# STAT 310 Lab 4 Residual Analysis

PrabhTalwar

2022-11-02

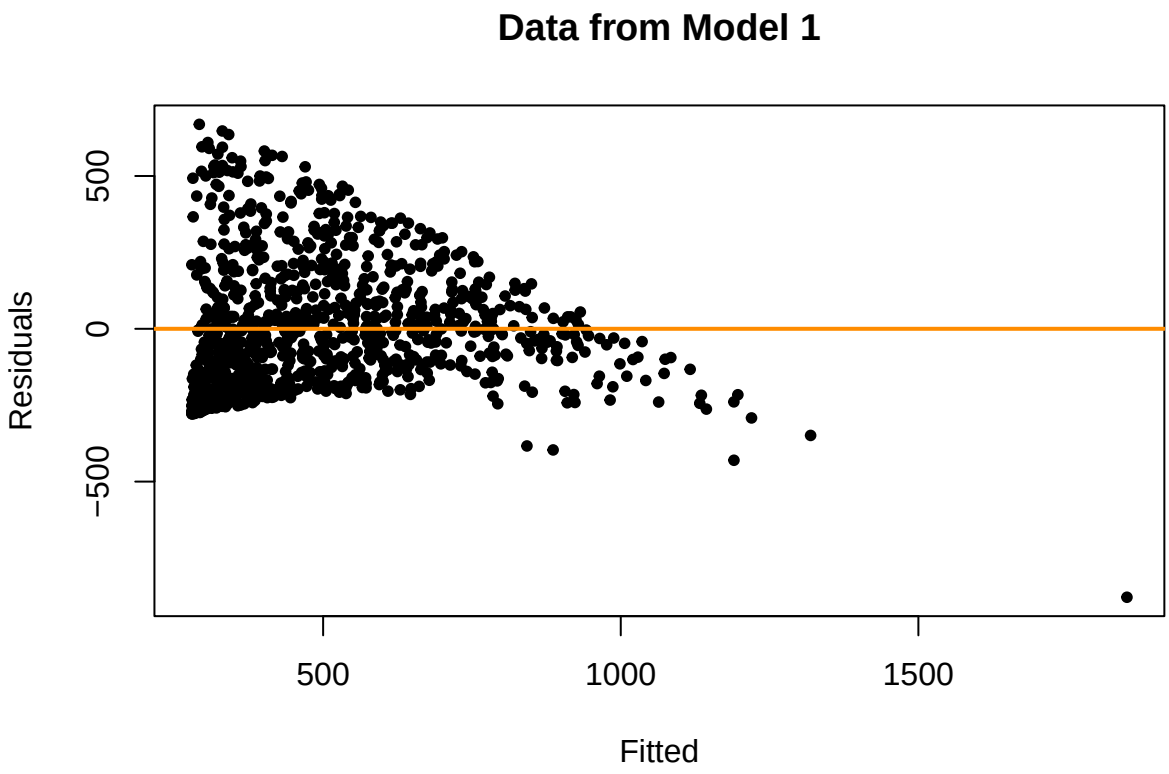
```
library(tidyverse)
library(lmtest)
library(car)
library(MASS)
```

## Generate the following data and fit the given model.

```
set.seed(0)

#define response variable
y <- c(1:1000)

#define three predictor variables
x1 <- c(1:1000)*runif(n=1000)
x2 <- (c(1:1000)*rnorm(n=1000))^2
x3 <- (c(1:1000)*rnorm(n=1000))^3

#fit multiple linear regression model
model <- lm(y~x1+x2+x3)
```

## Check for linearity heteroscedasticity with a standard residual plot.

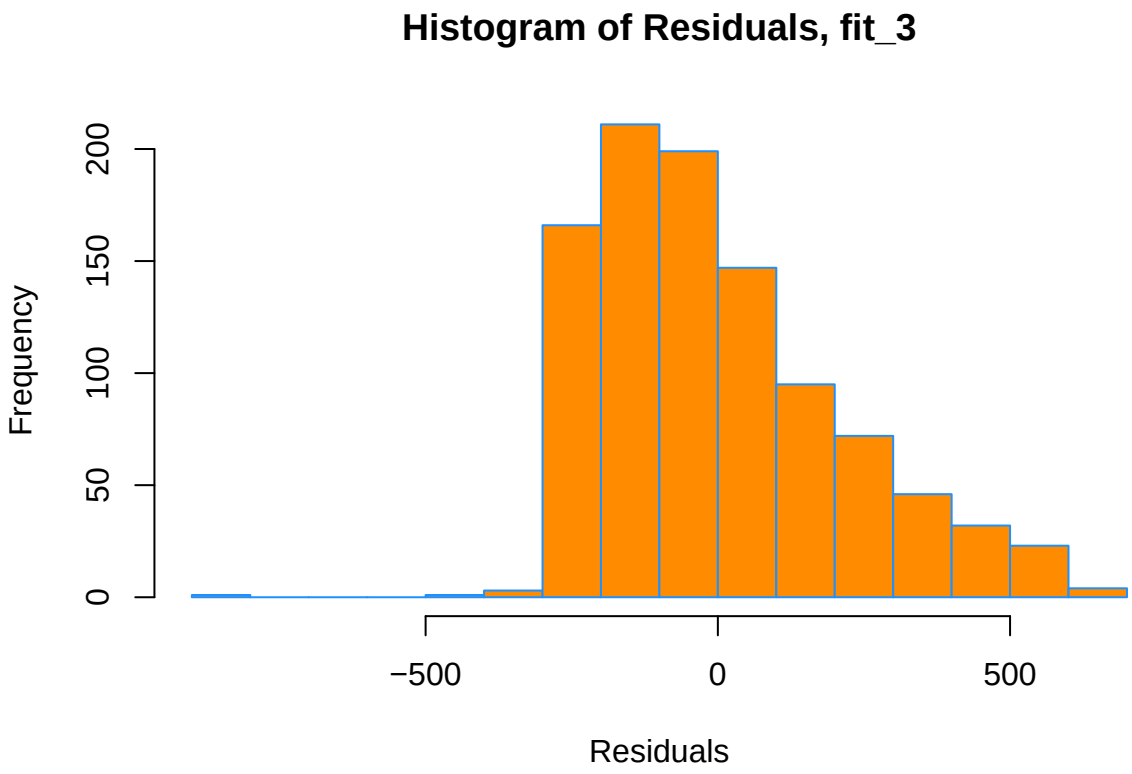
```
plot(fitted(model), resid(model), col = "black", pch = 20,
xlab = "Fitted", ylab = "Residuals", main = "Data from Model 1")
abline(h = 0, col = "darkorange", lwd = 2)
```



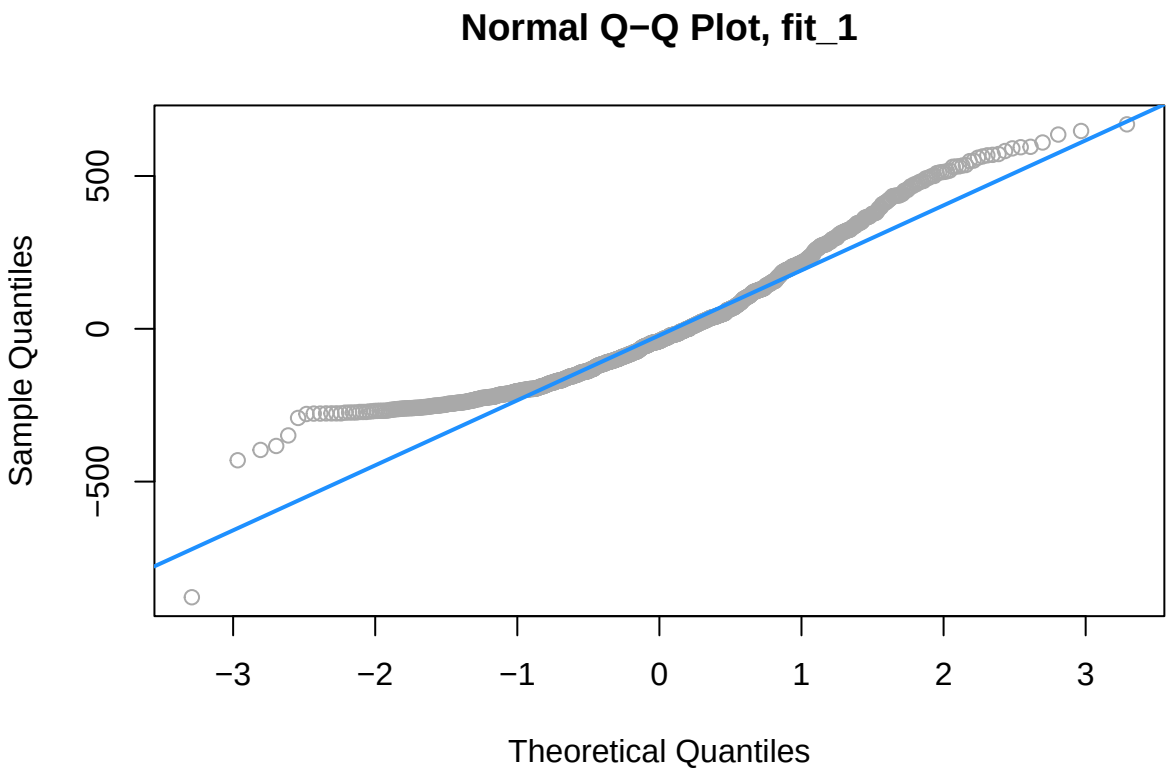

On the fitted versus residuals plot, For the fitted values, the residuals are centered at 0.The linearity assumption is not violated. However, for larger fitted values, the spread of the residuals is large. The constant variance assumption is violated here.

## . Check for normality of the residuals.

```
hist(resid(model),
xlab = "Residuals",
main = "Histogram of Residuals, fit_3",
col = "darkorange",
border = "dodgerblue",
breaks = 20)
```




```
qqnorm(resid(model), main = "Normal Q-Q Plot, fit_1", col = "darkgrey")
qqline(resid(model), col = "dodgerblue", lwd = 2)
```




we have a Q-Q plot here. We can say that the errors follow a normal distribution.

## Perform relevant statistical tests to verify your observations in the previous parts.

### Breusch-Pagan Test

There are many tests for constant variance, but here we will perform the Breusch-Pagan Test.

H0: Homoscedasticity. The errors have constant variance about the true model. HA: Heteroscedasticity. The errors have non-constant variance about the true model.

```
bptest(model)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model
## BP = 70.949, df = 3, p-value = 2.674e-15
```

Here, we see a small p-value, so we have enough evidence to reject the null hypothesis. The constant variance assumption is violated. This matches our findings with a fitted versus residuals plot.

### Shapiro-Wilk Test

The test is available to check the normality of our errors

H0: The data is sampled from a normal distribution HA: The data is not sampled from a normal distribution

```
shapiro.test(resid(model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(model)
## W = 0.93871, p-value < 2.2e-16
```

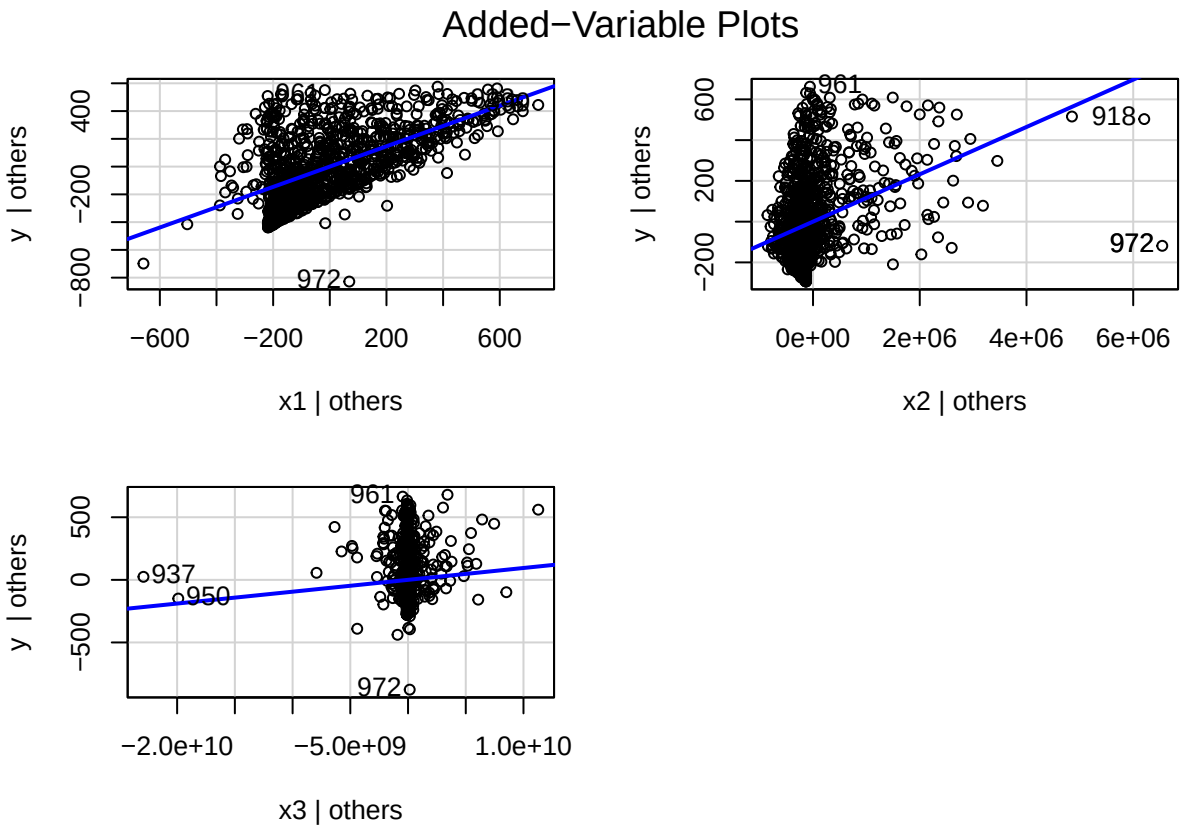
Here, we see a small p-value, so we have enough evidence to reject the null hypothesis. A small p-value indicates we believe there is only a small probability the data could have been sampled from a normal distribution.

## Use partial regression plots to check for heteroscedasticity and influential points.

```
model <- lm(y~x1+x2+x3)
summary(model)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -878.37 -164.86  -39.87  122.03  669.00
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.791e+02  1.008e+01  27.672   <2e-16 ***
## x1          7.313e-01  3.111e-02  23.506   <2e-16 ***
## x2          1.161e-04  1.031e-05  11.257   <2e-16 ***
## x3          9.468e-09  4.672e-09   2.026    0.043 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 210.2 on 996 degrees of freedom
## Multiple R-squared:  0.472,  Adjusted R-squared:  0.4704
## F-statistic: 296.8 on 3 and 996 DF,  p-value: < 2.2e-16
```

```
#  generating the AV plots

avPlots(model)
```

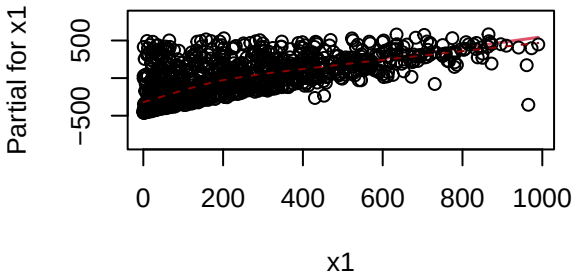


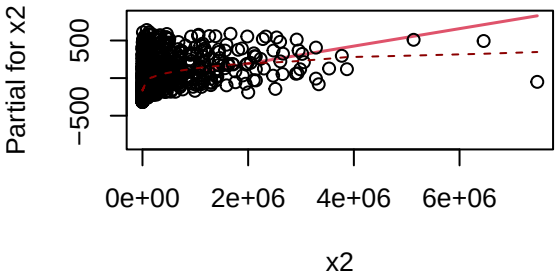

here, the importance of the variables in the model is indicated by the steepness of the slopes. Variable x3 has less steepness than x1 and x2. We can also see the potential influential points clearly. Observations 961 and 972 stand out in all the three plots.

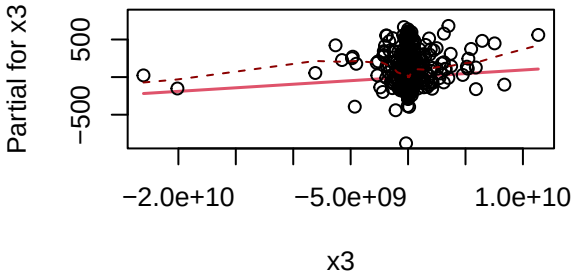
## Use partial residual plots to check for linearity among each variable. (Bonus) Can you recommend any transformations?

The partial residual plots are most commonly used to identify the nature of the relationship between Y and Xi

```
par(mfrow=c(2,2))
termplot(model, partial.resid=TRUE, col.res = "black", smooth=panel.smooth)
```



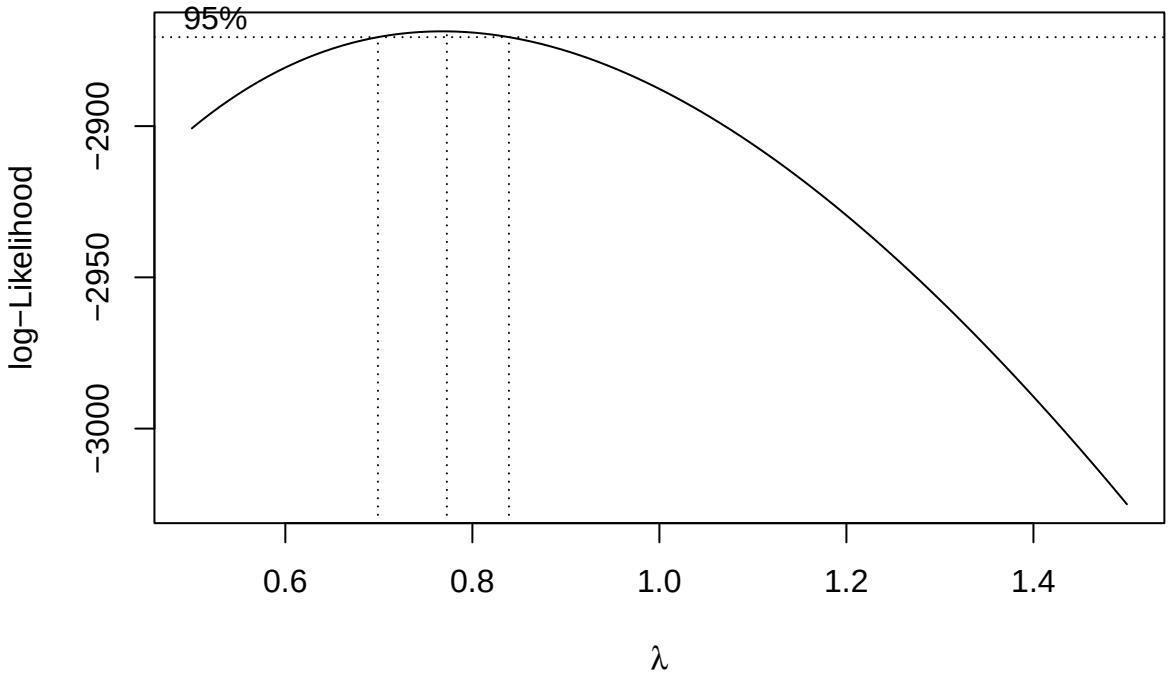




Here, x1 shows the linearity trend among the other variables in the data.

Box-Cox Transformations will check whether we need to transform the data or not.

```
boxcox(model, plotit = TRUE, lambda = seq(0.5, 1.5, by = 0.1))
```




The lambda comes out to be around 0.7. To transform the data we might use the square root.