Lab-5 Transformations

PrabhTalwar

2022-11-11

```
# Loading the data

library(tidyverse)
library(readr)
library(MASS)
library(faraway)

lab_9_data <- read_csv("lab_9_data.csv")
summary(lab_9_data)
```

```
##       ...1            X               Y
##  Min.   : 1.0   Min.   :20.00   Min.   : 3.00
##  1st Qu.: 3.5   1st Qu.:33.00   1st Qu.: 7.50
##  Median : 6.0   Median :42.00   Median :14.00
##  Mean   : 6.0   Mean   :43.91   Mean   :18.18
##  3rd Qu.: 8.5   3rd Qu.:53.00   3rd Qu.:25.50
##  Max.   :11.0   Max.   :70.00   Max.   :45.00
```

(a) Plot the points on a scatterplot. What type of relationship appears to exist between x and y?

```
# Building a model
model <- lm(Y ~ X, data = lab_9_data)
summary(model)
```

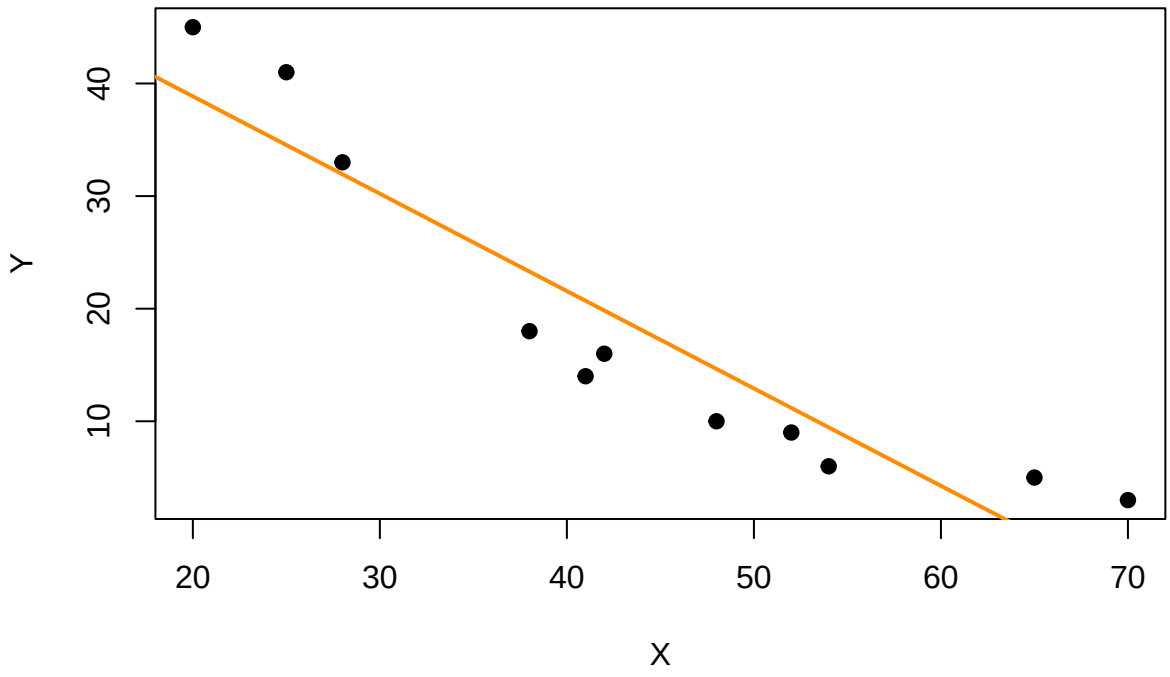
```
##
## Call:
## lm(formula = Y ~ X, data = lab_9_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.698 -4.238 -2.184  5.599  7.383
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  56.1573     5.2031  10.793 1.89e-06 ***
## X            -0.8649     0.1120  -7.723 2.93e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 5.643 on 9 degrees of freedom
## Multiple R-squared:  0.8689, Adjusted R-squared:  0.8543
## F-statistic: 59.65 on 1 and 9 DF,  p-value: 2.929e-05
```

```
# plotting the model

plot(Y ~ X, data = lab_9_data, col = "black",pch = 20, cex = 1.5)

abline(model, col = "darkorange", lwd = 2)
```



The above scatter plot shows a negative relationship between x and y. And adding the fitting line to the plot, we see that the linear relationship does not exist as it is not linear

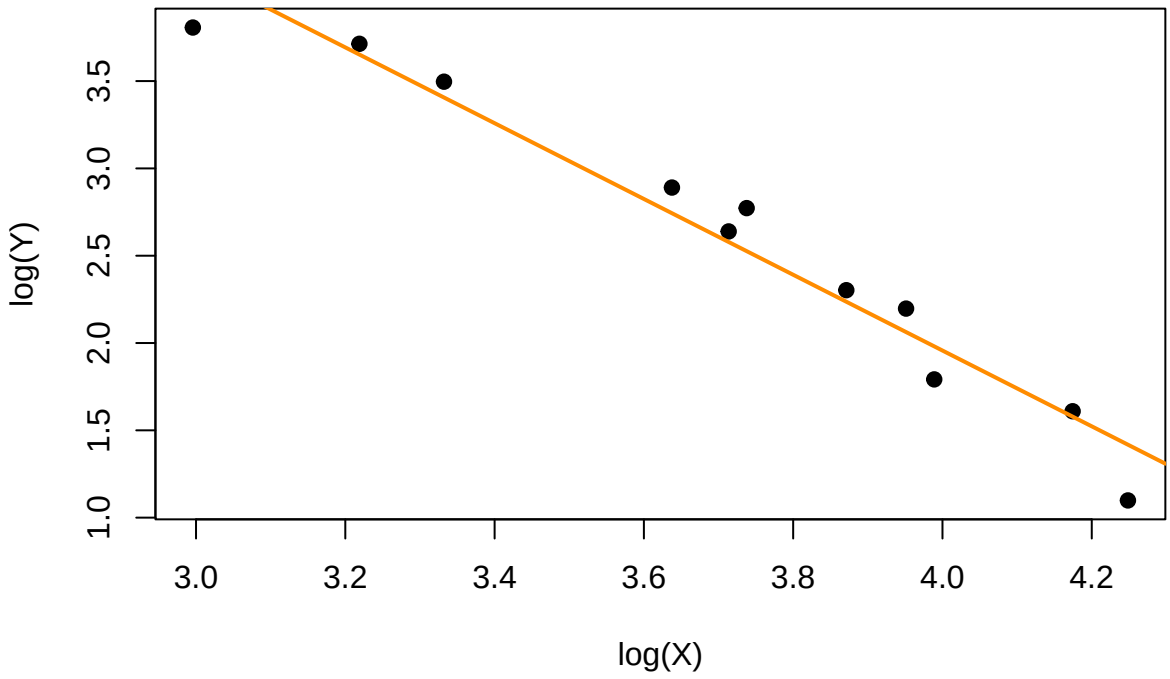
(b) For each observation, calculate ln x and ln y. Plot the log transformed data points on a scatterplot. What type of relationship appears to exist between ln x and ln y?

```
options(scipen = 1000)
model_log <- lm(log(Y) ~ log(X), data = lab_9_data)
summary(model_log)
```

```
##
```

```
## Call:
## lm(formula = log(Y) ~ log(X), data = lab_9_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32942 -0.07912  0.06168  0.11249  0.24640
##
## Coefficients:
##             Estimate Std. Error t value     Pr(>|t|)
## (Intercept)  10.6364     0.6028   17.64 0.0000000273 ***
## log(X)       -2.1699     0.1614  -13.44 0.0000002911 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2021 on 9 degrees of freedom
## Multiple R-squared:  0.9526, Adjusted R-squared:  0.9473
## F-statistic: 180.7 on 1 and 9 DF,  p-value: 0.0000002911
```

```
plot(log(Y) ~ log(X), data = lab_9_data, col = "black", pch = 20, cex = 1.5)
abline(model_log, col = "darkorange", lwd = 2)
```



The above scatter plot shows a negative relationship between ln(x) and ln(y). And adding the fitting line to the plot, we see that the linear relationship exist.

Fit the transformed model to the data. Is the model adequate?

The p value is less than the alpha value 0.05, which means our model is satistically significant.

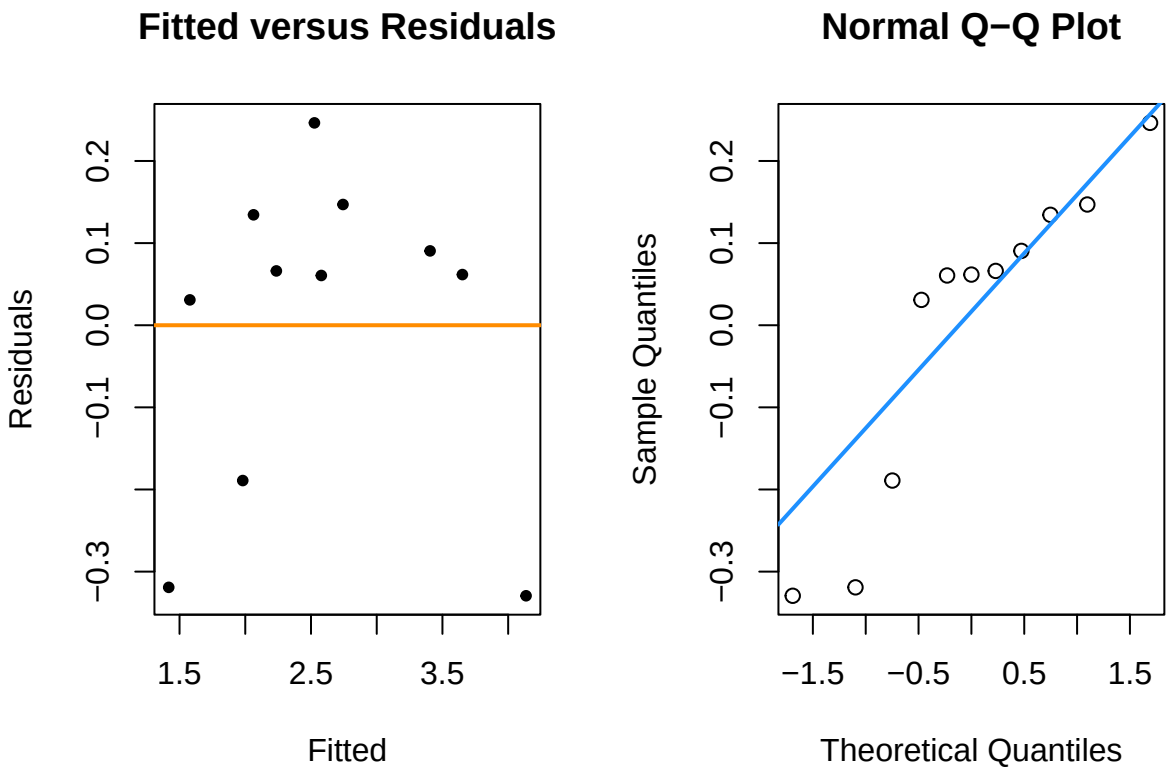
(d) Produce the appropriate residual plot(s) and qq-plot to verify the conditions are satisfied. Are they? Comment. Use the plot model function.

```
par(mfrow = c(1, 2))

plot(fitted(model_log), resid(model_log), col = "black",
pch = 20, xlab = "Fitted", ylab = "Residuals", main = "Fitted versus Residuals")

abline(h = 0, col = "darkorange", lwd = 2)

qqnorm(resid(model_log), main = "Normal Q-Q Plot", col = "black")
qqline(resid(model_log), col = "dodgerblue", lwd = 2)
```



For the residual and fitted plot, at any fitted value, the mean of the residuals should be roughly 0. In this case,the linearity assumption is valid.

At every fitted value, the spread of the residuals should be roughly the same. In this case, the constant variance assumption is voilated.

Here we have a suspect Q-Q plot. We would probably not believe the errors follow a normal distribution.

(e) If the conditions are not satisfied, then you have the wrong model. Try the following models instead. Which is the best model based on performance and the conditions being met? (Use the plot(model) function.)

Model_1

```
# Building a model

model_1 <- lm(Y ~ log(X), data = lab_9_data)
summary(model_1)
```

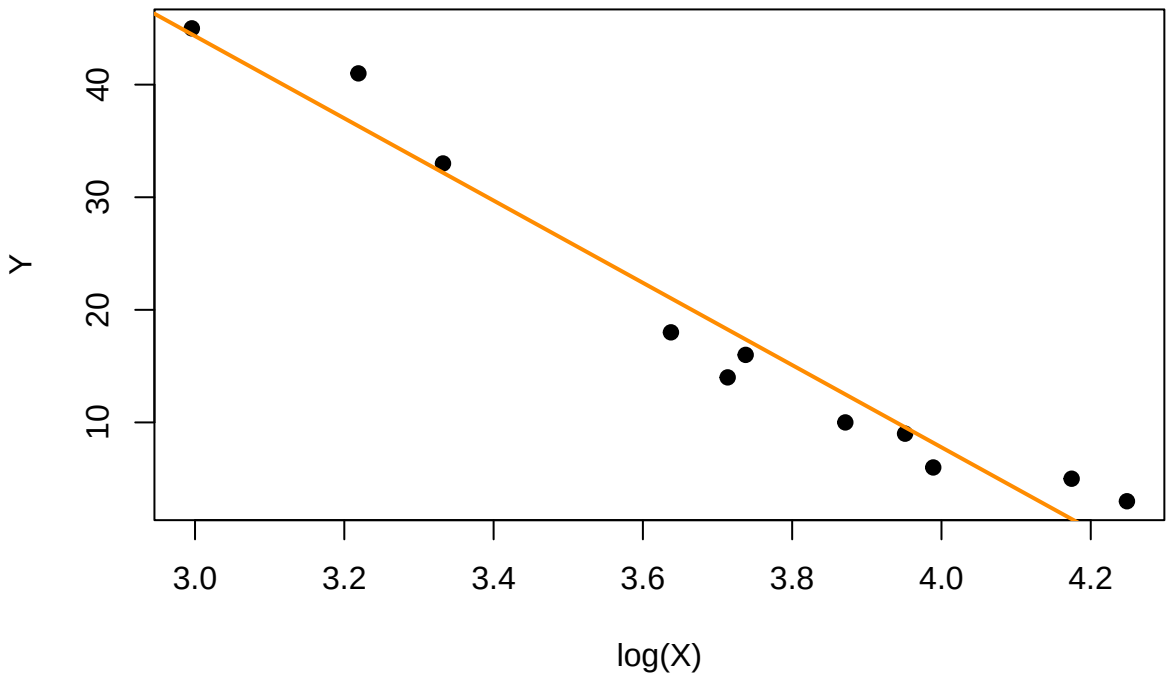
```
##
## Call:
## lm(formula = Y ~ log(X), data = lab_9_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2504 -2.3420 -0.5694  2.2005  4.6807
##
## Coefficients:
##             Estimate Std. Error t value     Pr(>|t|)
## (Intercept)  153.889      9.666   15.92 0.0000000672 ***
## log(X)       -36.525      2.588  -14.11 0.0000001915 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.241 on 9 degrees of freedom
## Multiple R-squared:  0.9568, Adjusted R-squared:  0.952
## F-statistic: 199.1 on 1 and 9 DF,  p-value: 0.0000001915
```

Adjusted R-squared: 0.952

```
# plotting the model

plot(Y ~ log(X), data = lab_9_data, col = "black",pch = 20, cex = 1.5)

abline(model_1, col = "darkorange", lwd = 2)
```

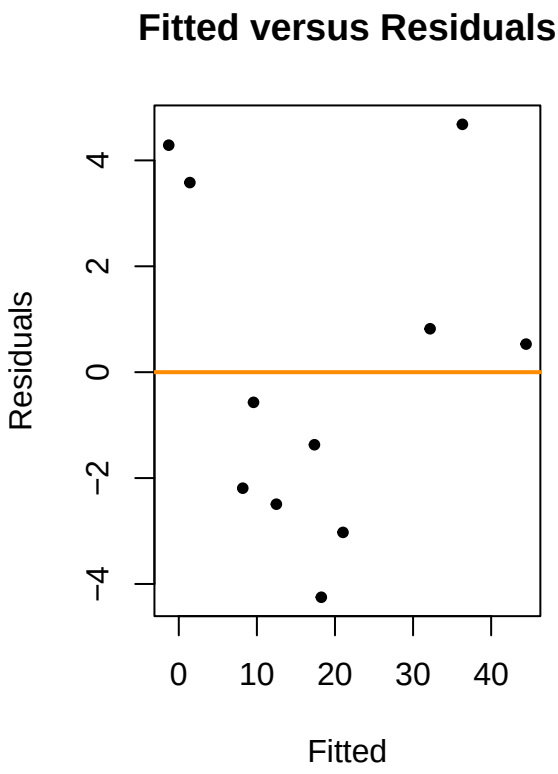


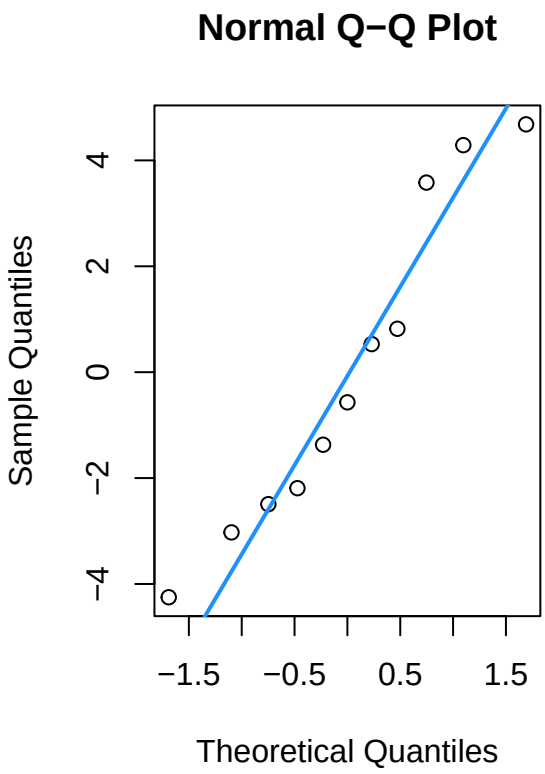

```
par(mfrow = c(1, 2))

plot(fitted(model_1), resid(model_1), col = "black",
pch = 20, xlab = "Fitted", ylab = "Residuals", main = "Fitted versus Residuals")

abline(h = 0, col = "darkorange", lwd = 2)

qqnorm(resid(model_1), main = "Normal Q-Q Plot", col = "black")
qqline(resid(model_1), col = "dodgerblue", lwd = 2)
```





Using the log scale on x variable and plotting the data and adding the fitted line, the variance looks much better but not great.

Model_2

```
# Building a model

model_2 <- lm(Y ~ I(1/X), data = lab_9_data)
summary(model_2)
```

```
##
## Call:
## lm(formula = Y ~ I(1/X), data = lab_9_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1763 -1.4704 -0.0625  0.9599  4.8607
##
## Coefficients:
##             Estimate Std. Error t value     Pr(>|t|)
## (Intercept)  -16.009      2.035  -7.865 0.0000253542 ***
## I(1/X)      1303.696     71.893  18.134 0.0000000215 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

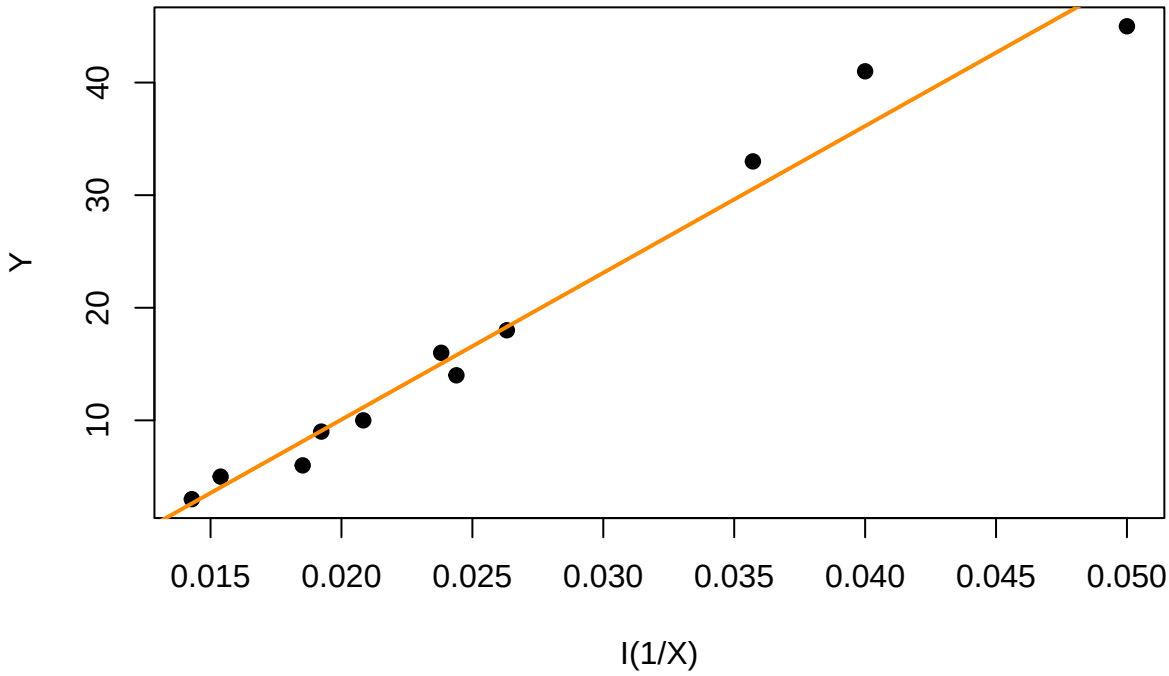
```
##
## Residual standard error: 2.544 on 9 degrees of freedom
## Multiple R-squared:  0.9734, Adjusted R-squared:  0.9704
## F-statistic: 328.8 on 1 and 9 DF,  p-value: 0.0000000215
```

Adjusted R-squared: 0.9704

```r
# plotting the model

plot(Y ~ I(1/X), data = lab_9_data, col = "black",pch = 20, cex = 1.5)

abline(model_2, col = "darkorange", lwd = 2)
```

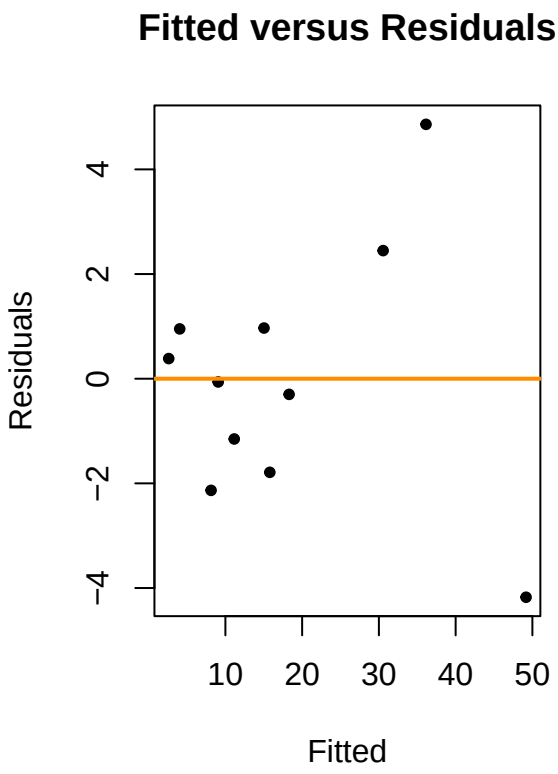


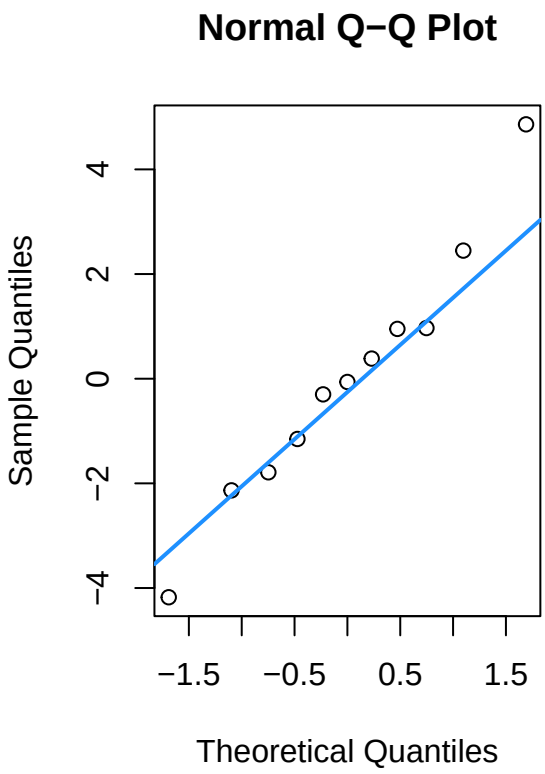

```r
par(mfrow = c(1, 2))

plot(fitted(model_2), resid(model_2), col = "black",
pch = 20, xlab = "Fitted", ylab = "Residuals", main = "Fitted versus Residuals")

abline(h = 0, col = "darkorange", lwd = 2)

qqnorm(resid(model_2), main = "Normal Q-Q Plot", col = "black")
qqline(resid(model_2), col = "dodgerblue", lwd = 2)
```





Model_3

```
# Building a model

model_3 <- lm(log(Y) ~ X, data = lab_9_data)
summary(model_3)
```

```
##
## Call:
## lm(formula = log(Y) ~ X, data = lab_9_data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.230086 -0.062371 -0.007593  0.079759  0.189925
##
## Coefficients:
##             Estimate Std. Error t value         Pr(>|t|)
## (Intercept)  4.97875    0.11290   44.10 0.00000000000792 ***
## X           -0.05476    0.00243  -22.53 0.00000000316249 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1224 on 9 degrees of freedom
## Multiple R-squared:  0.9826, Adjusted R-squared:  0.9807
```

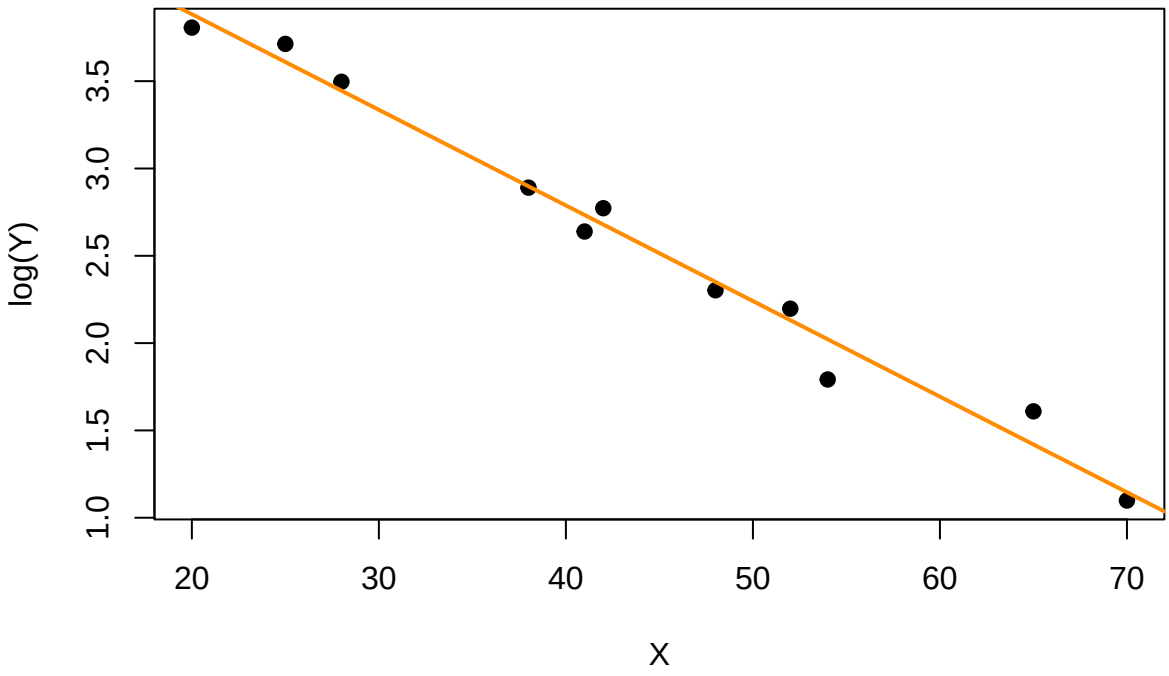
```
## F-statistic: 507.8 on 1 and 9 DF,  p-value: 0.000000003162
```

Adjusted R-squared: 0.9807

```
# plotting the model

plot(log(Y) ~ X, data = lab_9_data, col = "black",pch = 20, cex = 1.5)

abline(model_3, col = "darkorange", lwd = 2)
```

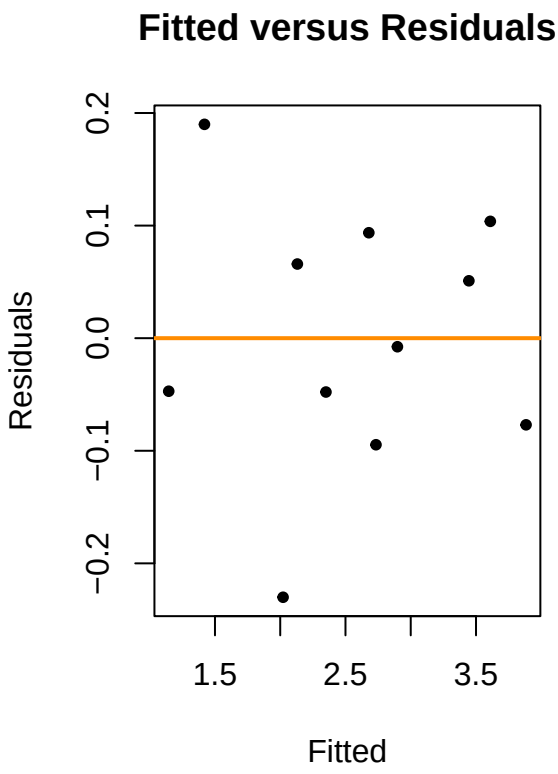


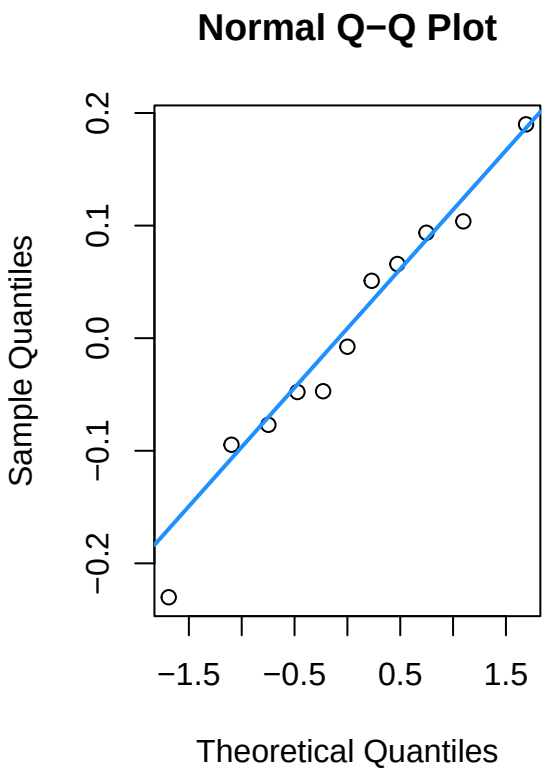

```
par(mfrow = c(1, 2))

plot(fitted(model_3), resid(model_3), col = "black",
pch = 20, xlab = "Fitted", ylab = "Residuals", main = "Fitted versus Residuals")

abline(h = 0, col = "darkorange", lwd = 2)

qqnorm(resid(model_3), main = "Normal Q-Q Plot", col = "black")
qqline(resid(model_3), col = "dodgerblue", lwd = 2)
```





Comparing Performances

```
summary(model_1) # log transformation of x
```

```
##
## Call:
## lm(formula = Y ~ log(X), data = lab_9_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2504 -2.3420 -0.5694  2.2005  4.6807
##
## Coefficients:
##             Estimate Std. Error t value     Pr(>|t|)
## (Intercept)  153.889      9.666   15.92 0.0000000672 ***
## log(X)       -36.525      2.588  -14.11 0.0000001915 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.241 on 9 degrees of freedom
## Multiple R-squared:  0.9568, Adjusted R-squared:  0.952
## F-statistic: 199.1 on 1 and 9 DF,  p-value: 0.0000001915
```

```r
summary(model_2) # 1/x transformation
```

```
##
## Call:
## lm(formula = Y ~ I(1/X), data = lab_9_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1763 -1.4704 -0.0625  0.9599  4.8607
##
## Coefficients:
##             Estimate Std. Error t value     Pr(>|t|)
## (Intercept)  -16.009      2.035  -7.865 0.0000253542 ***
## I(1/X)      1303.696     71.893  18.134 0.0000000215 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.544 on 9 degrees of freedom
## Multiple R-squared:  0.9734, Adjusted R-squared:  0.9704
## F-statistic: 328.8 on 1 and 9 DF,  p-value: 0.0000000215
```

```r
summary(model_3) # log transformation of y
```

```
##
## Call:
## lm(formula = log(Y) ~ X, data = lab_9_data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.230086 -0.062371 -0.007593  0.079759  0.189925
##
## Coefficients:
##             Estimate Std. Error t value         Pr(>|t|)
## (Intercept)  4.97875    0.11290   44.10 0.00000000000792 ***
## X           -0.05476    0.00243  -22.53 0.00000000316249 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1224 on 9 degrees of freedom
## Multiple R-squared:  0.9826, Adjusted R-squared:  0.9807
## F-statistic: 507.8 on 1 and 9 DF,  p-value: 0.000000003162
```

```r
# Model 1
sqrt(mean((lab_9_data$Y - fitted(model_1)) ^ 2))
```

```
## [1] 2.931172
```

```r
# Model 2
sqrt(mean((lab_9_data$Y - fitted(model_2)) ^ 2))
```

```
## [1] 2.300692
```

```r
# Model 3
sqrt(mean((lab_9_data$Y - exp(fitted(model_3))) ^ 2))
```

```
## [1] 1.900063
```

Looks the log transformation of Y is the better fit!

(f) Use the transformed model in part (e) to predict the value of y.

```r
# Model 3

Y = 4.97875 - 0.05476 * (lab_9_data$X)

# Predicting the model where X = 30

Y = 4.97875 - 0.05476 * (30)
Y
```

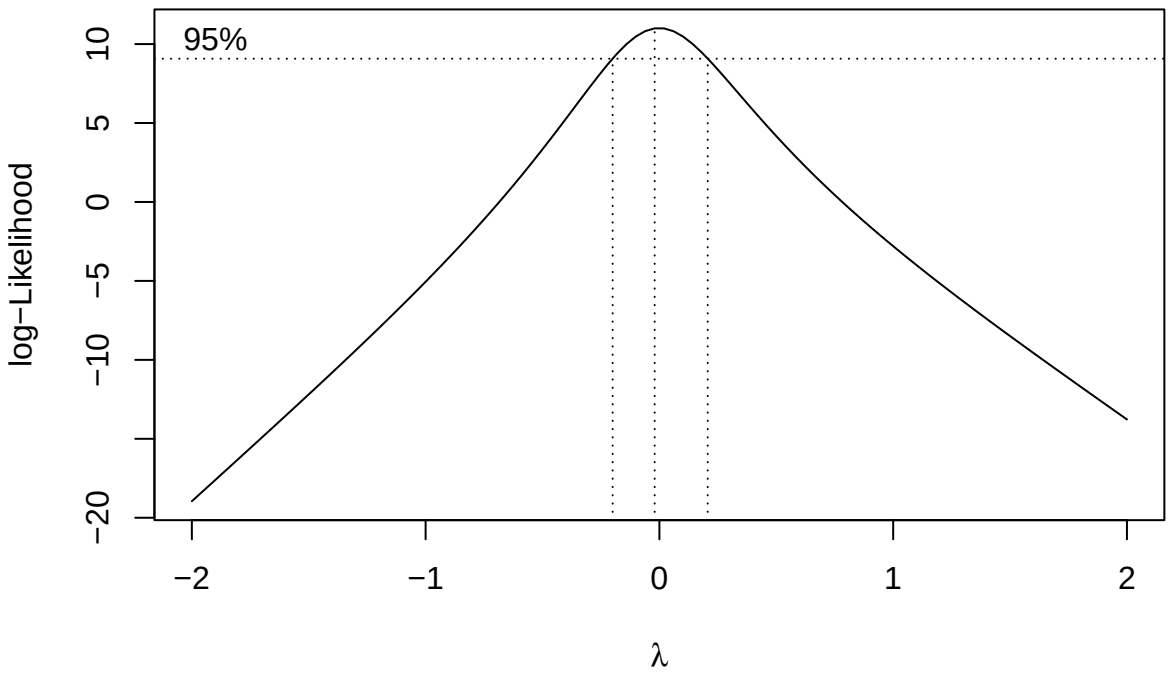
```
## [1] 3.33595
```

(g) Use the boxcox function (if applicable) to verify your choice of transform in part (e). If not applicable, just say "not applicable".

```r
model <- lm(Y ~ X, data = lab_9_data)
model
```

```
##
## Call:
## lm(formula = Y ~ X, data = lab_9_data)
##
## Coefficients:
## (Intercept)            X
##     56.1573      -0.8649
```

```r
boxcox(model, plotit = TRUE)
```



The above function to find the best transformation of the form considered by the Box-Cox method. Here we see that lambda = 0. The verification of choice for transformation is correct.

Transforming back

```
# Model 3

Y = 4.97875 - 0.05476 * (lab_9_data$X)

# Predicting the model where X = 30

Y = 4.97875 - 0.05476 * (30)

exp(Y)
```

```
## [1] 28.10507
```