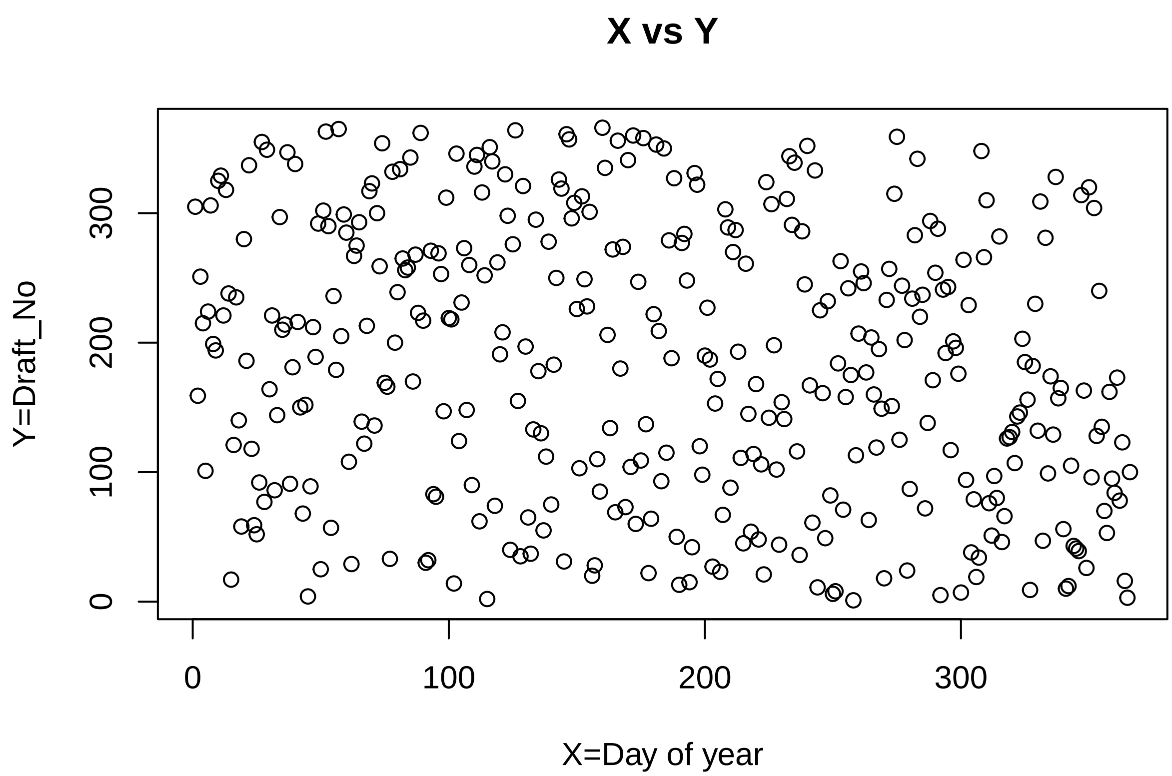
com_stats_lab05_group14

Akshath Srinivas,Samira Goudarzi

2022-12-10

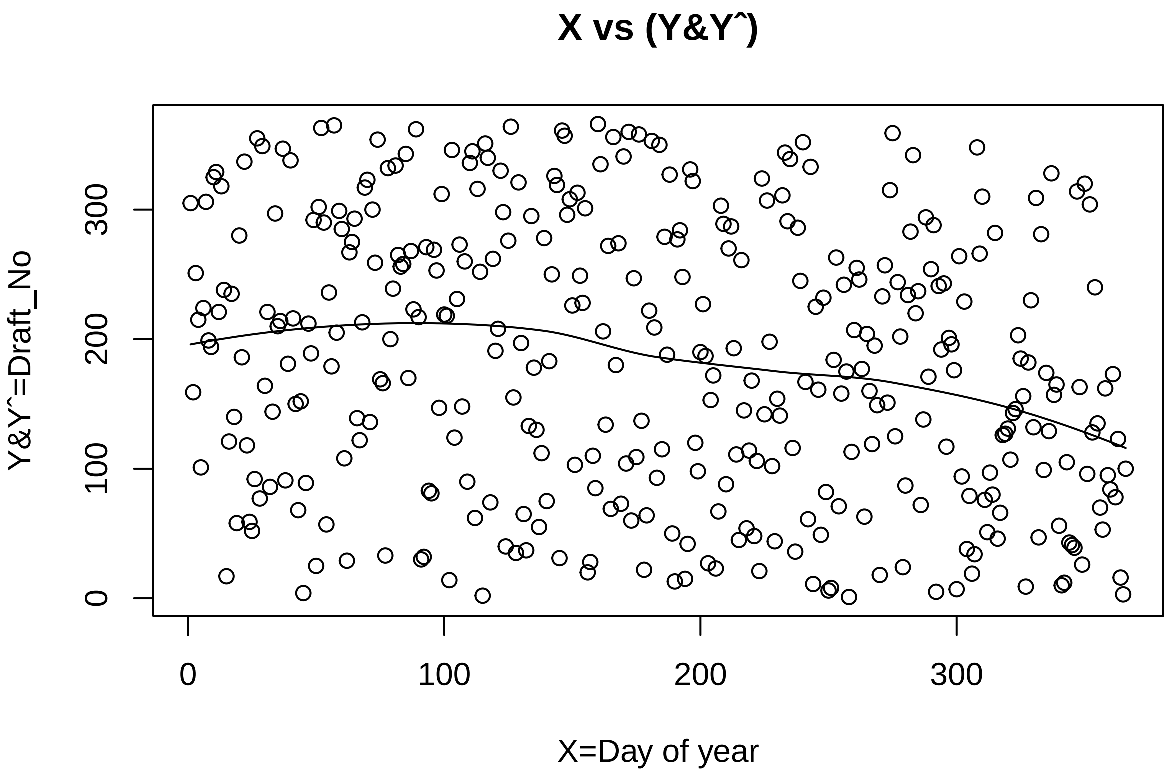
Question 1: Hypothesis testing

sub question 1



From the scatterplot above we can say that lottery is random.

sub question 2



From the scatter plot we can say that lottery is random,but when a line is plotted for $\hat{Y}$ vs X we can see a trend where $\hat{Y}$ predicted from loess() function decreases as the day of the year increases.

sub question 3

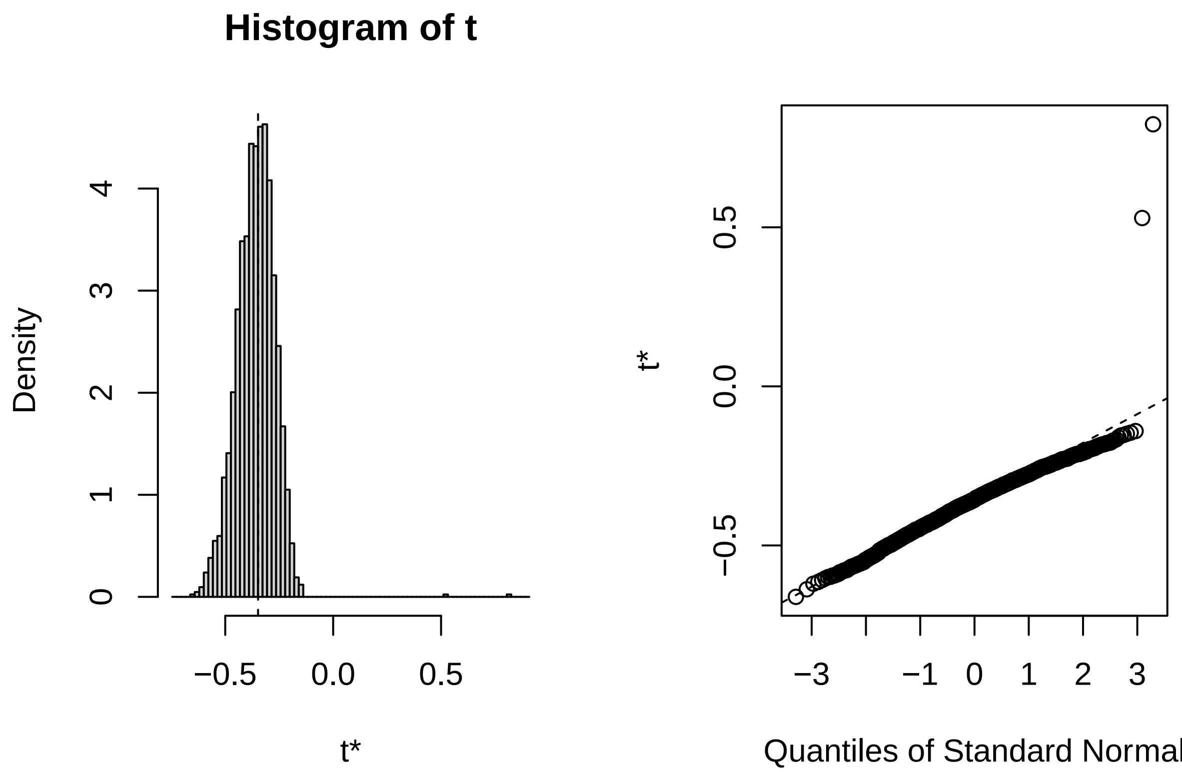
using **loess()** function to predict $\hat{Y}$ and then calculating T statistics will yield a value as shown below.

T statistics was found out by below formula.

$$T = \frac{\hat{Y}(X_b) - \hat{Y}(X_a)}{X_b - X_a}$$

T statistic value(-0.3479163) is significantly different from 0, we can say that lottery is non-random.

```
## T statistic value is: -0.3479163
```




```
## P quantile value is 0.999
```

sub question 4

```
## P value of this permutation test is: 0.1445
```

sub question 5

```
## P value for alpha=0.01, B=200 is : 0.405
```

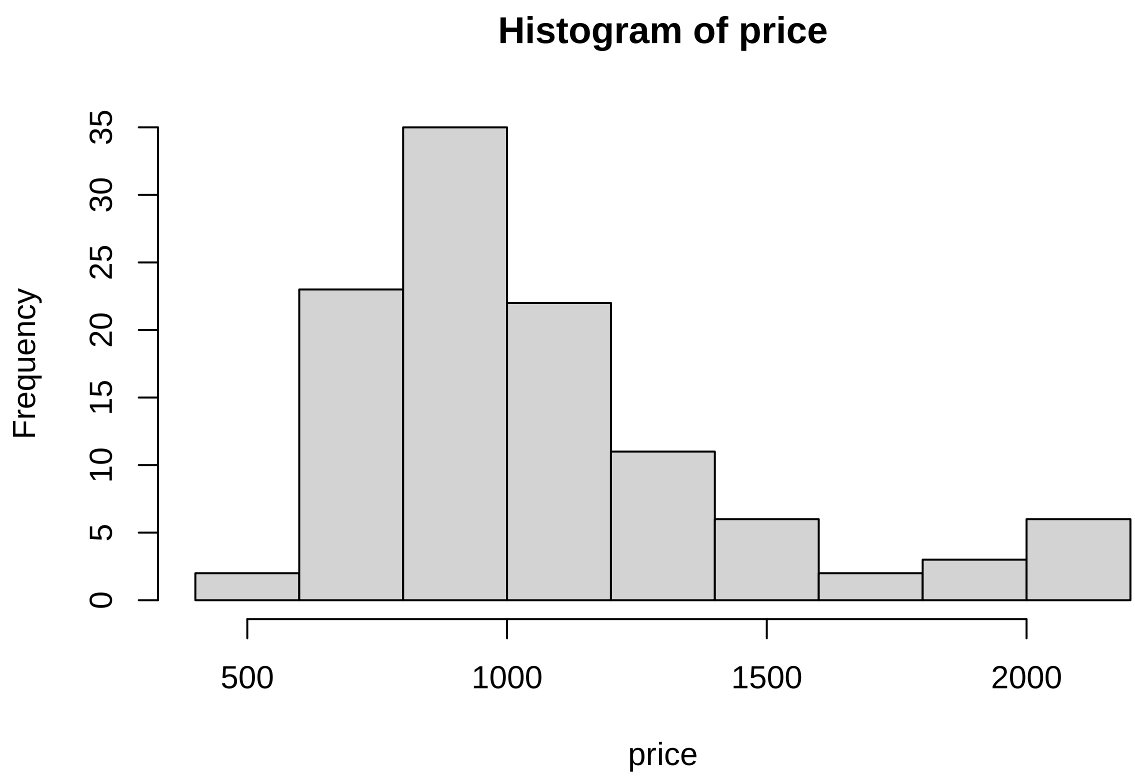
The p value when $\alpha = 0.01$ can be seen above, we can conclude that we failed to reject null hypothesis because the P value we found out is not statistically significant which is not less than 0.05.

```
## Power of the test is: 0.97
```

The power of our test can be seen above, we can conclude that power of our test is good. In this question we are generating a random dataset, it is expected to reject the null hypothesis in all the cases, but in our case around 95% is getting rejected.

Question 2: Bootstrap, jackknife and confidence intervals

sub question 1

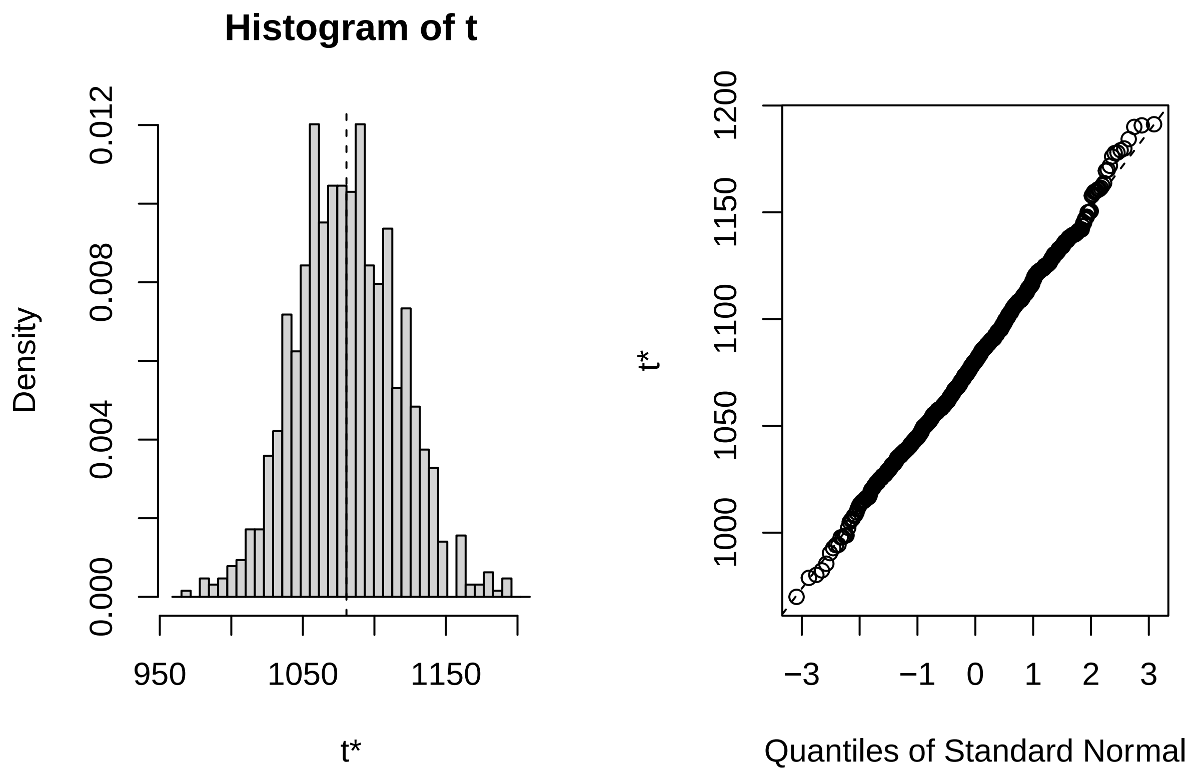



```
## mean price is: 1080.473
```

The above histogram somewhat looks like gamma distribution.

sub question 2

```
## Estimated mean price of houses using bootsrap 1080.853
```



Variance can be calculated by,

$$\widehat{Var[T(\cdot)]} = \frac{1}{B-1}\sum_{i=1}^{B}(T(D_i^*) - \overline{T(D^*)}\ )^2$$

Bias-correction can be calculated by,

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

```
## Bootstrap variance is : 1272.836
```

```
## bias-correction is: 1080.092
```

The confidence intervals for bootstrap percentile, bootstrap BCa, and first–order normal approximation are given below.

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = b, type = c("norm", "perc", "bca"))
##
## Intervals :
## Level      Normal              Percentile            BCa
## 95%   (1010, 1150 )   (1014, 1150 )   (1016, 1160 )
## Calculations and Intervals on Original Scale
```

sub question 3

Variance for jackknife can be calculated by,

$$\widehat{Var[T(\cdot)]} = \frac{1}{n(n-1)}\sum_{i=1}^{n}(T_i^* - J(T))^2$$

where

$$T_i^* = nT(D) - (n-1)T(D_i^*)$$

and

$$J(T) = \frac{1}{n}\sum_{i=1}^{n} T_i^*$$

```
## variance of mean price using Jackknife is : 1320.911
```

comparison of jackknife and bootstrap variance

```
## $jacknife
## [1] 1320.911
##
## $bootstrap
## [1] 1272.836
```

From the comparison above, we can conclude that jacknife overestimates the variance compared to Bootstrap This is because of the error in ensemble mean due to finite size of ensemble.

sub question 4

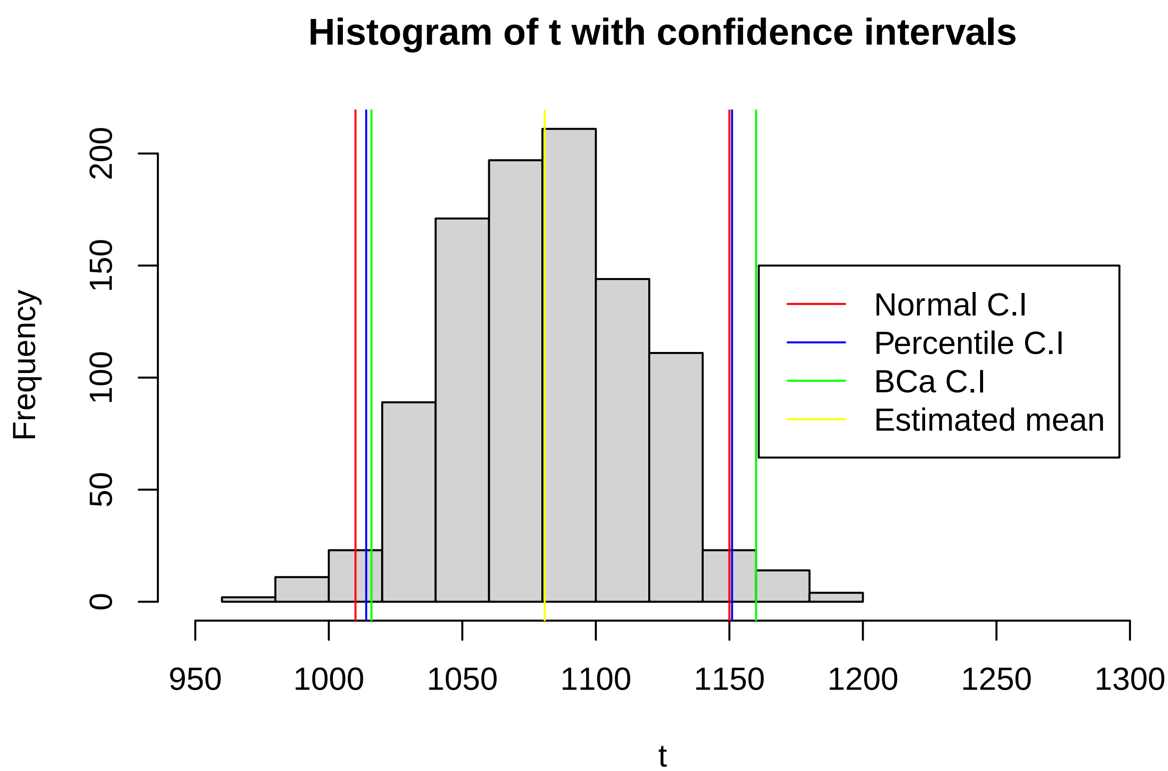
The estimated mean is 1080.473, it is located in all the three confidence intervals which is calculated as shown below.

Normal confidence intervals The location of the estimated mean is more appropriately located at the center for first–order normal approximation confidence intervals. These intervals are calculated by mean and standard deviation.

Percentile confidence interval The upper bound is same as normal approximation interval, but lower bound is slightly more than normal approximation interval. These confidence intervals are calculated by percentile values.

*__BCa confidence interval__ The BCa confidence intervals are slightly moved to right compared to normal and percentile C.I values.

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = b, type = c("norm", "perc", "bca"))
##
## Intervals :
## Level      Normal              Percentile            BCa
## 95%   (1010, 1150 )   (1014, 1150 )   (1016, 1160 )
## Calculations and Intervals on Original Scale
```



Appendix

```
knitr::opts_chunk$set(echo = TRUE)
library(boot)

################ Question 1 #############

lottery<-read.csv2('lottery.csv')
plot(x=lottery[,4],y=lottery[,5],xlab='X=Day of year',ylab='Y=Draft_No',main='X vs Y')


# estimate Y^ of the expected response as a function of X by using a #loess and adding line(y_hat) for 

lottery_lo<-loess(Draft_No~Day_of_year,lottery)
y_hat<-predict(lottery_lo,lottery)
plot(x=lottery[,4],y=lottery[,5],xlab='X=Day of year',ylab='Y&Y^=Draft_No',main='X vs (Y&Y^)')
lines(y_hat)


#Test statistics

set.seed(12345)
test_stat<-function(data,vn){
```

```r
  n_data<-as.data.frame(data[vn,])
  lo<-loess(Draft_No~Day_of_year,n_data)
  y_hat <- lo$fitted
  max_id <- which.max(y_hat)
  min_id <- which.min(y_hat)
  x_b<- n_data[,'Day_of_year'][max_id]
  x_a <- n_data[,'Day_of_year'][min_id]
  y_max <- y_hat[max_id]
  y_min <- y_hat[min_id]
  t <- (y_max - y_min) / (x_b - x_a)
  return(t)
}
act<-test_stat(lottery)
cat('T statistic value is:',act)
set.seed(12345)
m_boot<-boot(lottery,test_stat,R=2000)
plot(m_boot)

#calculating P quantile
cal_p <- m_boot$t[m_boot$t <= 0]
p_th_quantile<-length(cal_p) / length(m_boot$t)
cat("P quantile value is",p_th_quantile , "\n")


#permutation test

prem_t_test<-function(data,B){
  stat<-c()
  n=dim(data)[1]
  perm_data<-data
  stat_p <- c()
  for(i in 1:B){
    perm_data[,'Draft_No']<-sample(perm_data[,'Draft_No'], size = n)
    lo<-loess(Draft_No~Day_of_year,perm_data)
    y_hat <- lo$fitted

    max_id <- which.max(y_hat)
    min_id <- which.min(y_hat)
    x_b <- perm_data[,'Day_of_year'][max_id]
    x_a <- perm_data[,'Day_of_year'][min_id]
    y_max <- y_hat[max_id]
    y_min <- y_hat[min_id]
    stat <- c(stat,(y_max - y_min) / (x_b - x_a))
  }

  stat_p <- c(stat_p,mean(abs(stat) >= abs(test_stat(data))))
  return(list(stat_p,stat))
}
permutation_test<-prem_t_test(lottery,2000)

cat('P value of this permutation test is:', permutation_test[[1]])
```

```r
# power of the test
power_test<-function(alpha){
  Day_of_year<-lottery[,'Day_of_year']
  new_df<-data.frame(Day_of_year=Day_of_year)
  Draft_No<-c()
  for (x in 1:nrow(new_df)){
    beta <- rnorm(1, mean = 183, sd = 10)
    Draft_No <- c(Draft_No, max(0, min((alpha*x + beta), 366)))
  }
  new_df<-cbind(new_df,Draft_No=Draft_No)
  return(new_df)
}
p_value<-prem_t_test(power_test(0.01),200)[[1]]
cat('P value for alpha=0.01, B=200 is :',p_value)

#power
p_values<-c()
power_test_t<-c()
alpha<-seq(0.01,1,0.01)
for(i in 1:length(alpha)){
  p<-power_test(alpha[i])
  p_val<-prem_t_test(p,200)[[1]]
  p_values<-c(p_values,p_val)
  power_test_t<-c(power_test_t,ifelse(p_val>=0.05,'Accept','Reject'))
}

#power of test
power<-length(which(power_test_t=='Reject'))/length(power_test_t)

cat('Power of the test is:',power)


################ Question 2 #############

price<-read.csv2('prices1.csv')

hist(price[,1],main='Histogram of price',xlab='price')

cat('mean price is:',mean(price[,1]))


# bootstrapping
library(boot)
set.seed(12345)
stat<-function(m,i){
  return(mean(m[i]))
}
W<-1000
set.seed(12345)
b<-boot::boot(price[,1],stat,R=W)


cat('Estimated mean price of houses using bootsrap',mean(b$t))
```

```r
plot(b)

#variance
variance<-1/(W-1)*sum((b$t-mean(b$t))^2)
cat('Bootstrap variance is :',variance)
bias_correction<-2*(b$t0)-mean(b$t)
cat('bias-correction is:',bias_correction)



#confidence intervals
c_i<-boot.ci(b,type=c('norm','perc','bca'))
c_i


library(bootstrap)
n<-nrow(price)
param<-function(i,data){
  return(mean(data[-i]))
}
jackknife_result<-jackknife(1:n,param,price[,1])
j_t<-(1/n)*sum(jackknife_result$jack.values)
var_jack_knife<-(1/(n*(n-1)))*(sum((jackknife_result$jack.values-j_t)^2))
cat('variance of mean price using Jackknife is :',var_jack_knife)


list(jacknife=var_jack_knife,bootstrap=variance)


#confidence interval
c_i

#histogram of mean of t and with confidence intervals with normal approximation
hist(b$t, main='Histogram of t with confidence intervals', xlab='t',xlim = c(950,1300))
abline(v=1010,col='red')
abline(v=1150,col='red')
abline(v=mean(b$t),col='yellow')
abline(v=1014,col='blue')
abline(v=1151,col='blue')
abline(v=1016,col='green')
abline(v=1160,col='green')
legend(x = 1161, y = 150,
       legend = c('Normal C.I', 'Percentile C.I','BCa C.I','Estimated mean'),
       col = c('red', 'blue','green','yellow'), lty = c(1,1))
```