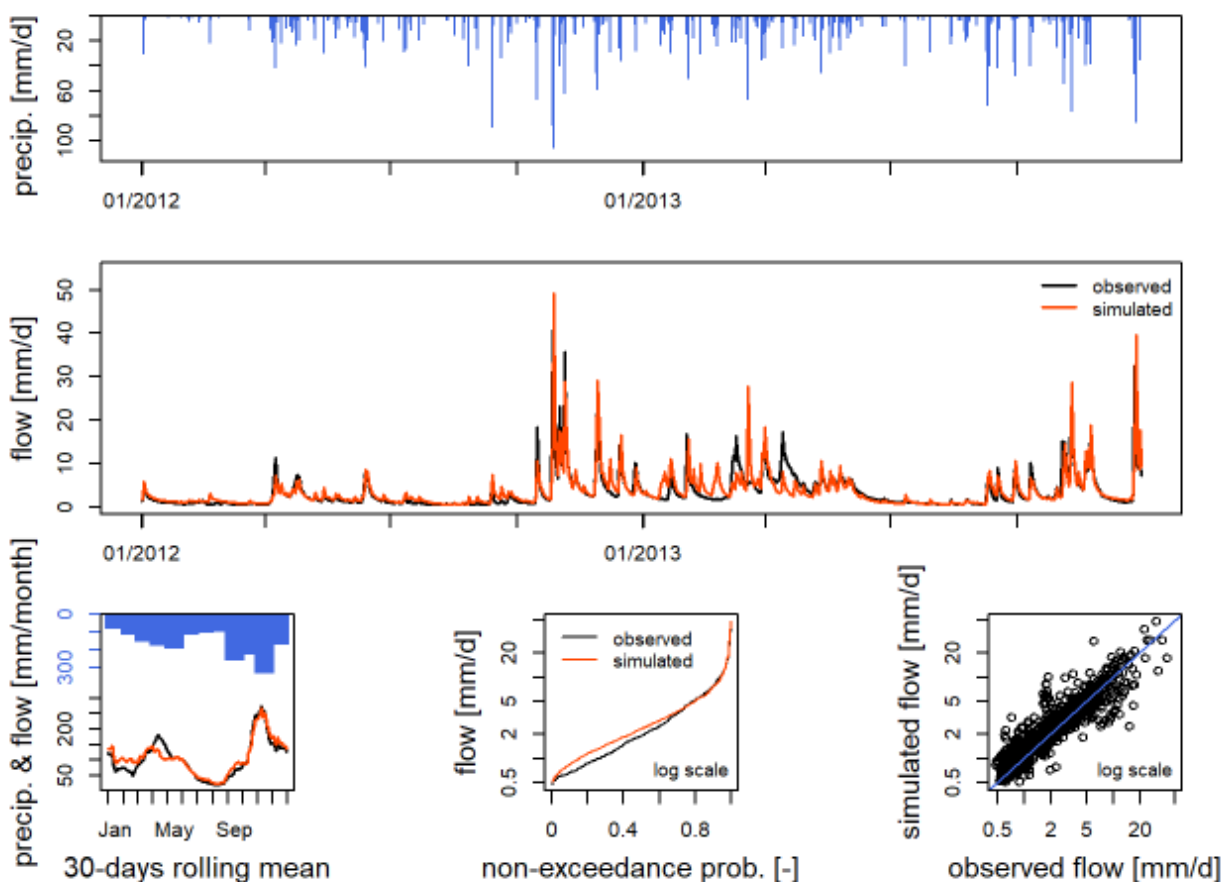
## Calibration completed (62 iterations, 797 runs)
## Param = 660.254, -0.664, 117.290, 0.500, 0.430
## Crit. NSE[Q] = 0.7406

ParamGR5J <- OutputsCalib$ParamFinalR
konec2 <- which(format(data[,1], format = "%d.%m.%Y")=="31.12.2013")
Ind_Run1 <- seq((konec+1),konec2)
RunOptions1 <- CreateRunOptions(FUN_MOD = RunModel_GR5J, InputsModel = Inputs
Model, IndPeriod_Run = Ind_Run1)

## Warning in CreateRunOptions(FUN_MOD = RunModel_GR5J, InputsModel = InputsM
odel, : model warm up period not defined: default configuration used
## the year preceding the run period is used

OutputsModel1 <- RunModel_GR5J(InputsModel = InputsModel, RunOptions = RunOpt
ions1, Param = ParamGR5J)
plot(OutputsModel1, Qobs = data[Ind_Run1,3])

```

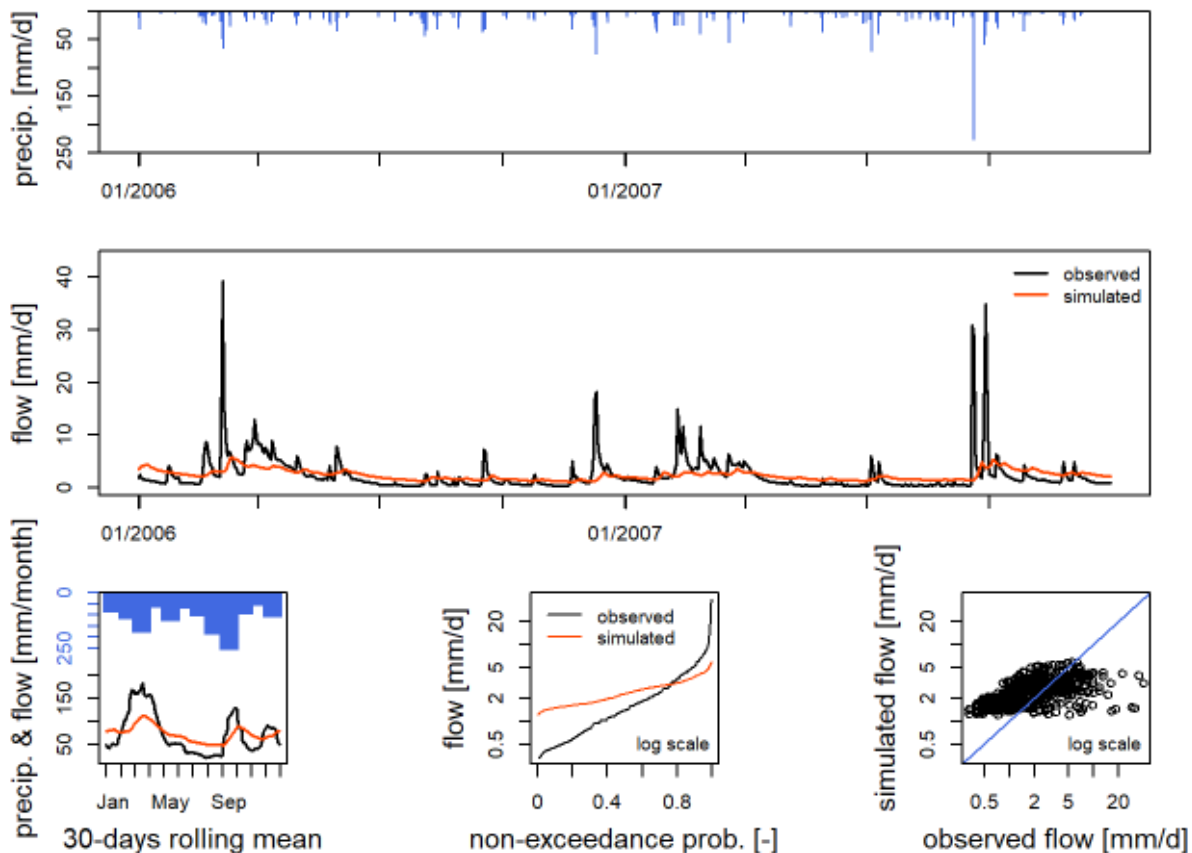


**Naloga 43: Zaženite model GR4J za obdobje 2006–2007 s podatki s porečja Selške Sore in z naključno generiranimi parametri modela (neumerjenimi), pri čemer morate pred naključnim generiranjem štirih števil uporabiti funkcijo `set.seed(15)`. Razponi parametrov so:  $x_1$ : 0–2500 mm,  $x_2$ : -5→5 mm;  $x_3$ : 0–1000 mm;  $x_4$ : 0–10 dni. Uporabite enakomerno porazdelitev. Model zaženite in primerjajte rezultate z meritvami.**

```

library(airGR)
data <- read.table("C:/Users/nbezak/OneDrive - Univerza v Ljubljani/Ucbenik/a
irGR/Data-Example.txt",header=T)
potET <- PE_Oudin(JD = as.POSIXlt(strptime(data[,1], "%d.%m.%Y"))$yday, Temp
= data[,5], Lat = 46.2, LatUnit = "deg")
InputsModel <- CreateInputsModel(FUN_MOD = RunModel_GR4J, DatesR = as.POSIXct
(strptime(data[,1], "%d.%m.%Y")), Precip = data[,4], PotEvap = potET)
konec <- which(format(data[,1], format = "%d.%m.%Y")==="31.12.2007")
Ind_Run <- seq(366, konec)
RunOptions <- CreateRunOptions(FUN_MOD = RunModel_GR4J, InputsModel = InputsM
odel, IndPeriod_Run = Ind_Run, IndPeriod_WarmUp = 1:365)
set.seed(15)
ParamGR4J <- c(runif(1,0,2500),runif(1,-5,5),runif(1,0,1000),runif(1,0,10))
OutputsModel <- RunModel_GR4J(InputsModel = InputsModel, RunOptions = RunOpti
ons, Param = ParamGR4J)
plot(OutputsModel, Qobs = data[Ind_Run,3])

```



**Naloga 44: Zaženite model TUWmodel še za obdobje validacije (2011–2016) ter rezultate prikažite grafično in izračunajte kriterij NSE.**

```

library(TUWmodel)
library(hydroGOF)

## Warning: package 'hydroGOF' was built under R version 4.1.3

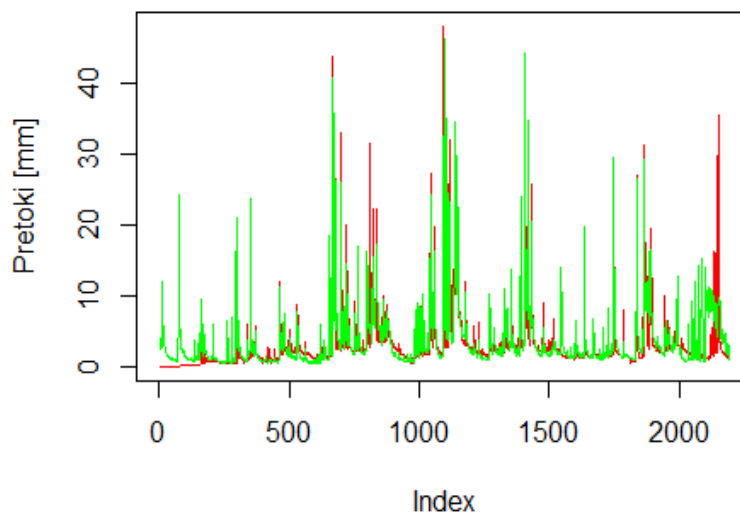
```



```

data <- read.table("C:/Users/nbezak/OneDrive - Univerza v Ljubljani/Ucbenik/a
irGR/Data-Example.txt",header=T)
preob1 <- data.frame(DatesR=as.POSIXct(strptime(data[,1], "%d.%m.%Y"),tz="UTC
"),P=data[,4], E=potET,Qmm=data[,3],T=data[,5])
load("C:/Users/nbezak/OneDrive - Univerza v Ljubljani/Ucbenik/Naloge-gradivo/
TUWmodelpar.Rdata")
zac <- which(format(data[,1], format = "%d.%m.%Y")==="31.12.2010")
zac <- zac+1
kon <- which(format(data[,1], format = "%d.%m.%Y")==="31.12.2016")
sim2 <- TUWmodel(prec=preob1[zac:kon,2], airt=preob1[zac:kon,5], ep=preob1[za
c:kon,3], area=1, param=calibrate_period1$optim$bestmem)
plot(as.numeric(sim2$q),type="l", col="red",ylab="Pretoki [mm]")
lines(preob1[zac:kon,4], col="green")

```



```
NSE(as.numeric(sim2$q),preob1[zac:kon,4])
```

```
## [1] 0.5844474
```

**Naloga 45: Z modelom TUWien izračunajte vrednosti pretokov (za isto obdobje kot pri nalogi 44) za različne kombinacije parametrov, ki so bile uporabljene med optimizacijo oziroma umerjanjem modela. Rezultate prikazite grafično.**

```

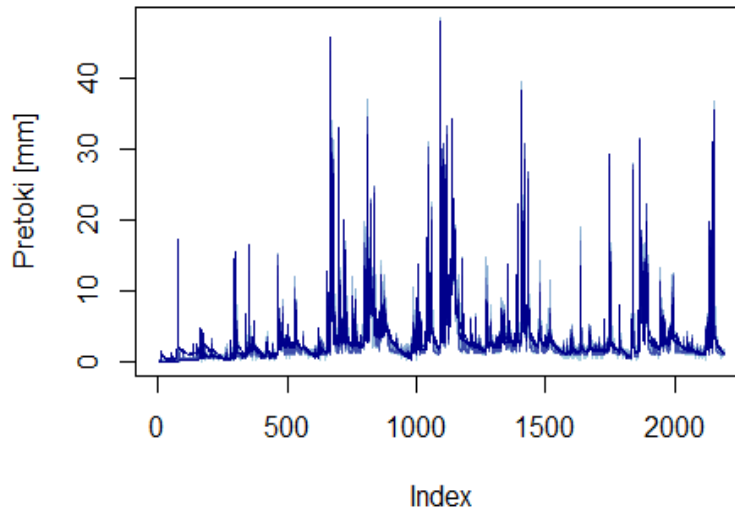
library(TUWmodel)
data <- read.table("C:/Users/nbezak/OneDrive - Univerza v Ljubljani/Ucbenik/a
irGR/Data-Example.txt",header=T)
preob1 <- data.frame(DatesR=as.POSIXct(strptime(data[,1], "%d.%m.%Y"),tz="UTC
"),P=data[,4], E=potET,Qmm=data[,3],T=data[,5])
load("C:/Users/nbezak/OneDrive - Univerza v Ljubljani/Ucbenik/Naloge-gradivo/
TUWmodelpar.Rdata")
zac <- which(format(data[,1], format = "%d.%m.%Y")==="31.12.2010")
zac <- zac+1
kon <- which(format(data[,1], format = "%d.%m.%Y")==="31.12.2016")
sim2 <- TUWmodel(prec=preob1[zac:kon,2], airt=preob1[zac:kon,5], ep=preob1[za
c:kon,3], area=1, param=calibrate_period1$optim$bestmem)
plot(as.numeric(sim2$q),type="l", col="red",ylab="Pretoki [mm]")

```

```

for(g in 1:dim(calibrate_period1$member$bestmemit)[1]){
sim3 <- TUWmodel(prec=preob1[zac:kon,2], airt=preob1[zac:kon,5], ep=preob1[za
c:kon,3], area=1, param=calibrate_period1$member$bestmemit[g,])
lines(as.numeric(sim3$q), col=colorRampPalette(c("lightblue", "darkblue"), al
pha = TRUE)(60)[g])
}

```



**Naloga 46:** Na podlagi simulacij CMIP6 grafično prikažite mesečno dinamiko v padavinah v Sloveniji in okolici ( $lon=c(13,17),lat=c(45,47)$ ) za obdobje 2061–2080 z uporabo modela CMCC-ESM2.

```

library(geodata, quietly=TRUE)
cmip6dod <- cmip6_tile(model="CMCC-ESM2", lon=c(13,17),lat=c(45,47), ssp="245
", time="2061-2080", var="prec", res=10, path=tempdir())
plot(cmip6dod)
plot(mean(cmip6dod))

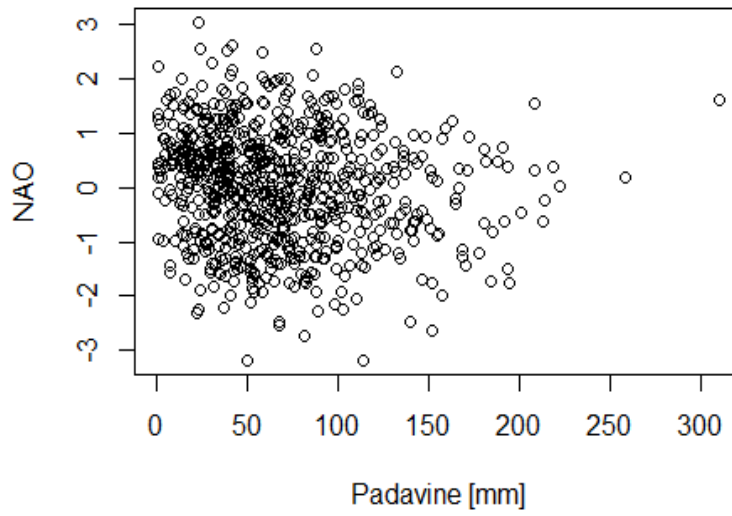
```

**Naloga 47:** S spletne strani ARSO prenesite podatke o mesečnih padavinah za postajo Murska Sobota, ki ste jih uporabili pri nalogi 37, in analizirajte, ali obstaja povezava med indeksom NAO in padavinami za izbrano postajo (za izbrano obdobje, glede na razpoložljivost podatkov ARSO).

```

library(rsos, quietly=TRUE)
nao <- download_nao()
murska <- read.delim("C:/Users/nbezak/OneDrive - Univerza v Ljubljani/Ucbenik
/Naloge-gradivo/MurskaS.txt",header=FALSE)
plot(murska[,2],as.numeric(unlist(nao[121:852,3])),xlab="Padavine [mm]",ylab=
"NAO")

```



```
cor(murska[,2],as.numeric(unlist(nao[121:852,3])))
```

```
## [1] -0.1243624
```