Exercise 3.14

$$R = \begin{bmatrix} 1 & .5 \\ .5 & 1 \end{bmatrix} \Rightarrow R^{-1} = \frac{1}{1 \times 1 - .5 \times .5} \begin{bmatrix} 1 & -.5 \\ -.5 & 1 \end{bmatrix} = \frac{4}{3} \cdot \begin{bmatrix} 1 & -.5 \\ -.5 & 1 \end{bmatrix}$$

$$\underline{\hat{\beta}}^* = R^{-1} r = \frac{4}{3} \begin{bmatrix} 1 & -.5 \\ -.5 & 1 \end{bmatrix} \begin{bmatrix} .4 \\ .8 \end{bmatrix} = \begin{bmatrix} 0 \\ .8 \end{bmatrix} \Rightarrow \boxed{\hat{\beta}_1^* = 0 \quad \hat{\beta}_2^* = .8}$$

$$\hat{\beta}_1 = \hat{\beta}_1^* \cdot \frac{S_y}{S_{x_1}} = \boxed{0} \qquad \hat{\beta}_2 = \hat{\beta}_2^* \cdot \frac{S_y}{S_{x_2}} = .8 \times \frac{5}{4} = \boxed{1}$$

$$\bar{y} = \hat{\beta}_0 + \hat{\beta}_1 \bar{x}_1 + \hat{\beta}_2 \bar{x}_2 \Rightarrow \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}_1 - \hat{\beta}_2 \bar{x}_2 = 10 - 0 - 1 \times 5 = \boxed{5}$$

Predictive Analytics 1 - Homework 3

Parth Patel, Kristian Nikolov, Jieda Li, Kristiyan Dimitrov

10/17/2019

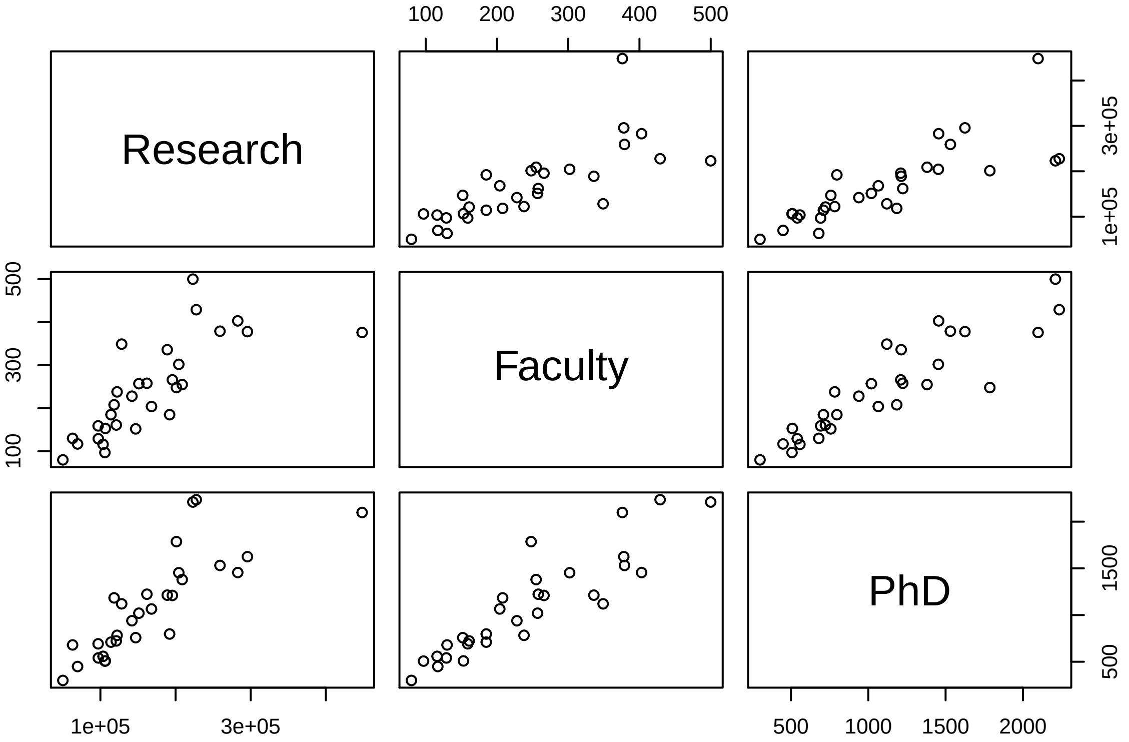
Exercise 3.13

a)

```
# First we need to import our data
data=read.csv("/Users/kristiyan/Documents/MSiA 401 - Predictive 1/Homeworks/Research.csv",stringsAsFact
# Below we transform the Research column to numeric by removing the $ sign and the commas
data$Research <- as.numeric(gsub(',','',substr(data$Research, start=2, stop=nchar(data$Research)-1), fi
# Finally, we don't really need the PhD Faculty column
data <- data[-4]
str(data)
```

```
## 'data.frame':    30 obs. of  4 variables:
##  $ University: chr  "MIT" "Stanford" "Berkeley" "CalTech" ...
##  $ Research  : num  448324 201160 208862 105985 195765 ...
##  $ Faculty   : int  376 248 255 97 266 378 500 379 302 429 ...
##  $ PhD       : int  2098 1786 1380 507 1210 1625 2210 1531 1453 2235 ...
```

```
# Matrix Scatterplot
plot(data[2:4])
```



There definitely appears to be a linear relationship b/w Research and # of Faculty / PhD Students.

```
# This produces the Correlation Matrix of the 3 variables
cor(data[2:4])
```

```
##           Research   Faculty       PhD
## Research 1.0000000 0.7648421 0.8174254
## Faculty  0.7648421 1.0000000 0.9036829
## PhD      0.8174254 0.9036829 1.0000000
```

b)

```
# Fitting a linear model
lmfit = lm(data$Research ~ data$Faculty + data$PhD , data = data)
summary(lmfit)
```

```
##
## Call:
## lm(formula = data$Research ~ data$Faculty + data$PhD, data = data)
##
## Residuals:
##    Min      1Q Median     3Q    Max
## -90804 -16921  -2921  13605 159743
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  23525.91   22034.47   1.068   0.2951
## data$Faculty   107.13     193.39   0.554   0.5842
## data$PhD       107.14      40.06   2.675   0.0125 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 49040 on 27 degrees of freedom
## Multiple R-squared:  0.6719, Adjusted R-squared:  0.6476
## F-statistic: 27.65 on 2 and 27 DF,  p-value: 2.923e-07
```

We note that Faculty has a t-statistic of 0.554, which means it is not significant. At the same time, PhD has a t-statistic 2.675, which makes it significant. This anomalous result is due to the causal dependence b/w Faculty & PhDs. i.e. our predictor variables are not independent of each other. i.e. there is some level of multicolinearity.

```
# install.packages("ppcor") # This package allows us to calculate Partial Correlation Coefficients
library("ppcor")
```

```
## Loading required package: MASS
```

```
pcor(data[2:4], method = "pearson")$estimate # The Partial Correlation Matrix
```

```
##           Research   Faculty       PhD
## Research 1.0000000 0.1060117 0.4576694
## Faculty  0.1060117 1.0000000 0.7504387
## PhD      0.4576694 0.7504387 1.0000000
```

```
pcor(data[2:4], method = "pearson")$statistic  # The t-statistics
```

```
##           Research   Faculty      PhD
## Research 0.0000000 0.5539747 2.674682
## Faculty  0.5539747 0.0000000 5.899769
## PhD      2.6746821 5.8997693 0.000000
```

We note that the t-statistic for Research w.r.t PhD is 2.6746, which is approx. what we got from the lmfit (2.675) Similarly, here we get a t-statistic for Research w.r.t to Faculty of 0.5539 while from the lmfit we got

(.554) i.e. They are the same!

Exercise 3.15

```
# First we import our data
data <- read.csv("/Users/kristiyan/Documents/MSiA 401 - Predictive 1/Homeworks/Sales Data.csv",stringsAs
str(data)
```

```
## 'data.frame':    10 obs. of  3 variables:
##  $ x1: int  31 46 40 49 38 49 31 38 33 42
##  $ x2: num  1.85 2.8 2.2 2.85 1.8 2.8 1.85 2.3 1.6 2.15
##  $ y : num  4.2 7.28 5.6 8.12 5.46 7.42 3.36 5.88 4.62 5.88
```

a)

```
# The correlation matrix R b/w x1 & x2
corr_mat=cor(data[1:2])
print(corr_mat)
```

```
##           x1        x2
## x1 1.0000000 0.9132577
## x2 0.9132577 1.0000000
```

```
print(cor(data[-2])[2]) # Correlation between y & x1
```

```
## [1] 0.9708553
```

```
print(cor(data[-1])[2]) # Correlation between y & x2
```

```
## [1] 0.9040219
```

```
r=c(cor(data[-2])[2],cor(data[-1])[2]) # Vector r with the correlation b/w y and x1, x2
print(c(cor(data[-2])[2],cor(data[-1])[2]))
```

```
## [1] 0.9708553 0.9040219
```

```
# Manual Calculation of partial correlation coefficient of y w.r.t. x2 given x1
par_corr_yx2_given_x1 = (.9040219-.9708553*.9132577) / sqrt((1-(.9708553)^2)*(1-(.9132577)^2))
print(par_corr_yx2_given_x1)
```

```
## [1] 0.1780168
```

```
# Confirming the above result with the pcor function
pcor(data)$estimate[3,2]
```

```
## [1] 0.1780173
```

```
# Manual Calculation of partial correlation coefficient of y w.r.t. x1 given x2
par_corr_yx1_given_x2 = (.9708553-.9040219*.9132577) / sqrt((1-(.9040219)^2)*(1-(.9132577)^2))
print(par_corr_yx1_given_x2)
```

```
## [1] 0.8340516
```

```
# Confirming the above result with the pcor function
pcor(data)$estimate[3,1]
```

```
## [1] 0.8340515
```

b)

```
# First we need to find the inverse of the R matirx
inv_corr_mat = solve(corr_mat)
print(inv_corr_mat)
```

```
##           x1        x2
## x1  6.025532 -5.502863
## x2 -5.502863  6.025532
```

```
# Now we find the standardized regression coefficients by multiplying R^(-1) %*% r
print(inv_corr_mat %*% r)
```

```
##         [,1]
## x1 0.8752107
## x2 0.1047290
```

In terms of "How do they compare with the partial correlation coefficients?" We see they are proportianally similar to the partial correlation coefficients i.e. we see that the beta_hat_star for x1 is .8752 and the partial corr. coeff. of y w.r.t x1 given x2 is 0.8340516 i.e. y appears to depend much more on x1 than on x2.

c)

```
sd_y = sd(data$y) # standard deviation of y
sd_x1 = sd(data$x1) # standard deviation of x1
sd_x2 = sd(data$x2) # standard deviation of x2
lmfit = lm(y ~ x1 + x2, data)
beta_1 = lmfit$coefficients[2] # This is beta_1: the coefficient of x_1 from the regression
beta_2 = lmfit$coefficients[3] # This is beta_2: the coefficient of x_2 from the regression
print(beta_1)
```

```
##        x1
## 0.1922408
```

```
print(beta_2)
```

```
##        x2
## 0.3406266
```

```
# Calculate the beta_hat_stars by scaling:
beta_1_star = (beta_1 / sd_y) * sd_x1
beta_2_star = (beta_2 / sd_y) * sd_x2
print(beta_1_star)
```

```
##        x1
## 0.8752107
```

```
print(beta_2_star)
```

```
##       x2
## 0.104729
```

```
# We double verify we got the same as before
print(inv_corr_mat %*% r)
```

```
##         [,1]
## x1 0.8752107
## x2 0.1047290
```

Indeed, we got the same results!

d) Which predictors are better: (beta__1, beta__2) or (beta__1__star, beta__2__star)?

The two sets of parameters ((beta_hat__1, beta_hat__2) & (beta_hat__1__star, beta_hat__2__star) are equivalent in terms of prediction i.e. we could use either set for our final model. However, the standardized (star) ones allow us to compare between x1 and x2 's influence on y. i.e. the standardized parameters are unitless and we can therefore compare them.

Exercise 3.16

a)

```
# We begin by importing our data
data <- read.csv("/Users/kristiyan/Documents/MSiA 401 - Predictive 1/Homeworks/salaries.csv",stringsAsF
str(data)
```

```
## 'data.frame':    46 obs. of  8 variables:
##  $ Salary   : int  38985 32920 29548 24749 41889 31528 38791 39828 28985 32782 ...
##  $ YrsEm    : int  18 15 5 6 22 3 21 18 0 0 ...
##  $ PriorYr  : int  7 3 6 2 16 11 4 6 1 1 ...
##  $ Education: int  9 9 1 0 7 3 5 5 4 7 ...
##  $ ID       : int  412 458 604 598 351 674 356 415 693 694 ...
##  $ Gender   : chr  "Female" "Male" "Female" "Male" ...
##  $ Dept     : chr  "Sales" "Sales" "Sales" "Sales" ...
##  $ Super    : int  5 4 0 1 7 6 9 5 4 0 ...
```

```
# Adding the log_10(Salary) variable as response
data <- data %>%
  mutate(log_Salary = log10(data$Salary))

str(data)
```

```
## 'data.frame':    46 obs. of  9 variables:
##  $ Salary    : int  38985 32920 29548 24749 41889 31528 38791 39828 28985 32782 ...
##  $ YrsEm     : int  18 15 5 6 22 3 21 18 0 0 ...
##  $ PriorYr   : int  7 3 6 2 16 11 4 6 1 1 ...
##  $ Education : int  9 9 1 0 7 3 5 5 4 7 ...
##  $ ID        : int  412 458 604 598 351 674 356 415 693 694 ...
##  $ Gender    : chr  "Female" "Male" "Female" "Male" ...
##  $ Dept      : chr  "Sales" "Sales" "Sales" "Sales" ...
##  $ Super     : int  5 4 0 1 7 6 9 5 4 0 ...
##  $ log_Salary: num  4.59 4.52 4.47 4.39 4.62 ...
```

```
data$Gender <- relevel(factor(data$Gender), "Male") # Choosing Male as our reference category for Gende
data$Dept <- relevel(factor(data$Dept), "Purchase") # Choosing Purchase as our reference category for D
lmfit = lm (log_Salary ~ YrsEm + PriorYr + Education + Gender + Dept + Super , data = data)
summary(lmfit)
```

```
##
## Call:
## lm(formula = log_Salary ~ YrsEm + PriorYr + Education + Gender +
##     Dept + Super, data = data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.089659 -0.024036 -0.004498  0.028587  0.089410
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.4287934  0.0213399 207.535  < 2e-16 ***
## YrsEm         0.0074788  0.0011931   6.269 2.72e-07 ***
## PriorYr       0.0016839  0.0019568   0.861 0.395039
## Education     0.0170345  0.0033360   5.106 1.02e-05 ***
## GenderFemale  0.0230683  0.0142917   1.614 0.115002
## DeptAdvertse -0.0387774  0.0249146  -1.556 0.128124
## DeptEngineer -0.0057292  0.0197703  -0.290 0.773597
## DeptSales    -0.0937783  0.0225745  -4.154 0.000185 ***
## Super         0.0003901  0.0008056   0.484 0.631115
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04586 on 37 degrees of freedom
## Multiple R-squared:  0.8634, Adjusted R-squared:  0.8338
## F-statistic: 29.22 on 8 and 37 DF,  p-value: 9.629e-14
```

Indeed our coefficients are the same as those specified in the book!

b)

```r
data$Gender <- relevel(factor(data$Gender), "Female") # Choosing Female as our reference category for G
data$Dept <- relevel(factor(data$Dept), "Sales") # Choosing Salesa s our reference category for Dept
lmfit2 = lm (log_Salary ~ YrsEm + PriorYr + Education + Gender + Dept + Super , data = data)
summary(lmfit2)
```

```
##
## Call:
## lm(formula = log_Salary ~ YrsEm + PriorYr + Education + Gender +
##     Dept + Super, data = data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.089659 -0.024036 -0.004498  0.028587  0.089410
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.3580834  0.0248414 175.436  < 2e-16 ***
## YrsEm         0.0074788  0.0011931   6.269 2.72e-07 ***
## PriorYr       0.0016839  0.0019568   0.861 0.395039
## Education     0.0170345  0.0033360   5.106 1.02e-05 ***
## GenderMale   -0.0230683  0.0142917  -1.614 0.115002
## DeptPurchase  0.0937783  0.0225745   4.154 0.000185 ***
## DeptAdvertse  0.0550009  0.0230111   2.390 0.022045 *
## DeptEngineer  0.0880491  0.0180562   4.876 2.07e-05 ***
## Super         0.0003901  0.0008056   0.484 0.631115
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04586 on 37 degrees of freedom
## Multiple R-squared:  0.8634, Adjusted R-squared:  0.8338
## F-statistic: 29.22 on 8 and 37 DF,  p-value: 9.629e-14
```

We notice the coefficient for Female in the first lmfit (.02306) is the same as the coefficient for Male in the second lmfit, but with a negative sign! (-.02306). The same holds for the coefficients for the dummy variables of Sales & Purchases (-.09377 & .09377).

The reason for this is we have simply changed the reference point. The marginal impact of being in Sales vs. being in Purchase is, of course, the same as being in Purchase vs. being in Sales.

Also note, for the Advertising & Engineering categories, the coefficients change, which is to be expected. The marginal impact of being in Advertising or Engineering changes when we change the reference from Sales to Purchase or vice-versa.

c)

The fact that Engg is non-significant when the reference is Purchase tells us that being in Engg does NOT make a significant marginal contribution to Salary vs being in Purchase. On the other hand, the fact that Engg is highly significant when the reference is Sales, tells us that there IS a significant marginal contribution to Salary when you are in Engg vs. being in Sales

d)

```r
lmfit3 = lm (log_Salary ~ YrsEm + Education + Dept , data = data)
summary(lmfit3)
```

```
##
## Call:
## lm(formula = log_Salary ~ YrsEm + Education + Dept, data = data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.114193 -0.028068 -0.002002  0.033938  0.081774
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.351412   0.019604 221.966  < 2e-16 ***
## YrsEm        0.007660   0.001208   6.341 1.57e-07 ***
## Education    0.018371   0.003124   5.881 6.95e-07 ***
## DeptPurchase 0.087593   0.022740   3.852 0.000414 ***
## DeptAdvertse 0.051105   0.023180   2.205 0.033287 *
## DeptEngineer 0.085085   0.018001   4.727 2.82e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04677 on 40 degrees of freedom
## Multiple R-squared:  0.8464, Adjusted R-squared:  0.8272
## F-statistic: 44.09 on 5 and 40 DF,  p-value: 3.099e-15
```

We note that: 1) The adjusted R_squared of our newest model (.8272) is not very different from that of our previous two models (.8338) This means that the predictive power of our model has not deteriorated when we removed the insignificant variable 2) The F-statistic for our latest model is 44.09, while for our first two models it was 29.22 Therefore, by removing the insignificant variables from our model we have actually improved the significant of our overall model quite a bit!

Exercise 4.4

```
# As always, we begin by importing our data
gpa <- read.csv("/Users/kristiyan/Documents/MSiA 401 - Predictive 1/Datasets/gpa.csv", stringsAsFactors
str(gpa)
```

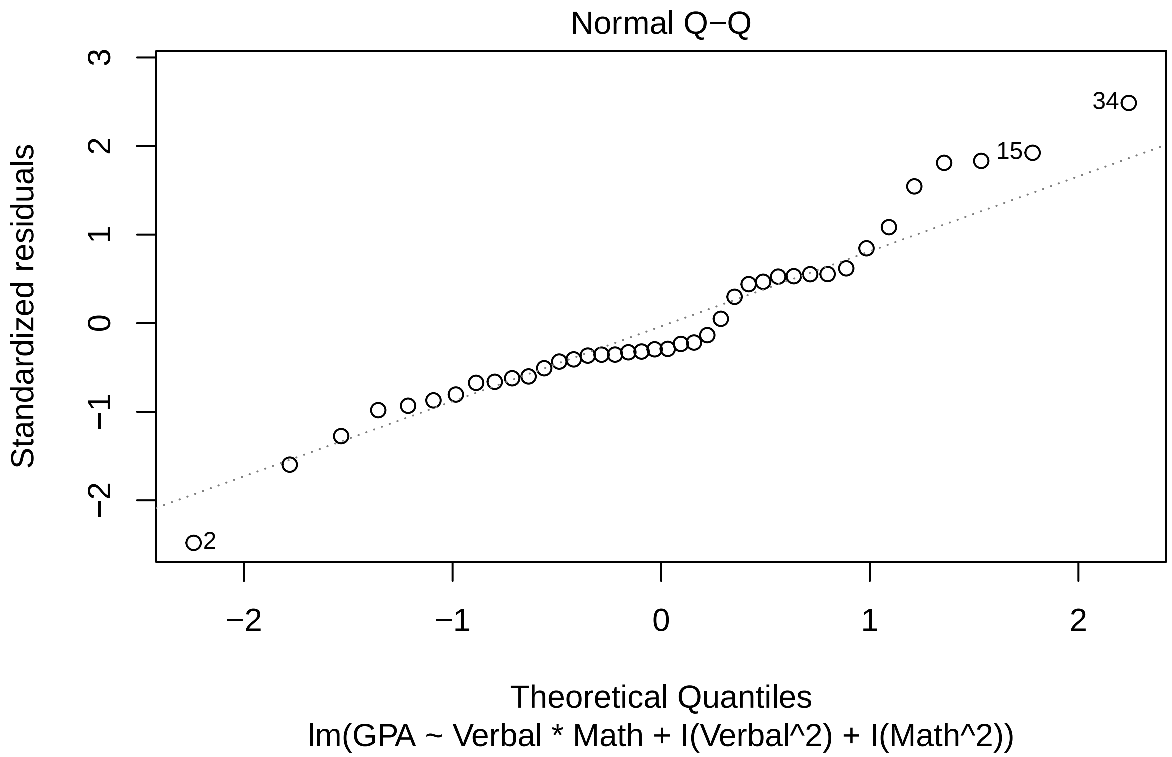
```
## 'data.frame':    40 obs. of  3 variables:
##  $ Verbal: int  81 68 57 100 54 82 75 58 55 49 ...
##  $ Math  : int  87 99 86 49 83 86 74 98 54 81 ...
##  $ GPA   : num  3.49 2.89 2.73 1.54 2.56 3.43 3.59 2.86 1.46 2.11 ...
```

```
# Fit with linear & quadratic terms of Verbal & Math
infit = lm(GPA ~ Verbal*Math + I(Verbal^2) + I(Math^2), data = gpa)
summary(infit)
```

```
##
## Call:
## lm(formula = GPA ~ Verbal * Math + I(Verbal^2) + I(Math^2), data = gpa)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.39203 -0.10935 -0.04432  0.09508  0.42601
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -9.9167631  1.3544134  -7.322 1.75e-08 ***
## Verbal       0.1668098  0.0212447   7.852 3.85e-09 ***
## Math         0.1375972  0.0267340   5.147 1.11e-05 ***
## I(Verbal^2) -0.0011082  0.0001173  -9.449 4.88e-11 ***
## I(Math^2)   -0.0008433  0.0001594  -5.290 7.23e-06 ***
## Verbal:Math  0.0002411  0.0001440   1.675    0.103
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1871 on 34 degrees of freedom
## Multiple R-squared:  0.9366, Adjusted R-squared:  0.9272
## F-statistic: 100.4 on 5 and 34 DF,  p-value: < 2.2e-16
```

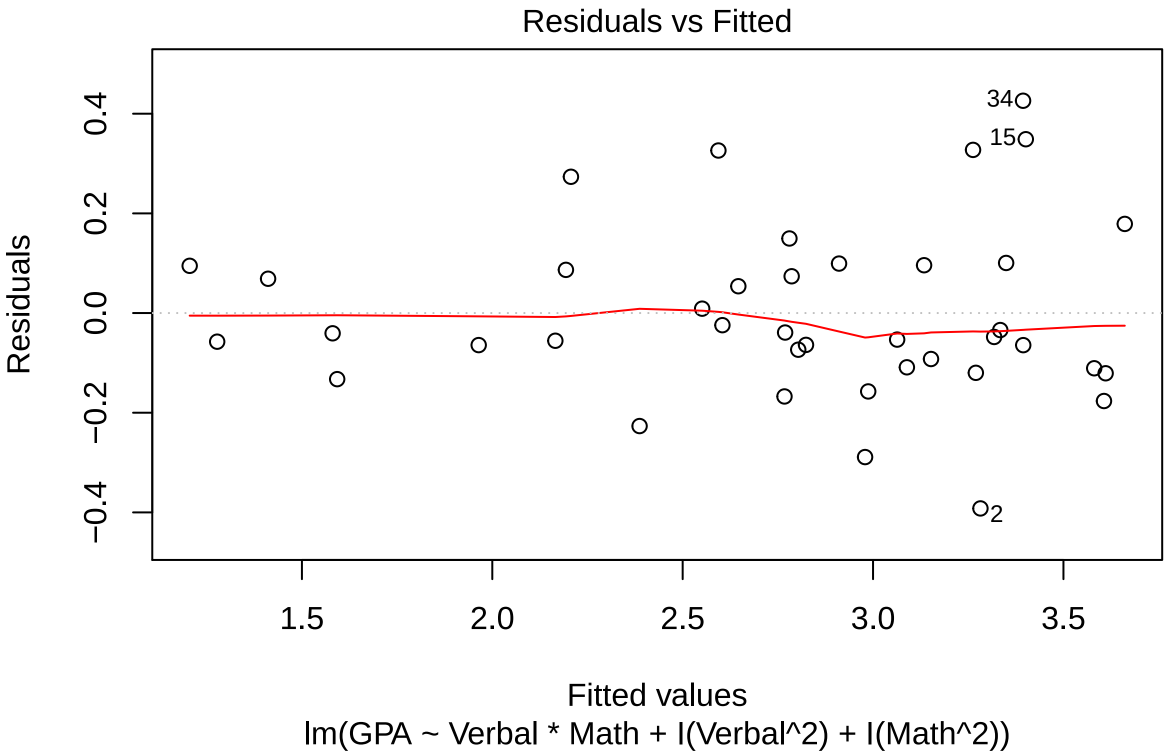
a)

```
plot(infit, which=2) # Making the Normal Q-Q plot
```



The Normal Q-Q plot exhibits a "step-wise" pattern and doesn't follow a straight line., especially at the tail ends. Therefore, the normality assumption appears to be violated.

```
plot(infit, which=1) # Making the Residuals ~ Fitted Values plot
```



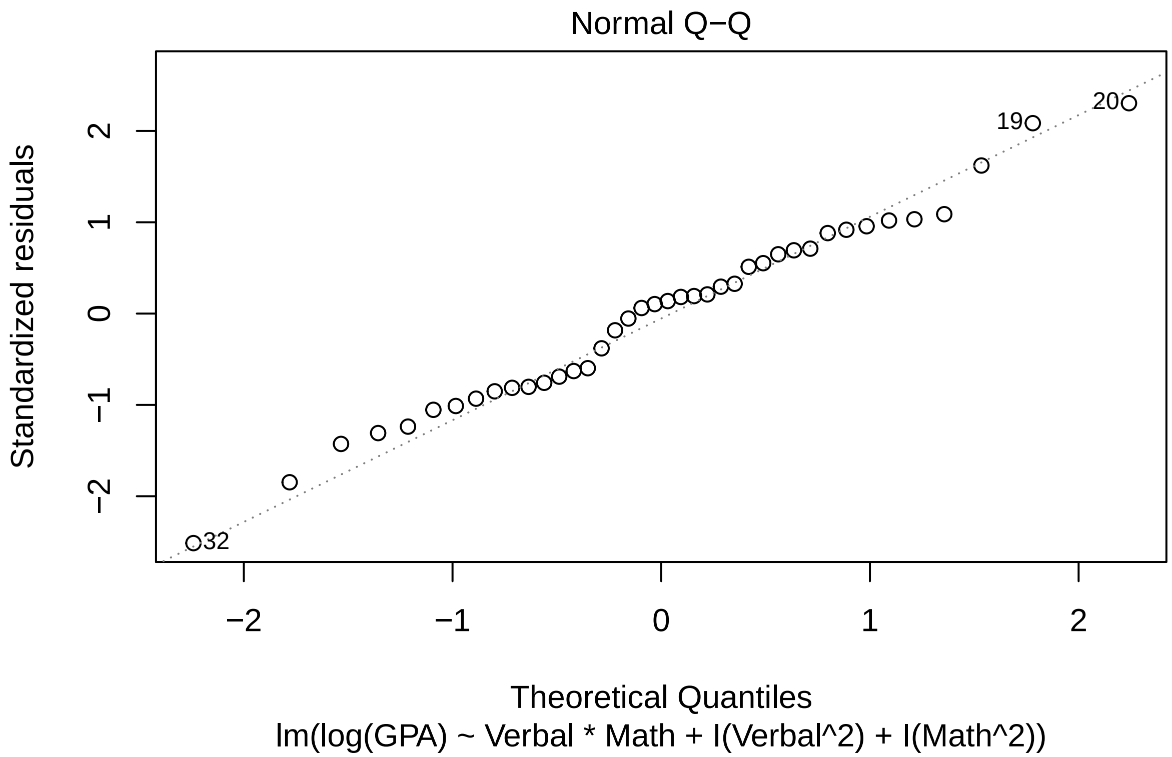
The residuals vs. fitted values plot shows us that the residuals increase with the fitted values in a linear fashion i.e. they are in a cone-shape. A linear relationship b/w fitted values and the standard deviation of the residuals, suggests a squared relationship between fitted values & variation of residuals i.e. fitted values ~ Var(residuals)^2.

Therefore, by the delta method discussed in class, this suggests the log-transformation of GPA as an appropriate variance stabilizing transform. ## b)

```r
# Performing the log Transformation
logfit = lm(log(GPA) ~ Verbal*Math + I(Verbal^2) + I(Math^2), data = gpa)
summary(logfit)
```

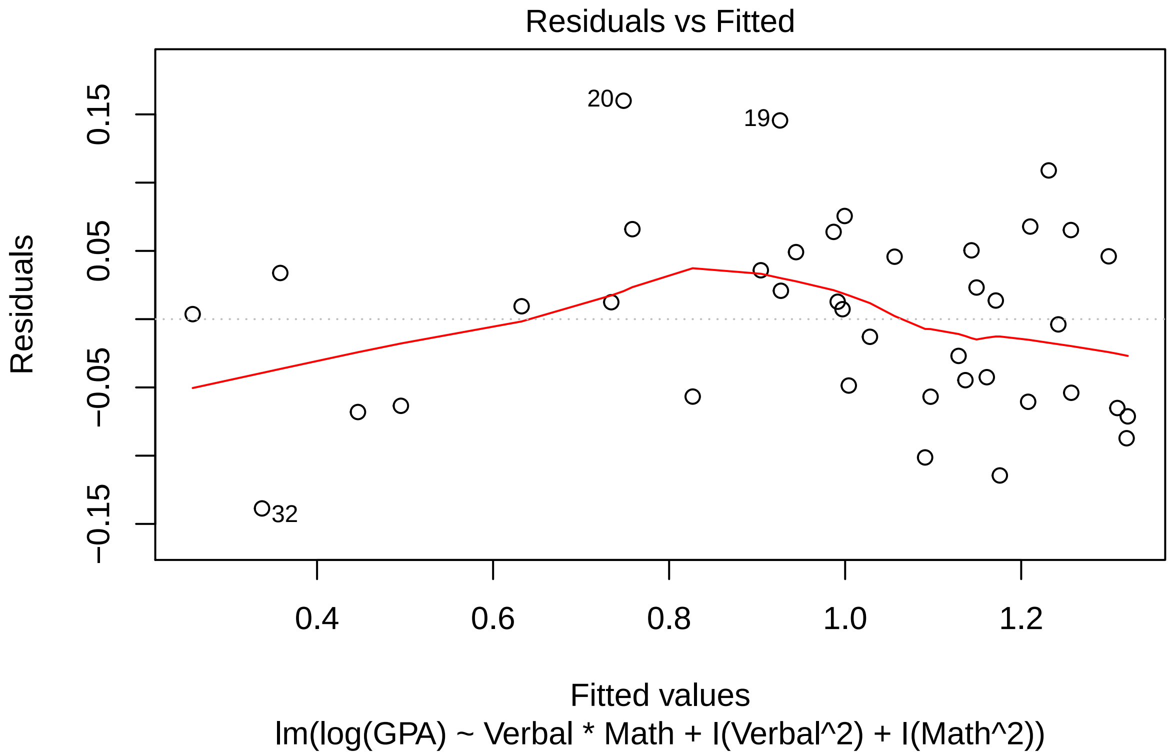
```
##
## Call:
## lm(formula = log(GPA) ~ Verbal * Math + I(Verbal^2) + I(Math^2),
##     data = gpa)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.138636 -0.056735  0.008366  0.046788  0.159979
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.479e+00  5.311e-01 -10.315 5.24e-12 ***
## Verbal       8.591e-02  8.331e-03  10.311 5.29e-12 ***
## Math         7.433e-02  1.048e-02   7.090 3.44e-08 ***
## I(Verbal^2) -5.248e-04  4.600e-05 -11.410 3.58e-13 ***
## I(Math^2)   -4.199e-04  6.252e-05  -6.717 1.02e-07 ***
## Verbal:Math -5.345e-07  5.646e-05  -0.009    0.993
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07339 on 34 degrees of freedom
## Multiple R-squared:  0.9465, Adjusted R-squared:  0.9386
## F-statistic: 120.2 on 5 and 34 DF,  p-value: < 2.2e-16
```

```r
plot(logfit, which=2) # Making the Normal Q-Q plot
```



The “step pattern” is now definitely mitigated and follows the normal line more closely. i.e. the normality assumption is now better satisfied.

```r
plot(logfit, which=1) # Making the Residuals ~ Fitted Values plot
```



The homoscedasticity assumption is still under question, especially for the tails, but the plot definitely exhibits an improvement in terms of the stability of the variance of the residuals w.r.t. the fitted values.

Exercise 4.6

```
# We begin by importing our data
data <- read.csv("/Users/kristiyan/Documents/MSiA 401 - Predictive 1/Homeworks/salaries.csv",stringsAsF
# Adding the log_10(Salary) variable as response
data <- data %>%
  mutate(log_Salary = log10(data$Salary))

str(data)
```

```
## 'data.frame':    46 obs. of  9 variables:
##  $ Salary    : int  38985 32920 29548 24749 41889 31528 38791 39828 28985 32782 ...
##  $ YrsEm     : int  18 15 5 6 22 3 21 18 0 0 ...
##  $ PriorYr   : int  7 3 6 2 16 11 4 6 1 1 ...
##  $ Education : int  9 9 1 0 7 3 5 5 4 7 ...
##  $ ID        : int  412 458 604 598 351 674 356 415 693 694 ...
##  $ Gender    : chr  "Female" "Male" "Female" "Male" ...
##  $ Dept      : chr  "Sales" "Sales" "Sales" "Sales" ...
##  $ Super     : int  5 4 0 1 7 6 9 5 4 0 ...
##  $ log_Salary: num  4.59 4.52 4.47 4.39 4.62 ...
```

```
data$Gender <- relevel(factor(data$Gender), "Male") # Choosing Male as our reference category for Gende
data$Dept <- relevel(factor(data$Dept), "Purchase") # Choosing Purchase as our reference category for D
lmfit = lm (Salary ~ YrsEm + Education + Dept, data = data) # Using Salary as variable
summary(lmfit)
```

```
##
## Call:
## lm(formula = Salary ~ YrsEm + Education + Dept, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11338.5  -2842.1   -956.2   3126.5   9642.2
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  25384.95    2003.45  12.671 1.38e-15 ***
## YrsEm          736.94     122.20   6.031 4.28e-07 ***
## Education     1790.72     316.01   5.667 1.39e-06 ***
## DeptAdvertse -3148.66    2560.52  -1.230 0.225996
## DeptEngineer    98.96    2027.01   0.049 0.961304
## DeptSales    -8297.02    2300.41  -3.607 0.000851 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4731 on 40 degrees of freedom
## Multiple R-squared:  0.8356, Adjusted R-squared:  0.815
## F-statistic: 40.65 on 5 and 40 DF,  p-value: 1.192e-14
```

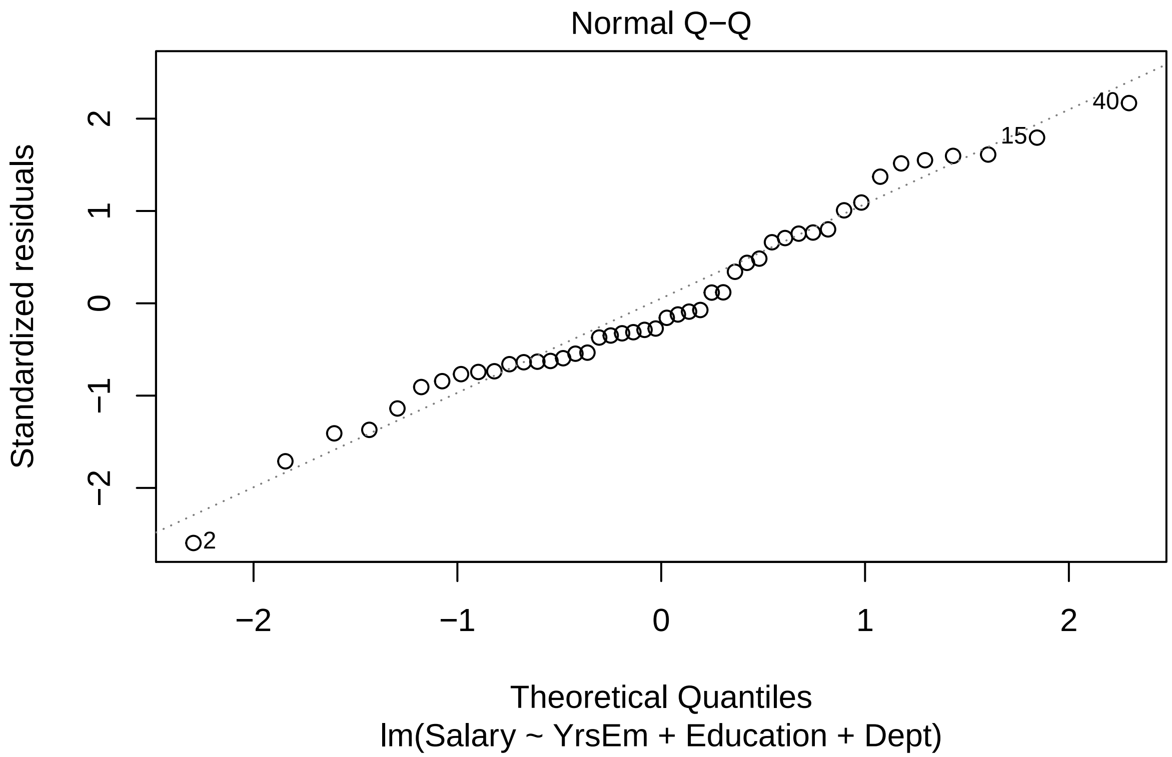
```
logfit = lm (log_Salary ~ YrsEm + Education + Dept, data = data) # Using log(Salary) as response variab
summary(logfit)
```

```
##
## Call:
## lm(formula = log_Salary ~ YrsEm + Education + Dept, data = data)
##
```

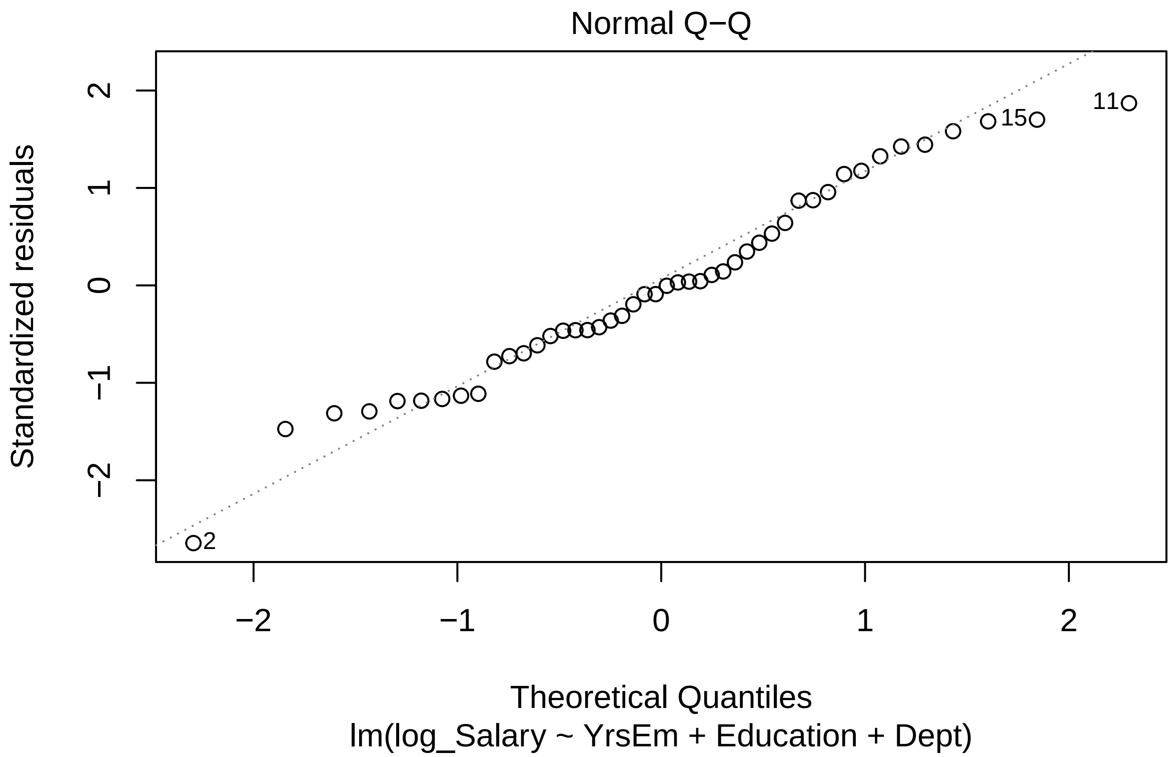
```
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.114193 -0.028068 -0.002002  0.033938  0.081774
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.439005   0.019804 224.142  < 2e-16 ***
## YrsEm         0.007660   0.001208   6.341 1.57e-07 ***
## Education     0.018371   0.003124   5.881 6.95e-07 ***
## DeptAdvertse -0.036488   0.025311  -1.442 0.157208
## DeptEngineer -0.002507   0.020037  -0.125 0.901046
## DeptSales    -0.087593   0.022740  -3.852 0.000414 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04677 on 40 degrees of freedom
## Multiple R-squared:  0.8464, Adjusted R-squared:  0.8272
## F-statistic: 44.09 on 5 and 40 DF,  p-value: 3.099e-15
```

a)

```
plot(lmfit, which=2) # Making the Normal Q-Q plot
```



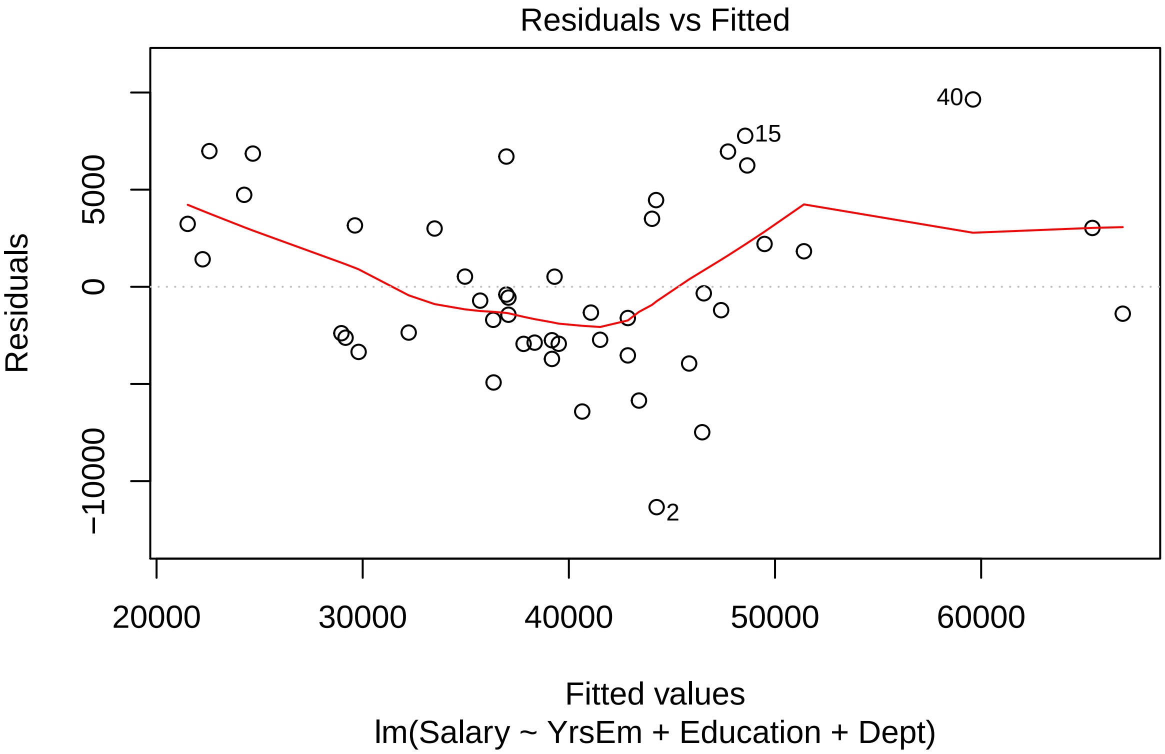
```
plot(logfit, which=2) # Making the Normal Q-Q plot
```



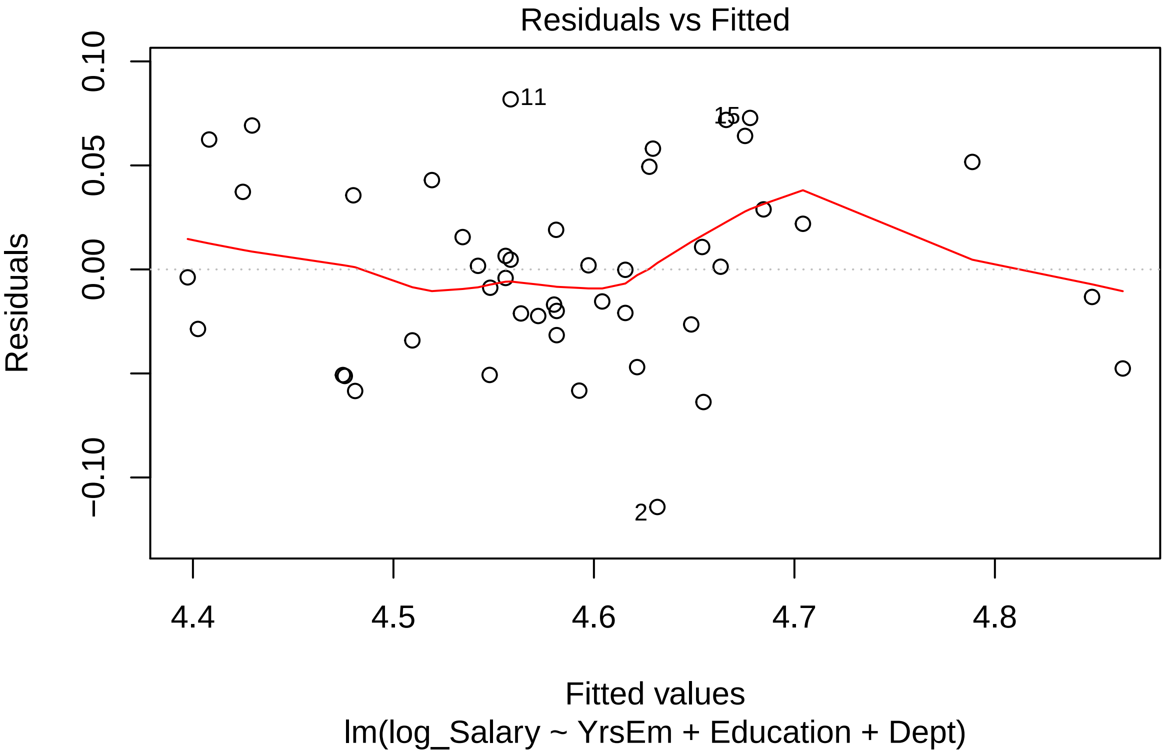
We note a slight improvement in terms of how the theoretical quantiles follow the normal line in the middle of the graph

b)

```r
plot(lmfit, which=1) # Making the Residuals vs. Fitted Values plot
```



```
plot(logfit, which=1) # Making the Residuals vs. Fitted Values plot
```



The residuals definitely appear to be following more of a parallel band. Therefore, our homoscedasticity assumption is better satisfied under a log-transformed response variable.