Lab0

Jas Sur

1/26/2022

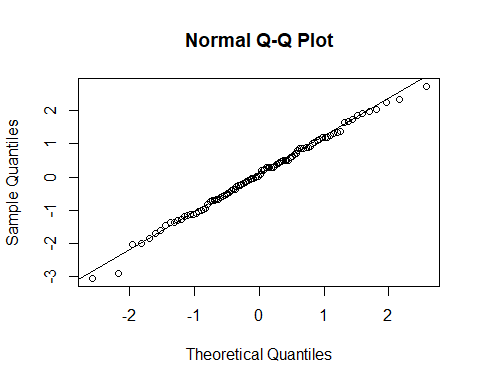
# Lab 0 Problems

## Question 1.a

### Generate a random sample of 100 from t-distribtion, degree of freedom 10.

### Check qqnorm(); qqline(), Shapiro test. Remarks

sample.t <- rt(100, df=10)  
qqnorm(sample.t)  
qqline(sample.t)



shapiro.test(sample.t)

##   
## Shapiro-Wilk normality test  
##   
## data: sample.t  
## W = 0.99472, p-value = 0.9672

1. Null hypothesis (H0 sample data is normal)
2. Alternative hypothesis (H1 sample data is not normal)

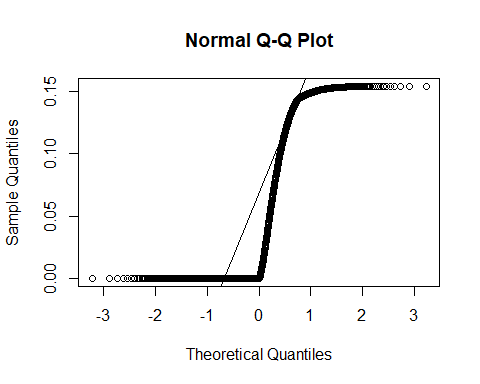
Since the pvalue > 0.05, which means that the null hypothesis that values are normal is accepted and the alternative hypothesis is rejected. \*\*\*

## Question 1.b

### Generate a random sample of 100 from a Chi-square distribution, df =5.

### Perform same procedures as in (a) . Remarks.

sample.1b <-seq(-4, 4, 0.01)  
sample.chi = dchisq(sample.1b, df=5)  
qqnorm(sample.chi)  
qqline(sample.chi)



shapiro.test(sample.chi)

##   
## Shapiro-Wilk normality test  
##   
## data: sample.chi  
## W = 0.728, p-value < 2.2e-16

1. Null hypothesis (H0 sample data is normal)
2. Alternative hypothesis (H1 sample data is not normal)

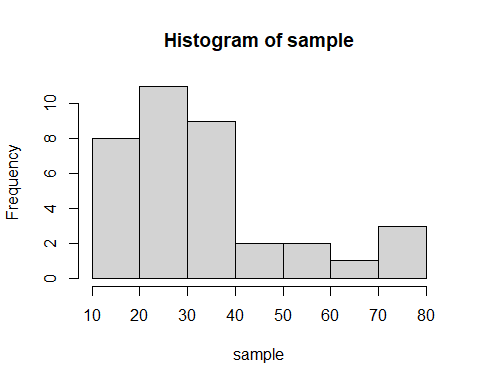
Since the pvalue < 0.05, which means that the null hypothesis that values are normal is rejected and the alternative hypothesis is accepted.

## Question 2.a

### Write a 95%-CI for the population mean.

Confidence interval for population mean can be determined using the following

sample <-c(26.4,23.5,25.4,22.9,25.2,39.2,25.5,31.9,26.0,44.6,35.5,38.6,  
 30.1,31.0,30.8,32.8,47.7,39.1,55.3,50.7,73.8,71.1,68.4,77.1,  
 19.4,19.3,18.7,19.0,23.2,21.3,23.2,19.9,18.9,19.8,19.6,21.9)  
hist(sample)



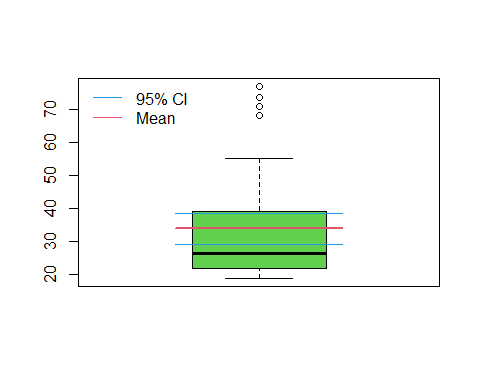
n<-length(sample)  
xbar<-mean(sample)  
s<-sd(sample)  
sprintf("mean %f, std dev %f, sample size %d", xbar, s, n)

## [1] "mean 33.800000, std dev 16.897489, sample size 36"

margin <- qt(0.95,df=n-1)\*s/sqrt(n)  
low\_ci <- xbar - margin  
  
high\_ci <- xbar + margin  
sprintf("confidence interval for the population mean is (%f, %f)", low\_ci, high\_ci)

## [1] "confidence interval for the population mean is (29.041745, 38.558255)"

boxplot(sample, col = 3)  
lines(c(0.75,1.25),c(low\_ci,low\_ci),col=4)  
lines(c(0.75,1.25),c(xbar,xbar),col=2,lwd=2)  
lines(c(0.75,1.25),c(high\_ci,high\_ci),col=4)  
legend("topleft", c("95% CI", "Mean"), lty=1,col = c(4, 2),bty ="n")



95 % Confidence interval is reasonable compare to the sample mean

### What assumption about population for the work, suppose the sample is random.

The assumptions would be that the data is normal and the errors are constant.

## Question 2.b

### Write a 95%- CI for population standard deviation.

Confidence interval for population standard deviation can be determined using the following formula confidence interval:

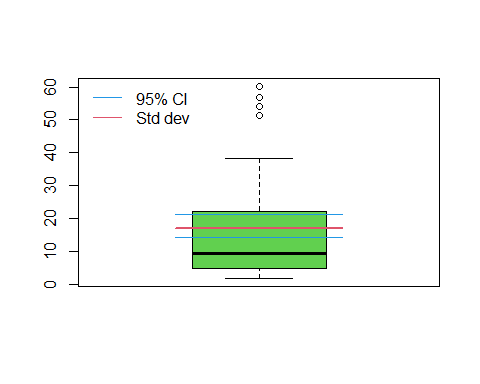
sample <-c(26.4,23.5,25.4,22.9,25.2,39.2,25.5,31.9,26.0,44.6,35.5,38.6,  
 30.1,31.0,30.8,32.8,47.7,39.1,55.3,50.7,73.8,71.1,68.4,77.1,  
 19.4,19.3,18.7,19.0,23.2,21.3,23.2,19.9,18.9,19.8,19.6,21.9)  
n <- length(sample)  
xbar<-mean(sample)  
s<-sd(sample)  
sprintf("population sd %f, size %d, mean %f", s, n, xbar)

## [1] "population sd 16.897489, size 36, mean 33.800000"

left <- qchisq(p=.05, df=n-1, lower.tail=FALSE)  
right <- qchisq(p=.95, df=n-1, lower.tail=FALSE)  
low\_ci <- sqrt((n-1)\*s^2/left)  
high\_ci <- sqrt((n-1)\*s^2/right)  
  
sprintf("Confidence interval for population std dev is %f, %f", low\_ci, high\_ci)

## [1] "Confidence interval for population std dev is 14.165551, 21.091275"

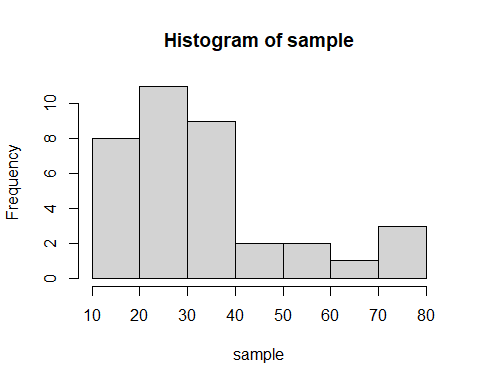
boxplot(sample-s, col = 3)  
lines(c(0.75,1.25),c(low\_ci,low\_ci),col=4)  
lines(c(0.75,1.25),c(s,s),col=2,lwd=2)  
lines(c(0.75,1.25),c(high\_ci,high\_ci),col=4)  
legend("topleft", c("95% CI", "Std dev"), lty=1,col = c(4, 2),bty ="n")

  ***A confidence interval for a standard deviation is a range of values that is likely to contain true population standard deviation with a certain level of confidence.***

# Question 3

### Run the following code

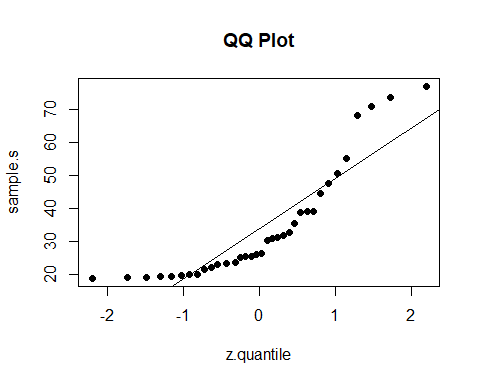
sample <-c(26.4,23.5,25.4,22.9,25.2,39.2,25.5,31.9,26.0,44.6,35.5,38.6,  
 30.1,31.0,30.8,32.8,47.7,39.1,55.3,50.7,73.8,71.1,68.4,77.1,  
 19.4,19.3,18.7,19.0,23.2,21.3,23.2,19.9,18.9,19.8,19.6,21.9)  
hist(sample)



sample.s <-sort(sample) #sort data increasing  
rank <- rank(sample.s) #rank data from 1 to 36  
size <- length(sample.s) # size of data  
p <- (rank-.5)/size #cummulative prob of data  
z.quantile <- qnorm(p) # Standard Normal quantiles with such probability  
  
data <- cbind(rank, sample.s, p, z.quantile)  
data

## rank sample.s p z.quantile  
## [1,] 1.0 18.7 0.01388889 -2.20041058  
## [2,] 2.0 18.9 0.04166667 -1.73166440  
## [3,] 3.0 19.0 0.06944444 -1.47994139  
## [4,] 4.0 19.3 0.09722222 -1.29754293  
## [5,] 5.0 19.4 0.12500000 -1.15034938  
## [6,] 6.0 19.6 0.15277778 -1.02459239  
## [7,] 7.0 19.8 0.18055556 -0.91324993  
## [8,] 8.0 19.9 0.20833333 -0.81221780  
## [9,] 9.0 21.3 0.23611111 -0.71886805  
## [10,] 10.0 21.9 0.26388889 -0.63140189  
## [11,] 11.0 22.9 0.29166667 -0.54852228  
## [12,] 12.5 23.2 0.33333333 -0.43072730  
## [13,] 12.5 23.2 0.33333333 -0.43072730  
## [14,] 14.0 23.5 0.37500000 -0.31863936  
## [15,] 15.0 25.2 0.40277778 -0.24616365  
## [16,] 16.0 25.4 0.43055556 -0.17495994  
## [17,] 17.0 25.5 0.45833333 -0.10463346  
## [18,] 18.0 26.0 0.48611111 -0.03482132  
## [19,] 19.0 26.4 0.51388889 0.03482132  
## [20,] 20.0 30.1 0.54166667 0.10463346  
## [21,] 21.0 30.8 0.56944444 0.17495994  
## [22,] 22.0 31.0 0.59722222 0.24616365  
## [23,] 23.0 31.9 0.62500000 0.31863936  
## [24,] 24.0 32.8 0.65277778 0.39283081  
## [25,] 25.0 35.5 0.68055556 0.46925288  
## [26,] 26.0 38.6 0.70833333 0.54852228  
## [27,] 27.0 39.1 0.73611111 0.63140189  
## [28,] 28.0 39.2 0.76388889 0.71886805  
## [29,] 29.0 44.6 0.79166667 0.81221780  
## [30,] 30.0 47.7 0.81944444 0.91324993  
## [31,] 31.0 50.7 0.84722222 1.02459239  
## [32,] 32.0 55.3 0.87500000 1.15034938  
## [33,] 33.0 68.4 0.90277778 1.29754293  
## [34,] 34.0 71.1 0.93055556 1.47994139  
## [35,] 35.0 73.8 0.95833333 1.73166440  
## [36,] 36.0 77.1 0.98611111 2.20041058

#scatterplot of x=Z quantiles, y= data sorted  
plot(x=z.quantile, y=sample.s, pch=16, main="QQ Plot")   
abline(lm(sample.s ~ z.quantile))



The QQ plot suggests that the data is not normal