Assignment 3

Kushal Rakeshbhai Kapadia

1213714799

Part 1

```r
cop = read.table("D:\\ASU Stuff\\SEM-1\\STP 530\\Copier Maintenance.txt")
x = cop$V2
y = cop$V1
```

Estimating the regression function

```r
b1 = sum(x*y)/sum(x*x)
b1

## [1] 14.94723
```

So, the regresion function is $\hat{Y}$=14.94723X.

Estimating the β_1 with 90% confidence interval

```r
res = y - 14.94723*x #residuals
res

##  [1]  -9.89446   0.21108   1.15831  11.10554  -2.94723 -12.47230  -6.73615
##  [8]  14.26385 -10.94723   2.10554   9.47493   6.52770   3.31662  -8.84169
## [15]  12.21108 -19.57784   0.36939  11.42216 -22.47230  -2.78892  -8.73615
## [22]  -3.63061   4.36939  -0.73615  -0.52507   7.36939 -11.89446  -1.73615
## [29]   6.36939   6.31662   3.42216  15.26385  -9.89446  -1.89446 -11.94723
## [36]  -2.78892  11.26385  -2.52507   7.36939  12.05277  -3.52507   4.10554
## [43]  -2.89446   1.21108   2.26385

MSE = sum((res)^2)/44
MSE

## [1] 77.72224

SEB = sqrt(MSE / sum(x*x))
SEB

## [1] 0.2264243
```

t(0.95,44)=1.684 and thus the confidence limits are 14.94723 - 1.684(0.2264) and 14.94723 + 1.684(0.2264). Thus, the 90% confidence limit is (14.5659,15.3284).

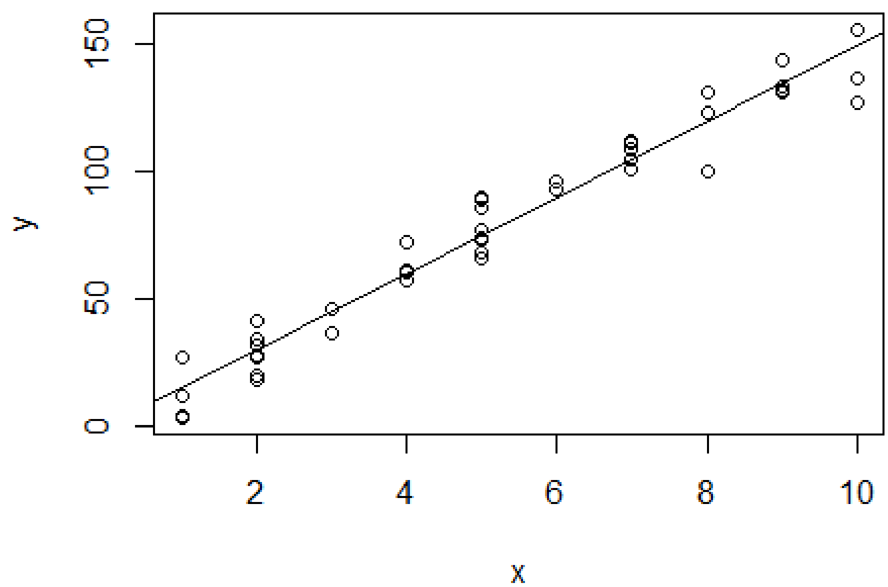
90% Prediction interval for 6 copiers

```r
SEBp = sqrt(MSE*(1 + (36/sum(x*x))))
```

The 90% prediction interval for 6 copiers is 89.68338-1.684(8.9200) and 89.68338+1.684(8.9200). Thus, the prediction interval is (74.6621,104.7047).

Plotting the regression line

```
reg = lm(y~x)
plot(y~x)
abline(lm(y~x))
```

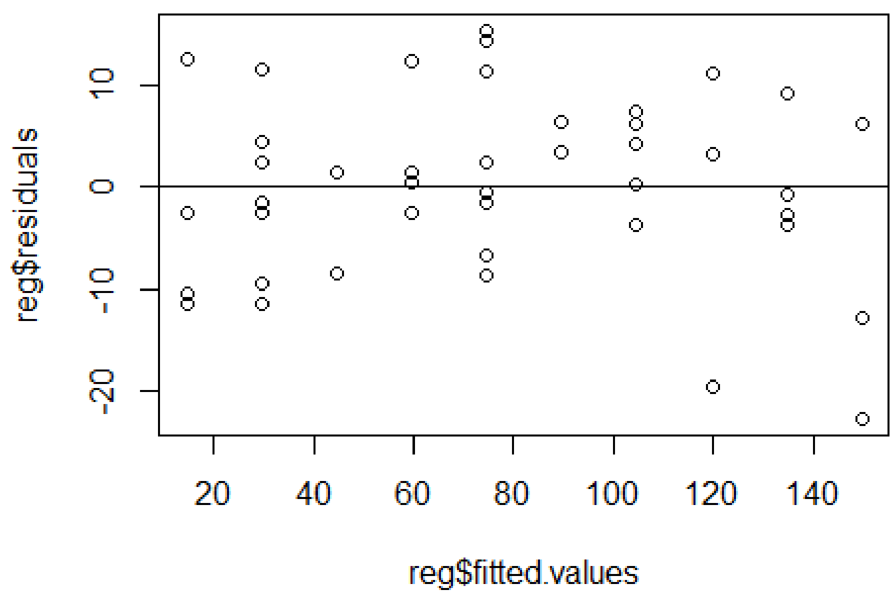


Yes, we can see from the plot that the regression line seems to give a good fit.

Residuals

```
res = y - 14.94723*x #residuals
res

##  [1]  -9.89446   0.21108   1.15831  11.10554  -2.94723 -12.47230  -6.73615
##  [8]  14.26385 -10.94723   2.10554   9.47493   6.52770   3.31662  -8.84169
## [15]  12.21108 -19.57784   0.36939  11.42216 -22.47230  -2.78892  -8.73615
## [22]  -3.63061   4.36939  -0.73615  -0.52507   7.36939 -11.89446  -1.73615
## [29]   6.36939   6.31662   3.42216  15.26385  -9.89446  -1.89446 -11.94723
## [36]  -2.78892  11.26385  -2.52507   7.36939  12.05277  -3.52507   4.10554
## [43]  -2.89446   1.21108   2.26385

sum(res)

## [1] -5.8629
```

```r
plot(reg$residuals ~ reg$fitted.values)
abline(h=0)
```



As we can see, the sum of residuals is not zero and from the residuals plot, there is no evidence of lack of fit or of strongly unequal variances.

Test for lack of fit

```r
cop = read.table("D:\\ASU Stuff\\SEM-1\\STP 530\\Copier Maintenance.txt")
x = cop$V2
y = cop$V1
a = data.frame(x,y)
#for loop for mean of y under different level of x as 1:10
Xi = rep(NA,10)
for(i in 1:10){
  Xi[i] = mean(a[x==i,]$y)
}
#for loop for mean of y under different level of x
  SSPEi=rep(NA,10)
  for(i in 1:10){
  SSPEi[i] = sum((a[x==i,]$y-mean(a[x==i,]$y))^2)
}
#Null hypothesis: E(Y)=beta1*X
  c=10
  n=length(x)
  SSPE=sum(SSPEi)
  SSLF = 622.12
```

```r
  Fs = (SSLF/(c-1))/(SSPE/(n-c))
  pvalue = pf(Fs,c-1,n-c,lower.tail = F)
  Fs

## [1] 0.8647788

  SSPE

## [1] 2797.658

  SSLF

## [1] 622.12

  pvalue

## [1] 0.5644336
```

Thus, we can see that $F^* \leq 2.9630$. Thus, conclude H_0. The p-value is 0.564.

Part 2

Joint confidence intervals

A family confidence coefficient corresponds to the probability, in advance of sampling, that the entire family of statements will be correct. So, a confidence coefficient of 90% would simply mean that if repeated samples are selected and interval estimates of B_0 and B_1 are calculated for each sample, 90% of the samples would lead to a family of estimates where *both* confidence intervals are correct. For the rest 10% of the samples, either one or both of the interval estimates would be incorrect. Also, it is not necessary that 5% of the time the confidence interval for β_0 will be incorrect. The percentage may vary.

b_0 and b_1 direction

Here, $\overline{x} = 5.111 > 0$ and thus $b_0 and b_1$ are negatively related meaning if one increases, the other one decreases. Thus, for our case as $\overline{x} > 0$, $b_0 and b_1$ tend to err in the opposite direction.

Bonferroni joint confidence intervals

```r
b1 = sum((x-mean(x))*(y-mean(y)))/sum((x-mean(x))^2)
b0 = mean(y) - b1*mean(x)

SEb1 = sqrt(MSE/sum((x-mean(x))^2))
SEb0 = sqrt(MSE*((1/45)+(mean(x)^2/sum((x-mean(x))^2))))
```

B = t(1-α/4; n-2) = t(0.9875,43) = 2.250

For β_0, 95% interval estimates are -0.5801 - 2.250(2.7732) and -0.5801 + 2.250(2.7732). Thus, interval is (-6.9198,5.9396).

For β_1, 95% interval estimates are 15.0352 - 2.250(0.4778) and 15.0352 + 2.250(0.4778). Thus, interval is (13.9601,16.1102).

And as suggested by the consultant, as $\beta_0 = 0$ is in the confidence interval range of β_0 and $\beta_1 = 14$ is also in the confidence interval range of β_1, we can say that the joint confidence intervals in part(b) support this view.

Part 3

```
x=c(7,12,10,10,14,25,30,25,18,10,4,6)
y=c(128,213,191,178,250,446,540,457,324,177,75,107)
```

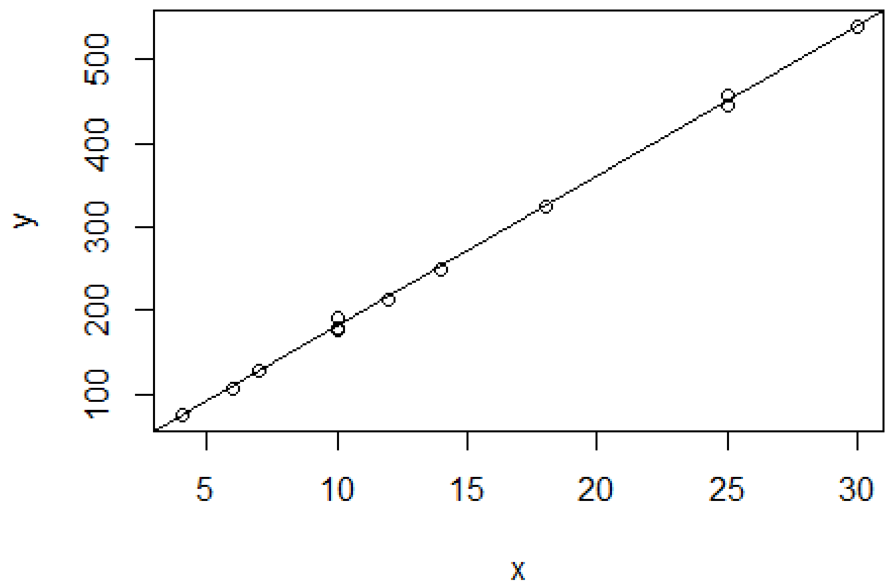
Fitting regression model and estimating regression function

```
b1 = sum(x*y)/sum(x*x)
b1

## [1] 18.0283
```

Thus, the estimated regression function is $\hat{Y} = 18.023X$.

Plotting the regression function

```
x=c(7,12,10,10,14,25,30,25,18,10,4,6)
y=c(128,213,191,178,250,446,540,457,324,177,75,107)
lm(y~x)

##
## Call:
## lm(formula = y ~ x)
##
## Coefficients:
## (Intercept)            x
##       1.088       17.970

plot(y~x)
abline(lm(y~x))
```



From the plot and the line passing through the plot, it is very clear that a linear regression function through the origin appears to give a good fit here because the points are not very scattered from the line.

Test for whether the standard should be revised or not

```
res1 = y - 18.023*x
res1

##  [1]  1.839 -3.276 10.770 -2.230 -2.322 -4.575 -0.690  6.425 -0.414 -3.230
## [11]  2.908 -1.138

MSE = sum((res1)^2)/11
MSE

## [1] 20.31952

SEb = sqrt(MSE/sum(x*x))
SEb

## [1] 0.07949984
```

$H_a: \beta_1 = 17.50, H_a: \beta_1 \neq 17.50$. So, $t^* = b1 - 17.50/SEb = 6.65$. and t(0.99;11) = 2.718 and thus $t^* \geq 2.718$ and so we conclude alternate hypothesis here.

Prediction interval

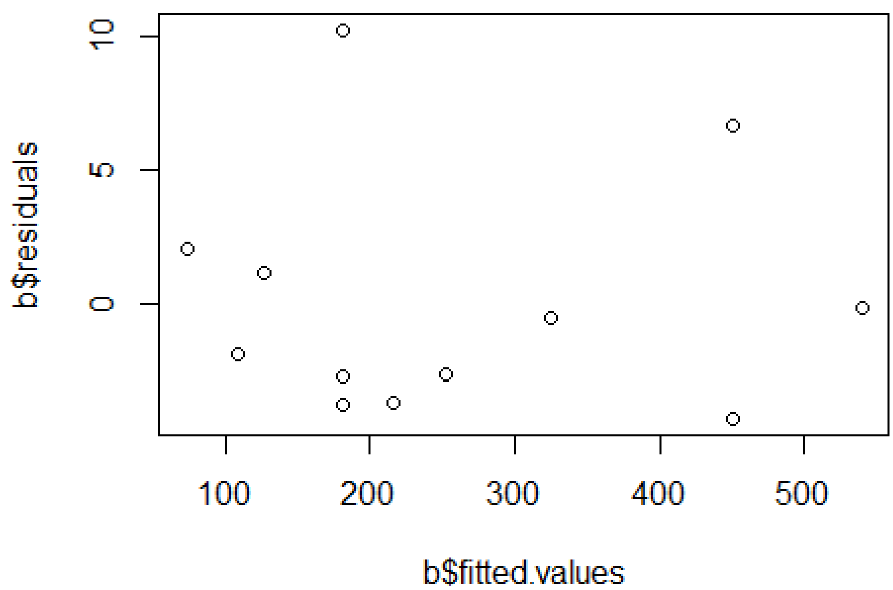
```
s.pred = sqrt(MSE*(1+(100/(sum(x*x)))))
s.pred
```

```
## [1] 4.577286
```

$\hat{y}_h$ = 180.23. Thus, the interval can be calculated as 180.23-2.718(4.578) and 180.23+2.718(4.578). Therefore, the prediction interval for the correction cost on a forthcoming job involving 10 galleys is (167.80,192.72).

Residuals

```
b= lm(y~x)
res1 = y - 18.023*x
res1

##  [1]  1.839 -3.276 10.770 -2.230 -2.322 -4.575 -0.690  6.425 -0.414 -3.230
## [11]  2.908 -1.138

res.sum = sum(res1)
res.sum

## [1] 4.067

plot(b$residuals~b$fitted.values)
```



From the sum of residuals, we can clearly see that it doesn't sum upto zero and from the residuals plot, there is no evidence of lack of fit or of strongly unequal variances.

Formal test for lack of fit

```
a = data.frame(x,y)
#for loop for mean of y under different level of x as 1:10
```

```r
Xi = rep(NA,10)
for(i in 1:10){
  Xi[i] = mean(a[x==i,]$y)
}
#for loop for mean of y under different level of x
  SSPEi=rep(NA,10)
  for(i in 1:10){
  SSPEi[i] = sum((a[x==i,]$y-mean(a[x==i,]$y))^2)
}
#Null hypothesis: E(Y)=beta1*X
  c=10
  n=length(x)
  SSPE=sum(SSPEi)
  SSLF = 40.924
  Fs = (SSLF/(c-1))/(SSPE/(n-c))
  pvalue = pf(Fs,c-1,n-c,lower.tail = F)
  Fs

## [1] 0.07454281

  SSPE

## [1] 122

  SSLF

## [1] 40.924

  pvalue

## [1] 0.9980049
```

$H_0: E(Y) = \beta_1 X, H_a: E(Y) \neq \beta_1 X$, F(0.99;8,3)=27.5 and here $F^* \geq 27.5$ and thus we conclude alternate hypothesis and also, the pvalue is 0.998.

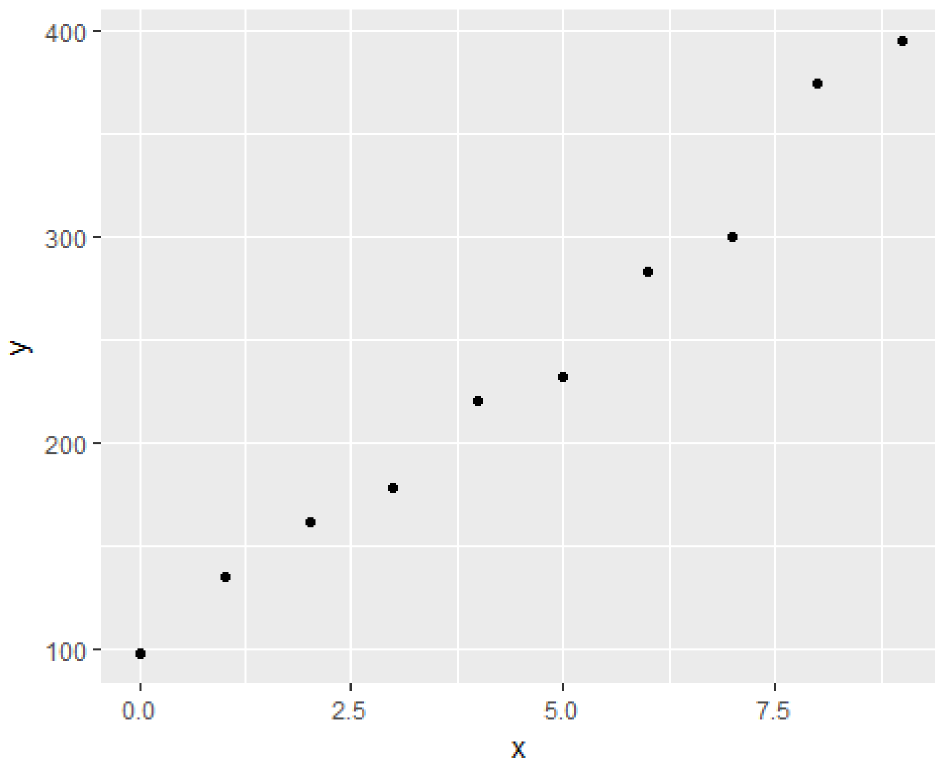
Part 4

```r
x = c(0,1,2,3,4,5,6,7,8,9)
y = c(98,135,162,178,221,232,283,300,374,395)
```

(a)

Scatterplot of the data

```r
library(ggplot2)
qplot(x,y)
```

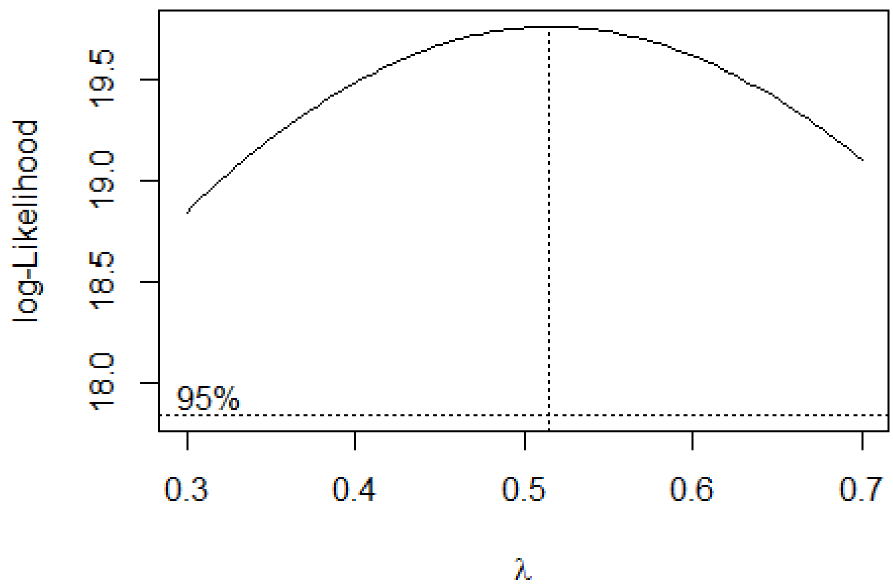


Yes, from the plot it is clear that a linear relation does appear adequate here.

(b)

Box-Cox procedure

```
library(MASS)
b = boxcox(y~x, lambda = c(0.3,0.4,0.5,0.6,0.7))
```

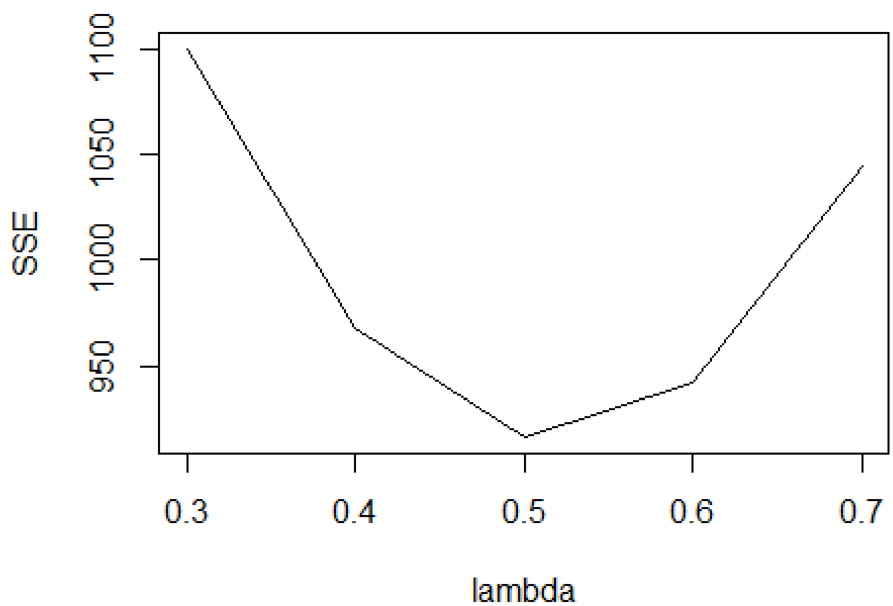


Evaluating SSE

```
resSS <- function(x, y, lambda){
  n <- length(y)
  k2 <- (prod(y))^(1/n)
  k1 <- 1/(lambda * (k2^(lambda - 1)))
  w <- rep(NA, n)
  for(i in 1:n){
    w[i] <- ifelse(lambda == 0, (k2 * log(y[i])), (k1 * (y[i]^lambda - 1)))}
  reg_fit <- lm(w ~ x)
  SSE <- deviance(reg_fit)
  return(SSE)
}

  lambda = c(0.3,0.4,0.5,0.6,0.7)
  SSE =  rep(NA, length(lambda))

  for(i in 1:length(lambda)){
    SSE[i] = resSS(x,y,lambda[i])

  }
  SSE

## [1] 1099.7093  967.9088  916.4048  942.4498 1044.2384
```

Thus, the values for SSE for lambda=0.3,0.4,0.5,0.6,0.7 are 1099.7093,967.9088,916.4048,942.4498,1044.2384.

```r
plot(lambda, SSE, type = "l")
```



From the plot, SSE is minimum at λ = 0.5.

(c)

Transformation of Y

```r
ydash = sqrt(y)
lr = lm(ydash~x)
lr

##
## Call:
## lm(formula = ydash ~ x)
##
## Coefficients:
## (Intercept)            x
##      10.261        1.076
```

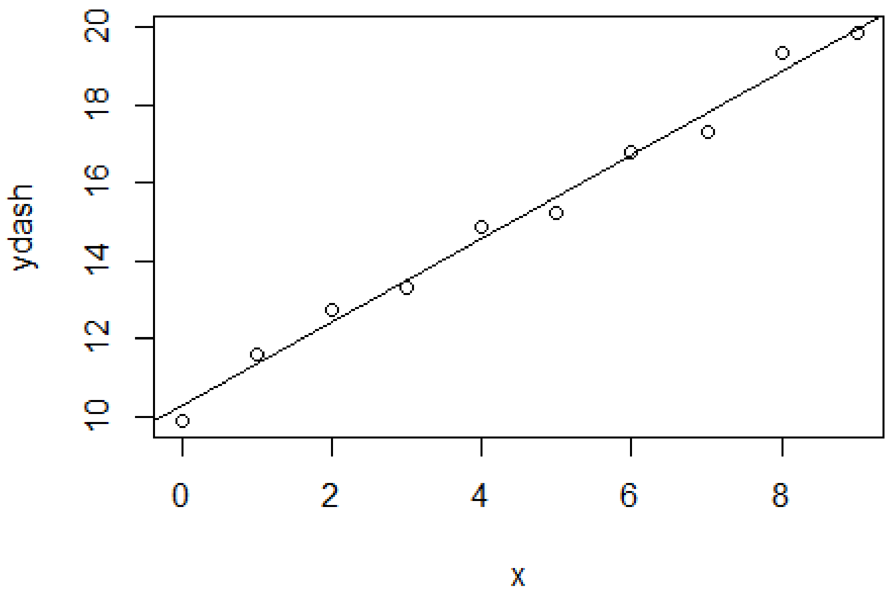
Thus the estimated regression fucntion is $Y_{i}^`$ = 10.261 + 1.076X_i.

(d)

Plotting the regression line and transformed data

```r
ydash = sqrt(y)
lr = lm(ydash~x)
```

```
plot(ydash~x)
abline(lm(ydash~x))
```

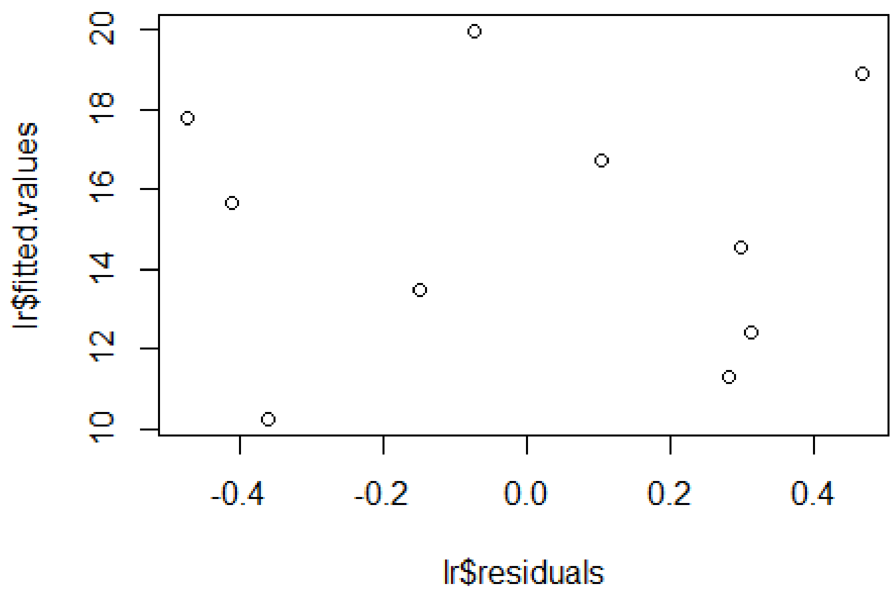


From the plot, it is clear that as the points are not very far away from the line, and so the regression line appears to give a good fit.

(e)

Residuals vs Fitted values plot

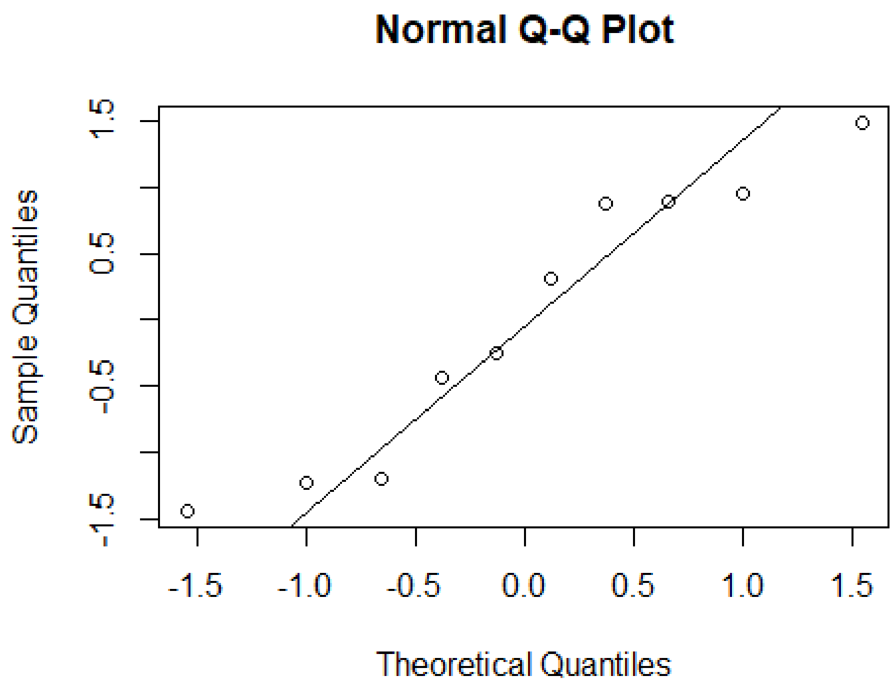
```
ydash = sqrt(y)
lr = lm(ydash~x)
plot(lr$fitted.values~lr$residuals)
```



From the residuals plot, there is no evidence of lack of fit or of strongly unequal variances.

Normal probability plot

```
ydash = sqrt(y)
lr = lm(ydash~x)
lrstdres = rstandard(lr)

qqnorm(lrstdres)
qqline(lrstdres)
```



From the normality plot, no substantial departures from normality are indicated.

(f)

Estimation of regression function in the original units

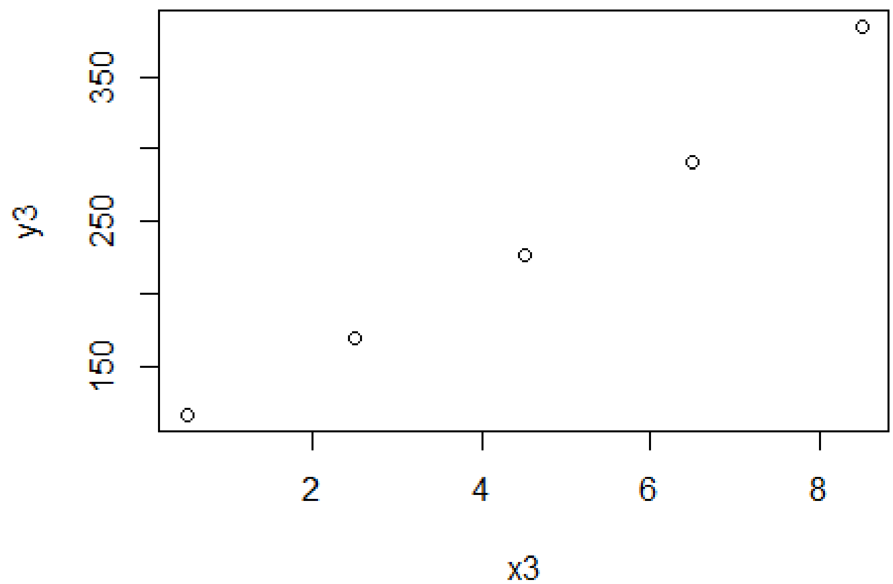
The transformed function is $E(\sqrt{(y)})$=10.26 + 1.076X. Now again transforming it will give E(Y) = 105.27+22.08X+1.158X^2.

Part 5

(a)

If we divide the range of X=-0.5 to 1.5 and Y=1.5 to 3.5, the median value of X will be (0.5,2.5,4.5,6.5,8.5) and median value of Y will be (116.5,170,226.5,291.5,384.5).

```
x3 = c(0.5,2.5,4.5,6.5,8.5)
y3 = c(116.5,170,226.5,291.5,384.5)
plot(x3,y3)
```

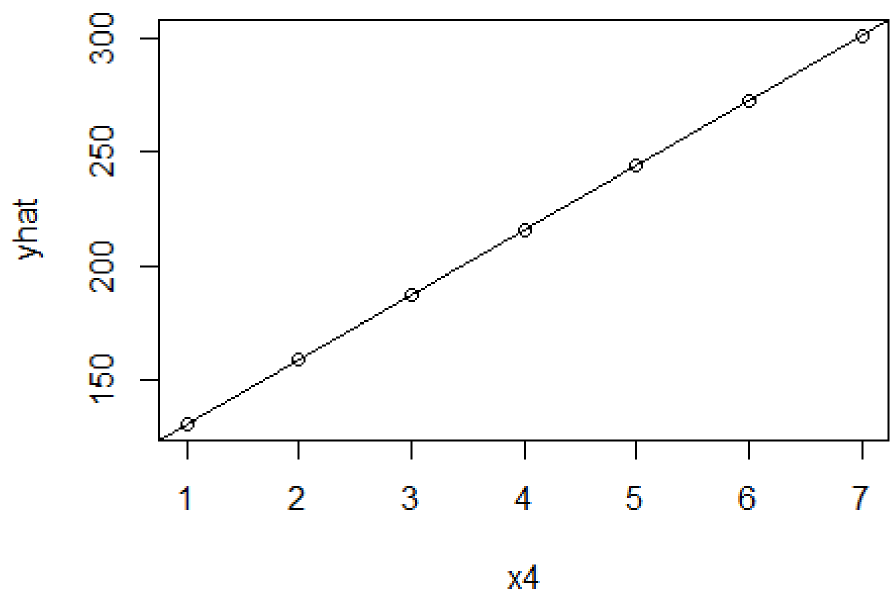


We can see from the plot that the data fits linearly well and thus the regression relation is linear.

(b)

If we divide the range of X=-0.5 to 2.5 and Y=0.5 to 3.5, the median value of X will be (1,2,3,4,5,6,7) and median value of Y will be (135,162,178,221,232,283,300).

```
x4 = c(1,2,3,4,5,6,7)
y4 = c(135,162,178,221,232,283,300)
m1 = lm(y4~x4)
yhat = m1$fitted.values
yhat

##        1        2        3        4        5        6        7
## 131.1071 159.3571 187.6071 215.8571 244.1071 272.3571 300.6071

plot(x4,yhat)
abline(lm(yhat~x4))
```



(c)

F(0.95;2,8) = 4.46 and W = sqrt(2F(1-alpha;2;n-2)) = 2.987

$X_h = 1: 124.061 + -2.987(7.4560), 101.731 \leq E(Y_h) \leq 146.391$

$X_h = 2: 156.456 + -2.987(6.2872), 137.778 \leq E(Y_h) \leq 175.338$

$X_h = 3: 189.055 + -2.987(5.3501), 173.074 \leq E(Y_h) \leq 205.036$

$X_h = 4: 221.552 + -2.987(4.8261), 207.132 \leq E(Y_h) \leq 235.931$

$X_h = 5: 254.049 + -2.987(4.8261), 239.671 \leq E(Y_h) \leq 268.428$

$X_h = 6: 286.546 + -2.987(5.3501), 270.562 \leq E(Y_h) \leq 302.527$

$X_h = 7: 319.043 + -2.987(6.2872), 300.006 \leq E(Y_h) \leq 337.823$

The simplified lowess smooth does entirely fall within the confidence band for the regression line and thereby supports the appropriateness of a linear regression function.

Part 6

$b_0 and b_1 relation$

Here, in the normal error regression model, the covariance between $b_0 and b_1$ is:

Cov(b_0, b_1) = $-\overline{X}\sigma^2 b_1$

So, when the predictor is coded that $\overline{X} = 0$, the covariance according to the equation above will be zero and thus $b_0 and b_1$ are independent which implies that the joint confidence intervals for $\beta_0 and \beta_1$ are also independent.

Deriving an extension

For two events X and Y, Bonferroni's inequality states that P($XY $) ≥ P(x) + P(Y) - 1.

Also, $X = \overline{A}_1 and Y = \overline{A}_2$ and thus, P($\overline{A}_1 \cap \overline{A}_2$) ≥ 1 - 2$\alpha$.

Now, let $\overline{X} = B_1, \overline{Y} = B_2 \cup B_3$ i.e. Y = $\overline{B}_2 \cap \overline{B}_1$ for some B_1, B_2, B_3 such that the confidence statements of coefficient for each of them is $1 - \alpha$.

Following this, P($\overline{B}_1 \cap (\overline{B}_2 \cap \overline{B}_3)$) ≥ $P(\overline{B}_1) + P(\overline{B}_2 \cap \overline{B}_3) - 1$

= $P(\overline{B}_1) - P(B_2 \cup B_3)$.

$P(B_2 \cup B_3) \leq P(B_2) + P(B_3)$ = 2α.

Therefore, $P(\overline{B}_1 \cap \overline{B}_2 \cap \overline{B}_3) \geq 1 - \alpha - 2\alpha$ = $1 - 3\alpha$.

So, the extenson is $P(\overline{B}_1 \cap \overline{B}_2 \cap \overline{B}_3) \geq 1 - 3\alpha$.

Proof of fitted least squares regression line through origin

Since

$\sum X_i Y_i = \sum X_i (b_1 X_i) \sum X_i Y_i - \sum X_i (b_1 X_i)$ = 0

and

$X_i e_i = X_i(Y_i-b_1X_i) = X_i Y_i - X_i(b_1X_i)=0 $

Therefore, $\sum X_i e_i = 0$.

Proof of $\hat{Y}$ as unbiased estimator of Y_i

First, we have to prove that,

$E(b_1) = E(\sum X_i Y_i / \sum X_i^2) = \sum X_i E(Y_i) / \sum X_i^2 = \sum X_i (\beta_1 X_i) / \sum X_i^2 = \beta_1$

then

$E(\hat{Y}_i) = E(b_1 X_1) = X_i E(b_1) = X_i \beta_1 = Y_i$

Thus, it is proved that $\hat{Y}$ is an unbiased estimator.

Part 7

```
cdi = read.table("D:\\ASU Stuff\\SEM-1\\STP 530\\CDI_data.txt")
x2 = cdi$V5
y2 = cdi$V8
```

(a)

```
cdi.lm = lm(y~x)

b1 = sum((x-mean(x))*(y-mean(y)))/sum((x-mean(x))^2)
b0 = mean(y) - b1*mean(x)

SEb1 = sqrt(MSE/sum((x-mean(x))^2))
SEb0 = sqrt(MSE*((1/440)+(mean(x)^2/sum((x-mean(x))^2))))
SEb1

## [1] 0.4962834

SEb0

## [1] 2.243591

qt(0.9875,438)

## [1] 2.249135
```

B = t(1-α/4; n-2) = t(0.9875,438) = 2.2249

For β_0, 95% joint confidence interval is (-188.7832,-32.48629).

For β_1, 95% joint confidence interval is (0.00268,0.002904).

(b)

And as suggested by the consultant, as $\beta_0 = -100$ is in the confidence interval range of β_0 and $\beta_1 = 0.0028$ is also in the confidence interval range of β_1, we can say that the joint confidence intervals in part(a) support this view.

(c)

```
#Bonferroni
xh = c(500000,1000000,5000000)
yh = cdi.lm$coefficients[1]+cdi.lm$coefficients[2]*xh
G = length(xh)
B = qt(1-0.1/(2*G), length(x2)-2)
s.yh = sqrt(MSE*((1/length(x2))+((xh-mean(x2))^2/sum((x2-mean(x2))^2))))
ub.yh = yh + B*s.yh
lb.yh = yh - B*s.yh
CI.bon = data.frame(lb.yh,ub.yh)
CI.bon

##       lb.yh     ub.yh
## 1  16248576  16248577
## 2  32497061  32497062
## 3 162484937 162484944

#Working Hotelling
w = sqrt(2*qf(1-0.1,2,length(x2)-2))
uw.yh = yh+w*s.yh
```

```r
lw.yh = yh-w*s.yh
ci.wh = data.frame(lw.yh,uw.yh)
ci.wh
```

```
##       lw.yh     uw.yh
## 1  16248576  16248577
## 2  32497061  32497062
## 3 162484936 162484944
```

(d)

The family of estimates for X=500000,1000000,5000000 for both the procedures is shown above. Here, we can say from the interval estimate range that bonferroni is more efficient than Working-Hotelling because the range of confidence interval of Bonferroni is comparatively lesser than Working-Hotelling.