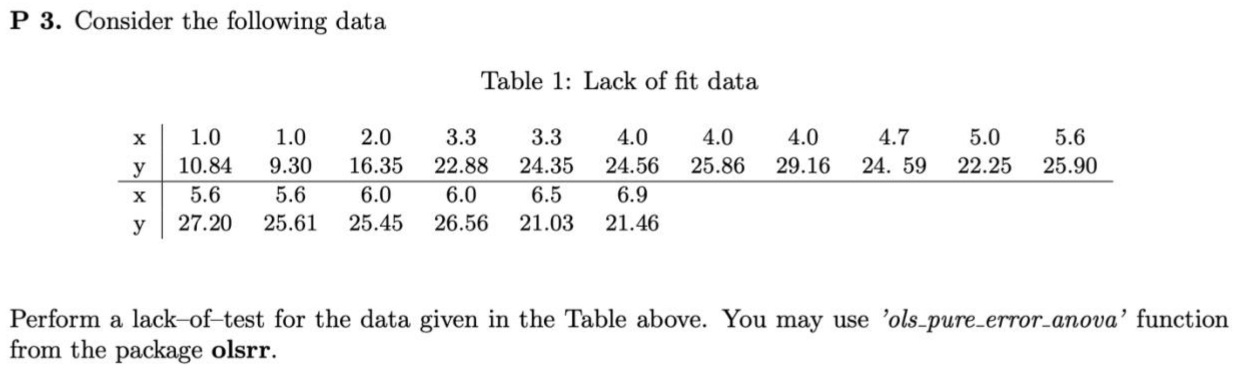
Lab Solutions

Name: Kaushik Raj V Nadar

Roll No.: 208160499

Date: 27/08/2024



P 3. Consider the following data

Table 1: Lack of fit data

x	1.0	1.0	2.0	3.3	3.3	4.0	4.0	4.0	4.7	5.0	5.6
y	10.84	9.30	16.35	22.88	24.35	24.56	25.86	29.16	24. 59	22.25	25.90
x	5.6	5.6	6.0	6.0	6.5	6.9					
y	27.20	25.61	25.45	26.56	21.03	21.46					

Perform a lack–of–test for the data given in the Table above. You may use *'ols_pure_error_anova'* function from the package **olsrr**.

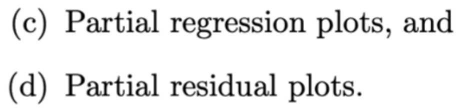
Code:

```
library(olsrr)
# Input the data
x <- c(1.0, 1.0, 2.0, 3.3, 3.3, 4.0, 4.0, 4.0, 5.6, 5.6, 6.0,
     6.0, 6.5, 6.9)
y <- c(10.84, 9.30, 16.35, 22.88, 24.35, 24.56, 25.86, 29.16,
     27.20, 25.61, 25.45, 26.56, 21.03, 21.46)
# Fit the linear model
model <- lm(y ~ x)
# Perform the lack-of-fit test
lack_of_fit_test <- ols_pure_error_anova(model)
# Print the results
print(lack_of_fit_test)
```

Output:

```
Lack of Fit F Test
--------------
Response :   y
Predictor:   x

                    Analysis of Variance Table
----------------------------------------------------------------------
              DF     Sum Sq     Mean Sq     F Value        Pr(>F)
----------------------------------------------------------------------
x              1     225.2168     225.2168     87.78663     7.204966e-07
Residual       12     245.0711     20.42259
 Lack of fit     6     229.6781     38.27968     14.92093      0.002250367
 Pure Error      6     15.39302     2.565503
----------------------------------------------------------------------
```

P 2. Consider the **Time Delivery** data. Analyse



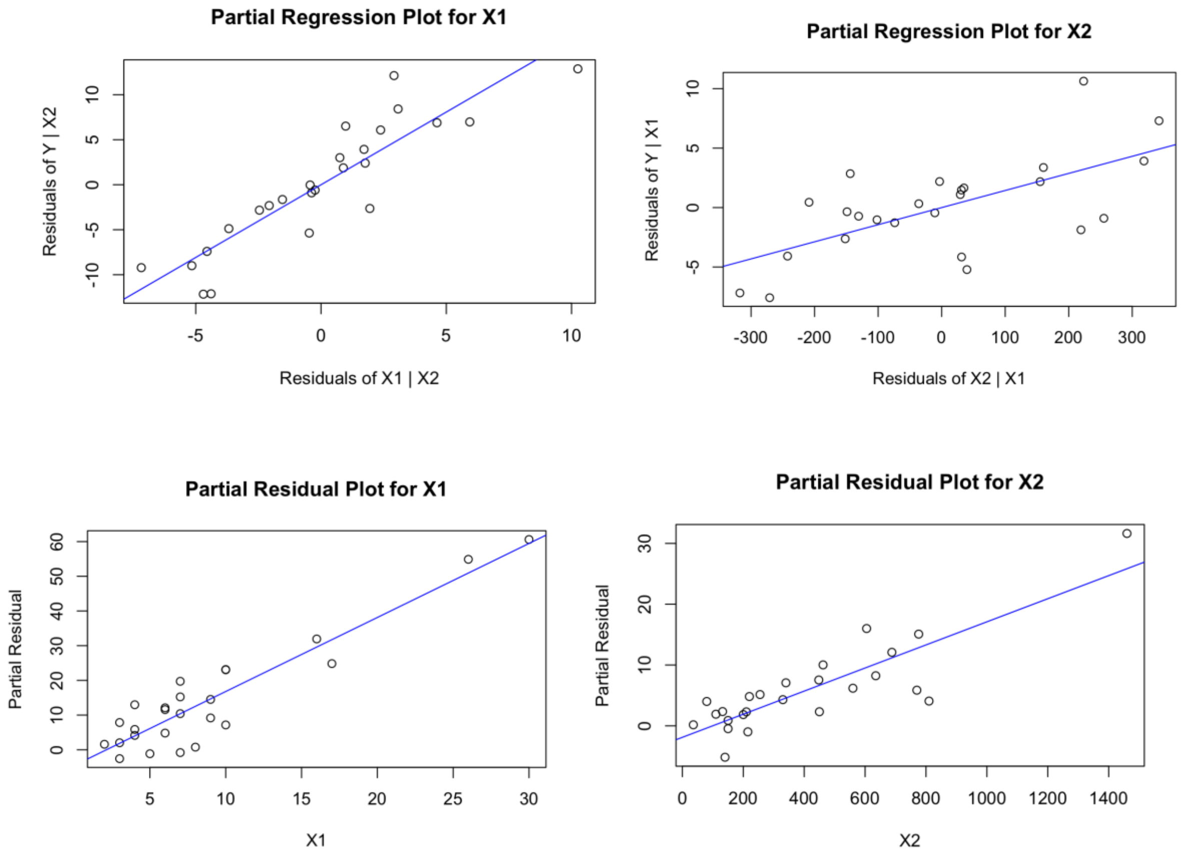
(c) Partial regression plots, and

(d) Partial residual plots.

Code:

```
# Plot Fitted Values vs. R-Student
# Load necessary library for reading Excel files
library(readxl)

# Load the data
data <- read_excel("TimeDeliveryData.xlsx")

# Convert the data to a data frame
df <- as.data.frame(data)

# Fit the linear model
model <- lm(Y ~ X1 + X2, data = df)

# Partial Regression for X1
# Regress Y on X2 and get residuals
residuals_Y_X2 <- residuals(lm(Y ~ X2, data = df))

# Regress X1 on X2 and get residuals
residuals_X1_X2 <- residuals(lm(X1 ~ X2, data = df))

# Plot residuals
plot(residuals_X1_X2, residuals_Y_X2,
   xlab = "Residuals of X1 | X2",
   ylab = "Residuals of Y | X2",
   main = "Partial Regression Plot for X1")
abline(lm(residuals_Y_X2 ~ residuals_X1_X2), col = "blue")

# Partial Regression for X2
# Regress Y on X1 and get residuals
residuals_Y_X1 <- residuals(lm(Y ~ X1, data = df))

# Regress X2 on X1 and get residuals
residuals_X2_X1 <- residuals(lm(X2 ~ X1, data = df))

# Plot residuals
plot(residuals_X2_X1, residuals_Y_X1,
   xlab = "Residuals of X2 | X1",
    ylab = "Residuals of Y | X1",
    main = "Partial Regression Plot for X2")
abline(lm(residuals_Y_X1 ~ residuals_X2_X1), col = "blue")

# Partial Residual Plot for X1
partial_residual_X1 <- residuals_Y_X2 + coef(model)["X1"] * df$X1
plot(df$X1, partial_residual_X1,
    xlab = "X1",
    ylab = "Partial Residual",
    main = "Partial Residual Plot for X1")
abline(lm(partial_residual_X1 ~ df$X1), col = "blue")

# Partial Residual Plot for X2
partial_residual_X2 <- residuals_Y_X1 + coef(model)["X2"] * df$X2
plot(df$X2, partial_residual_X2,
    xlab = "X2",
    ylab = "Partial Residual",
    main = "Partial Residual Plot for X2")
abline(lm(partial_residual_X2 ~ df$X2), col = "blue")
```

Output:



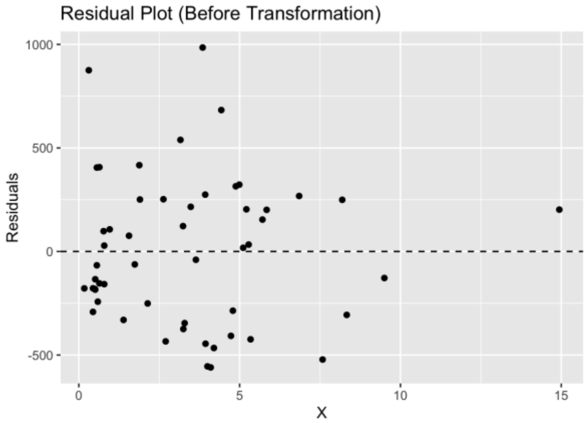
P 3. For the **Electricity Data** and **Windmill Data**, perform Box–Cox and Box–Tidwell transformations. Analyse the residual plots before and after the transformations.

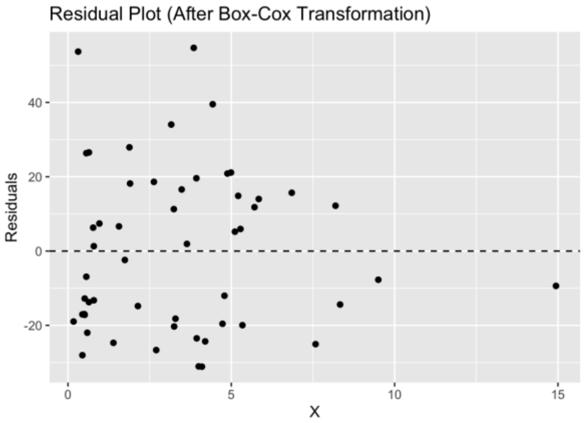
Code:

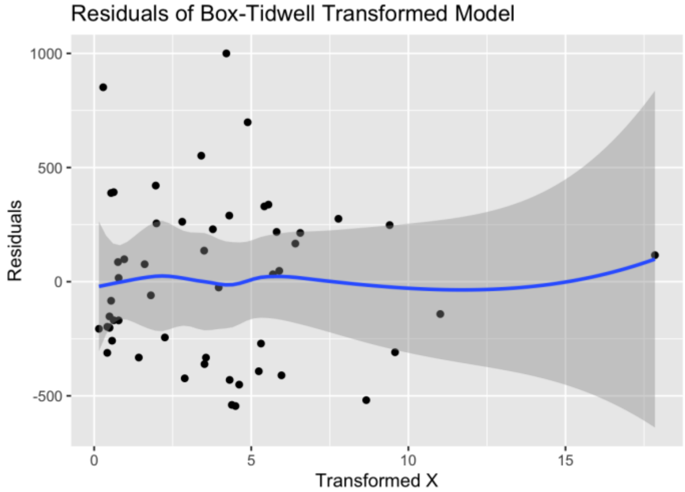
```
# Load necessary libraries
library(MASS)      # For Box-Cox transformation
library(car)       # For Box-Tidwell transformation
library(ggplot2)   # For plotting
library(dplyr)     # For data manipulation
library(readxl)

# Load the data
data <- read_excel("Electricity_Data.xlsx")
# Use "Wind_Mill_Data.xlsx" for Windmill Data

X <- data$X
Y <- data$Y

# Step 1: Initial Linear Model (before transformations)
model_initial <- lm(Y ~ X, data = data)
residuals_initial <- resid(model_initial)

# Plot initial residuals
ggplot(data, aes(X, residuals_initial)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(title = "Residual Plot (Before Transformation)", x = "X", y = "Residuals")

# Step 2: Box-Cox Transformation
boxcox_result <- boxcox(model_initial, plotit = FALSE)
lambda <- boxcox_result$x[which.max(boxcox_result$y)]

# Apply Box-Cox transformation
Y_boxcox <- (Y^lambda - 1) / lambda
data$Y_boxcox <- Y_boxcox

# Model after Box-Cox transformation
model_boxcox <- lm(Y_boxcox ~ X, data = data)
residuals_boxcox <- resid(model_boxcox)

# Plot residuals after Box-Cox transformation
ggplot(data, aes(X, residuals_boxcox)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(title = "Residual Plot (After Box-Cox Transformation)", x = "X", y = "Residuals")

# Box-Tidwell transformation
box_tidwell <- boxTidwell(Y ~ X, data = data)
transformed_X <- data$X ^ box_tidwell$result[1]
```

```
# Fit a linear model with the transformed X
model_boxtidwell <- lm(Y ~ transformed_X, data = data)

# Plot residuals of the Box-Tidwell transformed model
ggplot(data, aes(x = transformed_X, y = residuals(model_boxtidwell))) +
  geom_point() +
  geom_smooth(method = "loess") +
  labs(title = "Residuals of Box-Tidwell Transformed Model",
       x = "Transformed X",
       y = "Residuals")

# Step 4: Compare Models
summary(model_initial)
summary(model_boxcox)
summary(model_boxtidwell)
```

Output For Electricity Data:



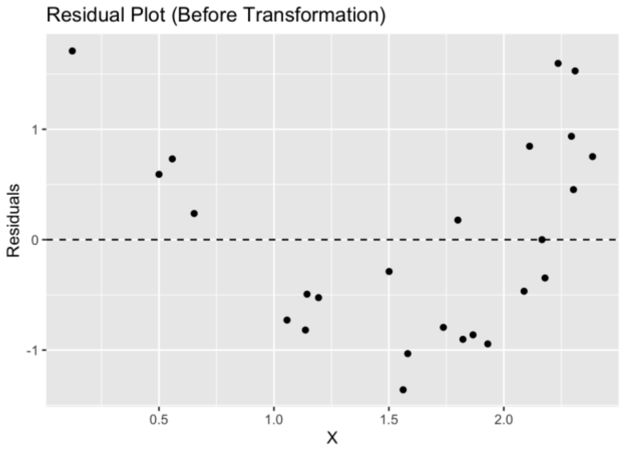


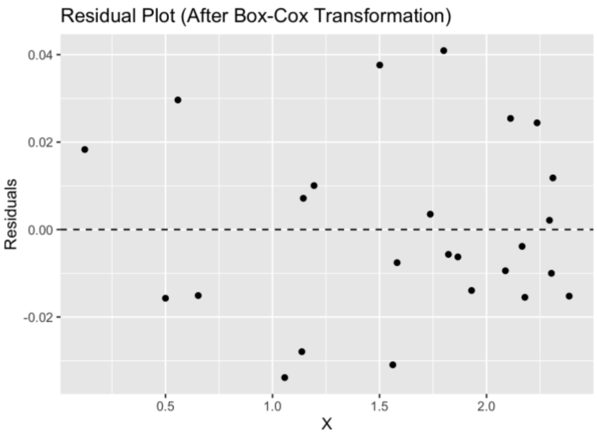


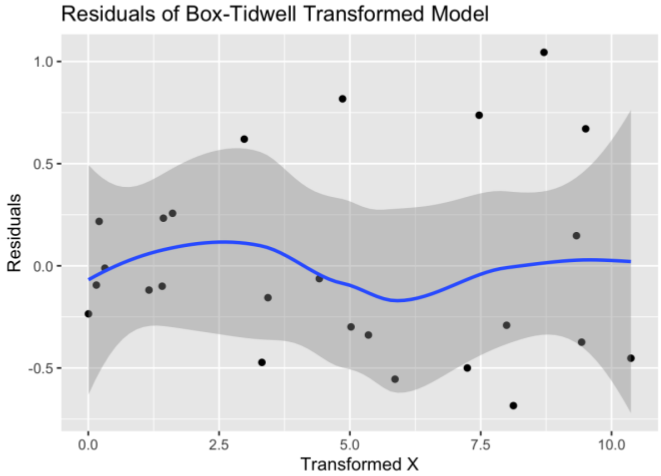
```
> summary(model_initial)
Call:
lm(formula = Y ~ X, data = data)


Residuals:
   Min     1Q Median     3Q    Max 
-559.9 -285.9  -39.9  249.6  984.9 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   499.48      77.09   6.479 3.62e-08 ***
X             191.32      17.35  11.030 4.11e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 359.5 on 51 degrees of freedom
Multiple R-squared:  0.7046,	Adjusted R-squared:  0.6988 
F-statistic: 121.7 on 1 and 51 DF,  p-value: 4.106e-15

> summary(model_boxcox)
Call:
lm(formula = Y_boxcox ~ X, data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-31.119 -18.214  -2.404  15.681  54.680 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   71.868      4.712   15.25  < 2e-16 ***
X             10.785      1.060   10.17 7.18e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 21.97 on 51 degrees of freedom
Multiple R-squared:  0.6698,	Adjusted R-squared:  0.6634 
F-statistic: 103.5 on 1 and 51 DF,  p-value: 7.183e-14

> summary(model_boxtidwell)
Call:
lm(formula = Y ~ transformed_X, data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-544.77 -270.94  -25.46  247.84  999.77 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)     535.74      74.48   7.193 2.70e-09 ***
transformed_X   162.95      14.75  11.050 3.84e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 359 on 51 degrees of freedom
```

```
Multiple R-squared:  0.7054,          Adjusted R-squared:  0.6996
F-statistic: 122.1 on 1 and 51 DF,  p-value: 3.842e-15
```

Output for WindMill Data:


Residual Plot (Before Transformation)


Residual Plot (After Box-Cox Transformation)


Residuals of Box-Tidwell Transformed Model

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

Residuals:
    Min      1Q  Median      3Q     Max
-1.3594 -0.7940 -0.2882  0.7315  1.7089

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.2950     0.4961   0.595    0.558
X             3.6264     0.2865  12.659 7.55e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9154 on 23 degrees of freedom
Multiple R-squared:  0.8745,          Adjusted R-squared:  0.869
```

```
F-statistic: 160.3 on 1 and 23 DF,  p-value: 7.546e-12

> summary(model_boxcox)
Call:
lm(formula = Y_boxcox ~ X, data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-0.03386 -0.01509 -0.00568  0.01181  0.04092

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.619871   0.011419   54.29   <2e-16 ***
X           0.223280   0.006593   33.86   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.02107 on 23 degrees of freedom
Multiple R-squared:  0.9803,	Adjusted R-squared:  0.9795
F-statistic:  1147 on 1 and 23 DF,  p-value: < 2.2e-16

> summary(model_boxtidwell)
Call:
lm(formula = Y ~ transformed_X, data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-0.68440 -0.33871 -0.09991  0.23315  1.04472

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    2.68260    0.16688   16.07 5.32e-14 ***
transformed_X  0.72024    0.02847   25.30  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4812 on 23 degrees of freedom
Multiple R-squared:  0.9653,	Adjusted R-squared:  0.9638
F-statistic: 640.1 on 1 and 23 DF,  p-value: < 2.2e-16
```