

```

# BOOTSTRAP
setwd("/Users/shirin/Desktop/Semi")
# sort_by_fibo ist eine Funktion, welche beim Aufruf zwei Übergabeparameter
erwartet. Diese beiden parameter werden in die Variablen table und search_for
# gespeichert.
main <- function(){
  lotto = read_file_csv2("data_lotto_number.csv") # ok
  length_source = nrow(lotto) # ok
  number_new_draw = 4294 # ok
  runs = 0 # ok
  lotto_new = create_new_lotto (runs,number_new_draw, length_source, lotto)
  main_2()
}

# Funktion namens "create_new_draw" (Neuen Zug erstellen) enthält zwei Variablen
(random_max & table) -> neue Tabelle wird erstellt
# Bedingung: (6 Durchläufe) r1 = Zahl, wert = Zufällige Ziehung einer Zeile von
allen Zeilen im Datensatz.
# r1 wird als neue Variable für den neuen Datensatz festgelegt. Ist die Schleife
z.B. im Durchlauf Nr. 1, so wird die zufällige
# Zeile mit dem Spaltenwert 1 ausgewählt. Da eine Lottoziehung 6 Zahlen beinhaltet
wird der Vorgang mit jeder Spalte durchgeführt.
# Durch Ziehen mit Zurücklegen wird so ein Bootstrap simuliert. Jeder Zahlenwert
wird in der Tabelle generiert und neue Ziehungen werden so simuliert.

create_new_draw <- function(random_max, table){
  table_new = c()
  for (i in 1:6){
    r1 = integer()
    wert = floor(runif(1, min=1, max=random_max))
    r1 = c(table[wert,i])
    table_new[i] <- r1
  }
  return(table_new) # ok
}

# Funktion namens "create_new_lotto" erstellt einen neuen Datensatz. Dieser greift
auf die Funktion "create_new_draw" zu.
# Funktion besteht aus 4 Variablen. Ein neuer Datensatz in Form einer Matrix wird
generiert (noch leer).
# while Schleife: wird solange durchgeführt, bis runs/count <= Anzahl der
Durchgänge ist. Diese greift auf die Funktion
#" create_new_draw zu, wobei dieser die Werte length_source und lotto zugewiesen
werden.

create_new_lotto <- function(count,number_new_draw,random_max, table){
  #m = matrix( c(t), nrow=count, ncol=6,)
  lotto_new = matrix()
  while (count <= number_new_draw) {
    new_draw = create_new_draw(random_max,table)
    if(!(anyDuplicated(new_draw))){
      FF <- as.matrix(t(new_draw))
      write.table(FF, file = "bootstrap_data_lotto_number.csv", sep = ";",
        col.names = FALSE,row.names = FALSE, append=TRUE)
      count = count+1
    }
  }
  return(lotto_new)
}

read_file_csv2 <- function(filename){

```

```

    file_content <- read.csv2(filename)
    return(file_content)
}

#test_lotto <- function(){
  # lotto = read_file("lotto.txt")
  #length_source <- nrow(lotto)
  #x = create_new_draw(length_source, lotto)
  #x = create_new_lotto(1,1000,length_source,lotto)
  #return(x)
#}

#t = test_lotto()
#t
#print(t)
#m = matrix( c(t), nrow=1, ncol=6,)
#print(m)
#Tage neu ziehen und an Tabelle anhängen

create_new_date <- function(spalte,table,ausgang){
  spalt <- table %>% select(spalte)
  #print(nrow(spalt))
  TT <- sample(spalt[0:4296,])
  #print(TT)
  write.table(TT, file = "bootstrap_data_lotto_date.csv", sep = ";",
              col.names = FALSE,row.names = FALSE, append=TRUE)
  new_date <- read.csv2("bootstrap_data_lotto_date.csv")
  #print(nrow(new_date))
  ausgang["Wochentag"] <- new_date
  write.table(ausgang, file = "bootstrap_data_lotto.csv", sep = ";",
              col.names = c("GewZahl_1",
"GewZahl_2", "GewZahl_3", "GewZahl_4", "GewZahl_5", "GewZahl_6", "Wochentag")
              ,row.names = FALSE, append=TRUE)
}

main_2 <- function(){
  #number_new_draw = 4297
  Tag <- read.csv2("data_lotto_date.csv")
  ausgang <- read.csv2("bootstrap_data_lotto_number.csv", header = FALSE)
  #print(nrow(ausgang))
  create_new_date(7,Tag,ausgang)
}

# Ziehungen ausführen
main()

# Dateipfad festlegen
#setwd("/Users/shisha/Dropbox/FOM/Semester_4/WissMethoden_quantitativeDatenanalyse")

#setwd("H:/R")

# CSV Datei auswählen
data_lotto <- read.csv2("bootstrap_data_lotto.csv")
# Liste von Zahlen, nach welchen gesucht und sortiert werden soll
search_list = list(1,2,21,34)
# Zufallsgenerator festlegen
#set.seed(1896)

```

```

# sort_by_fibo ist eine Funktion, welche beim Aufruf zwei Übergabeparameter
erwartet. Diese beiden parameter werden in die Variablen table und search_for
# gespeichert.
sort_by_fibo <- function(table, search_for){
  # leange ist eine variable, welche die laenge der variable table mittels der
funktion nrow() ermittelt und speichert
  laenge = nrow(table)
  # eine for-schleife, welche so oft durchläuft, wie die variable table Zeilen
hat. Die Anzahl der Durchläufe wird mit der Variable laenge bestimmt
  for (i in 1:laenge){
    # wert speichert die Zeile der Tabelle table. Es wird die jeweilige Zeile
gespeichert, in welchem Durchlauf wir uns befinden. Erster Durchlauf = erste Zeile
    # zweiter Durchlauf = Zweite Zeile, 50. Durchlauf = 50. Zeile usw. i ist der
aktuelle Durchlaufwert
    wert = table[i,]
    # eine for-schleife, welche so oft durch läuft, bis die Anzahl an Spalten
erreicht ist, welche die in wert ermittelte Zeile hat. Eine zeile besteht normal
auf 6
    # Spalten, für jede Lottozahl der Ziehung. Sie kann aber auch ein datum
enthalten, oder eine 7. Superzahl.
    for (j in 1:ncol(wert)){
      # zelle speichert den Inhalt der Spalte der Zeile, abhängig davon in welchem
Durchlauf sich die Schleife befindet. 1. Durchlauf Inhalt 1.Spalte der Zeile.
      # 6. Durchlauf Inhalt der 6. Spalte der Zeile
      zelle = wert[,j]
      # Ruft die Funktion check_fobi auf und übergibt ihr 3 Werte
      # Inhalt der aktuellen Zelle (zelle), die aktuelle Zeile (wert), und die
Liste mit Zahlen, nach welchen gesucht werden soll (search_for)
      check_fobi(zelle, wert, search_for)
    }
  }
}

# check_fobi ist eine Funktion, welche drei Werte als Übergabeparameter erwartet,
eine Zahl, eine Zeile und eine Liste an Einträgen, die mit der Zahl verglichen
# werden soll.
check_fobi <- function(zahl, zeile1, search_for){
  # if-Abfrage, welche überprüft, ob die gegeben Zahl (zahl) in der Liste an
Zahlen (search_for) enthalten ist. Wenn dem so ist, wird der Inhalt der If-Abfrage
# ausgeführt.
  if (zahl %in% search_for){
    # ruft die Funktion check() auf und übergibt ihr zwei Werte, den inhalt der
Variable zeile1, welches eine Zeile mit 6 Zellen ist und die variable zahl,
welches
    # eine der 6 Zellen ist.
    check(zeile1,zahl)
  }
}

# check ist eine funktion, welche zwei werte als Übergabeparameter erwartet,
zeile2, welches eine ganze Zeile ist, und zelle, welches der Wert einer Zelle der
Zeile ist
check <- function(zeile2, zelle){
  # filename ist der Name der Datei, in welche die Zeile geschrieben wird, welche
in der Variable zeile2 gespeichert ist. Die variable filename wird zusammen
gesetzt
  # aus der Zahl in der Variable zelle und dem String "_check_output.csv". Wenn
die Zelle eine 3 ist enthält die Variable filename den Wert "3_check_output.csv".
  filename = paste(zelle, "_Bootstrap_Sort_by_Fibo.csv",sep="")
  # write.table ist eine funktion, welche einen Wert (zeile2) in eine Datei
schreibt (filename), als Trennzeichen (seperator) wird das Zeichen ";" mitgegeben.
  # außerdem wird festgelegt, dass es keine Zeilenamen und keine Spaltennamen
gibt. Das was geschrieben wird (zeile2) wird an das bestehene File angehängt

```

```

(append = TRUE)
  write.table(zeile2, file = filename, sep = ";",
              col.names = FALSE, row.names = FALSE, append=TRUE)
}

# startet das Programm durch den Aufruf der Sort_by_fibo Funktion. Übergibt der
# Funktion die beiden zuvor festgelegten Variablen data_lotto und
# search_list.
n = sort_by_fibo(data_lotto, search_list)

# Dateipfad festlegen
#setwd("/Users/shisha/Dropbox/FOM/Semester_4/WissMethoden_quantitativeDatenanalyse")

#Gesamt-Lottoziehung einlesen
data_lotto_total <- read.csv2("bootstrap_data_lotto.csv")
datalength <- nrow(data_lotto_total)

#Listen einlesen
Fibo1 <- read.csv2("1_Bootstrap_Sort_by_Fibo.csv")
Fibo1 <- rbind(Fibo1)
Fibo2 <- read.csv2("2_Bootstrap_Sort_by_Fibo.csv")
Fibo2 <- rbind(Fibo2)
Fibo21 <- read.csv2("21_Bootstrap_Sort_by_Fibo.csv")
Fibo21 <- rbind(Fibo21)
Fibo34 <- read.csv2("34_Bootstrap_Sort_by_Fibo.csv")
Fibo34 <- rbind(Fibo34)
colnames(Fibo1) <- c("GewZahl_1",
"GewZahl_2", "GewZahl_3", "GewZahl_4", "GewZahl_5", "GewZahl_6", "Wochentag")
colnames(Fibo2) <- c("GewZahl_1",
"GewZahl_2", "GewZahl_3", "GewZahl_4", "GewZahl_5", "GewZahl_6", "Wochentag")
colnames(Fibo21) <- c("GewZahl_1",
"GewZahl_2", "GewZahl_3", "GewZahl_4", "GewZahl_5", "GewZahl_6", "Wochentag")
colnames(Fibo34) <- c("GewZahl_1",
"GewZahl_2", "GewZahl_3", "GewZahl_4", "GewZahl_5", "GewZahl_6", "Wochentag")

#Anteile von am häufigsten gemeinsam gezogenen Fibozahlen bestimmen.
quota <- function(data1,data2,quelle,filename1) {
  #Fibo x mit y [1]
  total_row <- nrow(quelle)
  test <- rbind(data1,data2)
  result <- which(duplicated(test))
  IDs <- c(result)
  ID_data <- test[IDs,]
  Sa <- ID_data %>% filter(Wochentag == "Samstag")
  Mi <- ID_data %>% filter(Wochentag == "Mittwoch")
  An <- ID_data %>% filter(Wochentag != "Samstag") %>% filter(Wochentag !=
"Mittwoch")
  Sam <- nrow(Sa)
  Mit <- nrow(Mi)
  And <- nrow(An)
  length <- length(result)
  quota <- (length)/(total_row)
  total_data <- data.frame(quota)
  leng <- data.frame(length)
  #print(quota)
  quota <- c(total_data,leng, Mit, Sam, And)
  write.table(quota, file = filename1, sep = ";",
              col.names = FALSE, row.names = FALSE, append=TRUE)
  #colnames("Anteil", "Ziehungsanzahl", "Ziehungen Mittwoch", "Ziehungen
Samstag", "Ziehungen andere Tage")
}

```

```
filename1 = "Bootstrap_Quota_Fibo_1_2.csv"
Fibo_1_2 <- quota(Fibo1,Fibo2,data_lotto_total,filename1)
filename2="Bootstrap_Quota_Fibo_21_34.csv"
Fibo_21_34 <- quota(Fibo21,Fibo34,data_lotto_total,filename2)

remove_bootstrap <- function(){
  file.remove("bootstrap_data_lotto_date.csv")
  file.remove("bootstrap_data_lotto_number.csv")
  file.remove("bootstrap_data_lotto.csv")
  file.remove("1_Bootstrap_Sort_by_Fibo.csv")
  file.remove("2_Bootstrap_Sort_by_Fibo.csv")
  file.remove("21_Bootstrap_Sort_by_Fibo.csv")
  file.remove("34_Bootstrap_Sort_by_Fibo.csv")
}

remove_bootstrap()
```