A logo with a lion and text

Description automatically generated

**MH3510 Regression Project**

**Team Members:**

**Joshua Shing (U2140212C)** **Lim Li Xiang (U2140560B)**

**Muhammad Firaz Khan Bin**  **Tiew Ding Xuan (U2140462J)**

**Mohmed Jherkan (U2140712K)**

**Lee Choong Wee, Basil (U2140042C)** **Lim Kai Zhe (U2140940E)**

**Ang Tse Pin (U2140353E)** **Ryan Chong (U2140942J)**

**Cai Ying Zhi (U2140245C)**

***R Code:***

library(dplyr)

library(ggplot2)

library(lmtest)

aadt\_raw = read.table('aadt.txt', header = FALSE)

aadt = data.frame(y = aadt\_raw$V1, x1 = aadt\_raw$V2, x2 = aadt\_raw$V3, x3 = aadt\_raw$V4, x4 = aadt\_raw$V5)

plot(aadt)

A screenshot of a computer

Description automatically generated

**Comment:** There are 4 variables used in this multi-linear regression model. X1 is population of county in which road section is located. X2 is number of lanes in road section. X3 is width of road section (in feet). X4 is two-category quality variable indicating whether or not there is control of access to road section. The scatter plot indicates a relatively sparse relation between response; Y and X1 predictors. There are relatively clear linear relationships for majority of the datapoints between the response and rest of predictor variables.

***R Code:***

# Fit MLR model

mlr = lm(y ~ x1 + x2 + x3 + x4, data = aadt)

summary(mlr)

A screenshot of a computer

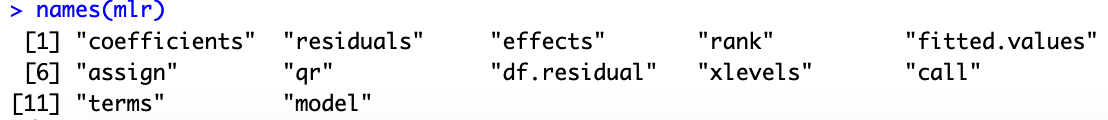
Description automatically generated

**Comment:** X3 shows the lowest significance.

***R Code:***

# Checking time order

names(mlr)



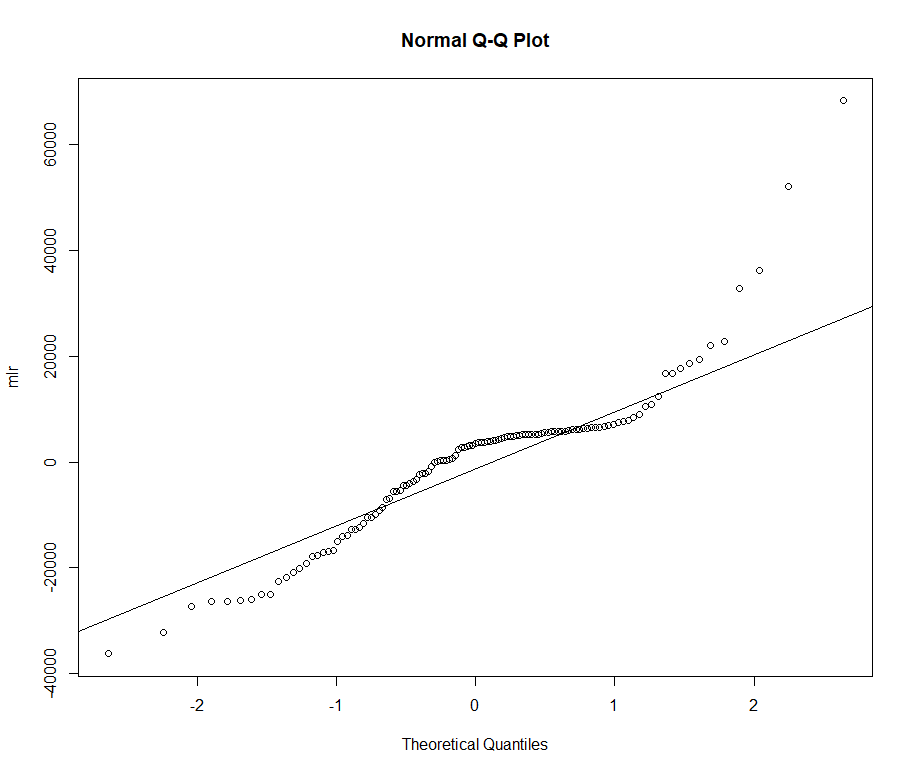
**Comment:** There is no time order.

***R Code:***

# Normality checking

qqnorm(residuals(mlr), ylab = "mlr")

qqline(residuals(mlr))



**Comment:** The residuals of first MLR fitting violates normality assumption.

***R Code:***

# Residual Plot

par(mfrow = c(1,5))

plot(residuals(mlr), fitted(mlr), ylab = "Residuals", xlab = "Fitted values")

plot(residuals(mlr), aadt$x1, ylab = "Residuals", xlab = "X1")

plot(residuals(mlr), aadt$x2, ylab = "Residuals", xlab = "X2")

plot(residuals(mlr), aadt$x3, ylab = "Residuals", xlab = "X3")

plot(residuals(mlr), aadt$x4, ylab = "Residuals", xlab = "X4")

par(mfrow = c(1,1))

A group of rectangular objects with numbers

Description automatically generated with medium confidence

**Comments:** There is expanding variance observed in fitted values. And bottom clustering observed in X1.

***R Code:***

# Checking for sequential dependence/ Durbin-Watson test

library(lmtest)

dwtest(y ~ x1 + x2 + x3 + x4, data = aadt)

A black text on a white background

Description automatically generated

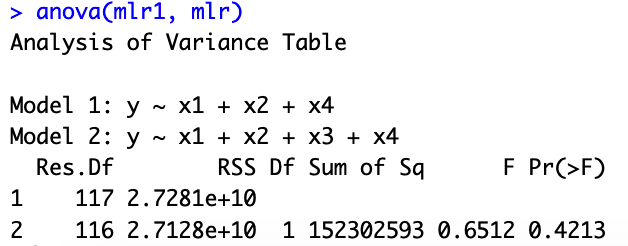
**Comments:** Successive residuals are positively serially correlated.

***R Code:***

# F-test if the beta values of non-significant predictors are 0. That is, .

mlr1 = lm(y ~ x1 + x2 + x4, data = aadt)

anova(mlr1, mlr)



**Comments:** Do not reject null hypothesis, can remove X3.

***R Code:***

# Consider transformation of y to obtain constant variance.

mlr2 = lm(I(y^(1/3)) ~ x1 + x2 + x4, data = aadt)

summary(mlr2)

A screenshot of a computer

Description automatically generated

**Comments:** Adjusted R squared = 0.8458.

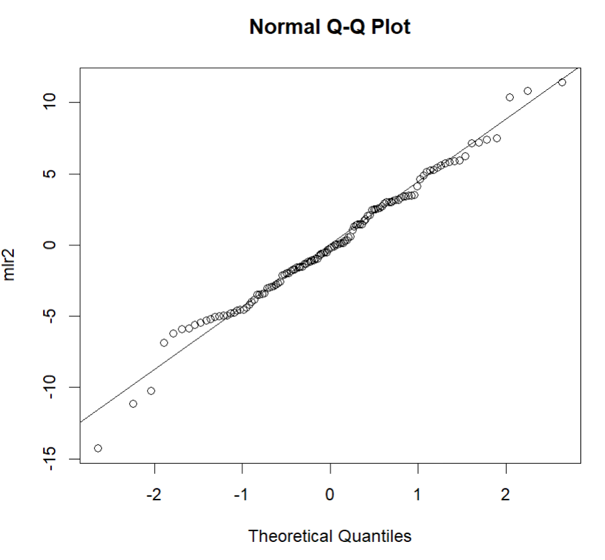
***R Code:***

# Normality Check on mlr 2

par(mfrow = c(1,1))

qqnorm(residuals(mlr2), ylab = "mlr2")

qqline(residuals(mlr2))



**Comments:** qq-plot appears to be much smoother than before.

***R Code:***

# Residual Plot for mlr2

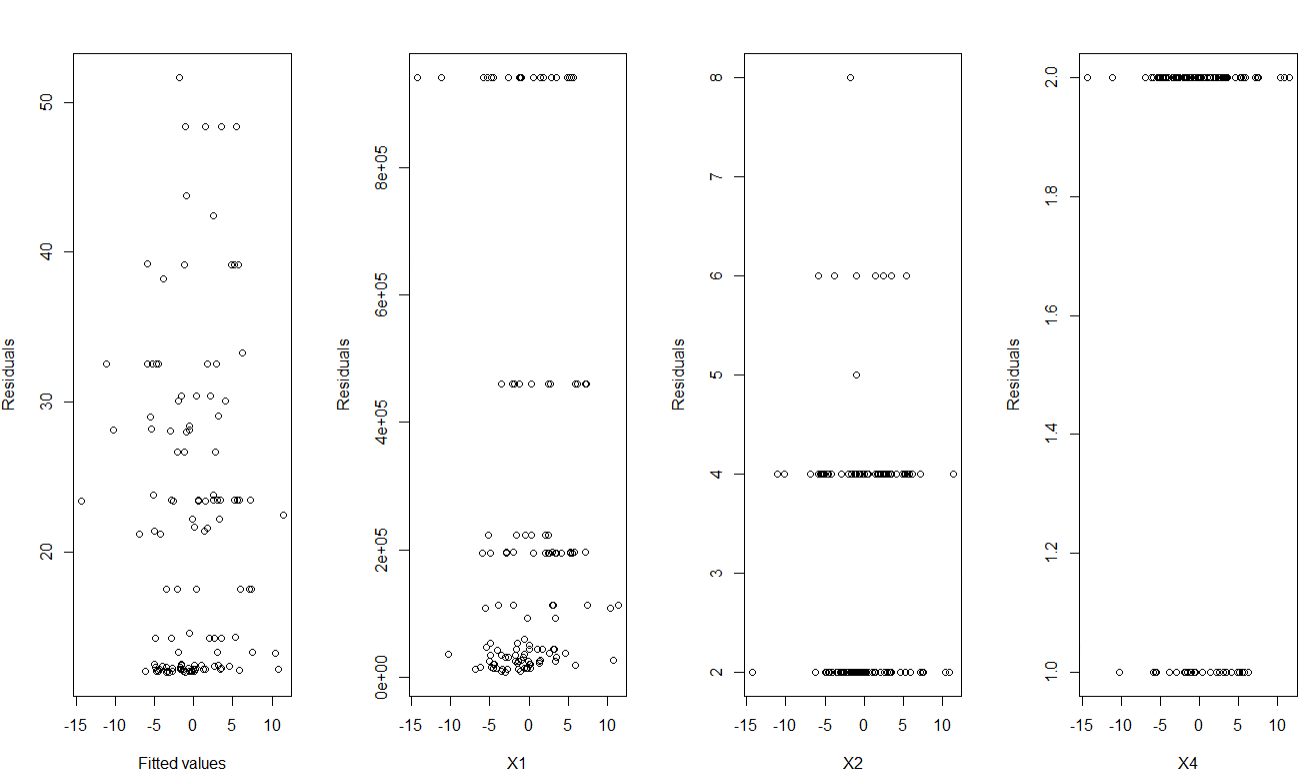
par(mfrow = c(1,4))

plot(residuals(mlr2), fitted(mlr2), ylab = "Residuals", xlab = "Fitted values")

plot(residuals(mlr2), aadt$x1, ylab = "Residuals", xlab = "X1")

plot(residuals(mlr2), aadt$x2, ylab = "Residuals", xlab = "X2")

plot(residuals(mlr2), aadt$x4, ylab = "Residuals", xlab = "X4")



**Comments:** Residuals vs fitted values appear more constant, but bottom clustering is still visible.

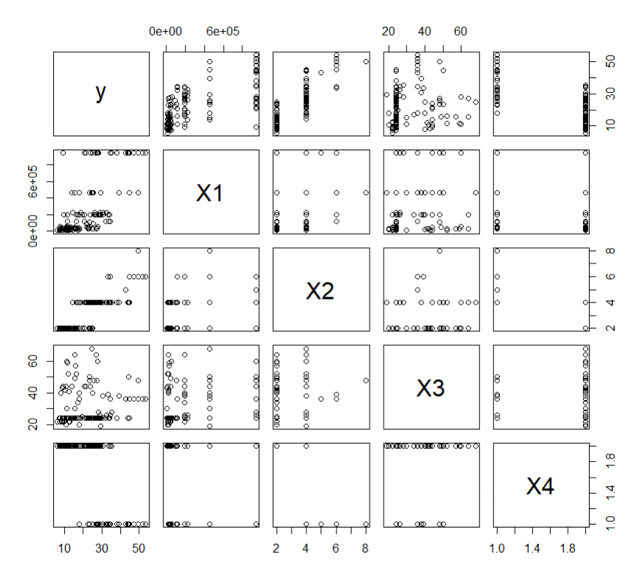
***R Code:***

# Observing transformed y^(1/3) vs untransformed x values

aadtv2 = aadt

aadtv2$y = aadtv2$y^(1/3)

plot(aadtv2)



**Comments:** y^(1/3) appears to be a logarithmic function of X1.

***R Code:***

# Consider transformation of X1, including y^(1/3)

mlr3 = lm(y^(1/3) ~ log(x1) + x2 + x4, data = aadt)

summary(mlr3)

A screenshot of a computer

Description automatically generated

**Comments:** Adjusted R squared = 0.859.

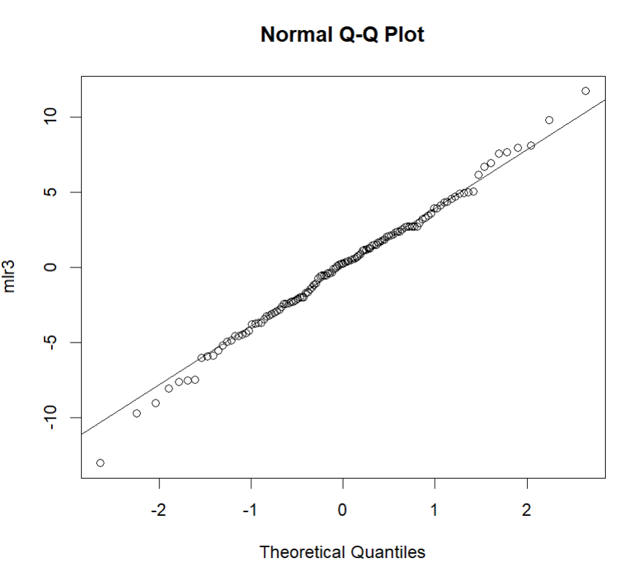
***R Code:***

# Normality check on mlr3

par(mfrow = c(1,1))

qqnorm(residuals(mlr3), ylab = "mlr3")

qqline(residuals(mlr3))



**Comments:** After applying transformations to both the response variable y and predictor variable X1, it generates a much smoother qq-plot.

***R Code:***

# Residual plot of mlr3

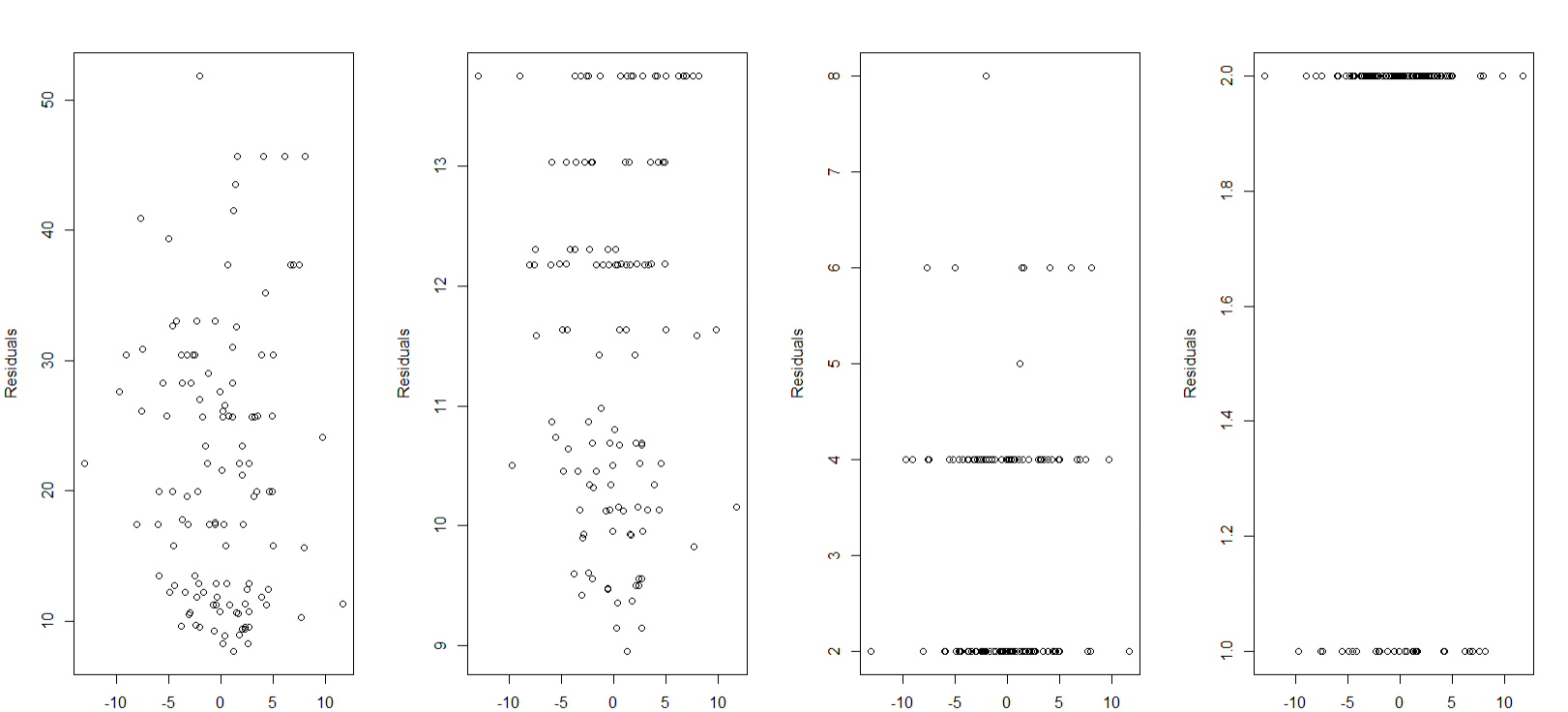
par(mfrow = c(1,4))

plot(residuals(mlr3), fitted(mlr3), ylab = "Residuals", xlab = "Fitted values")

plot(residuals(mlr3), log(aadt$x1), ylab = "Residuals", xlab = "X1")

plot(residuals(mlr3), aadt$x2, ylab = "Residuals", xlab = "X2")

plot(residuals(mlr3), aadt$x4, ylab = "Residuals", xlab = "X4")



**Comments:** There is much lesser bottom clustering for fitted vs residual plot.

***R Code:***

# Checking for Sequential Dependence/ Durbin-Watson Test

dwtest(y^(1/3) ~ log(x1) + x2 + x4, data = aadt)

A black text on a white background

Description automatically generated

**Comments:** Successive residuals are positively serially correlated.

***R Code:***

# Predictions (Manually)

mlrs = summary(mlr3)

con <- c(1,log(50000),3,2)

lhat <- sum(con\*coef(mlr3))

**Comments:** According to summary(mlr3), the degrees of freedom = 117.

***R Code:***

# Calculate the critical value with degree of freedom of 116t = qt(0.975,117)

c3 = 1

bm = t\*mlrs$sigma\*sqrt(con%\*%mlrs$cov.unscaled%\*%con + c3)

# Prediction interval of new response

c(lhat - bm, lhat + bm)^3

**A close up of numbers

Description automatically generated**

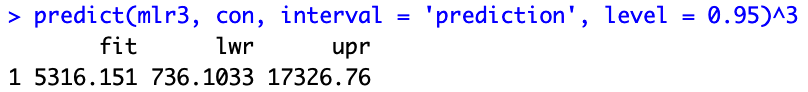
**Comments:** For predictions, we used the reduced model mlr3 with X1=50000, X2=3, x3=60, X4=2. Since the response variable is transformed into y^1/3, we take the cube of the interval.

***R Code:***

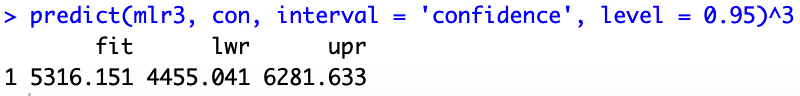
# Alternative method to get prediction interval

con = data.frame(x1 = 50000, x2 = 3, x3 = 60, x4 = 2)

predict(mlr3, con, interval = 'prediction', level = 0.95)^3

****

predict(mlr3, con, interval = 'confidence', level = 0.95)^3

****