lab5

Hugo Morvan, Daniele Bozzoli

2023-12-05

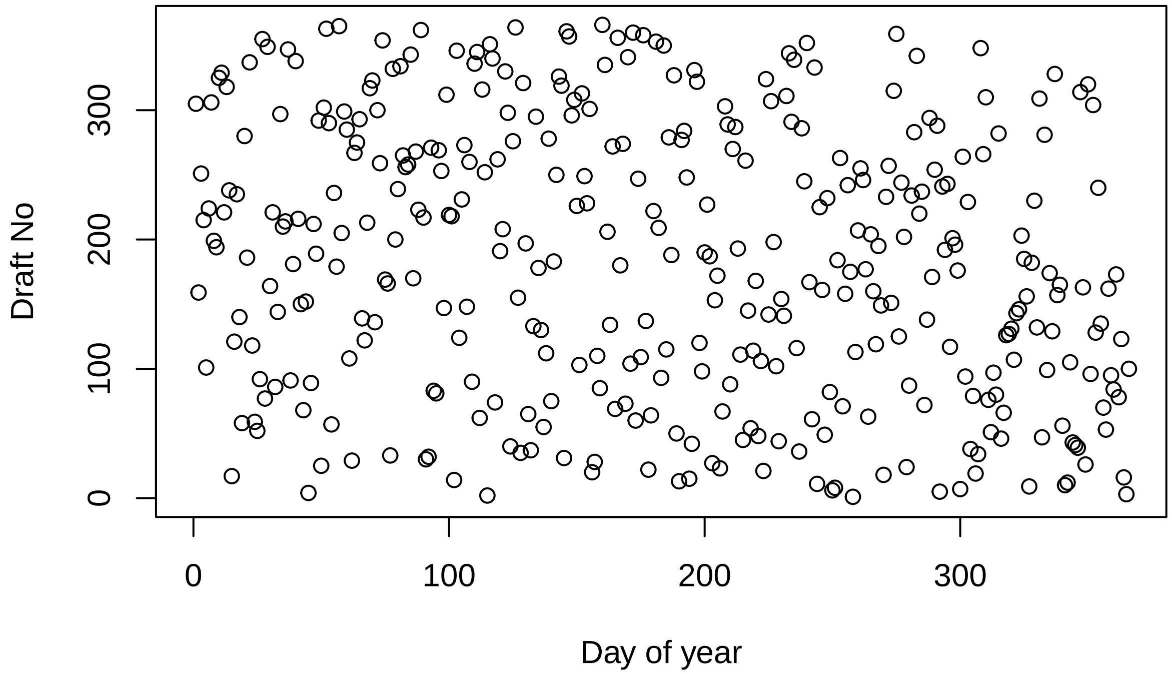
Question 1: Hypothesis testing

In 1970, the US Congress instituted a random selection process for the military draft. All 366 possible birth dates were placed in plastic capsules in a rotating drum and were selected one by one. The first date drawn from the drum received draft number one, the second date drawn received draft number two, etc. Then, eligible men were drafted in the order given by the draft number of their birth date. In a truly random lottery there should be no relationship between the date and the draft number. Your task is to investigate whether there can be doubts concerning the randomness of the selection of the draft numbers. The draft numbers (Y=Draft_No) sorted by day of year (X=Day_of_year) are given in the file lottery.csv. The data was originally published by the U.S. Government, and most conveniently made available online at http://jse.amstat.org/jse_data_archive.htm (see also Starr Norton (1997) Nonrandom Risk: The 1970 Draft Lottery, Journal of Statistics Education, 5:2, DOI: 10.1080/10691898.1997.11910534)

1.

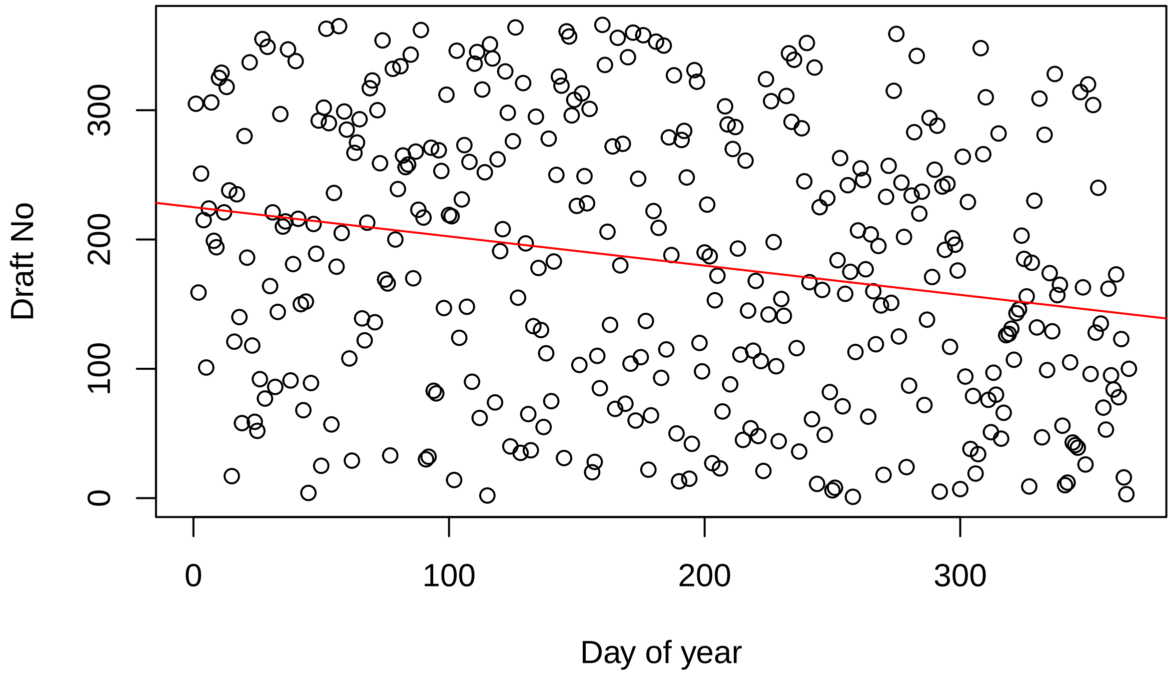
Create a scatterplot of Y versus X, are any patterns visible?

```
##       Day           Month             Mo.Number       Day_of_year    
##  Min.   : 1.00   Length:366         Min.   : 1.000   Min.   :  1.00  
##  1st Qu.: 8.00   Class :character   1st Qu.: 4.000   1st Qu.: 92.25  
##  Median :16.00   Mode  :character   Median : 7.000   Median :183.50  
##  Mean   :15.76                      Mean   : 6.514   Mean   :183.50  
##  3rd Qu.:23.00                      3rd Qu.: 9.750   3rd Qu.:274.75  
##  Max.   :31.00                      Max.   :12.000   Max.   :366.00  
##     Draft_No     
##  Min.   :  1.00  
##  1st Qu.: 92.25  
##  Median :183.50  
##  Mean   :183.53  
##  3rd Qu.:274.75  
##  Max.   :366.00  
```

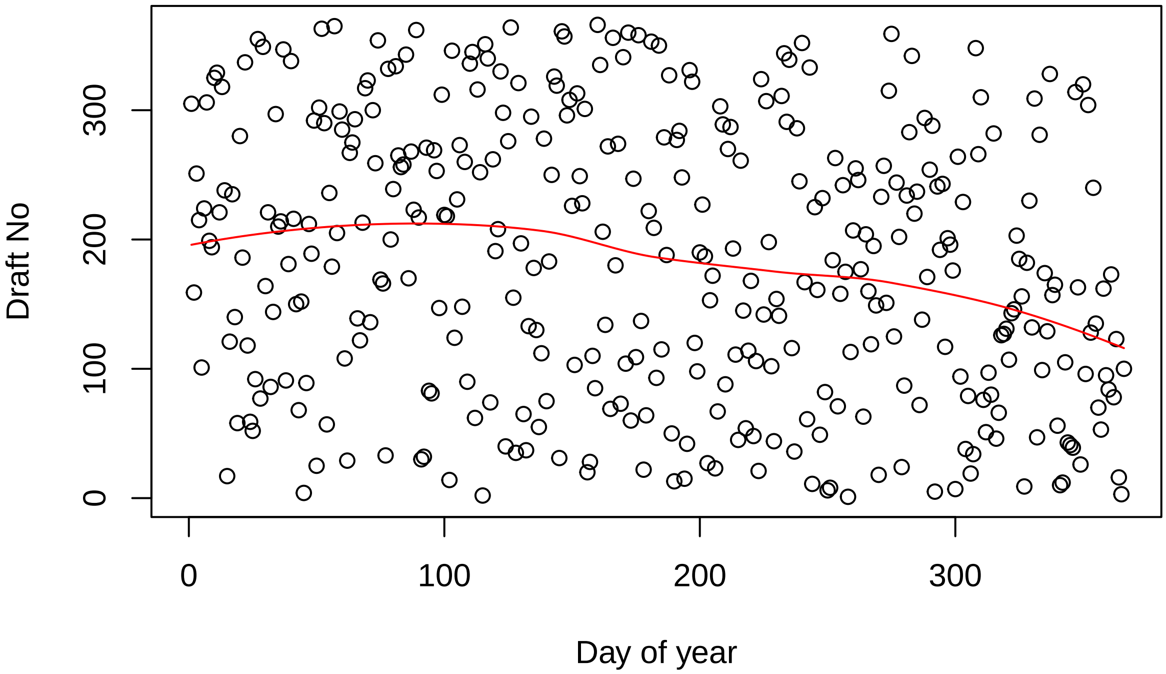


There is no clear pattern visible in this data. ## 2. Fit a curve to the data. First fit an ordinary linear model and then fit and then one using loess(). Do these curves suggest that the lottery is random? Explore how the resulting estimated curves are encoded and whether it is possible to identify which parameters are responsible for non–randomness.

```
##
## Call:
## lm(formula = Draft_No ~ Day_of_year, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -210.837  -85.629   -0.519   84.612  196.157
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 225.00922   10.81197  20.811  < 2e-16 ***
## Day_of_year  -0.22606    0.05106  -4.427 1.26e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 103.2 on 364 degrees of freedom
## Multiple R-squared:  0.05109,    Adjusted R-squared:  0.04849
## F-statistic:  19.6 on 1 and 364 DF,  p-value: 1.264e-05
```



```
## Call:
## loess(formula = Draft_No ~ Day_of_year, data = data)
##
## Number of Observations: 366
## Equivalent Number of Parameters: 4.35
## Residual Standard Error: 103
## Trace of smoother matrix: 4.73  (exact)
##
## Control settings:
##   span     :  0.75
##   degree   :  2
##   family   :  gaussian
##   surface  :  interpolate      cell = 0.2
##   normalize:  TRUE
##  parametric:  FALSE
## drop.square:  FALSE
```



3.

In order to check if the lottery is random, one can use various statistics. One such possibility is based on the expected responses. The fitted loess smoother provides an estimate $\hat{Y}$ as a function of X. It the lottery was random, we would expect $\hat{Y}$ to be a flat line, equaling the empirical mean of the observed responses, Y . The statistic we will consider will be

$$S = \sum_{i=1}^{n} |\hat{Y}_i - \bar{Y}|$$

If S is not close to zero, then this indicates some trend in the data, and throws suspicion on the randomness of the lottery. Estimate S's distribution through a non–parametric bootstrap, taking B = 2000 bootstrap samples. Decide if the lottery looks random, what is the p–value of the observed value of S.

```
## [1] 8238.649
```

```
## [1] 0.5825
```

the p_value is 0.556 on 364 degrees of freedom. We can't reject the null hypothesis that the lottery is random (?)

4.

We will now want to investigate the power of our considered test. First based on the test statistic S, implement a function that tests the hypothesis H0 : Lottery is random versus H1 : Lottery is non–random. The function should return the value of S and its p–value, based on 2000 bootstrap samples.

```
## $S
## [1] 8238.649
##
## $p_value
## [1] 0
```

5.

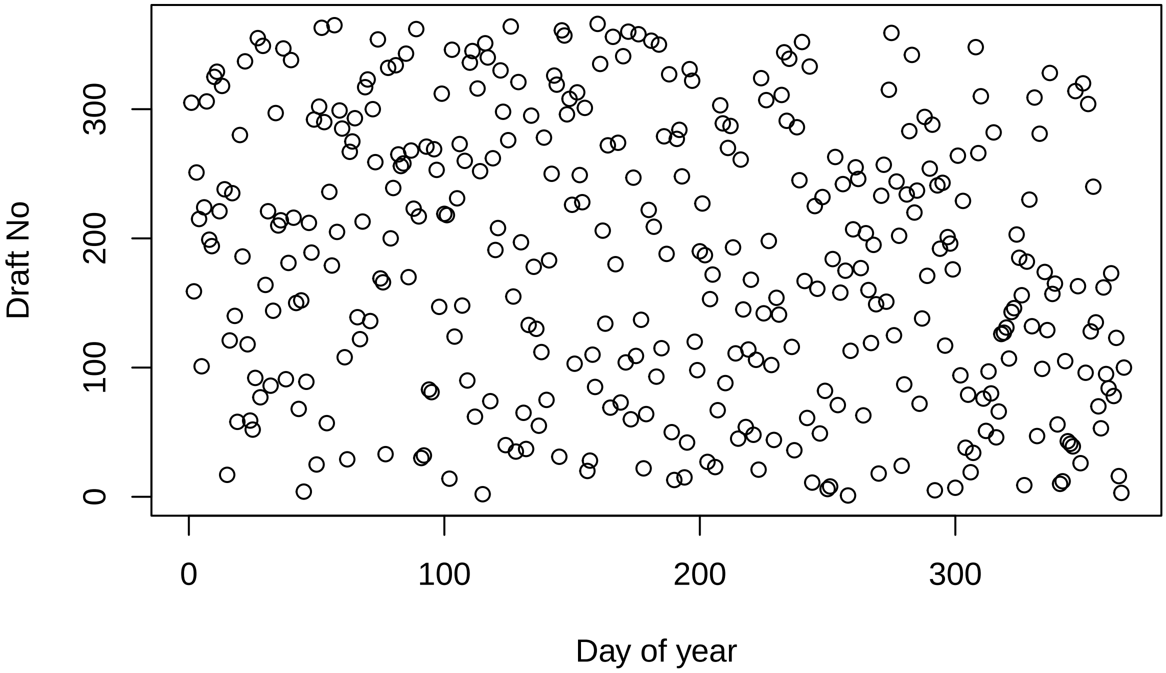
Now we will try to make a rough estimate of the power of the test constructed in Step 4 by generating more and more biased samples:

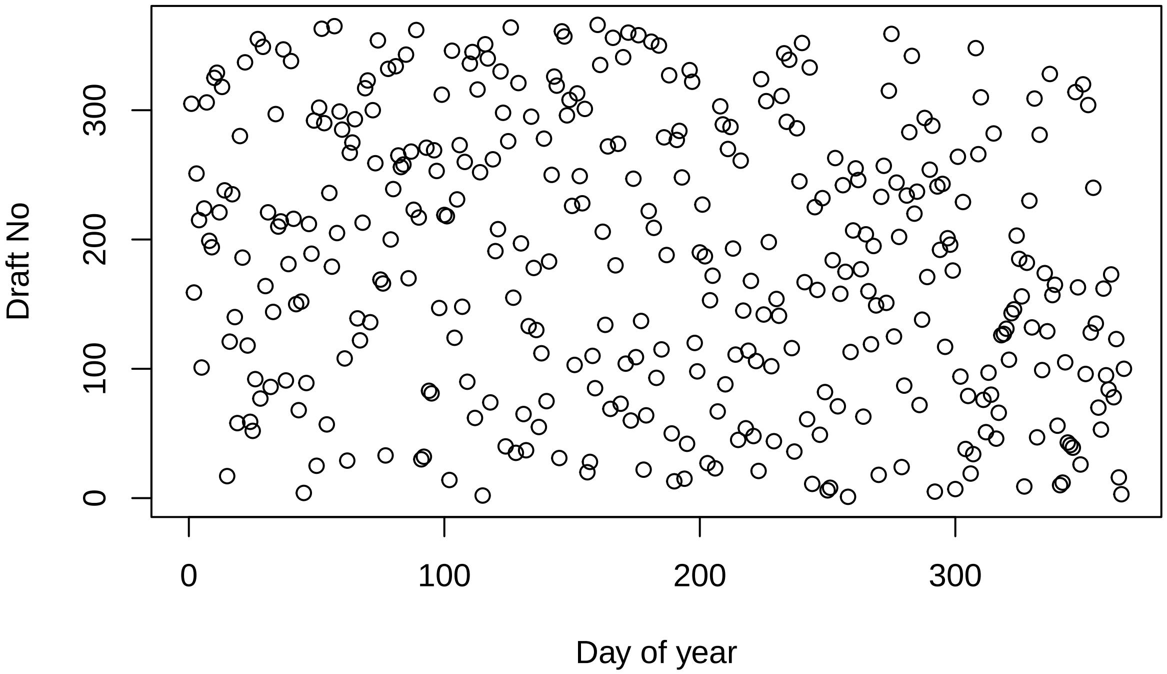
(a) Create a dataset of the same dimensions as the original data. Choose k, out of the 366, dates and assign them the end numbers of the lottery (i.e., they are not legible for the draw). The remaining 366 - k dates should have random numbers assigned (from the set $\{1, \ldots, 366 - k\}$). The k dates should be chosen in two ways:

i. k consecutive dates,

ii. as blocks (randomly scattered) of $\lfloor k/3 \rfloor$ consecutive dates (this is of course for $k >= 3$, and if k is not divisible by 3, then some blocks can be of length $\lfloor k/3 \rfloor + 1$).

*** We got stuck on this question. We are hopping to get some feedback and/or cues during the seminar. ***





(b) For each of the Plug the two new not–completely–random datasets from item 5a into the bootstrap test with B = 2000 and note whether it was rejected.

```
## [1] 0.5735
```

(c) Repeat Steps 5a–5b for k = 1, . . . , until you have observed a couple of rejections.

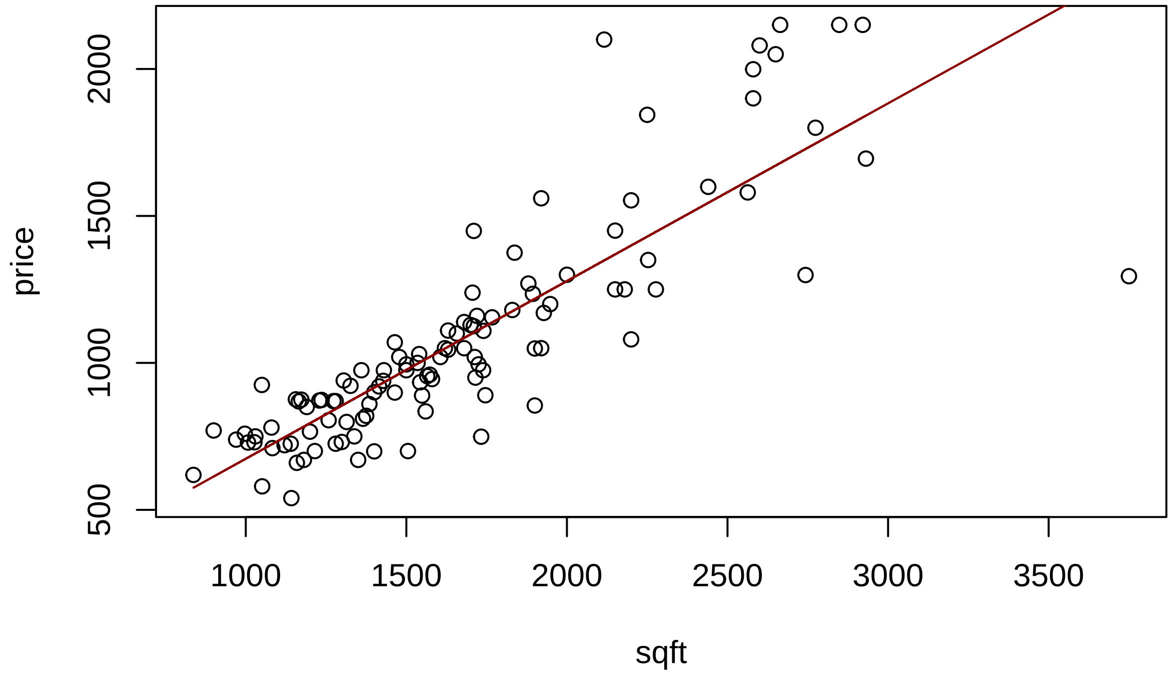
How good is your test statistic at rejecting the null hypothesis of a random lottery?

Question 2: Bootstrap, jackknife and coincidence intervals

The data you are going to continue analyzing is the database of home prices in Albuquerque, 1993. The variables present are Price; SqFt: the area of a house; FEATS: number of features such as dishwasher, refrigerator and so on; Taxes: annual taxes paid for the house. Explore the file prices1.xls. The source of the original is the Data and Story Library (https://dasl.datadescription.com/) and it can be recovered from (https://web.archive.org/web/20151022095618/http://lib.stat.cmu.edu/DASL/Datafiles/homedat.html).

1.

Create a scatter plot of SqFt versus Price. Fit a linear model to it. Does a straight line seem like a good fit?



A straight line seems to summarize the data pretty well.

2.

While the data do seem to follow a linear trend, a new sort of pattern seems to appear around 2000ft2. Consider a new linear model

$Price = b + a_1 * SqFt + a_2 * (SqFt - c) * 1_{SqFt>c}$ where c is the area value where the model changes. You can determine c using an optimizer, e.g., optim(), with the residual sum of squares (RSS) as the value to be minimized. For each value of c, the objective function should estimate b, a1, and a2; then calculate (and return) the resulting RSS.

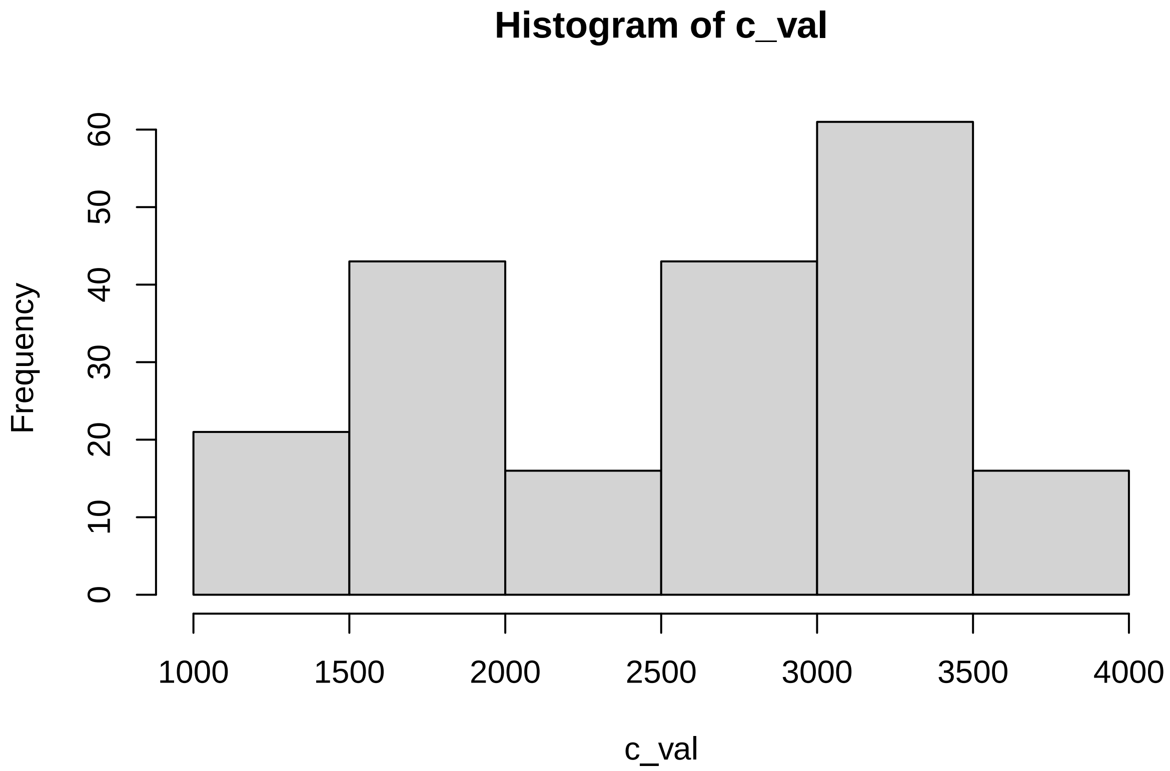
```
## [1] 3033.528
```

```
## [1] 3309413
```

3.

Using the bootstrap estimate the distribution of c. Determine the bootstrap bias{correction and the variance of c. Compute a 95% confidence interval for c using bootstrap percentile, bootstrap BCa, and first{order normal approximation

(Hint: use boot(),boot.ci(),plot.boot(),print.bootci())



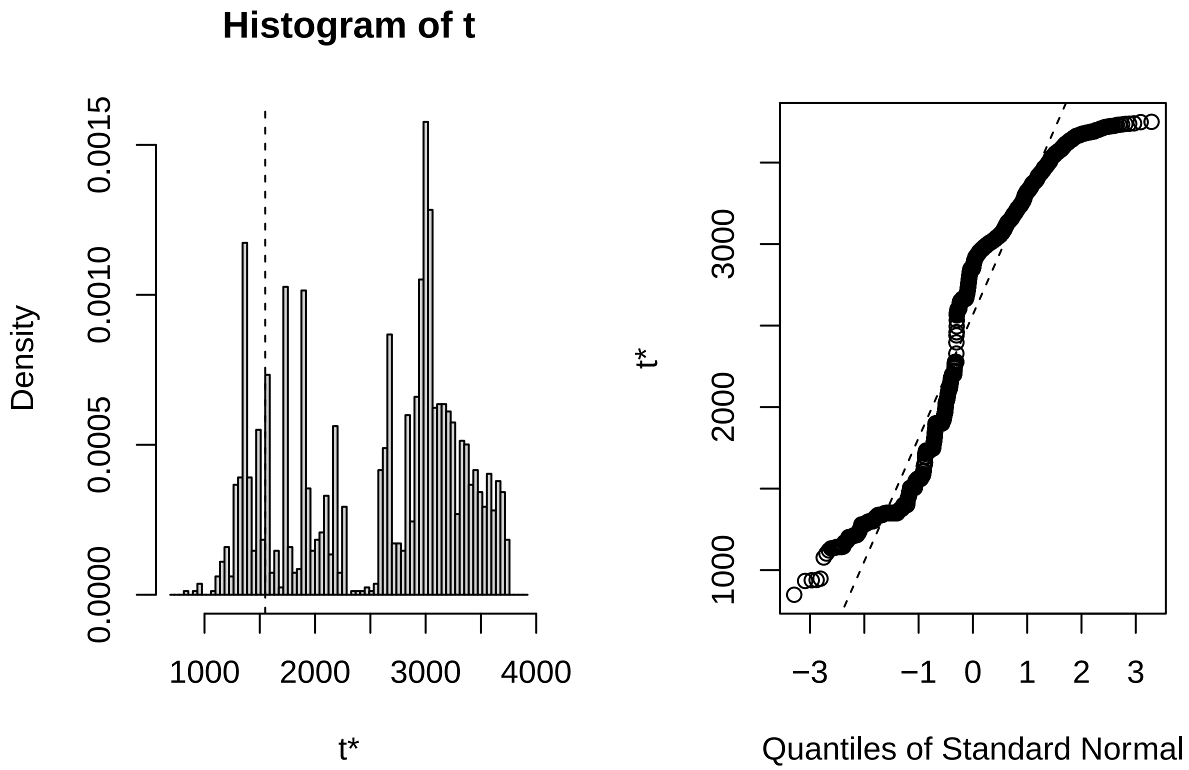
```
## [1] 5811.862
```

```
## [1] 58060.92
```

```
## Warning in boot.ci(boot_fun): bootstrap variances needed for studentized
## intervals
```

```
## Warning in norm.inter(t, adj.alpha): extreme order statistics used as endpoints
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_fun)
##
## Intervals :
## Level      Normal              Basic
## 95%   (-953, 2014 )   (-570, 1812 )
##
## Level     Percentile            BCa
## 95%   (1286, 3668 )   ( 849, 2688 )
## Calculations and Intervals on Original Scale
## Warning : BCa Intervals used Extreme Quantiles
## Some BCa intervals may be unstable
```



We seem to obtain two Gaussian-ish shaped distributions as the distribution of the optimization values.

4.

Estimate the variance of c using the jackknife and compare it with the bootstrap estimate.

```
##   [1] 1733.000 3087.634 2649.999 3217.507 2980.174 1505.000 3028.926 2741.827
##   [9] 2999.993 1505.000 1733.001 2847.999 2200.001 2276.999 2600.000 2200.000
##  [17] 2955.197 3249.475 1320.227 1549.000 1893.711 2277.000 3039.387 2689.974
##  [25] 3169.986 2166.266 3262.286 2600.000 1548.999 2982.432 1332.403 1739.000
##  [33] 3431.196 3704.210 2772.123 3005.800 2199.999 1560.000 2788.363 3098.276
##  [41] 3056.385 2664.000 3043.144 3043.440 2518.350 1746.000 1350.006 3043.092
##  [49] 3312.731 3050.074 3273.820 1334.620 3079.489 2942.375 3581.552 2970.637
##  [57] 2986.563 3721.745 2063.725  952.615 3656.104 1559.131 2963.614 1372.841
##  [65] 2945.194 3695.620 1373.253 3083.162 1900.000 1919.999 1505.000 1739.000
##  [73] 1350.000 2580.001 2037.272 3001.875 1142.000 1505.000 3363.134 3012.189
##  [81] 1383.055 1365.001 1733.000 1812.196 2933.131 1219.857 3069.566 3023.563
##  [89] 3126.375 3065.768 3314.565 3619.651 3165.703 1900.000 2964.809 1350.000
##  [97] 3004.050 3130.182 3007.624 1592.745 2963.847 3276.564 3010.916 3178.948
## [105] 2996.288 2961.478 2848.000 1350.000 2669.275 2848.000
```

```
## [1] 557070.8
```

(The values we obtain are not accurate, we will wait for feedback to understand what we did wrong.)

5.

Summarize the results of your investigation by comparing all of the confidence intervals with respect to their length and the location of c inside them.

Appendix

```r
knitr::opts_chunk$set(echo = FALSE)
#1.1
data=read.csv("lottery.csv", header = TRUE, sep = ";")
summary(data)
plot(data$Day_of_year, data$Draft_No, xlim = c(0, 366), ylim = c(0, 366), xlab = "Day of year", ylab =
#1.2
model=lm(Draft_No ~ Day_of_year , data=data)
summary(model)
plot(data$Day_of_year, data$Draft_No, xlim = c(0, 366), ylim = c(0, 366), xlab = "Day of year", ylab =
abline(model, col="red")

model2=loess(Draft_No ~ Day_of_year , data=data)
summary(model2)
plot(data$Day_of_year, data$Draft_No, xlim = c(0, 366), ylim = c(0, 366), xlab = "Day of year", ylab =
lines(data$Day_of_year, model2$fitted, col="red")

#1.3
S = sum(abs(model2$fitted-mean(data$Draft_No)))
B=2000
S_boot=rep(NA, B)
for (b in 1:B){
  data_boot=data[sample(1:nrow(data), replace = TRUE),]
  model_boot=loess(Draft_No ~ Day_of_year , data=data_boot)
  S_boot[b]=sum(abs(model_boot$fitted-mean(data_boot$Draft_No)))
}
p_value=mean(S_boot>S)
S
p_value
#1.4
#Permutation test:

test_hyp=function(data){
  model=loess(Draft_No ~ Day_of_year , data=data)
  S=sum(abs(model$fitted-mean(data$Draft_No))) #1: T(X) value of statistic from observed data
  B=2000
  S_boot=rep(NA, B)
  #generate B bootstrap samples without replacement
  for (b in 1:B){
    #2: Create permutations g_1, g_B of group variable {If the number of permutations is too large, sam
    data_boot = data
    data_boot$Day_of_year = sample(data$Day_of_year, replace = FALSE) #without replacement
    model_boot=loess(Draft_No ~ Day_of_year , data=data_boot) #fitting the model
    #3: Evaluate test statistic on each permutation
    S_boot[b] = sum(abs(model_boot$fitted-mean(data_boot$Draft_No)))
  }
  p_value=mean(S_boot>=S) #4: Estimate p-value: p^ = #{T(X_gb) >= T(X)}/B
  return(list(S=S, p_value=p_value))
}

test_hyp(data)
#1.5
#(a)
#i k consecutive dates,

#Create a dataset of the same dimensions as the original data.
data2=data

#Choose k, out of the 366, dates and assign them the end numbers of the lottery (i.e., they are not leg
#====================
k=300
#====================
#Randomize the first 366-k dates
# data2$Draft_No[366-k:366]=sample(1:(366-k), k, replace = FALSE)
model2=loess(Draft_No ~ Day_of_year , data=data2)
plot(data2$Day_of_year, data2$Draft_No, xlim = c(0, 366), ylim = c(0, 366), xlab = "Day of year", ylab =

#ii. as blocks (randomly scattered) of $\lfloor k/3 \rfloor$  consecutive dates (this is of course for 

data3=data

get_lotery_numbers_with_chunks <- function(k){
  #k is the number of dates that are not legible for the draw
  # 3 cases :
  # 1. n_chunks is an integer -> 3 normal chunks
  # 2. n_chunks is not an integer and has a remainder of 1 -> 2 normal chunks and 1 long chunk
  # 3. n_chunks is not an integer and has a remainder of 2 -> 1 normal chunk and 2 long

  #Case 1
  if(k%%3 == 0){
    chunk_size = floor(k/3)
    n_chunks = 3
    for (i in 1:n_chunks){
      #Sample n_chunks_normal chunks of length chunk_size
      #random starting point
      start = sample(1:366, 1, replace = TRUE)
      #sample chunk_size consecutive days
      data3$Draft_No[start:start+chunk_size-1] = sample(1:(366-k), chunk_size, replace = TRUE)
    }

  }

  # chunk_size = floor(k/3)
  # remainder = k%%chunk_size
  # n_chunks_normal = floor(k/chunk_size) - remainder
  # n_chunks_long = remainder
  # for (i in 1:n_chunks_normal){
  # #Sample n_chunks_normal chunks of lenght chunk_size
  # #random starting point
  # start = sample(1:366, 1, replace = TRUE)
  # #sample chunk_size consecutive days
  # data3$Draft_No[start:start+chunk_size-1] = sample(1:(366-k), chunk_size, replace = TRUE)
  # }
  # for (i in 1:n_chunks_long){
  #    #Sample n_chunks_long chunks of lenght chunk_size+1
  #   start = sample(1:366, 1, replace = TRUE)
  #   #sample chunk_size consecutive days
  #   data3$Draft_No[start:start+chunk_size] = sample(1:(366-k), chunk_size+1, replace = TRUE)
  # }
  # data3$Draft_No[n_chunks_normal*chunk_size+n_chunks_long*(chunk_size+1):366] = sample(1:(366-k), 366

}



model3=loess(Draft_No ~ Day_of_year , data=data3)
plot(data3$Day_of_year, data3$Draft_No, xlim = c(0, 366), ylim = c(0, 366), xlab = "Day of year", ylab
#(b)

S=sum(abs(model2$fitted-mean(data2$Draft_No)))
B=2000
S_boot=rep(NA, B)
for (b in 1:B){
  data_boot=data2[sample(1:nrow(data2), replace = TRUE),]
  model_boot=loess(Draft_No ~ Day_of_year , data=data_boot)
  S_boot[b]=sum(abs(model_boot$fitted-mean(data_boot$Draft_No)))
}
p_value=sum(S_boot>S)/B
p_value

#(c)

data <- read.csv("prices1.csv", sep=";")

## 1)

price <- data$Price
sqft <- data$SqFt

plot(sqft, price)
fit1 <- lm(Price ~ SqFt, data)
lines(sqft, fit1$fitted.values, col="darkred", type="l", lwd=1)
## 2)

fit2 <- function(c, data){
  data$new <- rep(0,110)
  for (i in 1:nrow(data)){
    if (data[i,2] > c) data[i,5] <- data[i,2] - c
    else data[i,5] <- 0
  }
  lm2 <- lm(Price ~ SqFt + new, data)
  sum(lm2$residuals^2)
}

RSS_opt <- optim(2000, fit2, data=data, method="L-BFGS-B", lower=0, upper=4500)
RSS_opt$par
RSS_opt$value
library(boot)

c_val <- c()

for (i in 1:200){ #2000 takes too long to run on my poor laptop
  if (i%%500==0) cat(i, "\r")
  id <- sample(1:110, 110, replace = TRUE)
  data_boot <- data[id,]
  c_opt <- optim(2000, fit2, data=data_boot, method="L-BFGS-B", lower=0, upper=4500)
  c_val <<- c(c_val, c_opt$par)
}

hist(c_val)

2 * RSS_opt$par - sum(c_val) / 2000      # bootstrap bias-correction
sum((c_val - mean(c_val))^2) / (2000-1)  # variance of c

opt_boot <- function(data, id){
  id <- sample(1:110, 110, replace = TRUE)
  data_boot2 <- data[id,]
  c_opt2 <- optim(2000, fit2, data= data_boot2, method = "L-BFGS-B", lower = 0, upper = 4500)
  c_opt2$par
}

boot_fun <- boot(data = data, opt_boot, R = 2E3)
boot.ci(boot_fun)
plot(boot_fun)

jack_c <-c()

# new function for the jackknife function with 109 rows instead of 110
fit3 <- function(c, data){
  data$new <- rep(0,109)
  for (i in 1:nrow(data)){
    if (data[i,2] > c) data[i,5] <- data[i,2] - c
    else data[i,5] <- 0
  }
  lm2 <- lm(Price ~ SqFt + new, data)
  sum(lm2$residuals^2)
}

for (i in 1:110){
  id <- sample(1:110, 110, replace = TRUE)
  data_boot3 <- data[id,]
  data_jack <- data_boot3[-i,]
  c_opt <- optim(2000, fit3, data=data_jack, method = "L-BFGS-B", lower = 0, upper = 4500)
  jack_c <<- c(jack_c, c_opt$par)
```

```
}

jack_c
var(jack_c)
```