ISE 789 HW-3

Deepak Kumar Tiwari
Master's in Financial Mathematics
NC State University

ISE789/OR791
Homework 3
Due: Oct. 23, 2019 before class

1. Consider the two models $\boldsymbol{y}_1 = \boldsymbol{X}_1\boldsymbol{\beta}_1 + \boldsymbol{\epsilon}_1$ and $\boldsymbol{y}_2 = \boldsymbol{X}_2\boldsymbol{\beta}_2 + \boldsymbol{\epsilon}_2$, where the $\boldsymbol{X}_i$'s are $n_i \times p$ matrices. Suppose that $\boldsymbol{\epsilon}_i \sim N(\boldsymbol{0}, \sigma^2\boldsymbol{I})$, where $i = 1, 2$ and $\boldsymbol{\epsilon}_1$ and $\boldsymbol{\epsilon}_2$ are independent. Show that the test statistic for $H_0 : \boldsymbol{\beta}_1 = \boldsymbol{\beta}_2$ can be written as

$$(\hat{\boldsymbol{\beta}}_1 - \hat{\boldsymbol{\beta}}_2)'[(\boldsymbol{X}_1'\boldsymbol{X}_1)^{-1} + (\boldsymbol{X}_2'\boldsymbol{X}_2)^{-1}]^{-1}(\hat{\boldsymbol{\beta}}_1 - \hat{\boldsymbol{\beta}}_2)/(p\hat{\sigma}^2),$$

where

$$(n_1 + n_2 - 2p)\hat{\sigma}^2 = \boldsymbol{y}_1'\boldsymbol{M}_1\boldsymbol{y}_1 + \boldsymbol{y}_2'\boldsymbol{M}_2\boldsymbol{y}_2,$$

and for $i = 1, 2$,

$$\hat{\boldsymbol{\beta}}_i = (\boldsymbol{X}_i'\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i'\boldsymbol{y}_i, \text{ and } \boldsymbol{M}_i = \boldsymbol{I}_{n_i} - \boldsymbol{X}_i(\boldsymbol{X}_i'\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i'.$$

2. Using the *teengamb* dataset, fit a model with gamble as the response and the other variables as predictors. Perform regression diagnostics on this model to answer the following questions. Display any plots that are relevant. Do not provide any plots about which you have nothing to say.

   (a) Check the normality assumption.

   (b) Check the constant variance assumption.

   (c) Check for large leverage points.

   (d) Check for outliers.

   (e) Check for influential points.

3. Researchers at National Institute of Standards and Technology (NIST) collected *pipeline* data (download from *faraway* library) on ultrasonic measurements of the depths of defects in the Alaska pipeline in the field. The depth of the defects were then measured

in the laboratory. These measurements were performed in six different batches. It turns out that this batch effect is not significant and so can be ignored in the analysis that follows. The laboratory measurements are more accurate than the in-field measurements, but more time consuming and expensive. We want to develop a regression equation for correcting the in-field measurements.

(a) Fit a regression model $Lab \sim Field$. Check for nonconstant variance.

(b) We wish to use weights to account for the nonconstant variance. Here we split the range of field into 12 groups of size nine (except for the last group which has only eight values). Within each group, compute the variance of Lab (varlab) and the mean of Field (meanfield). Suppose we guess that the variance in the response is linked to the predictor in the following way: $var(Lab) = a_0 Field^{a_1}$. Regress $log(varlab)$ on $log(meanfield)$ to estimate a_0 and a_1. (You might choose to remove the last point). Use this to determine appropriate weights in a WLS fit of Lab on Field. Show the regression summary.

(c) An alternative to weighting is transformation. Find transformations on Lab and/or Field so that in the transformed scale the relationship is approximately linear with constant variance. You may restrict your choice of transformation to square root, log, and inverse.

ISE 789 HW-3

Deepak Kumar Tiwari

1

$y_1 = X_1\beta_1 + \varepsilon_1$

$y_2 = X_2\beta_2 + \varepsilon_2$.

X_i's $= n_i \times p$ matrices.

$\varepsilon_i \sim N(0, \sigma^2 I)$ $i = 1, 2$.

$\varepsilon_1, \varepsilon_2$ independent.

$H_0 : \beta_1 = \beta_2$.

We know $\hat{\beta} = (X^TX)^{-1}X^Ty$.

so, for this problem

$$\hat{\beta} = \begin{bmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \end{bmatrix} = \begin{bmatrix} (X_1^TX_1)^{-1}X_1^Ty_1 \\ (X_2^TX_2)^{-1}X_2^Ty_2 \end{bmatrix}$$

$\Rightarrow$ $H_0 : C\beta = \gamma$ (under null hypothesis)

we reject H_0 when

$$\frac{(C\beta - \gamma)'(C(X^TX)^{-1}C^T)^{-1}(C\beta - \gamma)}{(y - X\beta)^T(y - X\beta)} \quad \text{--- (1)}$$ is large.

$\Rightarrow$ For this problem $C\beta - \gamma = \hat{\beta}_1 - \hat{\beta}_2$

$$C(X^TX)^{-1}C^T = (X_1^TX_1)^{-1} + (X_2^TX_2)^{-1}$$

Also, $(y - X\beta)^T(y - X\beta) = (y - Hy)^T(y - Hy) = y^T(I-H)(I-H)y$

$= y^T(I - H)y = y_1^T M_1 y_1 + y_2^T M_2 y_2$

$M_i = I_{n_i} - X_i(X_i^TX_i)^{-1}X_i^T \rightarrow H_i$

Here $H = X(X^TX)^{-1}X^T = \begin{bmatrix} X_1(X_1^TX_1)^{-1}X_1^T & 0 \\ 0 & X_2(X_2^TX_2)^{-1}X_2^T \end{bmatrix}$

If we denote

$$\hat{\sigma}^2 = (y - X\beta)^T(y - X\beta)/(n_1 + n_2 - 2p)$$

$$= (y_1^T M_1 y_1 + y_2^T M_2 y_2)/(n_1 + n_2 - 2p)$$

then ~~statistic for H_0~~ ① becomes as.

$$= \frac{(\hat{\beta}_1 - \hat{\beta}_2)^T \left[(X_1^TX_1)^{-1} + (X_2^TX_2)^{-1}\right]^{-1}(\hat{\beta}_1 - \hat{\beta}_2)}{(n_1 + n_2 - 2p)\hat{\sigma}^2}$$

Then ~~numer~~ using normal theory we know.

$$(\hat{\beta}_1 - \hat{\beta}_2)^T \left[(X_1^TX_1)^{-1} + (X_2^TX_2)^{-1}\right]^{-1}(\hat{\beta}_1 - \hat{\beta}_2)/\sigma^2 \sim \chi^2_p$$

and $(n_1 + n_2 - 2p)\hat{\sigma}^2/\sigma^2 \sim \chi^2_{n_1+n_2-2p}$

Dividing by proper degrees of freedom and taking ratio we can get F-test statistic as.

$$F = \frac{(\hat{\beta}_1 - \hat{\beta}_2)^T \left[(X_1^TX_1)^{-1} + (X_2^TX_2)^{-1}\right]^{-1}(\hat{\beta}_1 - \hat{\beta}_2)/p\sigma^2}{\frac{1}{\sigma^2}(n_1 + n_2 - 2p)\hat{\sigma}^2/(n_1 + n_2 - 2p)}$$

this F distribution with df as p and $n_1 + n_2 - 2p$

which can be simplified to get,

$$\boxed{\frac{(\hat{\beta}_1 - \hat{\beta}_2)^T \left[(X_1^TX_1)^{-1} + (X_2^TX_2)^{-1}\right]^{-1}(\hat{\beta}_1 - \hat{\beta}_2)}{p\hat{\sigma}^2}}$$

This is an R Markdown (http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

```

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

Let's get the data and check the head.

```
library(faraway)
```

```
## Warning: package 'faraway' was built under R version 3.5.3
```

```
attach(teengamb)
```

```
head(teengamb)
```

```
##   sex status income verbal gamble
## 1   1     51   2.00      8    0.0
## 2   1     28   2.50      8    0.0
## 3   1     37   2.00      6    0.0
## 4   1     28   7.00      4    7.3
## 5   1     65   2.00      8   19.6
## 6   1     61   3.47      6    0.1
```

```
gamb.lm <- lm(gamble ~ sex+status+income+verbal)
summary(gamb.lm)
```
```

```
##
## Call:
## lm(formula = gamble ~ sex + status + income + verbal)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.082 -11.320  -1.451   9.452  94.252
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  22.55565   17.19680   1.312   0.1968
## sex         -22.11833    8.21111  -2.694   0.0101 *
## status        0.05223    0.28111   0.186   0.8535
## income        4.96198    1.02539   4.839 1.79e-05 ***
## verbal       -2.95949    2.17215  -1.362   0.1803
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.69 on 42 degrees of freedom
## Multiple R-squared:  0.5267, Adjusted R-squared:  0.4816
## F-statistic: 11.69 on 4 and 42 DF,  p-value: 1.815e-06
```

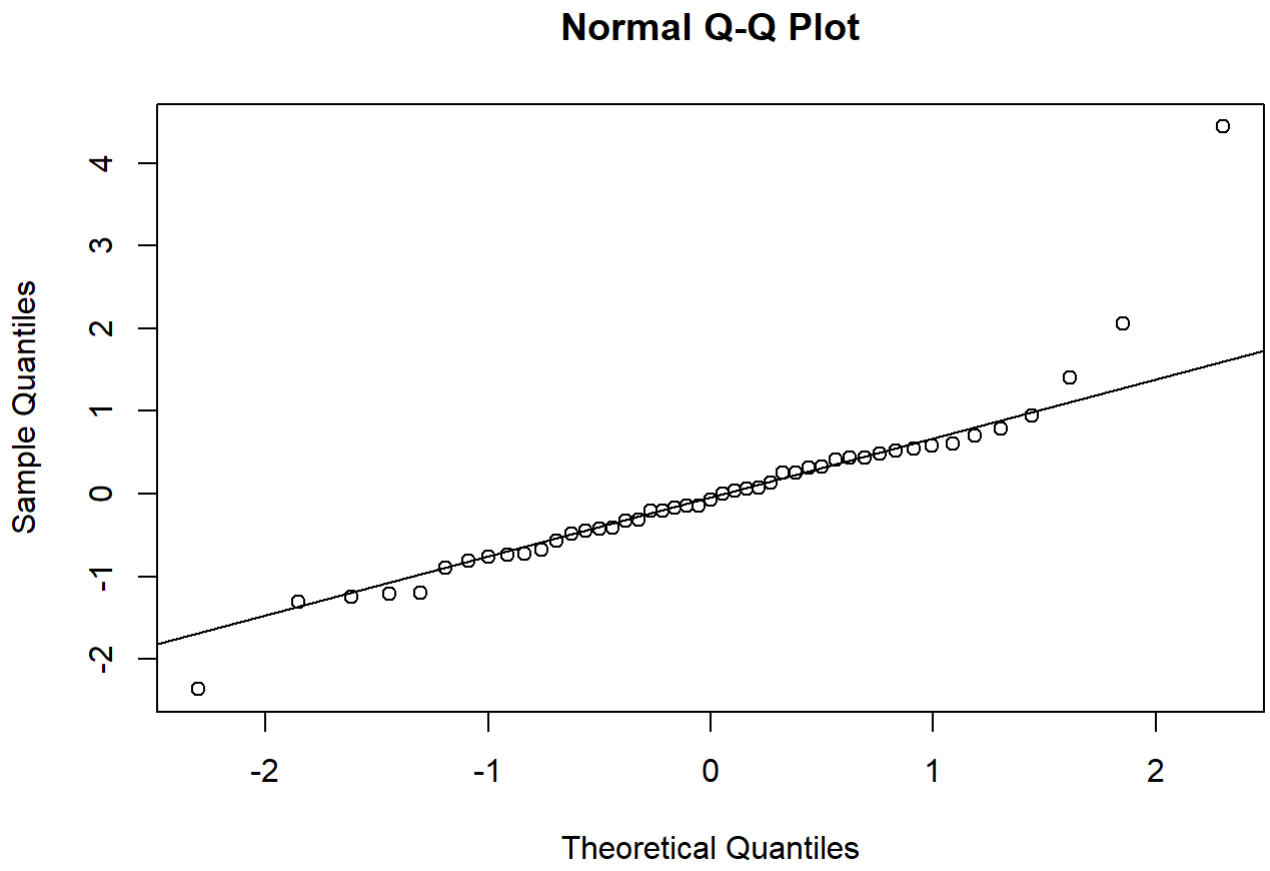
```
X=model.matrix(gamb.lm)
```

```
H=X%*%solve(t(X)%*%X)%*%t(X)
```

```
std.res=gamb.lm$res/(summary(gamb.lm)$sig*sqrt(1-diag(H)))
```

A. Let’s check Normality using QQ Plot now

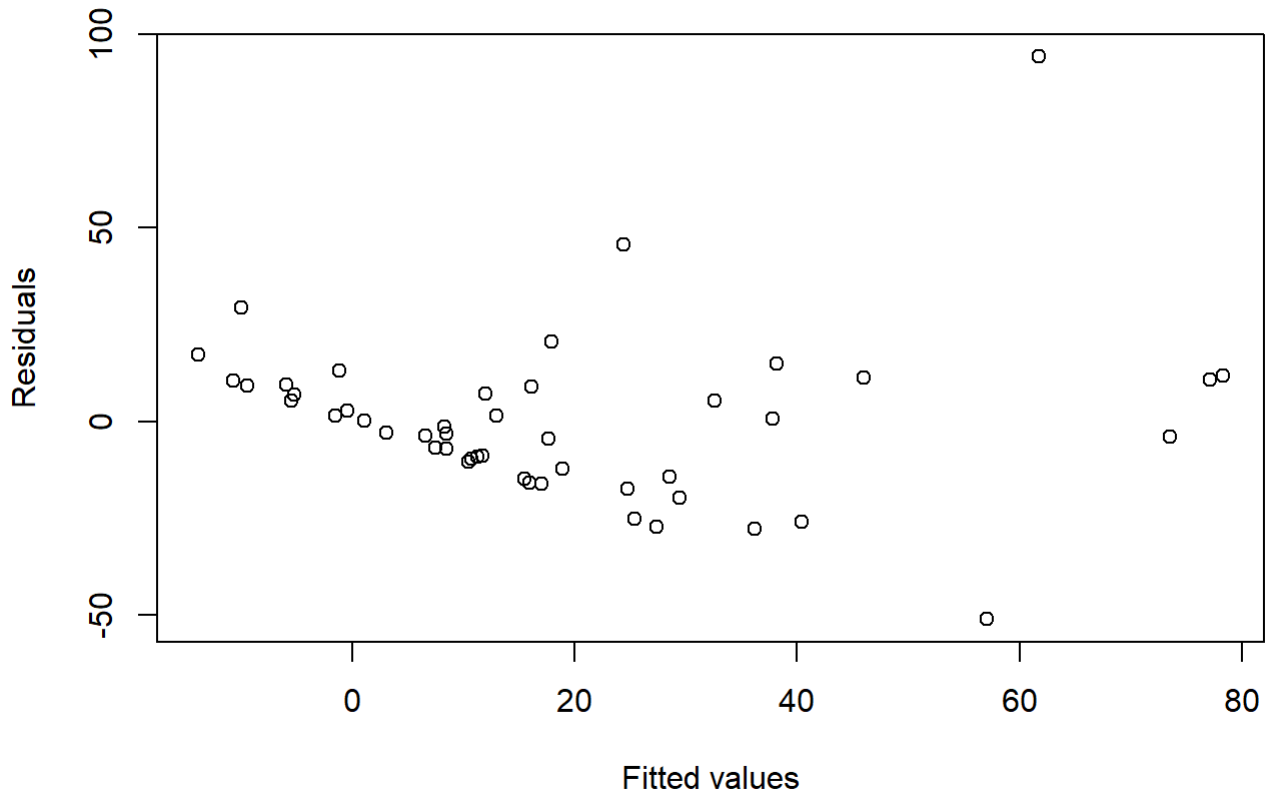
```
qqnorm(std.res)
qqline(std.res)
```



We can see that data fits the line perfectly well except few points here and there so, our normality assumptions hold true here.

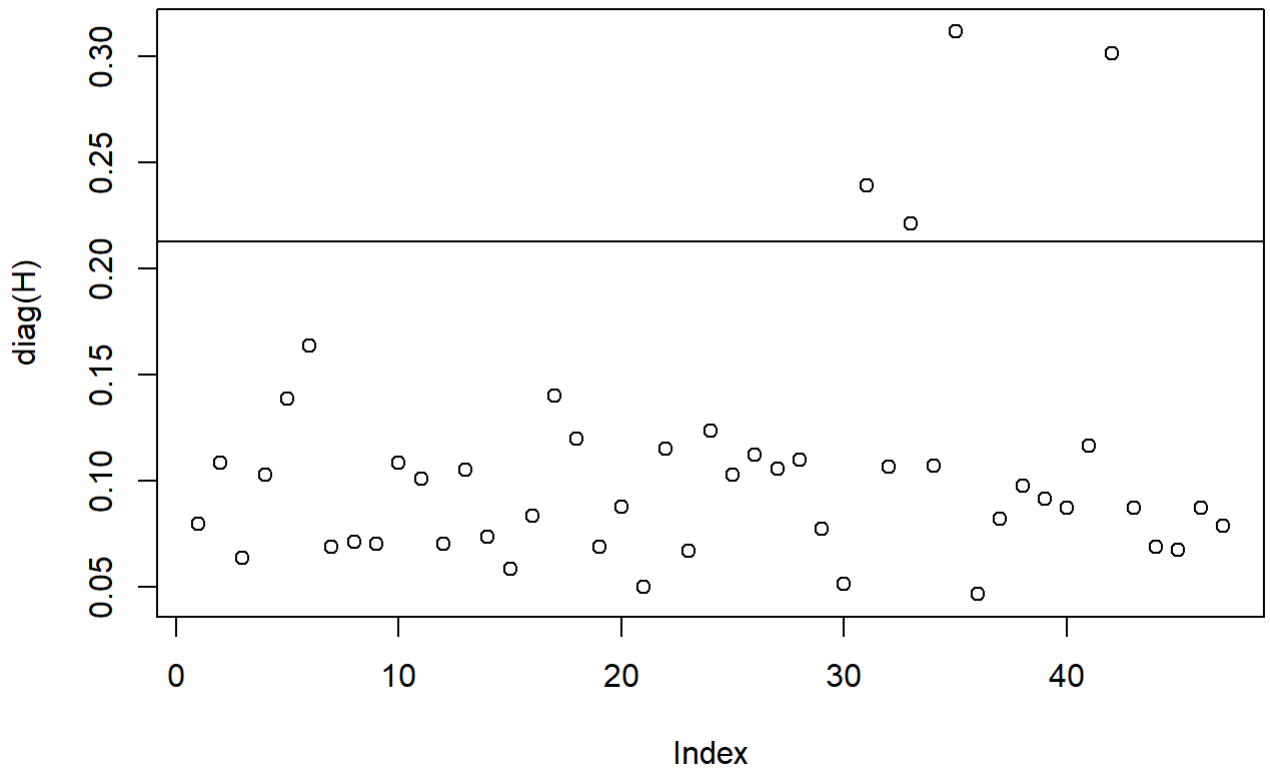
B.Let's check the constant variance by plotting the scatter plot residuals. As there are no pattern so, our assumptions hold true

```
plot(gamb.lm$fit,gamb.lm$res, xlab = "Fitted values", ylab = "Residuals")
```



C. Let's check for Leverage points now. So, we can see that we have four leverage points

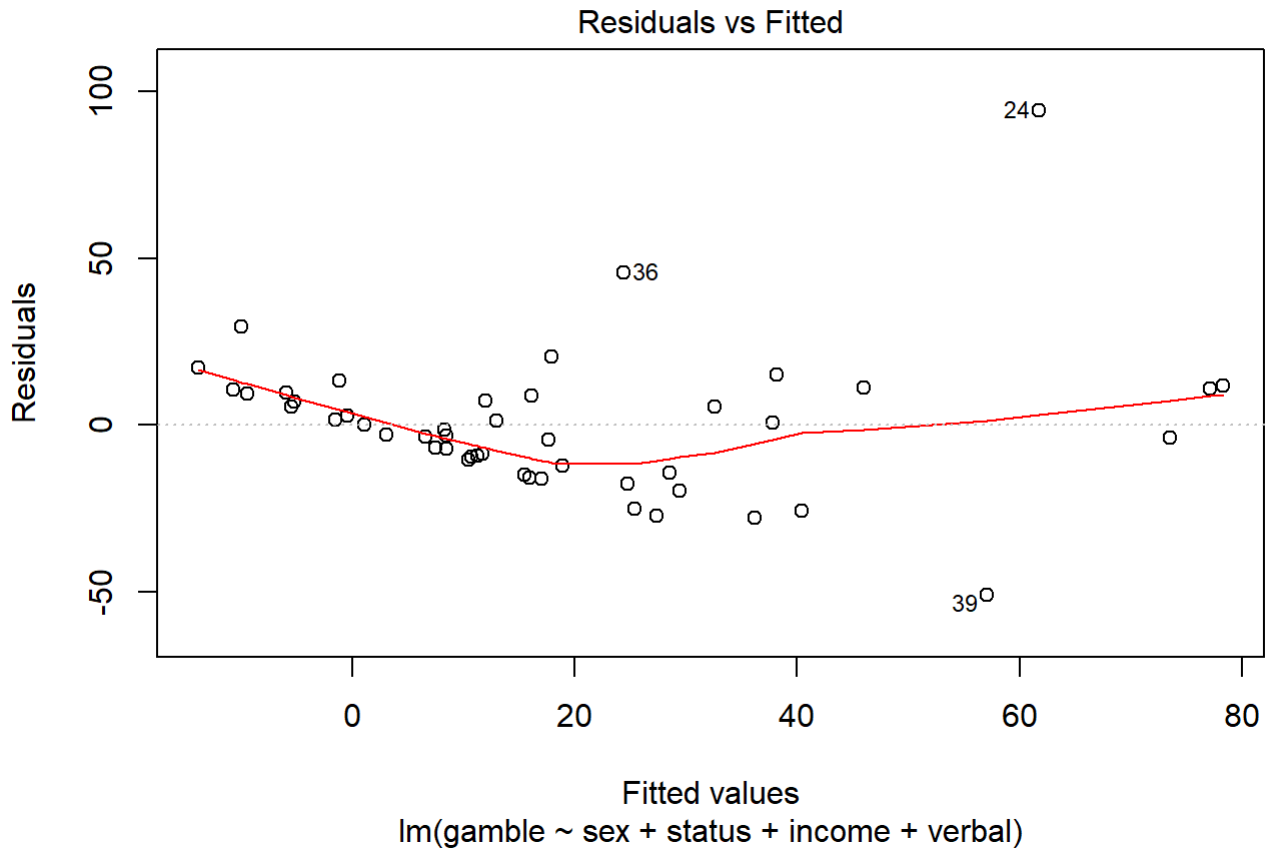
```
plot(diag(H))
abline(h=2*5/47,col=1)
identify(diag(H))
```

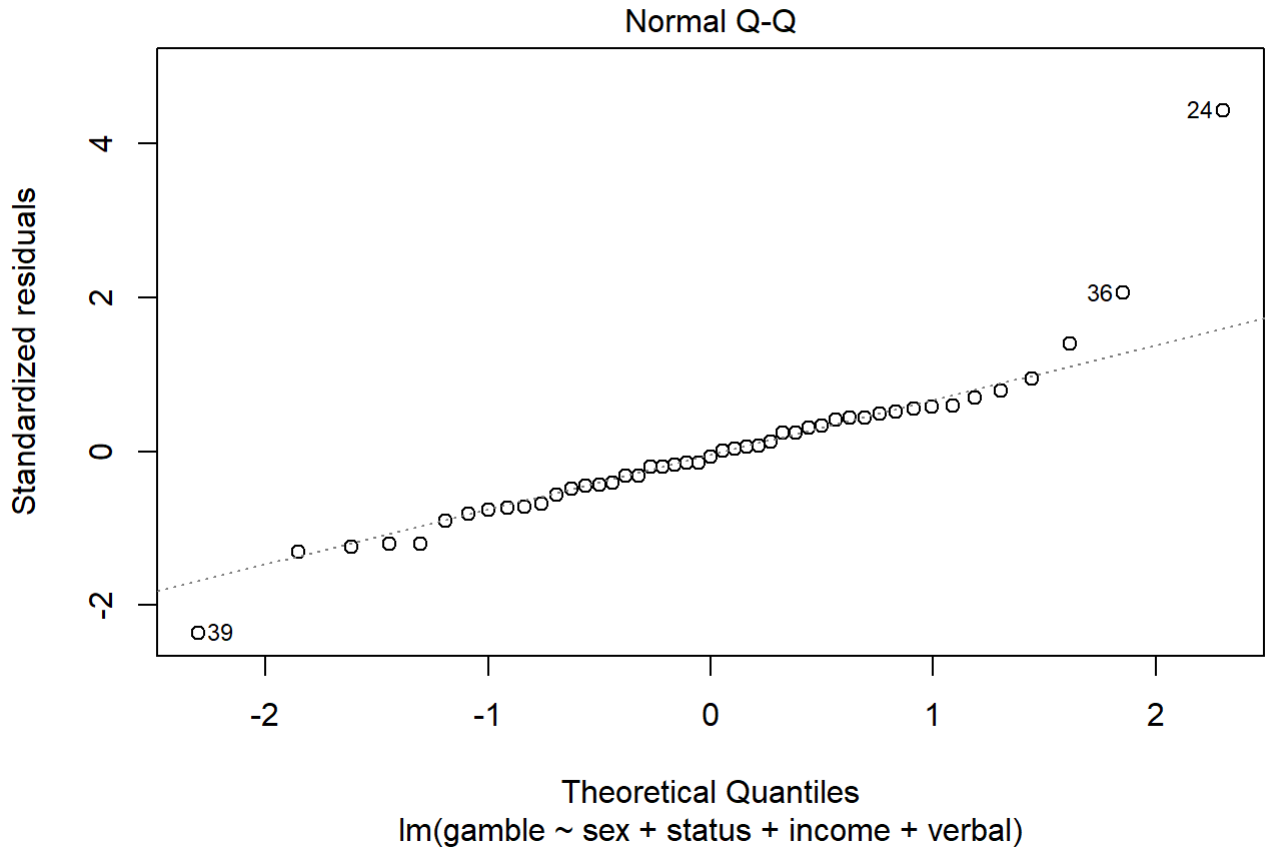


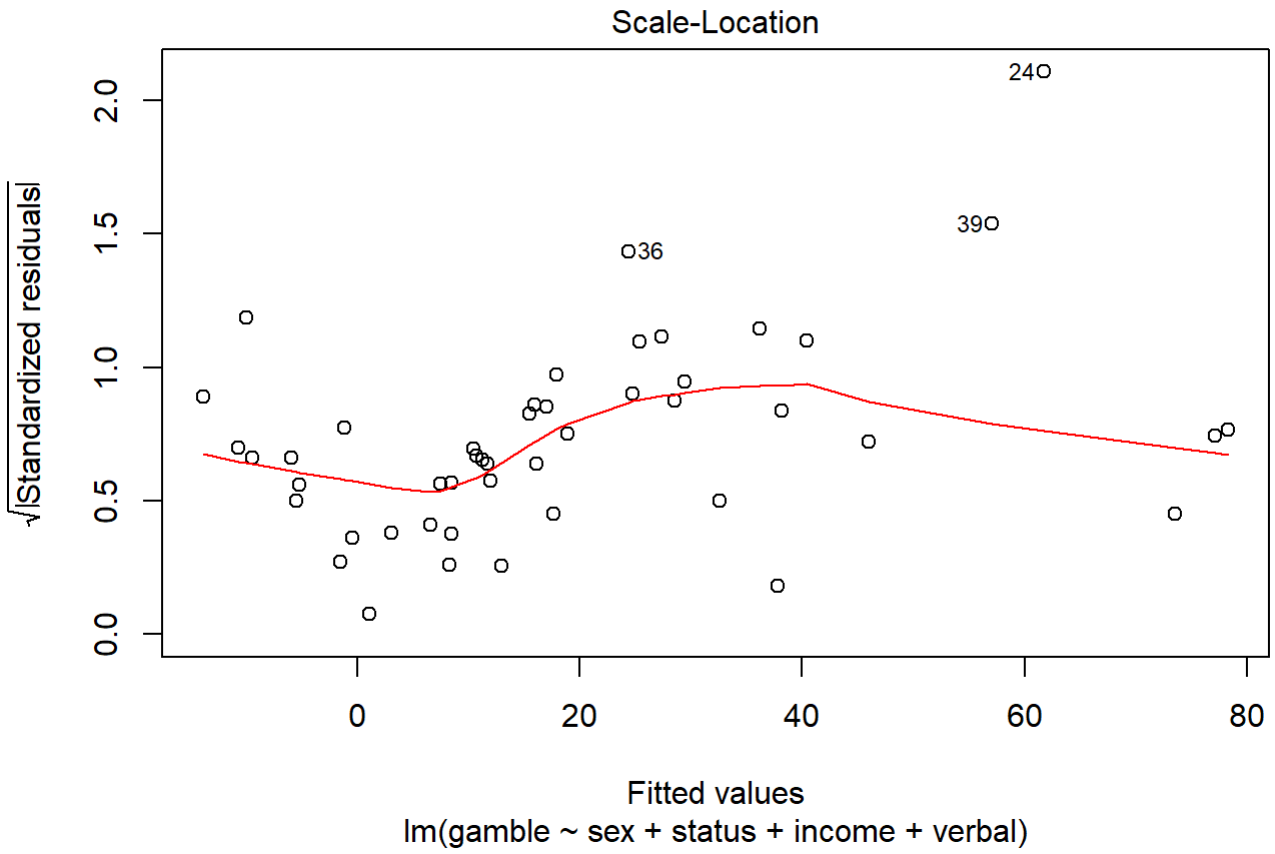

```
## integer(0)
```

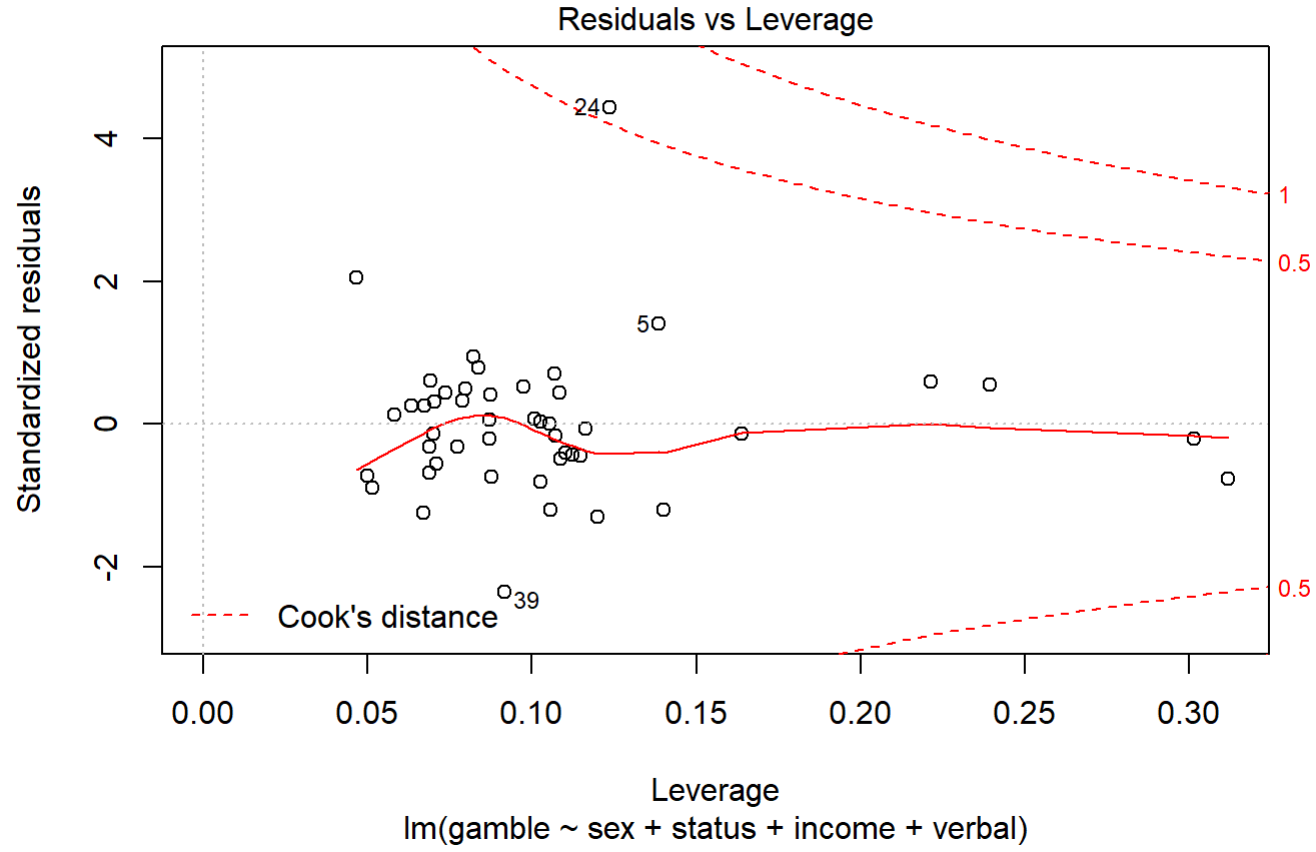
D. Now Let's check for outliers. we can see that we have one outlier, point 24 inside the cook distance.

```
plot(gamb.lm)
```


Residuals vs Fitted
Residuals
100
50
0
-50
24
36
39
0
20
40
60
80
Fitted values
lm(gamble ~ sex + status + income + verbal)

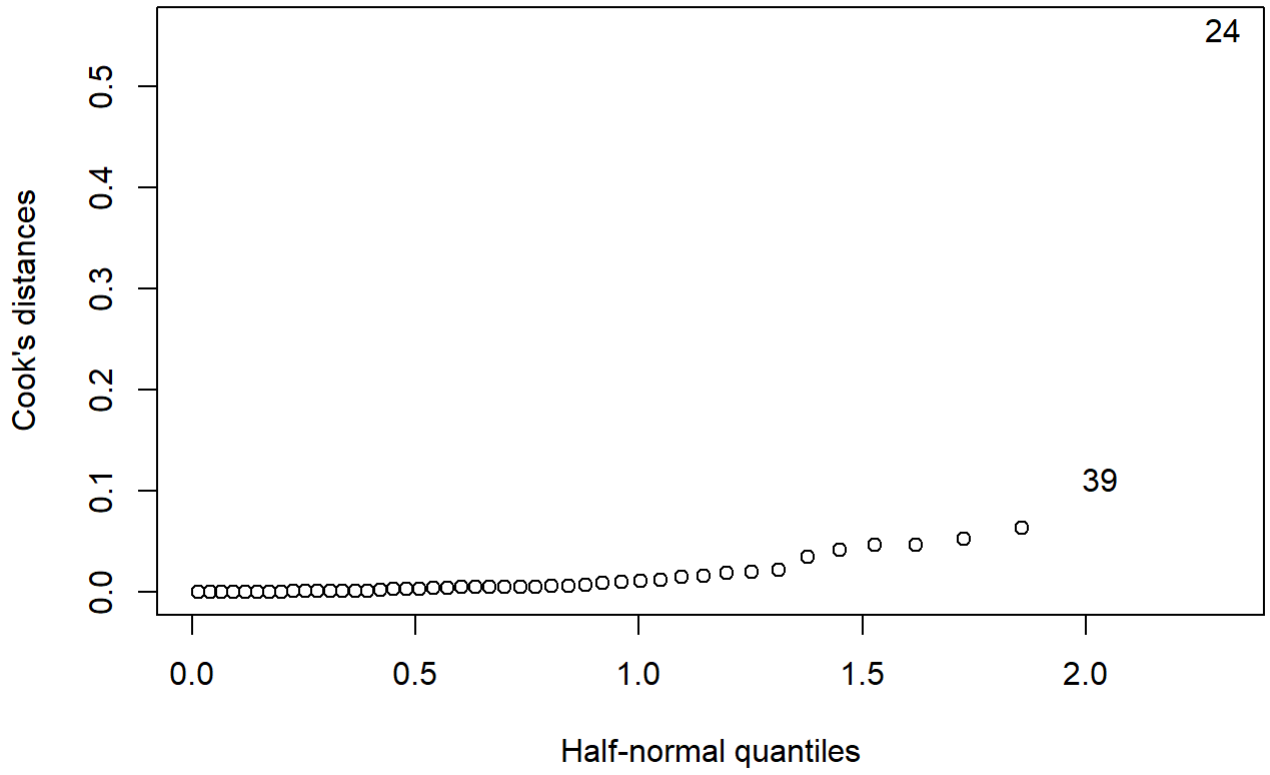

Normal Q-Q
Standardized residuals
4
2
0
-2
24
36
39
-2
-1
0
1
2
Theoretical Quantiles
lm(gamble ~ sex + status + income + verbal)





E. Let's check for influential point. We have one influential point that is point 24.

```
cook=cooks.distance(gamb.lm)
halfnorm(cook,2,ylab="Cook's distances")
```

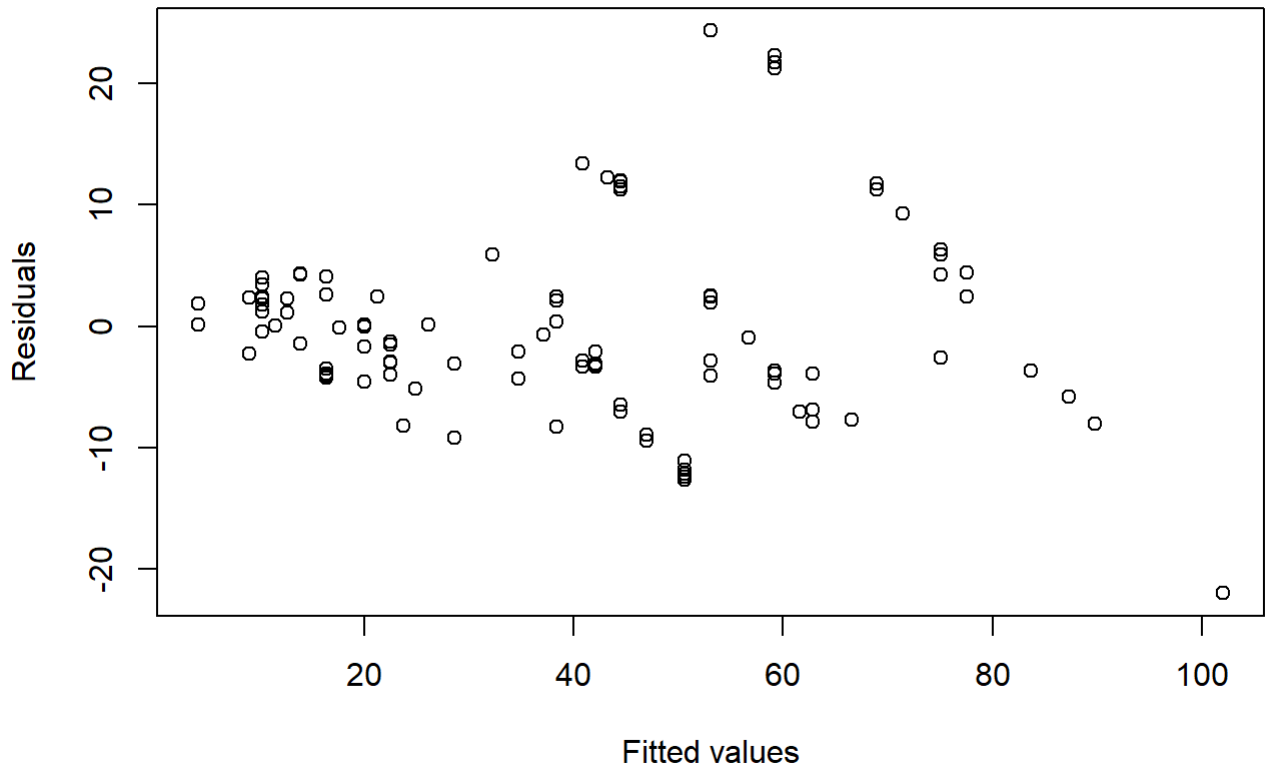


3. A. Let's fit the regression

```
library(faraway)
data(pipeline)
attach(pipeline)
fit = lm(Lab ~ Field)
summary(fit)
```

```
##
## Call:
## lm(formula = Lab ~ Field)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.985  -4.072  -1.431   2.504  24.334
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.96750    1.57479  -1.249    0.214
## Field        1.22297    0.04107  29.778   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.865 on 105 degrees of freedom
## Multiple R-squared:  0.8941, Adjusted R-squared:  0.8931
## F-statistic: 886.7 on 1 and 105 DF,  p-value: < 2.2e-16
```

```
plot(fit$fit,fit$res, xlab = "Fitted values", ylab = "Residuals")
```



We don’t have constant variance because we can see that spread of residuals increases with fitted values and thus there is trend.

B.

```
i=order(Field)
npipe=pipeline[i,]
ff=gl(12,9)[-108]
meanfield=unlist(lapply(split(npipe$Field,ff),mean))
varlab=unlist(lapply(split(npipe$Lab,ff),var))
```

```
fit2 = lm(log(varlab) ~ I(log(meanfield)))
summary(fit2)
```

```
##
## Call:
## lm(formula = log(varlab) ~ I(log(meanfield)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2038 -0.6729  0.1656  0.7205  1.1891
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -0.3538     1.5715  -0.225   0.8264
## I(log(meanfield))   1.1244     0.4617   2.435   0.0351 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.018 on 10 degrees of freedom
## Multiple R-squared:  0.3723, Adjusted R-squared:  0.3095
## F-statistic: 5.931 on 1 and 10 DF,  p-value: 0.03513
```

```
fit2 = lm(log(varlab) ~ I(log(meanfield))-1)
summary(fit2)
```

```
##
## Call:
## lm(formula = log(varlab) ~ I(log(meanfield)) - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1240 -0.6974  0.1401  0.7452  1.2352
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## I(log(meanfield))  1.02231    0.08256   12.38 8.42e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9734 on 11 degrees of freedom
## Multiple R-squared:  0.9331, Adjusted R-squared:  0.927
## F-statistic: 153.3 on 1 and 11 DF,  p-value: 8.417e-08
```

```
summary(fit2)
```

```
##
## Call:
## lm(formula = log(varlab) ~ I(log(meanfield)) - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1240 -0.6974  0.1401  0.7452  1.2352
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## I(log(meanfield))  1.02231    0.08256   12.38 8.42e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9734 on 11 degrees of freedom
## Multiple R-squared:  0.9331, Adjusted R-squared:  0.927
## F-statistic: 153.3 on 1 and 11 DF,  p-value: 8.417e-08
```

```
a0=1
a1=1
w=1/Field
wfit = lm(Lab~Field, weights = 1/w)
summary(wfit)
```

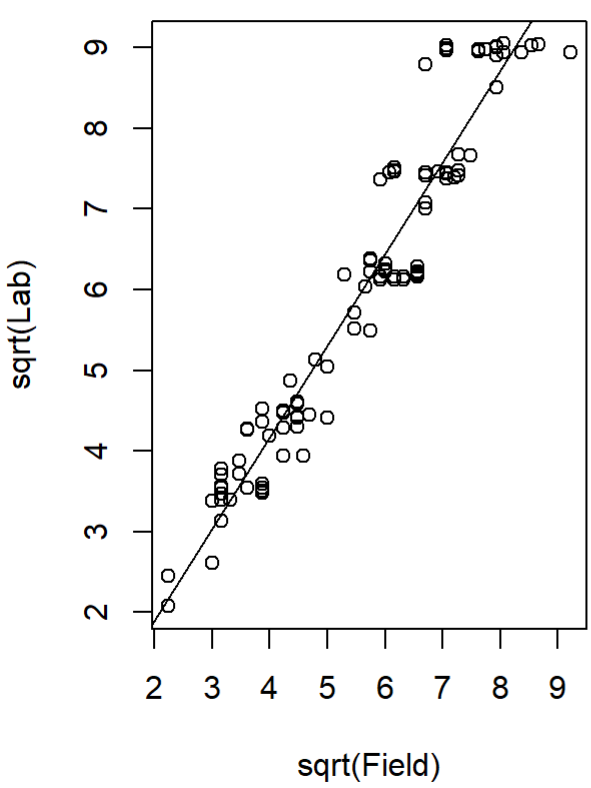
```
##
## Call:
## lm(formula = Lab ~ Field, weights = 1/w)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -189.71  -23.77   -9.41   11.66  163.51
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.47232    2.39210  -0.197    0.844
## Field        1.18882    0.05061  23.490   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 54 on 105 degrees of freedom
## Multiple R-squared:  0.8401, Adjusted R-squared:  0.8386
## F-statistic: 551.8 on 1 and 105 DF,  p-value: < 2.2e-16
```

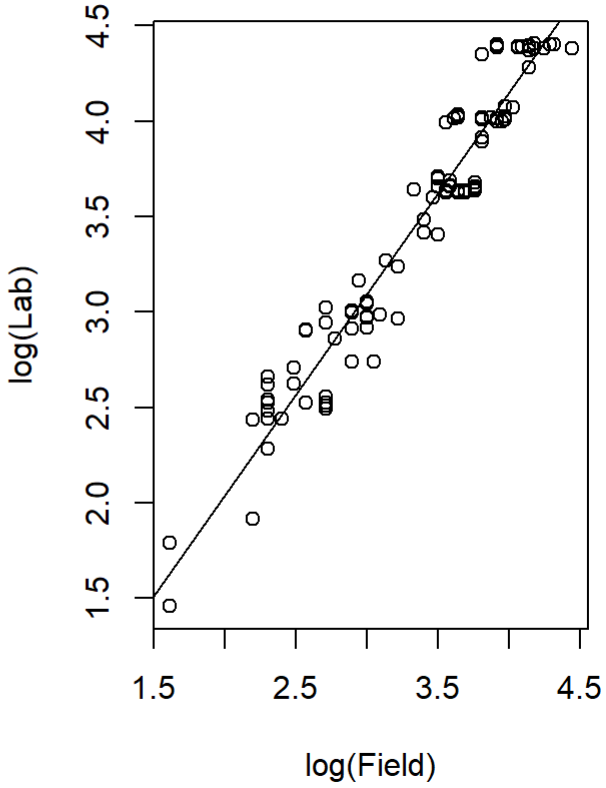
```
wfit = lm(Lab~Field-1, weights = 1/w)
summary(wfit)
```

```
##
## Call:
## lm(formula = Lab ~ Field - 1, weights = 1/w)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -186.81  -24.46  -10.69   10.51  163.14
##
## Coefficients:
##       Estimate Std. Error t value Pr(>|t|)
## Field  1.17956    0.01897   62.17   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 53.75 on 106 degrees of freedom
## Multiple R-squared:  0.9733, Adjusted R-squared:  0.9731
## F-statistic:  3865 on 1 and 106 DF,  p-value: < 2.2e-16
```

C.We can see that both square root and log transformations did well but log transformations did pretty better than squareroot transformation.

```
par(mfrow=c(1,2))
plot(sqrt(Field),sqrt(Lab))
abline(lm(sqrt(Lab)~I(sqrt(Field))))
plot(log(Field),log(Lab))
abline(lm(log(Lab)~I(log(Field))))
```






```
par(mfrow=c(1,1))
```