R code

```
Price=c(9,12,15,6,10,8,45,17,10,7,6,20,7,40,15,10,10,18,19,10,10,7,55,7,55,7,9,10,10,8,
7,6,7,40,17,7,6,11,50,7,6,40,7,17,70,7,10,55,48,7,10,7,48,7,60,15,8,7,7,7,15,12,6,20,55,2
0,8,25,15,7,6,6,6,7,6,7,20,10,49,15,14,10,7,10,7,10,19,9,7,15,9,10,9,9,15,15,55,10,9,9)
num = length(Price)
sprintf("Amount of data is %d",num) ###size
print("Price")
Price
###MeanData and Standard Diviation
meanData= mean(Price)
sdData= sd(Price)
sprintf("Mean of Data is %.4f",meanData)
sprintf("Standard diviaionis %.4f",sdData)
medianData= median(Price)
sprintf("Median of Data is %.4f",medianData)
###Normality Distribution
SE = sdData/sqrt(num)
sprintf("Standard Error is %f",SE)
E = qt(0.975, df=num-1)*SE
E # margin of error
meanData+ c(-E,E)
hist(Price, probability=TRUE)
### Normality test by Shapiro-Wilk's method



shapiro.test(Price)
### Compute Z
mufood= 2.6 # hypothesized value
z = (meanData- mufood) / SE
z ### Compute Critical value
alpha = .05
z.half.alpha= qnorm(1-alpha/2)
c(-z.half.alpha, z.half.alpha)
# test statistic
### Compute p-value
pvalDataFood= 2*pnorm(z) # lower tail
pvalDataFood # two?tailedp?value
##################################
### Correlation ##
Data2=c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,3,3,3,4,4)
Data3=c(4,5,4,5,2,3,3,4,5,4,4,4,4,4,3,3,4,5,5,4,5,5,5,5,5,3,4,4,4,4,5,5,4,4,4,3,4,4,5,4,3,5,
4,4,2,5,2,5,3,4,1,3,3,1,5,5,4,5,4,4,5,5,5,5,4,5,4,5,5,5,3,4,1,3,3,4,3,5,5,3,4,4,5,4,5,5,5,5,3,
5,5,5,5,3,5,5,5,5,2,5) ###Data like
print("Data2")
Data2
cor(Price,Data3)
```

```
### Linear Simple Regression ###
grade.lm = lm(Price ~ Data3)
coeffs= coefficients(grade.lm); coeffs
summary(grade.lm)$r.squared
summary(grade.lm)
sprintf("Correlation between groups %f",cor(Price,Data3))
sprintf("Simple regression equation is %f", coefficients(grade.lm))
sprintf("Coefficient of decision is %f",summary
 (grade.lm)$r.squared)
meanData3 = mean(Data3)
print("Data3")
Data3
muStore= 15.4 # hypothesized value
s = 2.5 # sample standard deviation
t = (meanData3 - muStore) / ( s / sqrt(num))
t # test statistic
### Compute Critical value
alpha = .05
t.half.alpha= qt(1-alpha/2, df=num-1)
c(-t.half.alpha, t.half.alpha) # critical value
### Compute p-value
pvalDataStore= 2 * pt(t, df=num-1) # lower tail
sprintf("Pvalof DataStore is %.4f",pvalDataStore) # two?tailedp?value
```