HOMEWORK -2

QUESTION 1:

In the model when we modify X and Y from the original values to a linear combination of them, the b0 - intercept and b1- slope change, but the correlation co-efficient remains same. We know that, the squared correlation coefficient (R^2) is the proportion of variance in Y that can be accounted for by knowing X. Conversely, it is the proportion of variance in X that can be accounted for by knowing Y. Thus in our case we noticed that the correlation coefficient is invariant (doesn't change) under a linear transformation of either X and/or Y. What this means essentially is that changing the scale of either the X or the Y variable will not change the size of the correlation coefficient, as long as the transformation conforms to the requirements of a linear transformation.

	b_0	b_1	R^2	FITTED EQUATION	SSE	OUTPUT
X Y	8.5209	0.0059	**0.2246**	$\widehat{Y}_i$ **= 8.5209 + 0.005891*X**	317.315	For a given X and Y, we obtained slope as 0.00589 and intercept as 8.52.
X*192 **Y*192**	1636.02	0.0059	**0.2246**	$\widehat{Y}_i$ **= 1636.019 + 0.005891*X**	11697502	Since both X and Y are multiplied by a constant > 0. The slope remains same. But the intercept is increased to 1636. SSE has increased since the dependent variable has been multiplied by a number.

X **Y*47**	400.48	0.2769	**0.2246**	$\widehat{Y}_i$= **400.4838 + 0.2769*X**	700948.9	Here since X is constant, but Y is multiplied by numeric value > 0. The slope changes also the y-intercept
X*12 **Y**	8.520931	0.000491	**0.2246**	$\widehat{Y}_i$= **8.520931 + 0.000491*X**	317.315	Here X is multiplied by 12, the Y intercept remains same, but the slope varies. Here the dependent variable is not changing and hence the SSE remains same as case 1.

Question 2:

For the data the SLR model is estimated by the regression equation

$$\hat{Y}_i = 168.6 + 2.034X$$

By observing the data of the model, the coefficient of determination R^2 = 0.9731, which is closer to 1 , is a measure of goodness of fit of the regression line. The F-statistic calculated by using R is 506, which is high. The correlation co-efficient is high and also the F-statistic, which is good. And p-value of the co-efficient(s) is less than significance level $\alpha = 0.05$, imply whether or not predictor variable is statistically significant in the model.

The scatter plot of the Plastic hardness in Brinell units (Y) vs the number of hours elapsed since the plastic was molded (X), reveals that the linear regression function is appropriate for the data being analyzed. The observations are independent. The relationship is linear.

Running Shapiro-Wilk test for normality gives the p-value greater than α- level = 0.05 hence conclude that H_0 , that the distribution of errors is normal.

```
#####################################################################
shapiro.test(lm_plasic$residuals)

data:  lm_plasic$residuals

W = 0.97348, p-value = 0.8914

#####################################################################
```

Also the residual plot, which is the plot of residuals vs the independent variable X suggests that

From the residual plot we observe that

- **Homoscedastic**: Residuals tend to fall within a horizontal band centered around regression line. And no systematic pattern of deviation around 0. The error terms seem to be homoscedastic.
- The error terms are independent, and random. They do not depend on predictor variable.
- **Outliers**: There were no outliers. The linear regression line passes through most of the points.
- **Normal Distributed**: The plot of histogram of the residuals reveals close to normal distribution. Running the Shapiro Wilk test. Conclude normally distributed. Also the histogram of residuals shows normal distribution.
- The number of observations were small. Small sample size.

RCode for plots

```
#####################################################################

scatterplot(df_plastic$Num_of_hrs                    ,
df_plastic$Plastic_hardness, xlab ="number of hrs"
, ylab="pastic ahrdness in Brinel units", main="plasit
hardness vs no of hrs")
  abline(lm_plasic)
hist(residuals(lm_plasic))
```

```
scatterplot(df_plastic$Num_of_hrs                    ,
residuals(lm_plasic),    xlab    ="number    of    hrs",
ylab="Residuals", main="Residual plot")
  abline(h=0)
  residualPlot(lm_plasic)
#####################################################################
```

R Results:

On running the Brown-Forsythe test to determine the whether or not error terms varies with the independent variable X. We divide the input into 2 groups, group1 being X <= 24 and group2 being X > 24. The 2 sample test statistic is given by

$$t_{BF}^{*} = \frac{\bar{d}_1 - \bar{d}_2}{s\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

and variance is

$$s^2 = \frac{\sum(d_{i1} - \bar{d}_1)^2 + \sum(d_{i2} - \bar{d}_2)^2}{n-2}$$

Using R calculated the value of s^2 is 2.97067 and test statistic t_{BF}^{*} is 0.8557853. The $t_{value} = qt(.975, 14)$ is 2.144787.

The **decision** rule is.

If $|t_{BF}^{*}| \leq$ 2.144787 conclude the error variance is constant.

If $|t_{BF}^{*}| >$ 2.144787 conclude the error variance is not constant.

Since $|t_{BF}^{*}| =$ 0.8557853$\leq$ 2.144787, we **conclude** that the error variance is constant and does not vary with the level of X.

The two sided P-value 0.4065253 is greater than specified significance value α-level = 0.05. We conclude H_0.

R-Code: Manually calculating BF test

```
#####################################################################
```

```
plastic_data <- read.table("plasticHardness.txt", sep
="", header=FALSE)
df_plastic<-data.frame(Num_of_hrs=plastic_data$V2,
Plastic_hardness=plastic_data$V1)
lm_plasic <- lm(Plastic_hardness ~ Num_of_hrs , data =
df_plastic)
df_plastic$Residuals <- residuals(lm_plasic)
Group1 <- df_plastic[which(df_plastic$Num_of_hrs <=
24),"Residuals"]
Group2 <- df_plastic[which(df_plastic$Num_of_hrs >
24),"Residuals"]
m1 <- median(Group1)
m2 <- median(Group2)
Di_1 <-abs(Group1 - m1)
Di_2 <-abs(Group2 - m2)
D1 <- mean(Di_1)
D2 <- mean(Di_2)
n<- length(Group1) + length(Group2)
n1 <- length(Group1)
n2 <- length(Group2)
sum((Di_1 - D1)^2)
sum((Di_2 - D2)^2)
n
s_2 <- (sum((Di_1 - D1)^2) + sum((Di_2 - D2)^2))/(n-2)
s <- sqrt(s_2)
T_BF <- (D1-D2)/(s*sqrt(1/n1 + 1/n2))
t_value <-qt( 0.975, n-2 )
p_value <-2 * ( 1 - pt( T_BF, 14))
#####################################################################
```

Alternatively used the levene test in R to get the test statistic and the p-value. The two sided P-value 0.4065 is greater than specified significance value α- level = 0.05. We conclude H_0. The error variance is constant and does not vary with the level of X.

R-code: Levene test

```
#####################################################################

library("MASS")
library("car")
```

```
library("lmtest")
df_plastic$group <- 0
df_plastic$group[df_plastic$Num_of_hrs <= 24] <- '1'
df_plastic$group[df_plastic$Num_of_hrs > 24] <- '2'
lm_plasic <- lm(Plastic_hardness ~ Num_of_hrs , data =
df_plastic)
leveneTest(lm_plasic$residuals, center = median, group =
df_plastic$group)
#####################################################################
```

R results:

Levene's Test for Homogeneity of Variance (center = median)

p-value =0.4065

QUESTION 3:

The co-efficient of the regression equation are calculated by using

$$b_1 = \frac{s_{xy}}{s_x}$$

$$b_0 = \bar{Y} - b_1\bar{X}$$

By using R, OLS estimates are $b_0 =$ -0.5801567 and $b_1 =$ 15.03525. The fitted regression equation is given by .

$$\hat{Y} = -0.5801567 + 15.03525\,X$$

By observing the data of the model, the coefficient of determination R^2 = 0.7501, which is closer to 1 , is a measure of goodness of fit of the regression line. The F-statistic calculated by using R is 174.1, which is quite high. The correlation co-efficient is high and also the F-statistic, which is good.

And p-value of the co-efficient(s) is less than significance level α- level = 0.05 , imply whether or not predictor variable is statistically significant in the model.

The graph of the Muscle mass (Y) vs Age (X), reveals that the linear regression function is appropriate for the data being analyzed. From the scatter plot, the relationship between X and Y is linear and are independent.

Observing the residual plot, which is the plot of residuals vs the independent variable X suggests that

Running Shapiro-Wilk test for normality gives the p-value greater than α- level = 0.05 hence conclude that H_0 , that the distribution of errors is normal.

```
######################################################################
shapiro.test(lm_mmdata$residuals)

data:  lm_mmdata$residuals

W = 0.97958, p-value = 0.4112

######################################################################
```

- **Homoscedastic**: Residuals tend to fall within a horizontal band centered around regression line. And no systematic pattern of deviation around 0. The error terms seem to be homoscedastic.
- The error terms are independent, and random. They do not depend on predictor variable.
- **Outliers**: There were no outliers. The linear regression line passes through most of the points.
- **Normal Distributed**: The plot of histogram of the residuals reveals close to normal distribution. Also running the Shapiro-Wilk test reveals normally distributed. Running the Shapiro Wilk test.
- The number of observations were around 60 in the sample, which is not that small.

Using Breush-Pagan test to determine the whether or not error terms varies with the independent variable X at specified significance value α- level = 0.01. Using R commands

Rcode:

```
######################################################################
mm_data <- read.table("muscleMass.txt", sep ="", header=FALSE)
```

```
mm_data_swap <- data.frame(mm_data$V2, mm_data$V1)
mm_data_swap$Age <- mm_data$V2
mm_data_swap$MMass <- mm_data$V1
lm_mmdata <- lm(MMass ~ Age , data = mm_data_swap)
ncvTest(lm_mmdata)
##Plotting code
scatterplot(mm_data_swap$Age , mm_data_swap$MMass, xlab
="Age"
     , ylab="Muscle Mass", main="plot of Age vs Muscle
mass ")
abline(lm_mmdata)
##Residual Plot
scatterplot(mm_data_swap$Age   ,   residuals(lm_mmdata),
xlab ="Age"
     , ylab="Residuals", main="Residual plot")
abline(h=0)
residualPlot(lm_mmdata)
hist(residuals(lm_mmdata))
qqPlot(lm_mmdata)
#####################################################################
```

R Results:

Chisquare = 3.817125 Df = 1 p = 0.05073122

qchisq (.99,1) is 6.634897

pchisq (.99,1) is 0.6802576

To control α- level = 0.01, we require $\text{Chisq}_{\text{value}}$ (.99; 1) = 6.634. Since test stat is $X^2_{BP} = 3.817125 \leq 6.634$, we conclude H_0, that the error variance is constant. The P-value of this test is 0.6802576 < α= 0.01, hence conclude H_0 constancy of the error variance.

Based on the Breush-Pagan test and the initial scatter plot, residual plot reveal that the error variance is constant and hence homoscedastic.

QUESTION 4:

E(Y_h), is a single mean response. Its estimated using the t distribution and significance level α. The confidence interval estimates the uncertainty in our estimate for conditional mean. $1 - \alpha$ confidence limits for mean response is given

by

$$\hat{Y}_h \pm t(1-\frac{\alpha}{2}; n-2)s(\hat{Y}_h) \quad (1)$$

$$s(\hat{Y}_h) = \sqrt{MSE\left[\frac{1}{n} + \frac{(\hat{Y}_h - \bar{X})^2}{\sum(X_i - \bar{X})^2}\right]} = 0.880281$$

$\hat{Y}_h$ is E(Y|X = X_h), conditional mean which is a single value not a random variable. We are trying to get the interval for the estimate of the mean response when the predictor variable is held constant at X = X_h. Here we are trying to predict the mean of the response.

When we try to predict an individual outcome, actual value of Y, we use prediction interval. The range of prediction interval at α significance level is given by

$$\hat{Y}_h \pm t\left(1-\frac{\alpha}{2}; n-2\right)s\{pred\} \quad (2)$$

Basic idea of prediction interval is to choose a range in distribution of Y wherein most of the observations fall, and then declare that the next observation will fall in this range. Here we take into account two elements.

- Variation in possible location of distribution of Y at X= X_h i.e. MSE
- Variation within the probability distribution of Y, i.e. $s^2\{\hat{Y}_h\}$

$$s^2\{pred\} = \text{MSE} + s^2\{\hat{Y}_h\}$$

$$3.412413 = 2.637518 + 0.7748946$$

$$s^2\{pred\} = \text{MSE}\left[1 + \frac{1}{n} + \frac{(X_h - \bar{X})^2}{\sum(X_i - \bar{X})^2}\right] = 1.847272$$

A confidence interval represents an inference on a parameter and is an interval that is intended to cover the value of the parameter. A prediction interval is a statement about the value taken by a random variable, the new observation. It is much harder to predict one response than to predict a mean response, so the prediction interval must be wider. The extra 1 in the formula for variance $s^2\{pred\}$ makes the interval wider. The MSE accounts for over 77.22% (2.637518/3.412413) of the estimated predicted variance.

When we compare formula (1) and (2) for the upper limits of CI and PI. We get the difference between (prediction upper limit – confidence upper limit)

$\left[\hat{Y}_h + t\left(1-\frac{\alpha}{2}; n-2\right) s\{pred\}\right]$ - $\left[\hat{Y}_h + t(1-\frac{\alpha}{2}; n-2) s(\hat{Y}_h)\right]$

Gives $t\left(1-\frac{\alpha}{2}; n-2\right)\left[s\{pred\} - s(\hat{Y}_h)\right]$

From R using qt (0.985, n-2) = t-value = 2.198346 , n = 110

2.198346 * (1.847272 - 0.880281) = 2.12578, this accounts for the difference in upper limit and lower limit of confidence interval.

R code:

```
#####################################################################
senic_data <- read.csv('SENIC_data.csv', header = FALSE)
length_of_stay <- senic_data$V2
infection_rate <- senic_data$V4
df_senic_data   <-   data.frame(X   =   infection_rate,
Y=length_of_stay)
lm_senice_data <- lm(Y ~ X , data = df_senic_data)
S_xy <- cov(df_senic_data$X, df_senic_data$Y)
S_xx <- var(df_senic_data$X)
b1 <- S_xy/S_xx
b0 <- mean(df_senic_data$Y) - b1*mean(df_senic_data$X)
df_senic_data$Y_hat <- b0 + b1 * df_senic_data$X
ei <- df_senic_data$Y - df_senic_data$Y_hat
df_senic_data$ei_sq <- ei^2
n <- length(df_senic_data$Y)
SSE <- sum(df_senic_data$ei_sq)
MSE = SSE/(n-2)
X_h <- 11.93
X_bar <- mean(df_senic_data$X)
numer <- (X_h -X_bar)^2
denom <- sum((df_senic_data$X - X_bar)^2)
s_y_h <- sqrt(MSE*((1/n) + (numer/denom)))
s_pred <- sqrt(MSE*(1+(1/n) + (numer/denom)))

qt(0.985, n-2)
2.198346 * (s_pred - s_y_h)
#####################################################################
```

QUESTION 5:

The box cox function is defined and corresponding linearBC() and bisectionBC() have been defined as follows in R. **The Signatures defined**

The BCTransform_GeomMean function which transforms the input Y – dependent variable using lambda

The LinearBC(dataframe) the dataframe has 2 columns namely X and Y

The bisectionBC(dataframe, tolerance, maxstep)

The dataframe has 2 columns namely X and Y, tolerance is the precision up to which the search is made, maxsteps is the max number of iterations allowed.

The stopping condition is `(delta > toler && steps < maxstep)`

`toler = 0.00001 and maxstep = 1000`

`######################################################################`

```
###Function to calculate geometric mean
geo_mean <- function(data) {
  log_data <- log(data)
  gm <- exp(mean(log_data[is.finite(log_data)]))
  return(gm)
}
###Function to calculate box cox transformation
BC_Transform_GeomMean = function(y, lambudah){
  W = NA
  n <- length(y)
  k2 <- geo_mean(y)

  if(lambudah != 0){
    num <- (y^lambudah)-1
    den <- lambudah * (k2)^(lambudah - 1)
    W <- num/den
  }
  if(lambudah == 0){
    W <- k2 * log(y)
  }
  W
```

```
}

####Function to calculate SSE for lambdah from X and Y
inpout data frame
Calculate_SSE_Lambdah = function(X, Y){
  df<- data.frame(X=X, Y=Y)
  lambdah_seq <- seq(from = -3.0, to = 3.0, by=0.1)
  n <- length(lambdah_seq)
  data_lambudah_sse   =   data.frame('lambda'=double(n),
'SSE'=double(n))
  for (i in seq(1:length(lambdah_seq))){
    l <- lambdah_seq[i]
    W <- BC_Transform_GeomMean(df$Y ,l)

    df$W <- W
    lm_trans <- lm(W~X, data=df)
    sse <- sum(residuals(lm_trans)^2)
    data_lambudah_sse$lambda[i] <- l
    data_lambudah_sse$SSE[i] <- sse
  }
  data_lambudah_sse
}

###Linear BC function to look for minimum Lambda from
data in adataframe
linearBC = function(df){
  names(df) <- c('Y','X')
  df_lambda_sse <- Calculate_SSE_Lambdah(df$X, df$Y)
  min_sse = df_lambda_sse$SSE[1]
  lambda_min_sse = 0
  for(i in seq(1:length(df_lambda_sse$lambda))){
    cur_sse <- df_lambda_sse$SSE[i]
    if (min_sse > cur_sse){
      min_sse <- cur_sse
      lambda_min_sse <- df_lambda_sse$lambda[i]
    }
  }
  lambda_min_sse
}
```

```
####Function to calculate the sse for particular lambda
func = function(lambduh, dfr){
  W <- BC_Transform_GeomMean(dfr$Y ,lambduh)
  dfr$W <- W
  lm_trans <- lm(W~X, data=dfr)
  sse <- sum(residuals(lm_trans)^2)

  sse
}

bisectionBC = function(df, tol = 0.000001, maxstep =
10000){
  steps <- 0
  toler <- tol
  names(df) <- c('Y','X')
  lambda_seq <- seq(from = -3.0, to = 3.0, by=0.001)
  a <- lambda_seq[1]
  b <- lambda_seq[length(lambda_seq)]
  c <- (a + b)/ 2
  fa <- func(a, df)
  fb <- func(b, df)
  delta <- abs( fa - fb)
 while (delta > toler && steps < maxstep){
      if (fa < fb){
        b <- c
        fb <- func(b, df)
        c <- (a + b)/ 2
      }
      else{
        a <- c
        fa <- func(a, df)
        c <- (a + b)/ 2
      }
    steps <- steps + 1
    delta <- abs(fa - fb)
 }
  return(b)
}
```