Lab 5 Solutions

Name: Kaushik Raj V Nadar

Roll No.: 208160499

Date: 03/09/2024

P 1. Consider the **Electricity** data set.

(a) Observe the residual plot.

(b) Apply Box–Cox transformation on the response variable.

(c) Fit the transformed model and then observe the residual plot.

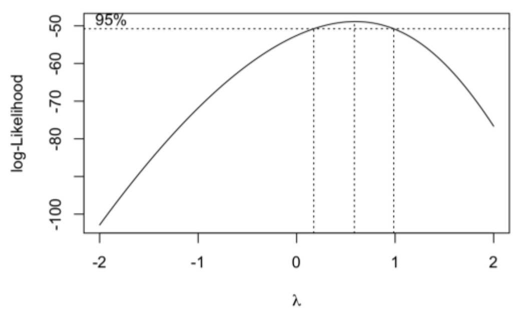
Code:

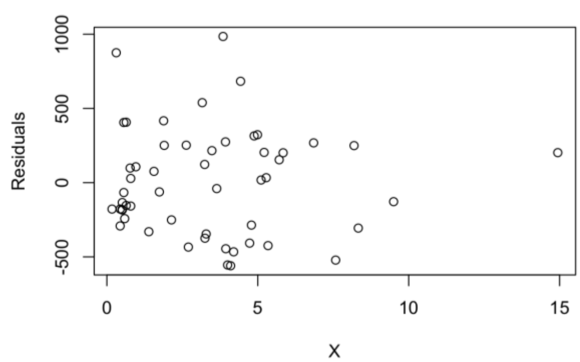
```
library(readxl)
library(tidyr)
library(MASS)
library(ggplot2)

data = read_excel('Electricity_Data.xlsx')
model = lm(Y~X, data)

g1 = ggplot()+
  geom_point(aes(model$fitted.values, resid(model) ))+
  labs(title = 'Residual Plot',
      x = 'Yhat',
      y = 'e')

#find optimal lambda for Box-Cox transformation
box_cox <- boxcox(Y ~ X, data = data)
(lambda <- box_cox$x[which.max(box_cox$y)])

#transformed model
box_cox_transform <- function(Y, lambda) { if (lambda == 0) {
  return(log(Y)) } else {
    return((Y^lambda - 1) / lambda)
  } }

Y1 <- box_cox_transform(data$Y, lambda)
model_bc = lm(Y1~data$X)

g2 = ggplot()+
  geom_point(aes(model_bc$fitted.values, model_bc %>% resid() ))+
  labs(title = 'Residual Plot after Box-Cox transformation',
      x = 'Yhat',
      y = 'e')
```

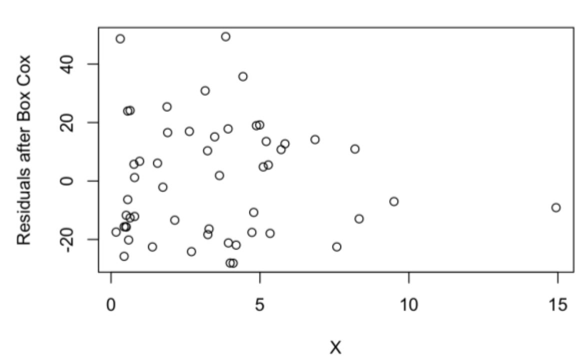
```
plot(data$X, model %>% resid(), xlab = "X", ylab="Residuals")
plot(data$X,model_bc %>% resid(), xlab="X", ylab="Residuals after Box Cox")
plot(g1)
plot(g2)
plot(data$X, data$Y)
abline(model, col="red")
plot(data$X, Y1, xlab="X", ylab="transformed Y")
abline(model_bc, col="red")
```

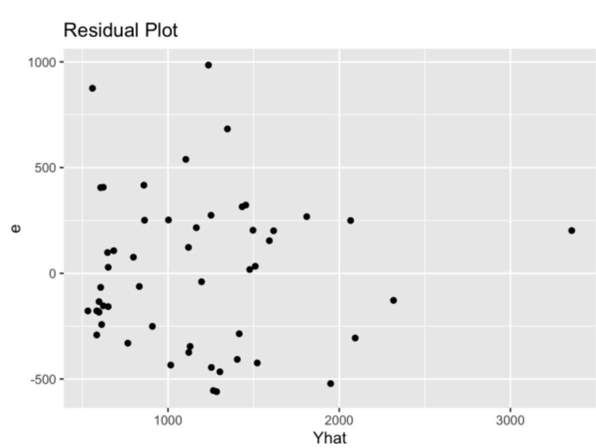
Output:

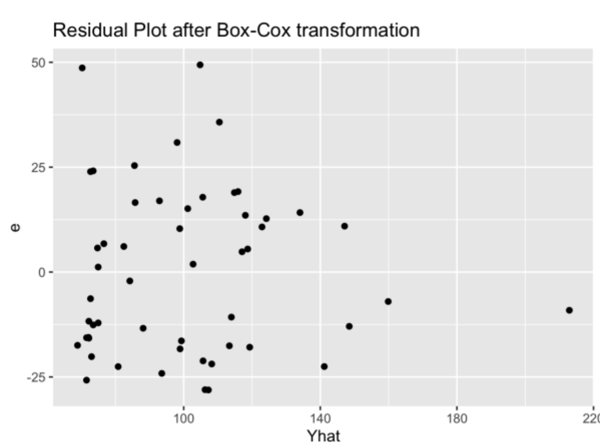
```
> box_cox <- boxcox(Y ~ X, data = data)
> (lambda <- box_cox$x[which.max(box_cox$y)])
[1] 0.5858586
```

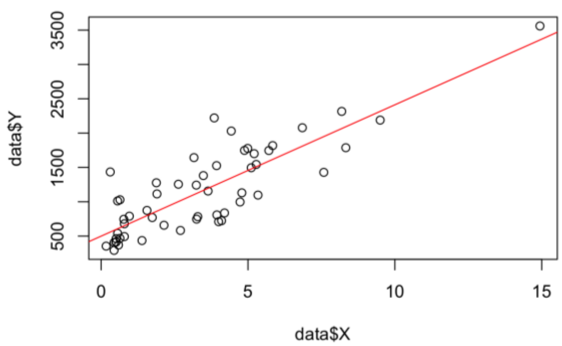


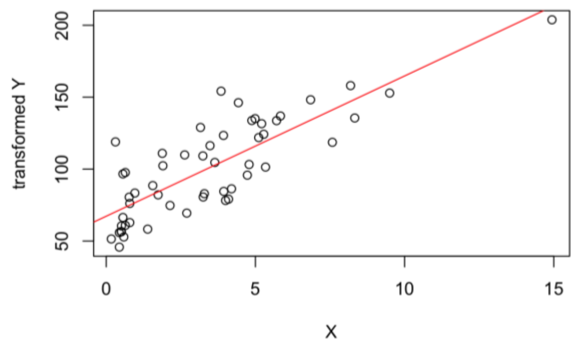












P 2. Consider the **Windmill** data set.

(a) Observe the residual plot.

(b) Apply Box–Tidewell transformation on the regressor variable.

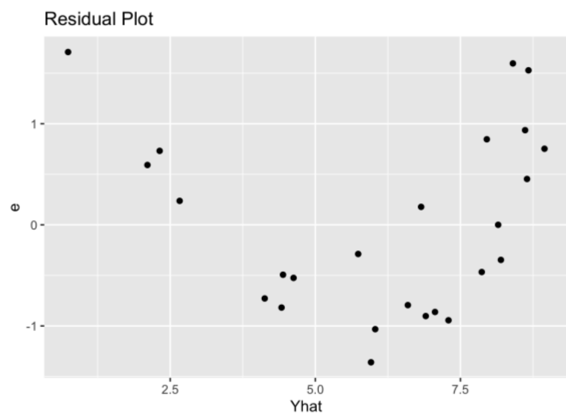
(c) Fit the transformed model and then observe the residual plot.

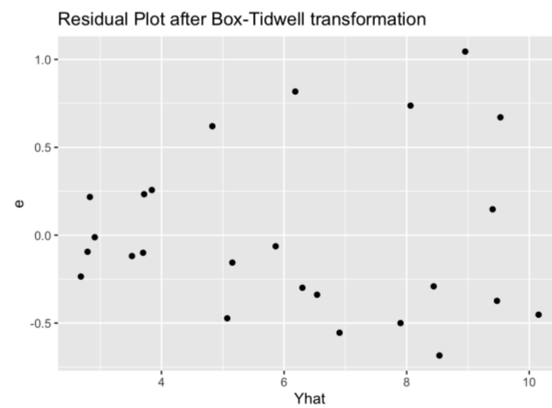
Code:

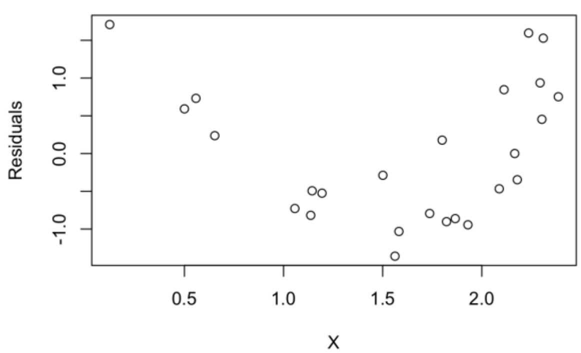
```
library(readxl)
library(ggplot2)
library(car)

data = read_excel('Wind_Mill_Data.xlsx')
model = lm(Y~X, data)

g3 = ggplot()+
  geom_point(aes(model$fitted.values, model %>% resid() ))+
  labs(title = 'Residual Plot',
       x = 'Yhat',
       y = 'e')

#optimal value of alpha for box-tidwell transformation
bt = boxTidwell(Y~X, data = data)

box_tidwell_tranform <- function(X, lambda) {
  if (lambda == 0) {
  return(log(X)) } else {
    return(X^lambda) }
}

X1 = box_tidwell_tranform(data$X, bt$result[1])

model_bt = lm(data$Y~X1)

g4 = ggplot()+
  geom_point(aes(model_bt$fitted.values, model_bt %>% resid() ))+
  labs(title = 'Residual Plot after Box-Tidwell transformation',
```

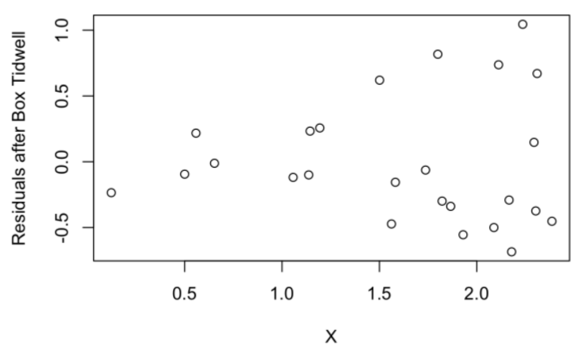
```
    x = 'Yhat',
    y = 'e')

plot(g3)
plot(g4)

plot(data$X, data$Y)
abline(model, col="red")

plot(X1, data$Y, xlab="Tranformed X", ylab = "Y")
abline(model_bt, col="red")
```

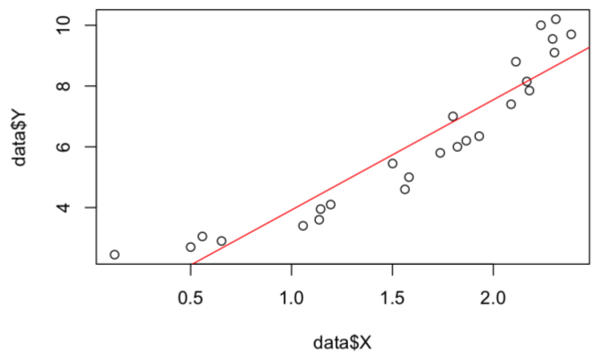
Output:

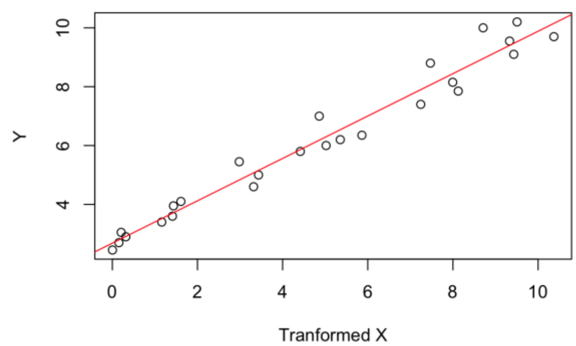












P 3. Consider the **Weighted_Least_Squares** data set.

(a) Observe the residual plot.

(b) Divide the data set in appropriate groups by observing the regressor values.

(c) Compute the sample mean and variance of each groups. Observe their scatter plot.

(d) Fit an appropriate regression regression model (may be a quadratic term required).

(e) Based in the above fitted model, compute the fitted values by inserting the original regressor values.

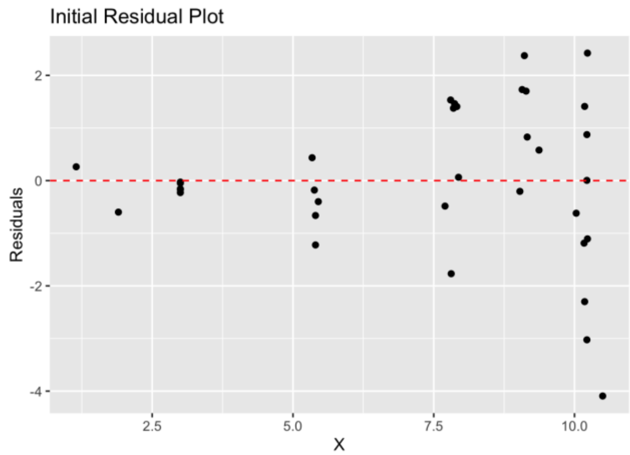
(d) Use theses fitted values as reciprocal of the weights and fit a weighted least squares model. Observe the residual plot. If there as negative weights then take the absolute.

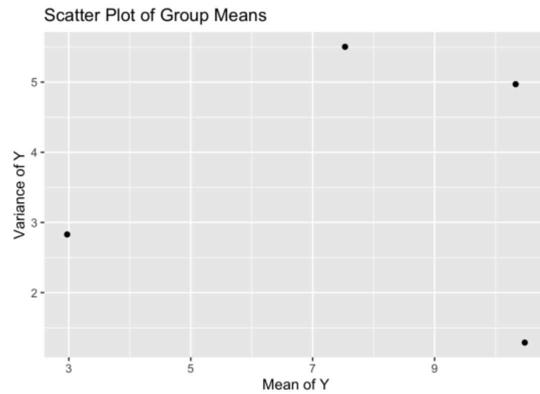
Code:

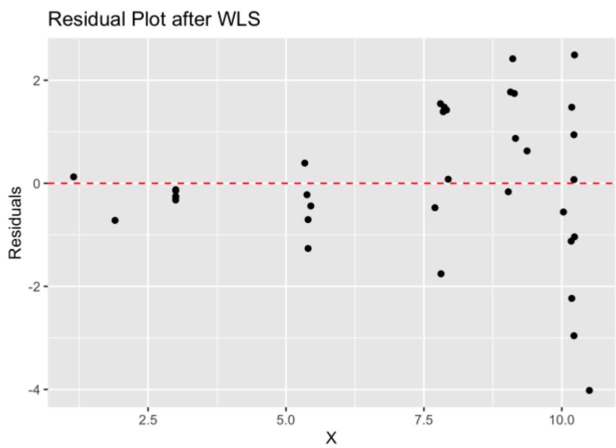
```
# Load necessary libraries
library(readxl)
library(dplyr)
library(ggplot2)

# Step 1: Load the data
data <- read_excel("Weighted_Least_Squares_Data.xlsx")
x_og <- data$X
y_og <- data$Y

# Step 2: Plot the residuals from an initial linear model
initial_model <- lm(Y ~ X, data = data)
initial_residuals <- residuals(initial_model)
summary(initial_model)

# Plot the residuals
ggplot(data, aes(x = X, y = initial_residuals)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Initial Residual Plot", x = "X", y = "Residuals")

# Step 3: Divide data into groups based on regressor values
# Here, we'll assume splitting the data into quartiles for demonstration.
```

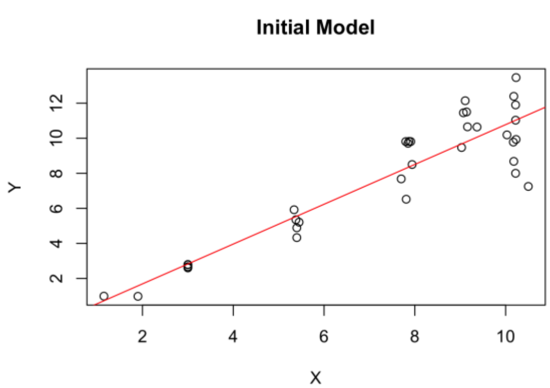
```r
data <- data %>%
  mutate(group = ntile(X, 4))

# Step 4: Compute the sample mean and variance of each group
group_stats <- data %>%
  group_by(group) %>%
  summarize(mean_X = mean(X), mean_Y = mean(Y), var_Y = var(Y), var_X=var(X))

# Step 5: Scatter plot of the group means and variances
ggplot(group_stats, aes(x = mean_Y, y = var_Y)) +
  geom_point() +
  labs(title = "Scatter Plot of Group Means", x = "Mean of Y", y = "Variance of Y")

# Step 6: Fit an appropriate regression model (including a quadratic term)
quad_model <- lm(Y ~ poly(X, 2), data = data)
summary(quad_model)

# Step 7: Compute the fitted values
fitted_values <- fitted(quad_model)

# Step 8: Use these fitted values as reciprocal of the weights
weights <- 1 / abs(fitted_values)

# Step 9: Fit a weighted least squares model
wls_model <- lm(Y ~ X, data = data, weights = weights)
summary(wls_model)

# Step 10: Observe the residual plot of the WLS model
wls_residuals <- residuals(wls_model)

ggplot(data, aes(x = X, y = wls_residuals)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Residual Plot after WLS", x = "X", y = "Residuals")

plot(x_og, y_og, xlab="X", ylab="Y", main="Initial Model")
abline(initial_model, col="red")
plot(x_og, y_og, xlab="X", ylab="Y", main="WLS Model")
abline(wls_model, col="red")
```

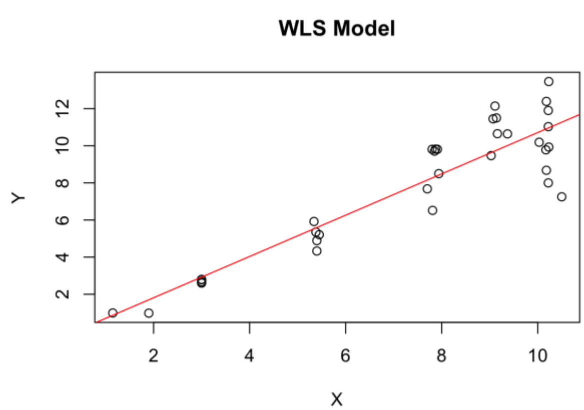
Output:












```
> summary(model_initial)
Call:
lm(formula = Y ~ X, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-4.0928 -0.6087 -0.0473  1.1256  2.4238

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.57895    0.67919  -0.852      0.4
```

```
X           1.13540    0.08622  13.169 1.09e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.457 on 33 degrees of freedom
Multiple R-squared:  0.8401,           Adjusted R-squared:  0.8353
F-statistic: 173.4 on 1 and 33 DF,  p-value: 1.089e-14

> summary(quad_model)
Call:
lm(formula = Y ~ poly(X, 2), data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-3.7010 -0.8106 -0.0589  1.1044  2.7108

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   7.7569     0.2438  31.823  < 2e-16 ***
poly(X, 2)1  19.1820     1.4421  13.302 1.39e-14 ***
poly(X, 2)2  -1.8628     1.4421  -1.292    0.206
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.442 on 32 degrees of freedom
Multiple R-squared:  0.8481,           Adjusted R-squared:  0.8386
F-statistic:  89.3 on 2 and 32 DF,  p-value: 8.07e-14

> summary(wls_model)
Call:
lm(formula = Y ~ X, data = data, weights = weights)

Weighted Residuals:
    Min      1Q  Median      3Q     Max
-1.2145 -0.2421 -0.0746  0.3968  0.7709

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.41505    0.18977  -2.187   0.0359 *
X            1.11276    0.04378  25.419   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.483 on 33 degrees of freedom
Multiple R-squared:  0.9514,           Adjusted R-squared:  0.9499
F-statistic: 646.1 on 1 and 33 DF,  p-value: < 2.2e-16
```