# Model Adequecy Check

### B Aditi \|\| MM19B022

## Problem 4.2

```{r}
#Getting table from 3.1
tb1 <- read.csv('Table_B1.csv',header=TRUE,
                      stringsAsFactors=FALSE)
print(tb1)
```

```{r}
#model fittin as per 3.1
model2 <- lm(Y ~ X2 + X7 + X8, data = tb1)
summary(model2)
```

#### Part a: QQ plot of residuals

```{r}
#probability plot
stdres2 = rstandard(model2)
qqnorm(stdres2,
       ylab="Standardized Residuals",
       xlab="Normal Scores",
       main="Residual distribution plot")
qqline(stdres2)
```

From the above graph we can conclude that there is a slight issue with the normality of the residuals.

#### Part b: Residual vs Prediction

```{r}
plot(model2, which = c(1,1))
```

This plot looks balanced.

#### Part c: Residuals vs Regressor plots

```{r}
install.packages("ggplot2")

```

```{r}
library(ggplot2)
```

```{r}
res2 <- resid(model2)
```

```{r}
ggplot(tb1, aes(x = X2, y = res2)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  xlab("Predictor") +
  ylab("Residuals") +
  ggtitle("Residuals vs. X2 Plot")

```

```{r}
ggplot(tb1, aes(x = X7, y = res2)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  xlab("Predictor") +
  ylab("Residuals") +
  ggtitle("Residuals vs. X7 Plot")
```

```{r}
ggplot(tb1, aes(x = X8, y = res2)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  xlab("Predictor") +
  ylab("Residuals") +
  ggtitle("Residuals vs. X8 Plot")
```

While the plot for X8 shows somewhat constant variance, the plot for X7 shows non-constant variance and the plot for X2 is partially constant.

#### Part d: Partial regressor plots

```{r}
library(car)
```

```{r}
#create partial residual plots
crPlots(model2)
```

#### Part e: Studentised Residuals

```{r}
library(MASS)
```

```{r}
stud_resids2 <- studres(model2)
print(stud_resids2)
```

These can be used fo find outliers in the model.

## Problem 4.5

```{r}
#Dataset from 3.7
tb4 <- read.csv('Table_B4.csv',header=TRUE,
                      stringsAsFactors=FALSE)
print(tb4)
```

```{r}
#fitting model
model5 = lm(X ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9, data = tb4)
summary(model5)
```

#### Part a: QQ plot of residuals

```{r}
plot(model5, which = c(1,2))
```

Based on the plot, the assumption of normality of residuals seems valid without any problems.

#### Part b: Residual vs Predicted values

```{r}
plot(model5, which = c(1,1))
```

While the variance is fairly uniform, there is a slight upward shift in the graph.

#### part c: Partial regression plots

```{r}
library(gridExtra)
```

```{r}
plot5 = crPlots(model5)
```

From these graphs we can see that pther than X1 rest deviate from the values and are thus not necessary.

#### Part d: studentised and R-studentised residuals

```{r}
# Compute the studentized residuals
studentized_res5 <- rstandard(model5)

# Compute the R-student residuals
rstudent_res5 <- rstudent(model5)

```

```{r}
print(studentized_res5)
```

```{r}
print(rstudent_res5)
```

## Problem 4.9

```{r}
#inputting dataset
tb9 <- read.csv('table_2_ozone.csv',header=TRUE,
                      stringsAsFactors=FALSE)
print(tb9)
```

```{r}
#Fitting model
model9 = lm(Days ~ Index, data = tb9)
summary(model9)
```

#### Part a: QQ plot for Residuals

```{r}
plot(model9, which = c(1,2))
```

The normal assumptions of residuals seem to be valid.

#### Part b: Residuals vs Predictions

```{r}
plot(model9, which = c(1,1))
```

```

There is a clear pattern that is being captured in this plot

#### Part c: Residual vs time plot

```{r}
res9 = resid(model9)
time_order <- 1:nrow(tb9)

plot(time_order, res9,
     ylab="Residuals", xlab="Time order",
     main="Residual vs Time plot")
```

The graph shows a positive autocorrelation.

## Problem 4.16

```{r}
#inputting dataset
tb16 <- read.csv('table_B8.csv',header=TRUE,
                      stringsAsFactors=FALSE)
print(tb16)
```

```{r}
#building model
model16 = lm(y ~ x1 + x2, data = tb16)
summary(model16)
```

#### Part a: QQ plot for Residuals

```{r}
stdres16 = rstandard(model16)
qqnorm(stdres16,
       ylab="Standardized Residuals",
       xlab="Normal Scores",
       main="Residual distribution plot")
qqline(stdres16)
```

The plot is abnormal near the tails

#### Part b: Residual vs Predictions plot

```{r}
plot(model16, which = c(1,1))
```

The fit seems to be pretty good.

#### Part c: PRESS values

```{r}
model16_2 = lm(y ~ x2, data = tb16)
summary(model16_2)
```

```{r}
#calculating residula
r16_2 <- resid(model16_2)
r16 = resid(model16)

```

```{r}

pr16 <- (resid(model16)/(1 - lm.influence(model16)$hat))^2
print(sum(pr16))
```

```{r}
pr16_2 <- (resid(model16_2)/(1 - lm.influence(model16_2)$hat))^2
print(sum(pr16_2))
```

Based on the press values of both values, the model with both x1 and x2 will work better.

## Problem 4.20

```{r}
#importing data
tb20 <- read.csv('table_4_20.csv',header=TRUE,
                      stringsAsFactors=FALSE)
print(tb20)
```

```{r}
#fitting the model
model20 = lm(y ~ Acid.Temp. + Acid.Conc. + WaterTemp. + Sulfide.Conc., data = tb20)
summary(model20)
```

#### Part a: Model adequecy test

```{r}
#QQ plot
plot(model20, which = c(1,2))
```

Clearly, there is a problem with the normal assumption

```{r}
res20 = resid(model20)

# produce residual vs. fitted plot
plot(fitted(model20), res20)

# add a horizontal line at 0
abline(0,0)
```

We also note that there is non constant variance.

```{r}
plot20 = crPlots(model20)
```

#### Part b: Lack of fit test

There is no test for lack of fit since there are no replicate points. It is possible to use the near-neighbor approach.

## Problem 4.21

Please refer the pdf file with the name "Problem4_21"
```

# PROBLEM 4.21

To find:-

$\mathbb{E}[MS_{PE}] = ?$

$\mathbb{E}[MS_{LOF}] = ?$

Working:-

$$\mathbb{E}[MS_{PE}] = \frac{1}{n-m} \mathbb{E}\left[\sum_{i=1}^{m}\sum_{j=1}^{n} (y_{ij} - \bar{y}_i)^2\right] \text{ ~~where~~ } \text{——— } (1)$$

$$\mathbb{E}\left[\sum_{i=1}^{m}\sum_{j=1}^{n}(y_{ij}-\bar{y}_i)^2\right] = E\left[\sum_{i=1}^{m}\sum_{j=1}^{n}(y_{ij}^2 - 2y_{ij}\bar{y}_i + \bar{y}_i^2)\right]$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{n}\left\{E(y_{ij}^2) - 2E\left(y_{ij}\sum_{j'=1}^{n_i}\frac{y_{ij'}}{n_i}\right) + E(\bar{y}_i^2)\right\}$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{n}\left\{\sigma^2 - 2E\left[y_{ij}\sum_{j'=1}^{n_i}\frac{y_{ij'}}{n_i}\right] + \frac{\sigma^2}{n_i}\right]$$

$$= n\sigma^2 + m\sigma^2 - 2\sum_{i=1}^{m}\left(\sum_{j=1}^{}\sum_{j'=1}^{n_i}\frac{y_{ij}y_{ij'}}{n_i}\right)$$

$$= n\sigma^2 + m\sigma^2 - 2\sum_{i=1}^{m}\frac{n_i\sigma^2}{n_i} = n\sigma^2 + m\sigma^2 - 2m\sigma^2$$

$$= (n-m)\sigma^2 \text{ ——— } (2)$$

Sub (2) in (1) :-

$$\mathbb{E}[MS_{PE}] = \frac{1}{n-m} \times (n-m)\sigma^2 = \boxed{\sigma^2} \Rightarrow \boxed{\mathbb{E}[MS_{PE}] = \sigma^2}$$

$$SS_{Res} = SS_{LOF} + SS_{PE} \Rightarrow \boxed{SS_{LOF} = SS_{Res} - SS_{PE}}$$

$$\Rightarrow \mathbb{E}[SS_{LOF}] = \mathbb{E}[SS_{Res}] - \mathbb{E}[SS_{PE}] = (n-2)\sigma^2 + \sum_{i=1}^{m}\left(\mathbb{E}(y_i) - \beta_0 - \beta_1 x_i\right)^2 - (n-m)\sigma^2$$

$$= (m-2)\sigma^2 + \sum_{i=1}^{m}\left[\mathbb{E}[y_i) - \beta_0 - \beta_1 x_i\right)^2$$

$$\mathbb{E}[MS_{LOF}] = \frac{\mathbb{E}[SS_{LOF}]}{m-2} \Rightarrow \mathbb{E}[MS_{LOF}] = \sigma^2 + \frac{\sum_{i=1}^{m}\left[\mathbb{E}(y_i) - \beta_0 - \beta_1 x_i\right)^2}{m-2}$$

$\Rightarrow$