Computer Lab 5

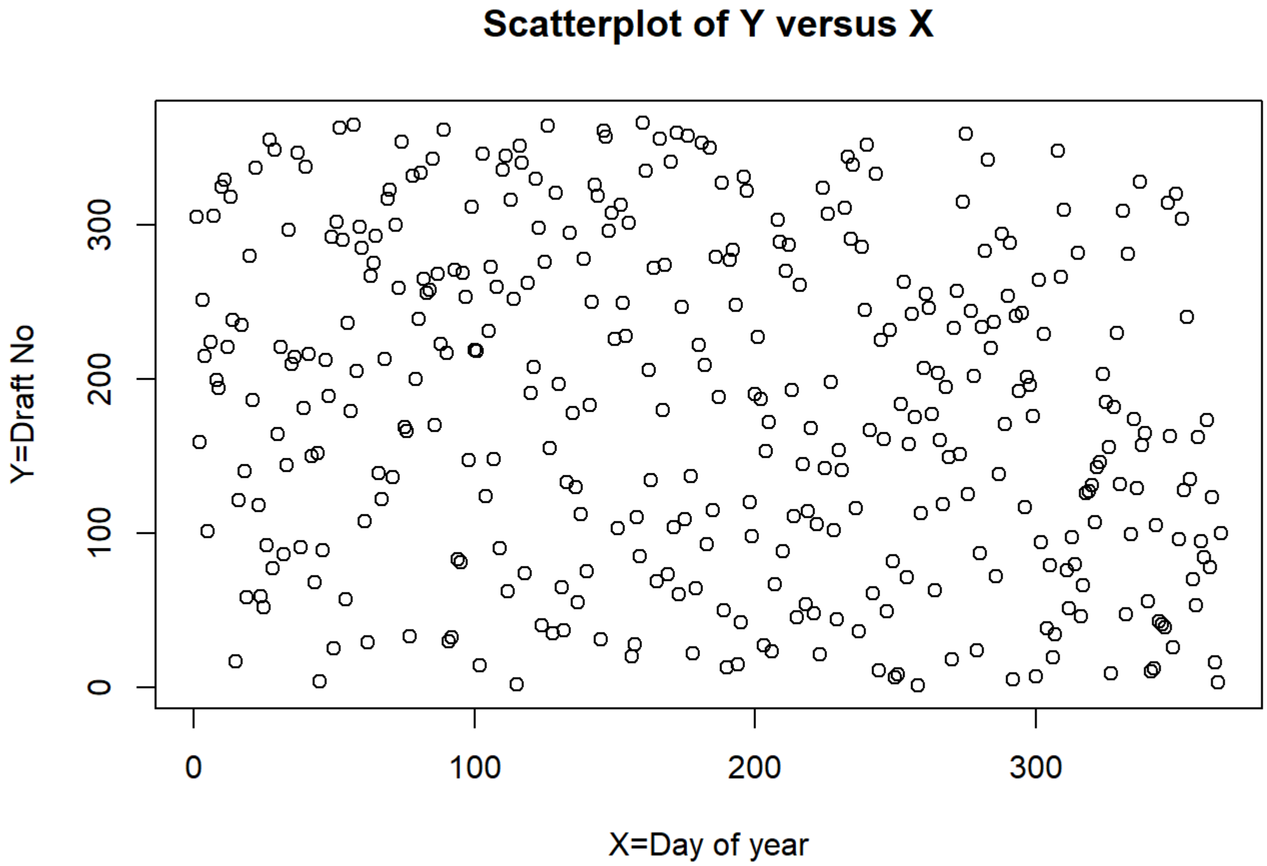
Group-19 - Dinesh and Umamaheswarababu

2023-02-08

Statement of Contribution : Question 1 was mostly done by Umamaheswarababu and Question 2 was mostly done by Dinesh

Question 1: Hypothesis testing

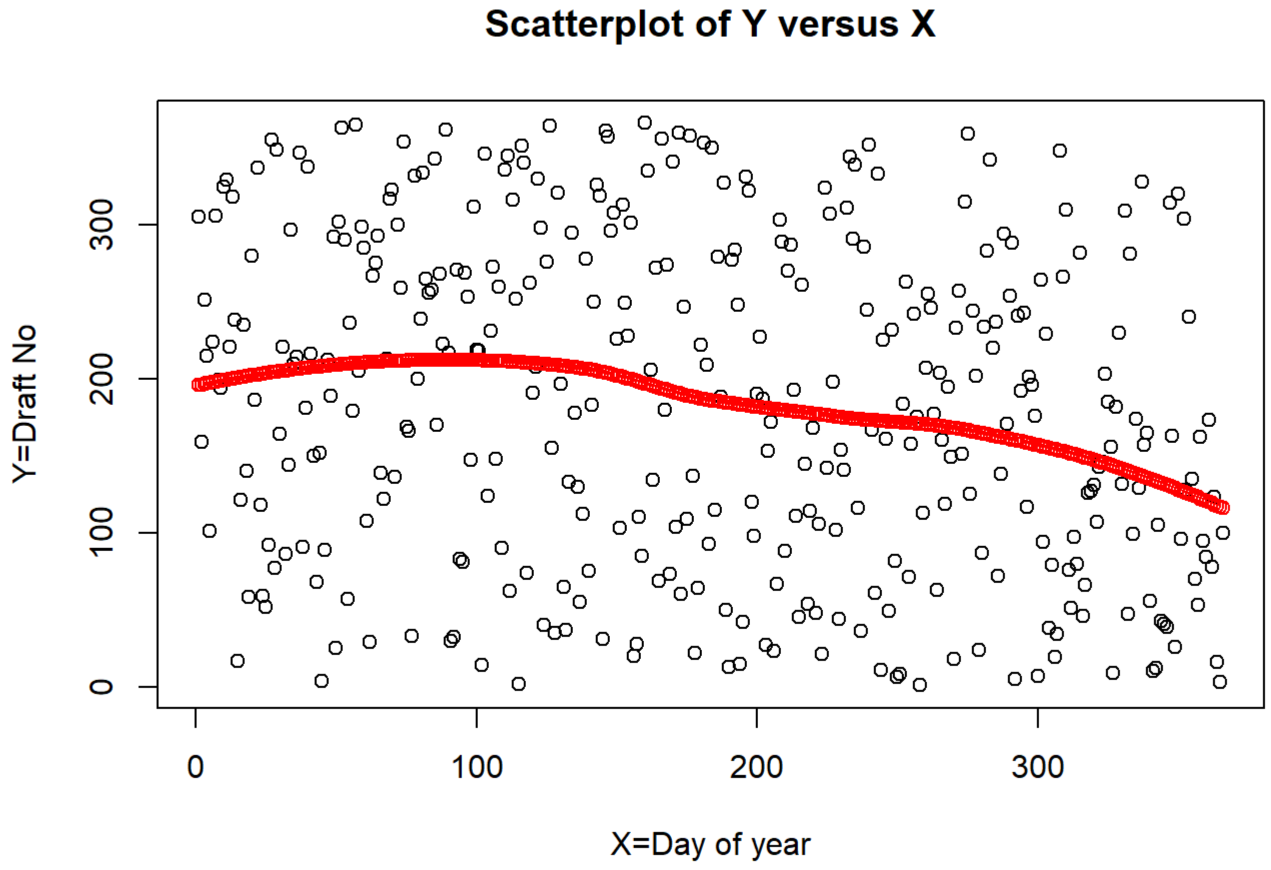
Question 1.1: Make a scatterplot of Y versus X and conclude whether the lottery looks random.



From the plot we can say that the lottery is random because we observe Draft No(Y) has no pattern with respect to Day of year(X).

Question 1.2 : Compute an estimate Y_hat of the expected response as a function of X by using a loess smoother (use loess()), put the curve Y_hat

versus X in the previous graph and state again whether the lottery looks random.



We observe that the estimate $\hat{Y}$ almost flat till Day of year(X) is 100 and then is decreasing with increase in Day of year(X). From this we can say that the chance of getting small draft is higher for the people who were born at the end of the year. So the lottery may not be random.

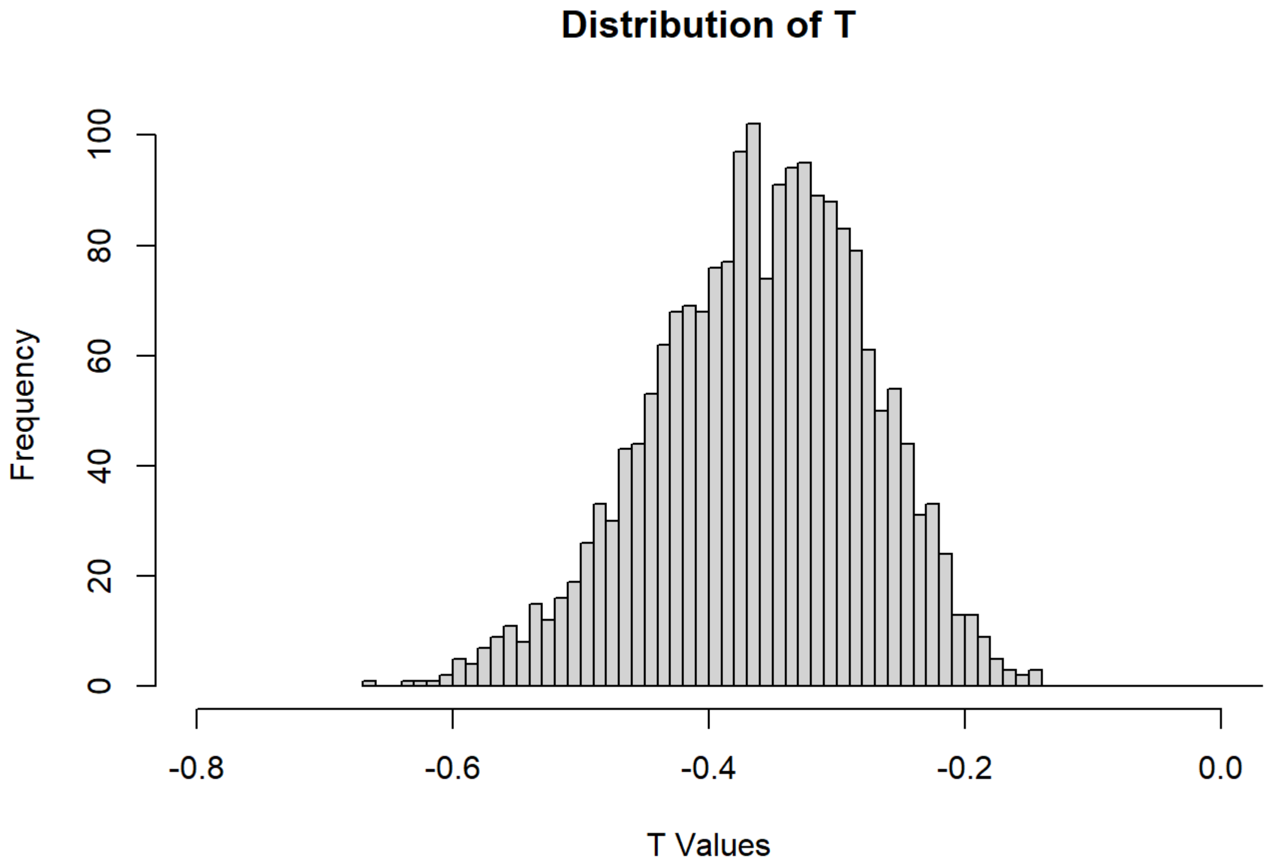
Question 1.3 : To check whether the lottery is random, it is reasonable to use test statistics. If this value is significantly different from zero and not greater than zero then there should be a trend in the data and the lottery is not random. Estimate the distribution of T by using a non–parametric bootstrap with B = 2000 and comment whether the lottery is random or not. What is the value of p, in which t = 0 is the p-quantile of T, given your bootstrap samples?

The value of test statistic is

```
## [1] -0.3479163
```

Since the test statistic value is significantly different from zero, we can say there is a trend in the data and the lottery is not random.

Estimation of the distribution of T by using a non–parametric bootstrap with B = 2000



From the above plot(Distribution of T), we observe that the T values is significantly different from zero, we can say there is a trend in the data and the lottery is not random.

The value of p, in which t = 0 is the p-quantile

```
## p-value is  0.999
```

Question 1.4 : Implement a function depending on data and B that tests the hypothesis by using a permutation test with statistics T. The function is to return the p–value of this test. Test this function on

our data with B = 2000.

```
set.seed(12345)
#a function for permutation test
permut_test <- function(data,B){
  k<-dim(data)[1]
  result<-numeric(B)
  for (i in 1:B){
    samp<-sample(data$Day_of_year,k,replace = FALSE)
    sampled_data<-data
    sampled_data$Day_of_year<-samp
    result[i]<-t_stat(sampled_data,1:k)
  }
  test_stat<-t_stat(data,1:dim(data)[1])
  p_value<-mean(abs(result)>=abs(test_stat))
  return(p_value)
}
p_value<-permut_test(data=data,B=2000)
cat("p-value:",p_value)
```

```
## p-value: 0.156
```

Question 1.5 : Make a crude estimate of the power of the test constructed in Step 4:

Question 1.5.a

Generating data set with n=366

```
set.seed(12345)
gen_dataset<-function(data, alpha){
  Day_of_year<-data$Day_of_year
  Draft_No<-numeric(length(data$Day_of_year))
  b<-rnorm(366, mean = 183, sd=10)
  for (i in 1:366){
    Draft_No[i]<-max(0,min(alpha*i+b[i],366))
  }
  result<-data.frame(Day_of_year,Draft_No)
  return(result)
}

new_data<-gen_dataset(data=data,alpha = 0.1)
```

Question 1.5.b : Plug these data into the permutation

test with B = 200 and note whether it was rejected.

```
set.seed(12345)
new_data<-gen_dataset(data=data,alpha = 0.1)
p_value<-permut_test(new_data,B=200)
cat("The p-value:",p_value)
```

```
## The p-value: 0.005
```

The p-value is lesser than 0.05(threshold at 95% Confidence Interval) so we can reject the null hypothesis.

Question 1.5.c : Repeat Steps 5a–5b for α = 0.01, 0.02, . . . , 1.

```
set.seed(12345)
alpha<-seq(0.01,1,0.01)
teststat=c()
pvalues<-c()
for (i in alpha){
  dataset<-gen_dataset(data=data,alpha = i)
  pvalues<-c(pvalues,permut_test(dataset,B=200))
}
```

The value of the power of the test is given by

```
power<- sum(pvalues<=0.05)/length(pvalues)
cat("Power of the test is ",power)
```

```
## Power of the test is  0.95
```

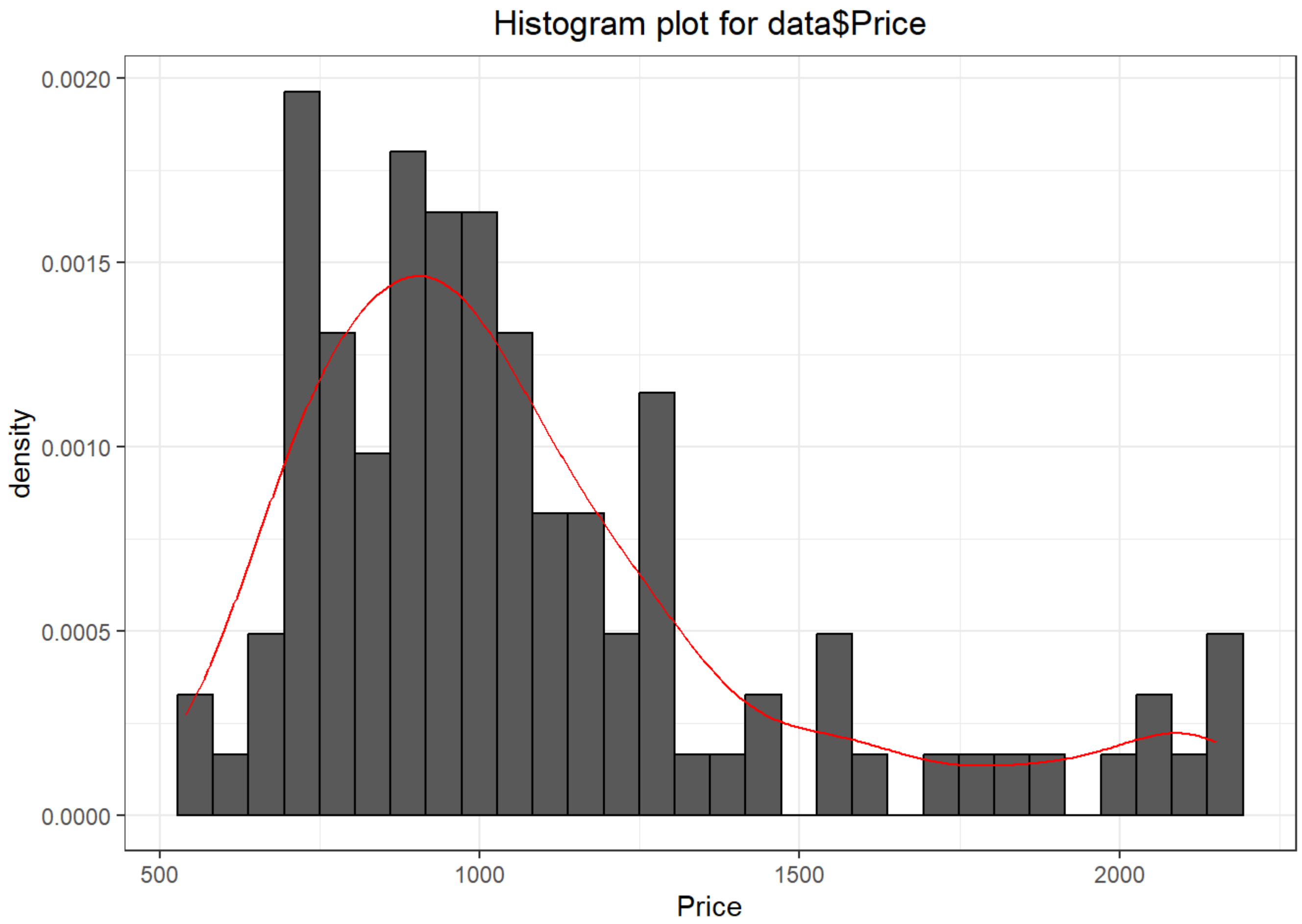
The power of the test is high which means there is high probability of rejecting the null hypothesis when it is really false. So we can say that the quality of our test statistics is good.

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.

1. Plot the histogram of Price. Does it remind any conventional

distribution? Compute the mean price.



The gamma distribution is a conventional choice for this data based on the shape of the density curve observed.

```
#Mean of the price
mean_price<- round(mean(data$Price),3)
cat(paste("mean value of the price is", mean_price))
```

```
## mean value of the price is 1080.473
```

2. Estimate the distribution of the mean price of the house using bootstrap. Determine the bootstrap bias correction and the variance of the mean price. Compute a 95% confidence interval for the mean price using bootstrap percentile, bootstrap BCa, and first{order normal approximation

To correct the bias, we subtract the bootstrap bias estimate from the sample estimate.

$$T = 2D - \frac{1}{B}\sum_{i-1}^{B} T_i^*$$

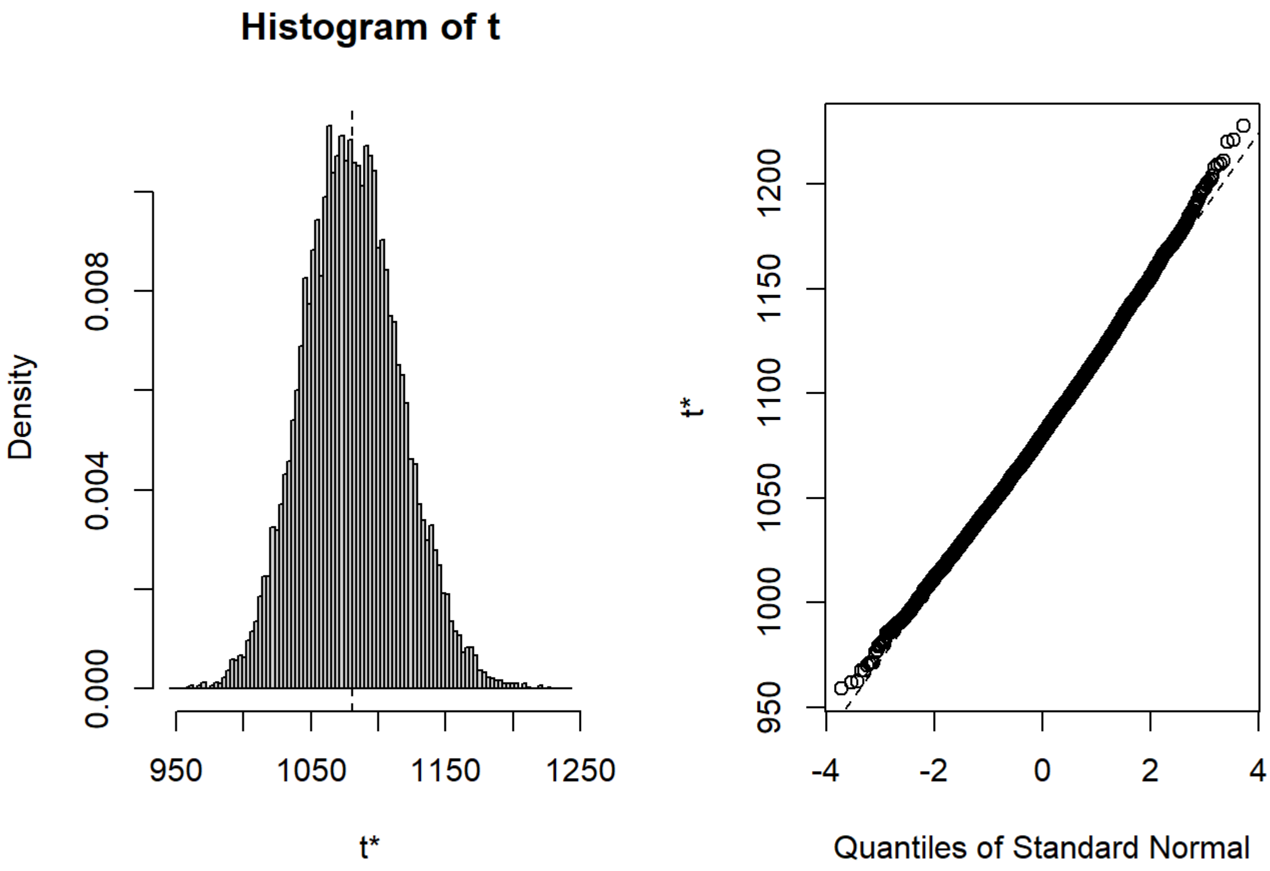
```r
library(boot)
set.seed(12345)
# distribution of the mean price of the house
my_mean <- function(data,indices){
  d <- data[indices,]
  mu <- mean(d$Price)
  return(mu)
}

# Generate bootstrap to estimate distribution of mean price using boot()
boot_res <- boot(data = data, statistic = my_mean, R=10000)
boot_res
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = data, statistic = my_mean, R = 10000)
##
##
## Bootstrap Statistics :
##     original    bias    std. error
## t1* 1080.473 0.2593873    35.99705
```

```
## variance of the mean price is 1295.788
```

```r
# Plotting the bootstrap for estimation of distribution of mean price
plot(boot_res)
```



```
#bootstrap bias correction
bias_cor <- 2*(mean(data$Price)) - mean(boot_res$t)
cat("Bootstrap bias correction : ", bias_cor)
```

```
## Bootstrap bias correction :  1080.213
```

```
# Compute 95% confidence interval for the mean price using bootstrap percentile, BCa and first o
rder normal
con_int <- boot.ci(boot_res,type=c('perc','bca','norm'))
print(con_int)
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 10000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_res, type = c("perc", "bca", "norm"))
##
## Intervals :
## Level      Normal              Percentile            BCa
## 95%   (1010, 1151 )   (1013, 1153 )   (1017, 1159 )
## Calculations and Intervals on Original Scale
```

3. Estimate the variance of the mean price using the jackknife and compare it with the bootstrap estimate

The jacknife is a leave-one-out resampling method that calculates a statistic of interest; it does this successively or iteratively until each observation has been removed. The variance of the estimator as

$$var_{jackknife} = \frac{n-1}{n}\sum_{i=1}^{n}\left(\bar{x}_i - \text{mean}(\bar{x})\right)^2$$

```
#variance of the mean price using the jackknife
mean_jackknife <- c()
n <- length(data$Price)
for(i in 1:n){
  new_price <- data$Price[-i]
  mean_jackknife[i] <- mean(new_price)
}

# Computing variance of the mean price using jackknife
var_jackknife <- ((n-1)/n) * sum((mean_jackknife-mean(mean_jackknife))^2)
print(var_jackknife)
```

```
## [1] 1320.911
```

```
## Variance of the mean price using jackknife :  1320.911
```

```
##
##  Variance of the mean price using bootstrap :  1295.788
```

The standard error of the Jackknife method is larger than that of the Bootstrap method, indicating that the Jackknife method will tend to overestimate the variance and result in a larger error compared to the Bootstrap method.

4. Compare the confidence intervals obtained with respect to their length and the location of the estimated mean in these intervals.

Comparison of the confidence intervals

	BCa	PERCENTILE	NORMAL APPROXIMATION
Length Of Interval	142.39900	140.03500	141.10600
Mid Point	1088.00000	1083.00000	1080.50000
Estimate mean-lower bound	63.87206	67.01727	70.81232
Estimate mean-upper bound	78.52713	73.01750	70.29354

By observing the table. we can say that the bootstrap BCa method has the largest interval among the comparison methods, while the bootstrap percentile method has the smallest interval. The estimated mean for the BCa and percentile methods is closer to the lower bound of the interval and is located on the left side of the interval. In contrast, the estimated mean for the normal approximation method is nearly at the center of the interval.

Appendix

```r
knitr::opts_chunk$set(echo = TRUE)
#loading the data
data<-read.csv("lottery.csv",sep = ";")

#scatter plot
plot(data$Day_of_year,data$Draft_No, xlab = "X=Day of year",ylab = "Y=Draft No",main = "Scatterp
lot of Y versus X")
loess_Y<- loess(Draft_No~Day_of_year, data = data)
Y_hat<-predict(loess_Y,data$Day_of_year)
plot(data$Day_of_year,data$Draft_No, xlab = "X=Day of year",ylab = "Y=Draft No",main = "Scatterp
lot of Y versus X")
points(data$Day_of_year,Y_hat,col="red")
#test statistic
t_stat<-function(data, sample){
  newdata<-data[sample,]
  loess_reg<-loess(Draft_No ~ Day_of_year, data = newdata)
  Y_hat<-as.vector(predict(loess_reg, newdata))
  X_b<-newdata$Day_of_year[which.max(Y_hat)]
  X_a<-newdata$Day_of_year[which.min(Y_hat)]
  Y_hat_a<-Y_hat[which.min(Y_hat)]
  Y_hat_b<-Y_hat[which.max(Y_hat)]
  result<-(Y_hat_b-Y_hat_a)/(X_b-X_a)
  return(result)
}
t_stat(data)
#Non-parametric bootstrapping
set.seed(12345)
library(boot)
bootstrap <-boot(data=data, statistic = t_stat, R=2000)
hist(bootstrap$t, breaks = 200, xlim = c(-0.8,0), main = "Distribution of T", xlab = "T Values")
p=length(bootstrap$t[bootstrap$t<=0])/length(bootstrap$t)
cat("p-value is ",p)
set.seed(12345)
#a function for permutation test
permut_test <- function(data,B){
  k<-dim(data)[1]
  result<-numeric(B)
  for (i in 1:B){
    samp<-sample(data$Day_of_year,k,replace = FALSE)
    sampled_data<-data
    sampled_data$Day_of_year<-samp
    result[i]<-t_stat(sampled_data,1:k)
  }
  test_stat<-t_stat(data,1:dim(data)[1])
  p_value<-mean(abs(result)>=abs(test_stat))
  return(p_value)
}
p_value<-permut_test(data=data,B=2000)
cat("p-value:",p_value)
set.seed(12345)
gen_dataset<-function(data, alpha){
  Day_of_year<-data$Day_of_year
  Draft_No<-numeric(length(data$Day_of_year))
  b<-rnorm(366, mean = 183, sd=10)
  for (i in 1:366){
    Draft_No[i]<-max(0,min(alpha*i+b[i],366))
  }
  result<-data.frame(Day_of_year,Draft_No)
  return(result)
}

new_data<-gen_dataset(data=data,alpha = 0.1)
set.seed(12345)
new_data<-gen_dataset(data=data,alpha = 0.1)
p_value<-permut_test(new_data,B=200)
cat("The p-value:",p_value)
set.seed(12345)
alpha<-seq(0.01,1,0.01)
teststat=c()
pvalues<-c()
for (i in alpha){
  dataset<-gen_dataset(data=data,alpha = i)
  pvalues<-c(pvalues,permut_test(dataset,B=200))
}
power<- sum(pvalues<=0.05)/length(pvalues)
cat("Power of the test is ",power)
#Read the data
data<-read.csv2("prices1.csv")

library(ggplot2)
ggplot(data=data) +
geom_histogram(aes(x = Price, y=..density..), bins=30, col=I("black")) +
geom_density(aes(Price), color = "red") +
ggtitle("Histogram plot for data$Price") +
theme_bw() +
theme(plot.title = element_text(hjust = 0.5), legend.position = "right")

#Mean of the price
mean_price<- round(mean(data$Price),3)
cat(paste("mean value of the price is", mean_price))
library(boot)
set.seed(12345)
# distribution of the mean price of the house
my_mean <- function(data,indices){
  d <- data[indices,]
  mu <- mean(d$Price)
  return(mu)
}

# Generate bootstrap to estimate distribution of mean price using boot()
boot_res <- boot(data = data, statistic = my_mean, R=10000)
boot_res
#variance of the mean price
variance_bootstrap <- round(var(boot_res$t),3)
```

```r
cat(paste("variance of the mean price is", variance_bootstrap))
# Plotting the bootstrap for estimation of distribution of mean price
plot(boot_res)

#bootstrap bias correction
bias_cor <- 2*(mean(data$Price)) - mean(boot_res$t)
cat("Bootstrap bias correction : ", bias_cor)

# Compute 95% confidence interval for the mean price using bootstrap percentile, BCa and first order normal
con_int <- boot.ci(boot_res,type=c('perc','bca','norm'))
print(con_int)

#variance of the mean price using the jackknife
mean_jackknife <- c()
n <- length(data$Price)
for(i in 1:n){
  new_price <- data$Price[-i]
  mean_jackknife[i] <- mean(new_price)
}

# Computing variance of the mean price using jackknife
var_jackknife <- ((n-1)/n) * sum((mean_jackknife-mean(mean_jackknife))^2)
print(var_jackknife)
cat("Variance of the mean price using jackknife : ", var_jackknife)
cat("\n Variance of the mean price using bootstrap : ", variance_bootstrap)

boot_ci<-boot.ci(boot_res)
bca<-boot_ci$bca
perc<-boot_ci$percent
norm<-boot_ci$normal
knitr::kable(
  cbind('BCa'=c('Length Of Interval'=round(bca[5]-bca[4],3),'Mid Point'=(round(bca[5]) + round(bca[4]))/2,
                'Estimate mean-lower bound'=abs(boot_res$t0-bca[4]),
                'Estimate mean-upper bound'=abs(boot_res$t0-bca[5])),
        'PERCENTILE'=c(round(perc[5]-perc[4],3),(round(perc[5]) + round(perc[4]))/2,
                       abs(boot_res$t0-perc[4]),abs(boot_res$t0-perc[5])),
        'NORMAL APPROXIMATION'=c(round(norm[3]-norm[2],3),(round(norm[3]) + round(norm[2]))/2 ,abs(boot_res$t0-norm[2]),abs(boot_res$t0-norm[3]))),
  caption = "Comparison of the confidence intervals")
```