Group2

Greeshma Jeev Koothuparambil(greko370), Sangeeth Sankunny Menon(sansa237)

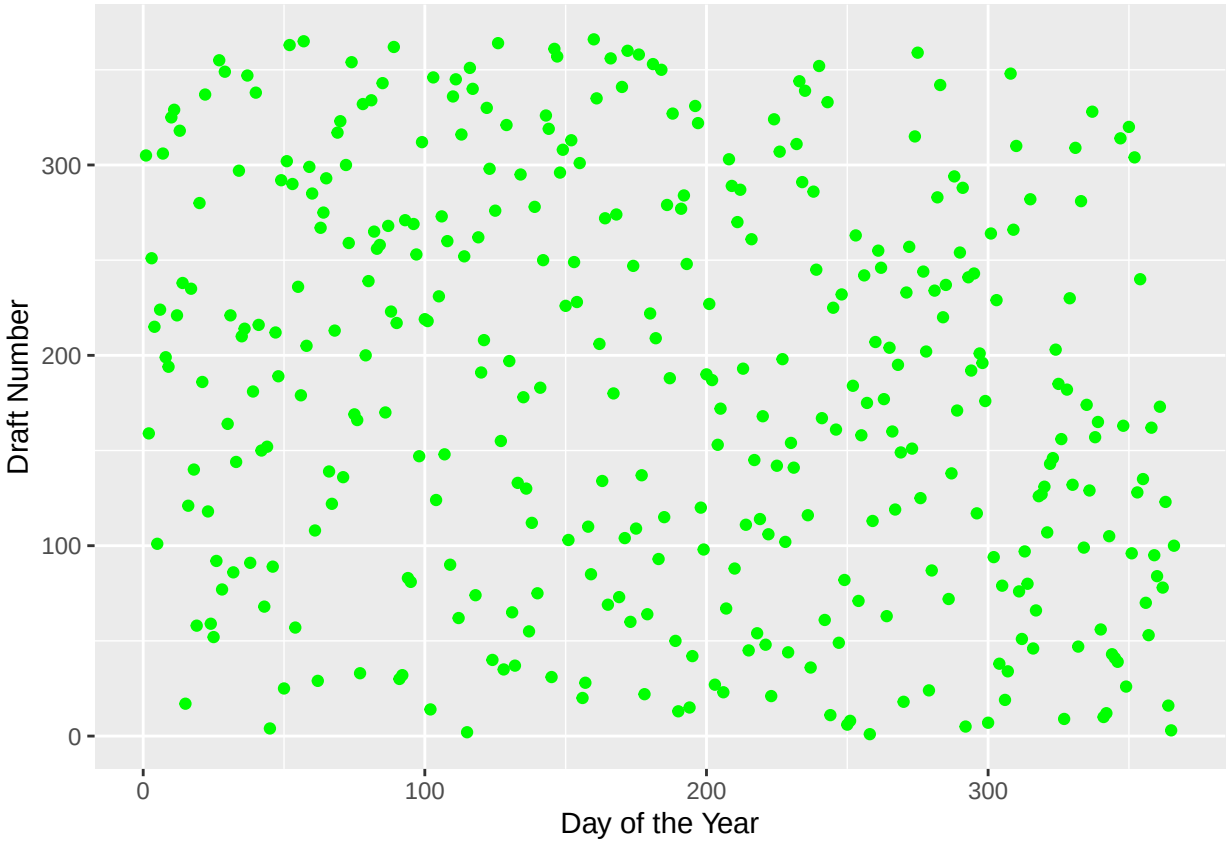
2023-12-05

Question 1: Hypothesis testing

In 1970, the US Congress instituted a random selection process for the military draft. All 366possible 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.xls. 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?

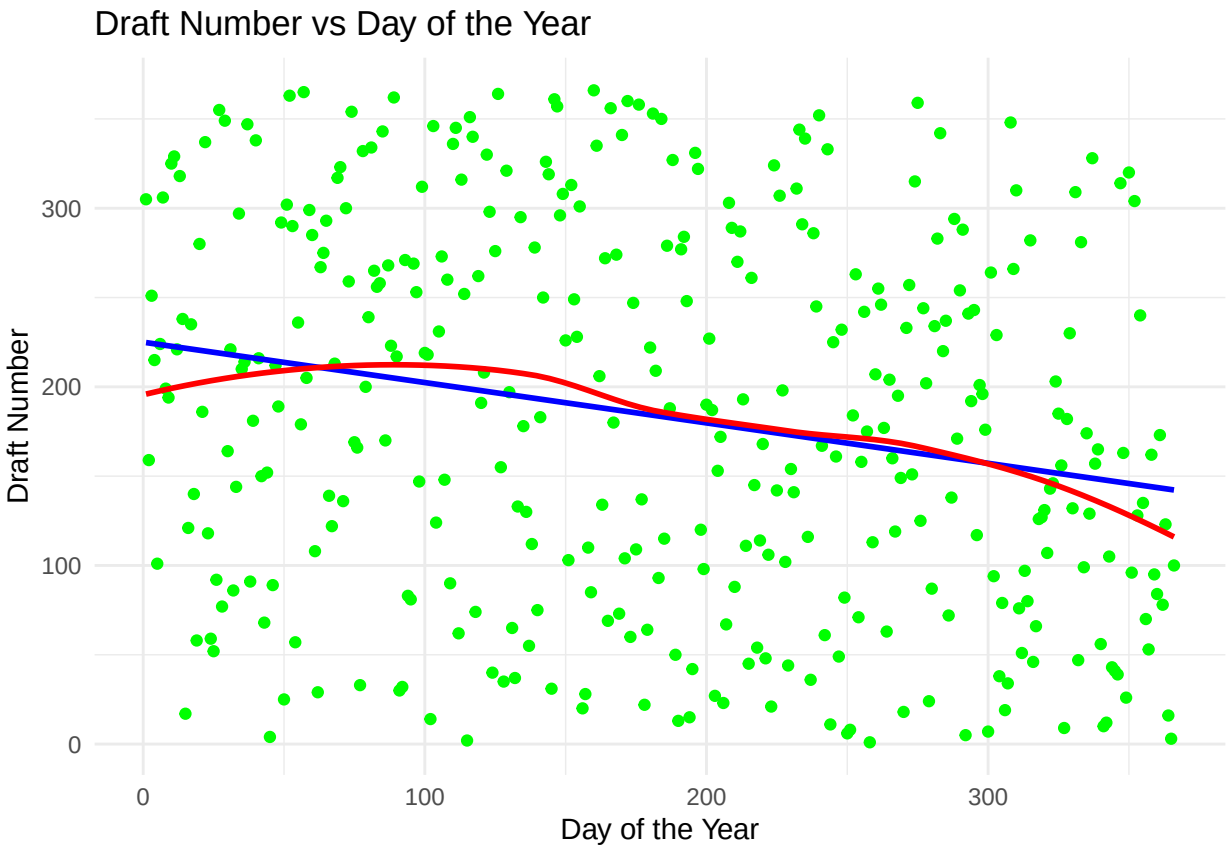
```
library(ggplot2)

data <- read.csv("C:/Users/sange/OneDrive/Documents/lottery.csv",sep = ";",header = TRUE)

ggplot(data, aes(x = Day_of_year, y = Draft_No)) +
  geom_point(color = "green") +
  labs(x = "Day of the Year", y = "Draft Number")
```



There is no visible pattern observed as the data points are spreaded.

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

```
linear_model <- lm(Draft_No ~ Day_of_year, data = data)

loess_model <- loess(Draft_No ~ Day_of_year, data = data)

scatterplot <- ggplot(data, aes(x = Day_of_year, y = Draft_No)) +
  geom_point(color="green") +
  labs(x = "Day of the Year", y = "Draft Number") +
  ggtitle("Draft Number vs Day of the Year")

scatterplot +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE, color = "blue") +
  geom_smooth(method = "loess", se = FALSE, color = "red") +
  theme_minimal()
```



The red line represents the prediction of the loess model.From a perspective based on the lines plotted, a distinct relationship among the parameters is noticeable. There exists a correlation between X and Y.

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, equalling the empirical mean of the observed responses, $\overline{Y}$. The statistic we will consider will be

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

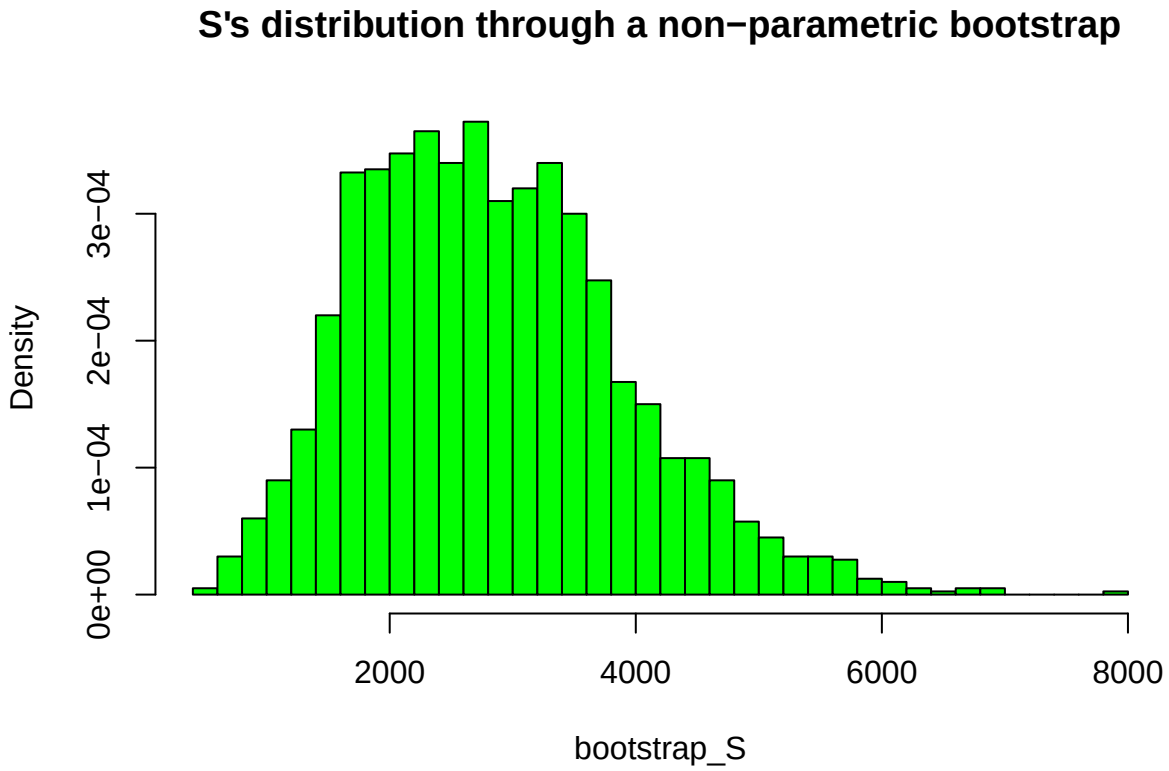
If S is not close to zero, then this indicates some trend in the data, and throws suspicion on the randomness of the lottery. Estimate S's distribution through a non–parametric bootstrap, taking $B = 2000$ bootstrap samples. Decide if the lottery looks random, what is the p–value of the observed value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

```
# Calculating the observed statistic S and estimate its distribution through non-parametric bootstrap
predicted_values <- predict(loess_model)
observed_S <- sum(abs(predicted_values - mean(data$Draft_No)))
calculate_S <- function(index) {
  sampled_draft_numbers <- sample(data$Draft_No, size = nrow(data),replace = TRUE)
  new_df <- as.data.frame(cbind(Day_of_year = data$Day_of_year, Draft_No = sampled_draft_numbers))
  mean_Y <- mean(sampled_draft_numbers)
  loess_model <- loess(Draft_No ~ Day_of_year, data = new_df)
  predicted_values <- predict(loess_model, newdata = new_df)
  sum(abs(predicted_values - mean_Y))
}
# Non-parametric bootstrap
```

```r
B <- 2000
bootstrap_S <- numeric(B)
for (i in 1:B) {
  bootstrap_S[i] <- calculate_S(i)
}

observed_S <- sum(abs(predicted_values - mean(data$Draft_No)))
p_value <- sum(bootstrap_S >= observed_S) / B
print(p_value)
```

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

```r
hist(bootstrap_S, breaks = 40, col = "green", freq = F, main = "S's distribution through a non-parametri
```



The p–value of the observed value of S is 0.

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.

```r
hypothesis_test <- function(data, loess_model, B = 2000) {
  n <- nrow(data)

  predicted_values <- predict(loess_model)
  observed_S <- sum(abs(predicted_values - mean(data$Draft_No)))
```

```
  bootstrap_S <- numeric(B)



  for (i in 1:B) {
    bootstrap_S[i] <- calculate_S(i)
  }

  p_value <- sum(bootstrap_S >= observed_S) / B

  return(list(observed_S = observed_S, p_value = p_value))
}
```

5. Now we will try to make a rough estimate of the power of the test constructed in Step 4by 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, . . . , 366 − k}). The k dates should be chosen in two ways:
i. k consecutive dates,
ii. as blocks (randomly scattered) of bk/3c consecutive dates (this is of course for k 3, and if k is not divisible by 3, then some blocks can be of length bk/3c+ 1).

(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.

```
#5.a b
#5.a b
create_biased_consecutive <- function(original_data, k) {
  n <- nrow(original_data)
  new_data <- original_data


  d <- sample(1:(366-(k-1)), 1)
  non_legible_dates <- d:(d+k-1)
  new_data$Draft_No <- NA
  new_data$Draft_No[non_legible_dates] <- original_data$Draft_No[non_legible_dates]
  new_data$Draft_No[is.na(new_data$Draft_No)] <- sample(c(1:(d-1),(d+k):366),replace = F)
  return(new_data)
}

# Create a biased dataset with k consecutive dates having non-legible end numbers
k_consecutive <- 10  # Choose the number of consecutive dates to be biased
biased_data_consecutive <- create_biased_consecutive(data, k_consecutive)


#Create a dataset with blocks of consecutive dates with non-legible end numbers:


create_biased_blocks <- function(original_data, k) {
  if(k>=3){
  n <- nrow(original_data)
  new_data <- original_data
```

```
  num_blocks <- floor(k / 3)
  remaining_dates <- k %% 3
  block_size <- floor(k / 3)

  if (remaining_dates > 0) {
    block_size <- block_size + 1
  }

  print(k)
  new_data$Draft_No <- NA
  block_start <- sample(1:367- block_size, 1)

  block_dates <- block_start:(block_start + (block_size-1 ))
  new_data$Draft_No[block_dates] <- original_data$Draft_No[block_dates]

  t <- which(is.na(new_data$Draft_No))
  na_draft <- sample(t,replace = F)
  new_data$Draft_No[is.na(new_data$Draft_No)] <- na_draft

  return(new_data)
}
}
```

(c) Repeat Steps 5a–5b for $k = 1, \ldots,$ until you have observed a couple of rejections. How good is your test statistic at rejecting the null hypothesis of a random lottery?

```
#5.c


test_for_k_values <- function(original_data, loess_model, max_k = 366, B = 2000) {
  k <- 1
  rejnum <- 0
  rejected <- FALSE

  while (k <= max_k && rejnum<5) {

    biased_data <- create_biased_consecutive(original_data, k)

    test_result <- hypothesis_test(biased_data, loess_model, B)

    cat("Testing on simulated data on Consecutive dates...","\n")
    if (test_result$p_value < 0.05) {
      rejected <- TRUE

      cat("Null hypothesis rejected for k =", k, "\n")
      cat("Observed value of S:", test_result$observed_S, "\n")
      cat("P-value:", test_result$p_value, "\n")
    }


    k <- sample(1:357, 1)
    rejnum <- rejnum+1
  }
```

```
    k <- 3
    rejnum <- 0
    rejected <- FALSE

    while (k <= max_k && rejnum<5) {

      biased_data <- create_biased_blocks(original_data, k)

      test_result <- hypothesis_test(biased_data, loess_model, B)

      cat("Testing on simulated data on Consecutive blocks...","\n ")
      if (test_result$p_value < 0.05) {
        rejected <- TRUE

        cat("Null hypothesis rejected for k =", k, "\n")
        cat("Observed value of S:", test_result$observed_S, "\n")
        cat("P-value:", test_result$p_value, "\n")
      }


      k <- sample(1:357, 1)
      rejnum <- rejnum+1
  }
}

test_for_k_values(data, loess_model)
```

```
## Testing on simulated data on Consecutive dates...
## Null hypothesis rejected for k = 1
## Observed value of S: 8239.001
## P-value: 0
## Testing on simulated data on Consecutive dates...
## Null hypothesis rejected for k = 235
## Observed value of S: 8855.644
## P-value: 0
## Testing on simulated data on Consecutive dates...
## Null hypothesis rejected for k = 259
## Observed value of S: 15766.62
## P-value: 0
## Testing on simulated data on Consecutive dates...
## Null hypothesis rejected for k = 67
## Observed value of S: 8441.361
## P-value: 0
## Testing on simulated data on Consecutive dates...
## Null hypothesis rejected for k = 244
## Observed value of S: 17431.63
## P-value: 0
## [1] 3
## Testing on simulated data on Consecutive blocks...
##  Null hypothesis rejected for k = 3
## Observed value of S: 8252.473
## P-value: 0
## [1] 79
```

```
## Testing on simulated data on Consecutive blocks...
##  Null hypothesis rejected for k = 79
## Observed value of S: 8354.55
## P-value: 0
## [1] 326
## Testing on simulated data on Consecutive blocks...
##  Null hypothesis rejected for k = 326
## Observed value of S: 16503.86
## P-value: 0
## [1] 7
## Testing on simulated data on Consecutive blocks...
##  Null hypothesis rejected for k = 7
## Observed value of S: 8299.1
## P-value: 0
## [1] 31
## Testing on simulated data on Consecutive blocks...
##  Null hypothesis rejected for k = 31
## Observed value of S: 8463.257
## P-value: 0
```

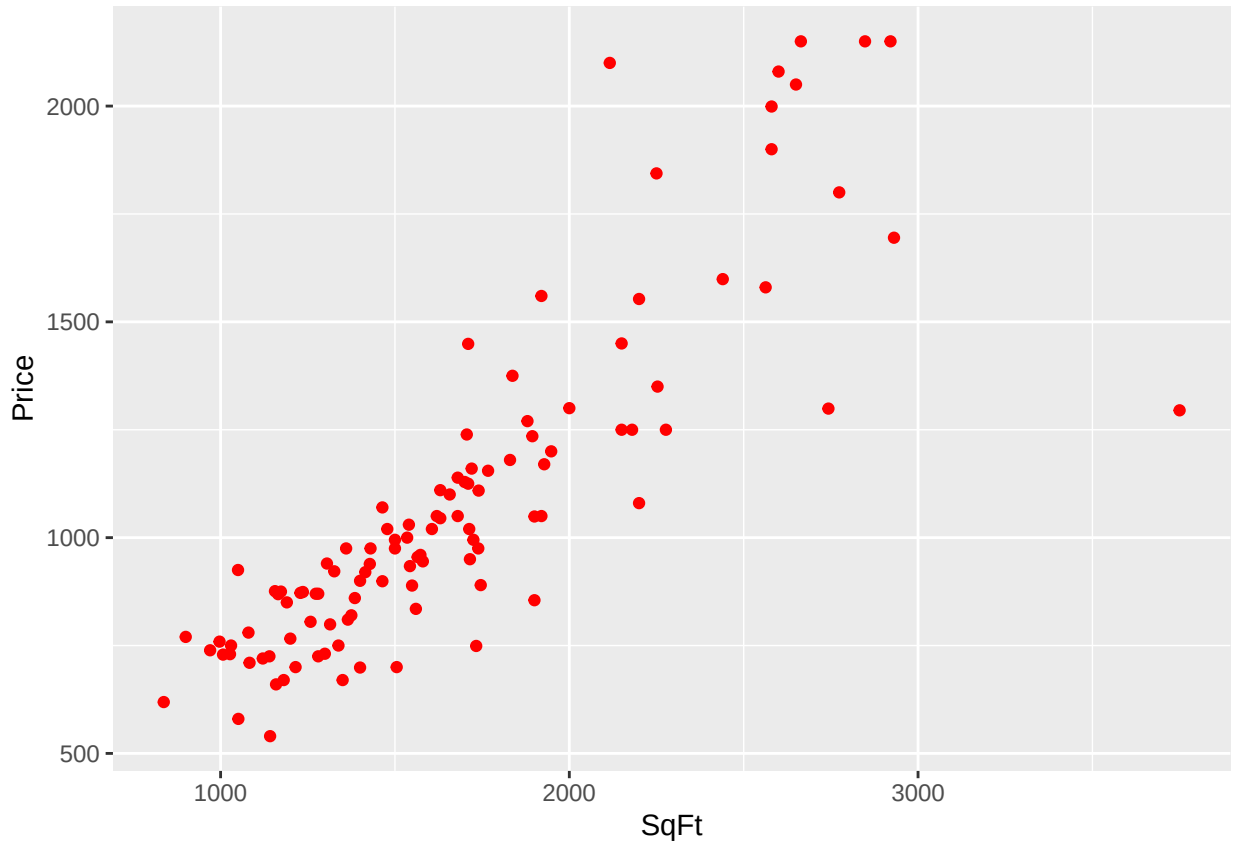
The p-value is very small (less than 0.05), it suggests that the observed test statistic is unlikely to occur by random chance alone, leading to the rejection of the null hypothesis.

Question 2: Bootstrap, jackknife and confidence intervals

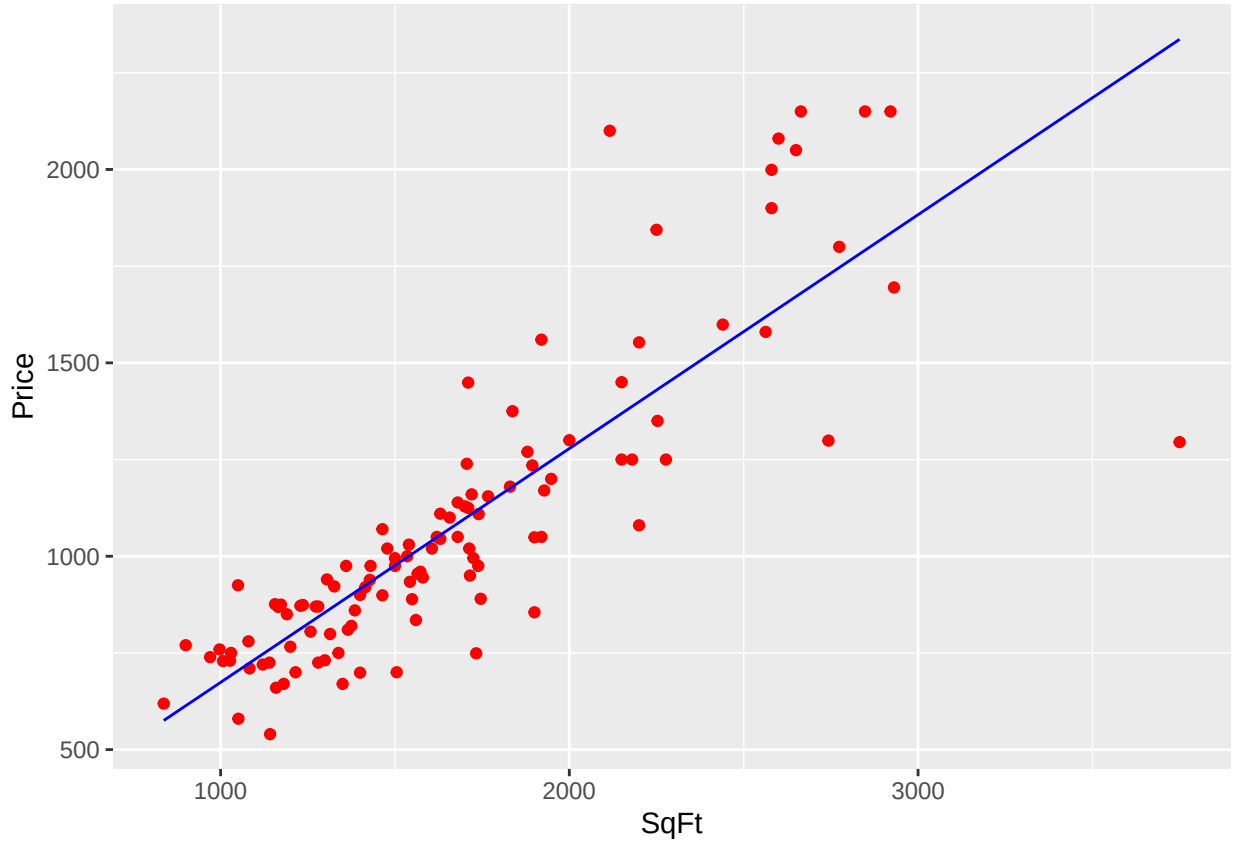
The data you are going to continue analyzing is the database of home prices in Albuquerque, 1993. The variables present are Price; SqFt: the area of a house; FEATS: number of features such as dishwasher, refrigerator and so on; Taxes: annual taxes paid for the house. Explore the file prices1.xls. The 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?

```r
library(ggplot2)
library(boot)

# Loading the dataframe:
data <- read.csv("C:/Users/sange/OneDrive/Documents/prices1.csv", sep = ";")

#2.1

#Plotting the scatterplot

ggplot(data, mapping = aes(x= SqFt, y= Price ))+
  geom_point( colour = "red")
```




```
#Fit a linear model

lmmodel <- lm(Price~SqFt, data)
data$fit <- lmmodel$fitted.values

#Adding the fit line
ggplot(data, mapping = aes(x= SqFt, y= Price ))+
  geom_point( colour = "red")+
  geom_line(mapping = aes(y = fit), colour = "blue")
```



In the plot, till 2000 the linear model seem to fit. But after that it does not seem like a better option.

2. While the data do seem to follow a linear trend, a new sort of pattern seems to appear around $2000 ft^2$*. Consider a new linear mode*

$$Price = b + a1 \cdot SqFt + a2 \cdot (SqFt - c)1_{SqFt>c},$$

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.

```
#2.2

#Defining RSS function
RSS <- function(pars, data){
  b <- pars[1]
  a1 <- pars[2]
  a2 <- pars[3]
  c <- pars[4]
  NewPrice <- b+(a1* data$SqFt)+ (a2* (data$SqFt-c)*(data$SqFt > c))
  RSSvalue <- sum((data$Price - NewPrice)^2)

  return(RSSvalue)
}

#Initialising parameter vlaues
initialpar <- c(1000, 0.1, 0.5, 2000)
```

```r
#Optimising function

optimvals <- optim(initialpar, fn = RSS,data =data)

# Calculate RSS

RSSOptimal <- RSS(optimvals$par, data)
```

The optimised parameters are as follows:

b =-74.4459145

a1 =0.7035742

a2 =-1.5126491

c =2895.3062007

With these values the calculated RSS value is : 3.3148654×10^6

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())

```r
#2.3
#Optimising function
optimizer <- function(data, w){
  paras <- optim(initialpar, fn = RSS,data =data[w,])
  return(paras$par[4])
}
parabootstrap <- boot(data =data, statistic = optimizer, R = 1000)


#Bias - Correction Estimator:
BiasCorr <- 2* optimvals$par[4]- mean(parabootstrap$t)

#Normal CI
normal <- boot.ci(parabootstrap, type = "norm")
#Percentile CI
percent <- boot.ci(parabootstrap, type = "perc")
#Percentile CI
bca <- boot.ci(parabootstrap, type = "bca")
```

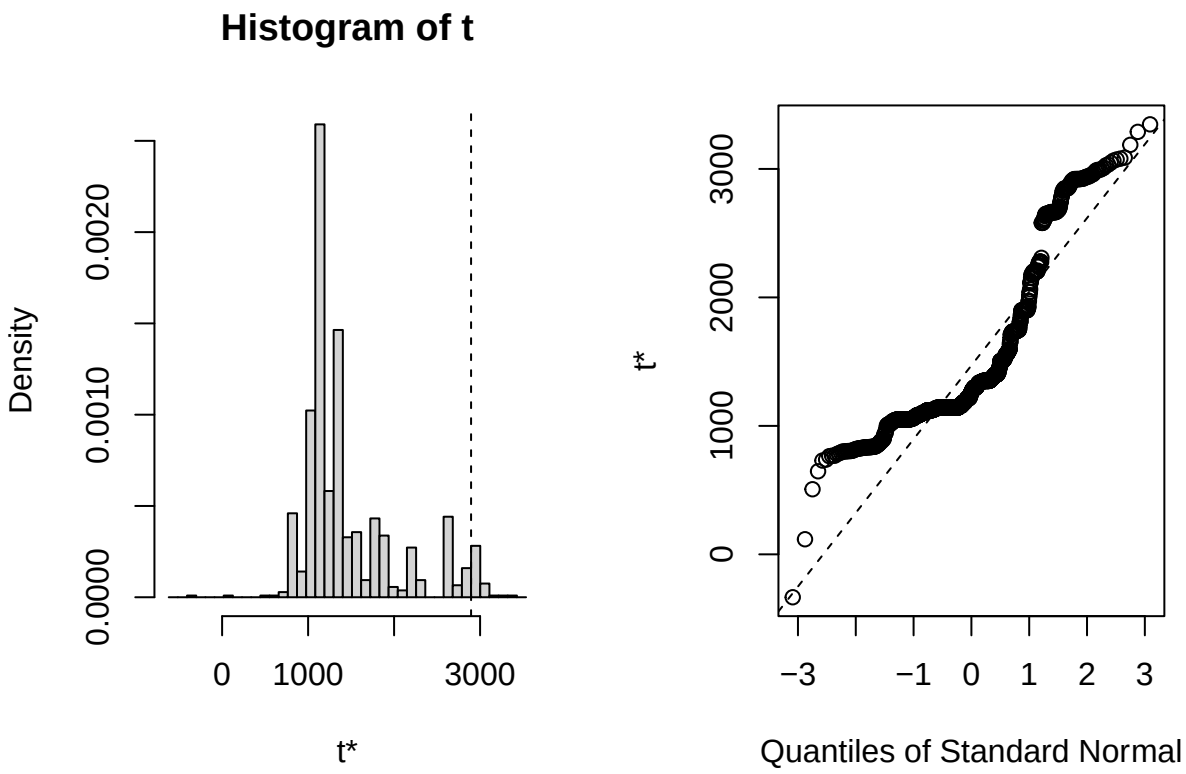
The bootstrap Bias-Correction is 4320.2796192

The Normal CI is : 3195.8406269, 5444.7186116

The Percentile CI is : 823.3562825, 2932.8439839

The BCA CI is: 2689.9910292, 3346.9408747

The histogram and Quantiles of Standard Normal of the C value is shown below:



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

```
#2.4
#Jackknife function:
jackknife <- function(data, B){
  cvals <- c()
  for (i in 1:B) {
    paras <- optim(initialpar, fn = RSS,data =data[-i,])
    cvals[i] <- paras$par[4]
  }
  return(cvals)
}
#c value generation
jack_cs <- jackknife(data, 1000)

#bootstrap Variance:
mean_c <- mean(parabootstrap$t)
sse <- sum((parabootstrap$t- mean_c)^2)
varboot <- sse/(length(parabootstrap$t)-1)


#jackknife Variance:
size <- length(jack_cs)
Ti <- (size*optimvals$par[4])- ((size-1)*jack_cs)
Jt <- mean(Ti)
ssejack <- sum((Ti - Jt)^2)
```

```
varjack <- ssejack/(size*(size-1))
```

The bootstrap variance of C is: 3.2913617×10^5

The jackknife variance of C is: 2.2583365×10^8

The variances are really high. Also the difference is extremely high between the variances.

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.

```
cval <- c(normal$t0, percent$t0, bca$t0)
interval_from <- c(normal$normal[2], percent$percent[4], bca$bca[4])
interval_to <- c(normal$normal[3], percent$percent[5], bca$bca[5])
length <- interval_to - interval_from

tablec <- data.frame(interval_from, interval_to, length, cval)
colnames(tablec) <- c("Interval From", "Interval To", "Length", "C-Value")
```

The length and C-values for each estimate is as tabulated:

Interval From	Interval To	Length	C-Value
3195.8406	5444.719	2248.8780	2895.306
823.3563	2932.844	2109.4877	2895.306
2689.9910	3346.941	656.9498	2895.306