

```

#Permutation
# Dateipfad festlegen
setwd("/Users/shirin/Desktop/Semi")

date_list =
c("Montag", "Dienstag", "Mittwoch", "Donnerstag", "Samstag", "Sonntag")
prob_days <- read.csv2("data_otto_date.csv")
colnames(prob_days) <- c("GewZahl_1",
"GewZahl_2", "GewZahl_3", "GewZahl_4", "GewZahl_5", "GewZahl_6", "Wochentag")

quota_day <- function(day, table, colu){
  prob_day <- table %>% filter(table[,colu] == day)
  quota_day <- nrow(prob_day)/nrow(table)
}

prob_monday = (quota_day("Montag", prob_days, 7))
prob_tuesday = quota_day("Dienstag", prob_days, 7)
prob_wednesday = quota_day("Mittwoch", prob_days, 7)
prob_thursday = quota_day("Donnerstag", prob_days, 7)
prob_friday = quota_day("Freitag", prob_days, 7)
prob_saturday = quota_day("Samstag", prob_days, 7)
prob_sunday = quota_day("Sonntag", prob_days, 7)

probs = c(prob_monday, prob_tuesday, prob_wednesday, prob_thursday,
prob_saturday, prob_sunday)

try1 = nrow(prob_days)

main_3 <- function(date_list, try1, probs){
  lotto_prob <- do(try1)*sample(1:49, 6, replace = FALSE)
  lotto_prob_date <- do(try1)*sample(date_list, 1, replace =
FALSE, prob=probs)
  #print(lotto_prob_date)
  #is.data.frame(lotto_prob)
  colnames(lotto_prob) <- c("GewZahl_1",
"GewZahl_2", "GewZahl_3", "GewZahl_4", "GewZahl_5", "GewZahl_6")
  colnames(lotto_prob_date) <- c("Wochentag")
  lotto_prob["Wochentag"] <- lotto_prob_date
  write.table(lotto_prob, file = "permutation_data_otto.csv", sep =
";",
              col.names = TRUE, row.names = FALSE, append=TRUE)
}

main_3(date_list, try1, probs)

#Sort by Fibo

# CSV Datei auswählen
data_otto <- read.csv2("permutation_data_otto.csv")
# Liste von Zahlen, nach welchen gesucht und sortiert werden soll
search_list = list(1, 2, 21, 34)

# 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, "_Permutation_Sort_by_Fibo.csv",sep="")
  # write.table ist eine funktion, welche einen Wert (zeile2) in eine
  Datei schreibt (filename), als Trennzeichen (separator) 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)

#Anteilsvergleich

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

#Listen einlesen
Fibo1 <- read.csv2("1_Permutation_Sort_by_Fibo.csv")
Fibo1 <- rbind(Fibo1)
Fibo2 <- read.csv2("2_Permutation_Sort_by_Fibo.csv")
Fibo2 <- rbind(Fibo2)
Fibo21 <- read.csv2("21_Permutation_Sort_by_Fibo.csv")
Fibo21 <- rbind(Fibo21)
Fibo34 <- read.csv2("34_Permutation_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 = "Permutation_Quota_Fibo_1_2.csv"
Fibo_1_2 <- quota(Fibo1,Fibo2,data_lotto_total,filename1)
filename2="Permutation_Quota_Fibo_21_34.csv"
Fibo_21_34 <- quota(Fibo21,Fibo34,data_lotto_total,filename2)

remove_permutation <- function(){
  file.remove("permutation_data_lotto.csv")
  file.remove("1_Permutation_Sort_by_Fibo.csv")
  file.remove("2_Permutation_Sort_by_Fibo.csv")
  file.remove("21_Permutation_Sort_by_Fibo.csv")
  file.remove("34_Permutation_Sort_by_Fibo.csv")
}

remove_permutation()

```