Assignment 4

```
# packages
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------------------------------------------------ 
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(faraway)
```

```
## Warning: package 'faraway' was built under R version 4.0.3
```

```
library(car)
```

```
## Loading required package: carData
```

```
## Registered S3 methods overwritten by 'car':
##   method                           from
##   influence.merMod                 lme4
##   cooks.distance.influence.merMod lme4
##   dfbeta.influence.merMod          lme4
##   dfbetas.influence.merMod         lme4
```

```
##
## Attaching package: 'car'
```

```
## The following objects are masked from 'package:faraway':
##
##     logit, vif
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

2.

a).

```
# import data
assign4_i <- read.csv("assign4_i.csv")
```

```r
attach(assign4_i)
assign4_i <- select(assign4_i,x1:y)

# unit length scaling
assign4_imean <- apply(assign4_i, 2, FUN = mean)
assign4_isd <- apply(assign4_i, 2, FUN = sd)
sjj_sqrt_4i <- sqrt(9-1)*assign4_isd
w1_4i <- (x1-assign4_imean[1])/sjj_sqrt_4i[1]
w2_4i <- (x2-assign4_imean[2])/sjj_sqrt_4i[2]
sst_4i <- sum((y-assign4_imean[3])^2)
yi_4i <- (y-assign4_imean[3])/sqrt(sst_4i)
assign4_i_scaled <- lm(yi_4i ~ -1+ w1_4i + w2_4i)
(assign4_i_sum <- summary(assign4_i_scaled))
```

```
##
## Call:
## lm(formula = yi_4i ~ -1 + w1_4i + w2_4i)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.10404 -0.07079 -0.05422  0.09243  0.12643
##
## Coefficients:
##       Estimate Std. Error t value Pr(>|t|)
## w1_4i   0.9137     0.1008   9.067 4.06e-05 ***
## w2_4i   0.3068     0.1008   3.044   0.0187 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1008 on 7 degrees of freedom
## Multiple R-squared:  0.9289, Adjusted R-squared:  0.9086
## F-statistic: 45.74 on 2 and 7 DF,  p-value: 9.575e-05
```

The standardized regression coefficients are 0.9137 and 0.3068.

b).

```
# (W'W)^-1 matrix
W_4i <- cbind(w1_4i,w2_4i)
solve(t(W_4i) %*% W_4i) * assign4_i_sum$sigma^2
```

```
##            w1_4i      w2_4i
## w1_4i 0.01015452 0.00000000
## w2_4i 0.00000000 0.01015452
```

The non-diagonal elements are 0, the covariance between estimates of the coefficients for x1 and x2 is 0, so the estimates of the coefficients for x1 and x2 are independent.

c).

```
# diagnol of (W'W)^-1 matrix and VIF
diag(solve(t(W_4i) %*% W_4i))
```

```
## w1_4i w2_4i
##     1     1
```

```
vif(assign4_i_scaled)
```

```
## Warning in vif.default(assign4_i_scaled): No intercept: vifs may not be
## sensible.
```

```
## w1_4i w2_4i
##     1     1
```

VIF is 1.

d).

```
# import data
assign4_ii <- read.csv("assign4_ii.csv")
attach(assign4_ii)
```

```
## The following objects are masked from assign4_i:
##
##     Obs, x1, x2, y
```

```
assign4_ii <- select(assign4_ii,x1:y)

# unit length scaling
assign4_iimean <- apply(assign4_ii, 2, FUN = mean)
assign4_iisd <- apply(assign4_ii, 2, FUN = sd)
sjj_sqrt_4ii <- sqrt(9-1)*assign4_iisd
w1_4ii <- (x1-assign4_iimean[1])/sjj_sqrt_4ii[1]
w2_4ii <- (x2-assign4_iimean[2])/sjj_sqrt_4ii[2]
sst_4ii <- sum((y-assign4_iimean[3])^2)
yi_4ii <- (y-assign4_iimean[3])/sqrt(sst_4ii)
assign4_ii_scaled <- lm(yi_4ii ~ -1+ w1_4ii + w2_4ii)
(assign4_ii_sum <- summary(assign4_ii_scaled))
```

```
##
## Call:
## lm(formula = yi_4ii ~ -1 + w1_4ii + w2_4ii)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.234756 -0.058329 -0.002059  0.113727  0.267447
##
## Coefficients:
##        Estimate Std. Error t value Pr(>|t|)
## w1_4ii  0.89411    0.22504   3.973  0.00537 **
## w2_4ii -0.01155    0.22504  -0.051  0.96050
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1746 on 7 degrees of freedom
## Multiple R-squared:  0.7865, Adjusted R-squared:  0.7255
## F-statistic:  12.9 on 2 and 7 DF,  p-value: 0.004494
```

The standardized regression coefficients are 0.89411 and -0.01155.

e).

```
# (W'W)^-1 matrix
W_4ii <- cbind(w1_4ii,w2_4ii)
solve(t(W_4ii) %*% W_4ii) * assign4_ii_sum$sigma^2
```

```
##             w1_4ii      w2_4ii
## w1_4ii  0.05064141 -0.03194144
## w2_4ii -0.03194144  0.05064141
```

The non-diagonal elements are not 0, the covariance between estimates of the coefficients for x1 and x2 are -0.03194144, so the estimates of the coefficients for x1 and x2 are dependent.

f).

```
# diagnol of (W'W)^-1 matrix and VIF
diag(solve(t(W_4ii) %*% W_4ii))
```

```
##   w1_4ii   w2_4ii
## 1.660661 1.660661
```

```
vif(assign4_ii_scaled)
```

```
## Warning in vif.default(assign4_ii_scaled): No intercept: vifs may not be
## sensible.
```

```
##   w1_4ii   w2_4ii
## 1.660661 1.660661
```

VIF is 1.660661.

g).

The design from assign4_i are preferable, because the estimates of the coefficients for x1 and x2 are independent, and VIF is 1 which means x1 and x2 are not correlated.

3.

```
# generate Y
attach(assign4_i)
```

```
## The following objects are masked from assign4_ii:
##
##     x1, x2, y
```

```
## The following objects are masked from assign4_i (pos = 4):
##
##     x1, x2, y
```

```
# n=9
x1 <- c(1,2,3,1,2,3,1,2,3)
x2 <- c(1,1,1,2,2,2,3,3,3)
for(i in 1:1000){
  eps <- rnorm(9, mean=0, sd=20)
  y =  2*x1 + 2*x2 + eps
  reg1 <- lm(y ~ -1+x1+x2)
  reg_sum1 <- summary(reg1)
}

# n=18
x1 <- c(1,2,3,1,2,3,1,2,3)
x2 <- c(1,1,1,2,2,2,3,3,3)
x1 <- cbind(rep(x1,2))
x2 <- cbind(rep(x2,2))
for(i in 1:1000){
  eps <- rnorm(n = 18, mean=0, sd=20)
  y <- 2*x1 + 2*x2 + eps
  reg2 <- lm(y ~ -1 + x1 + x2)
  reg_sum2 <- summary(reg2)
  reg_sum2
}

# n=27
x1 <- c(1,2,3,1,2,3,1,2,3)
x2 <- c(1,1,1,2,2,2,3,3,3)
x1 <- cbind(rep(x1,3))
x2 <- cbind(rep(x2,3))
for(i in 1:1000){
  eps <- rnorm(n = 27, mean=0, sd=20)
  y <- 2*x1 + 2*x2 + eps
  reg3 <- lm(y ~ -1 + x1 + x2)
  reg_sum3 <- summary(reg3)
  reg_sum3
}

# n=36
x1 <- c(1,2,3,1,2,3,1,2,3)
x2 <- c(1,1,1,2,2,2,3,3,3)
x1 <- cbind(rep(x1,4))
x2 <- cbind(rep(x2,4))
for(i in 1:1000){
```

```
  eps <- rnorm(n = 36, mean=0, sd=20)
  y <- 2*x1 + 2*x2 + eps
  reg4 <- lm(y ~ -1 + x1 + x2)
  reg_sum4 <- summary(reg4)
  reg_sum4
}

# n= ...
```

4.

a).

```
# import data
pig <- read.csv("guinea_pig.csv")
attach(pig)
```

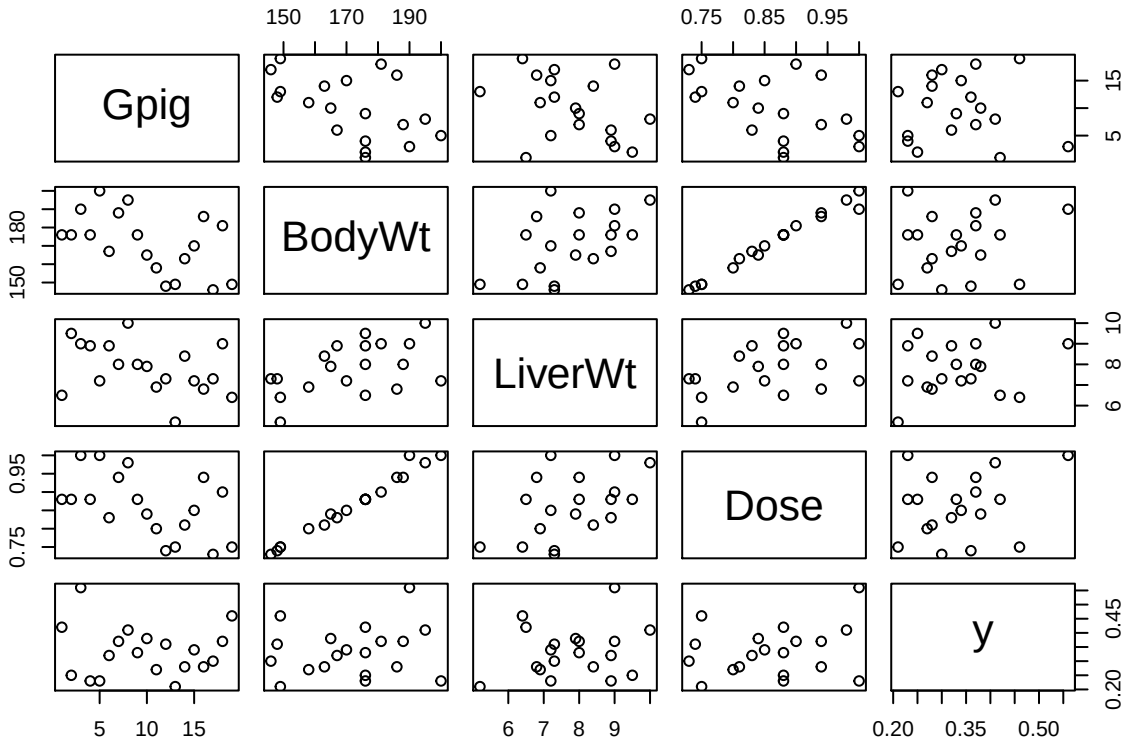
```
## The following object is masked _by_ .GlobalEnv:
##
##     y
```

```
## The following object is masked from assign4_i (pos = 3):
##
##     y
```

```
## The following object is masked from assign4_ii:
##
##     y
```

```
## The following object is masked from assign4_i (pos = 5):
##
##     y
```

```
pairs(pig)
```



(i)Comment on the relationship between the predictors. Do you observe any issues?

Since Gpig is just the observation number, we will not discuss it. From the pair plots, we can observe that there is a strong linear relationship between BodyWt and Dose. and the relationships between BodyWt and LiverWt, LiverWt and Dose seems are slightly linear. We will further discuss multicollinearity later.

(ii)Comment on the relationship between the predictors and the response variable.

BodyWt and Dose seems have a slightly linear relationship with response variable, but LiverWt is not linear to response variable at all.

b).

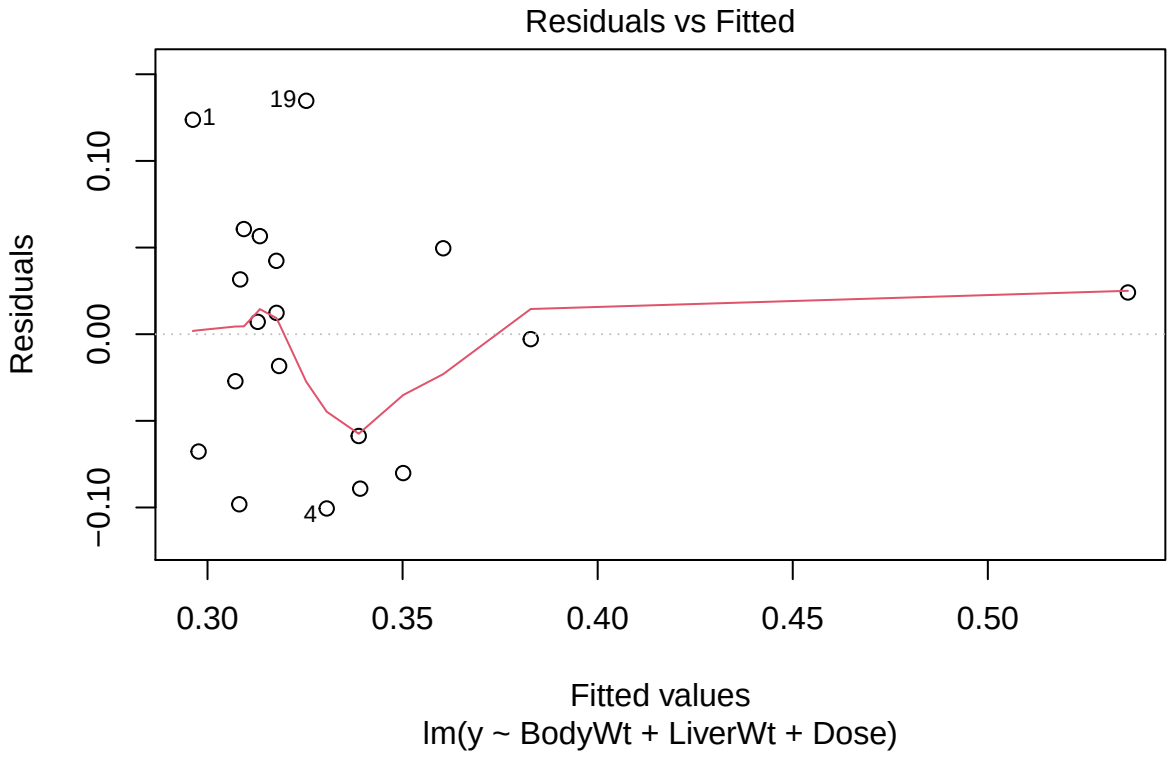
```
pig.lm <- lm(data = pig, y ~ BodyWt + LiverWt + Dose)
summary(pig.lm)
```

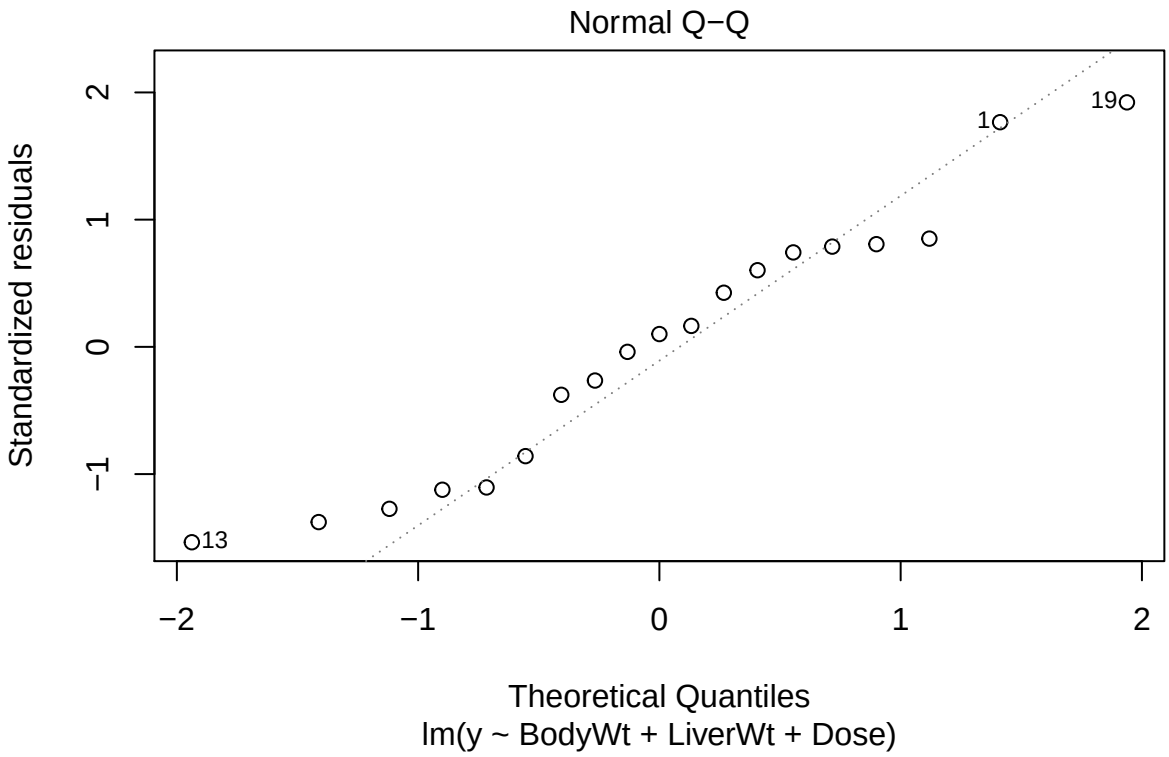
```
##
## Call:
## lm(formula = y ~ BodyWt + LiverWt + Dose, data = pig)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.100557 -0.063233  0.007131  0.045971  0.134691
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.265922   0.194585   1.367   0.1919
## BodyWt      -0.021246   0.007974  -2.664   0.0177 *
## LiverWt      0.014298   0.017217   0.830   0.4193
## Dose         4.178111   1.522625   2.744   0.0151 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07729 on 15 degrees of freedom
## Multiple R-squared:  0.3639, Adjusted R-squared:  0.2367
## F-statistic:  2.86 on 3 and 15 DF,  p-value: 0.07197
```

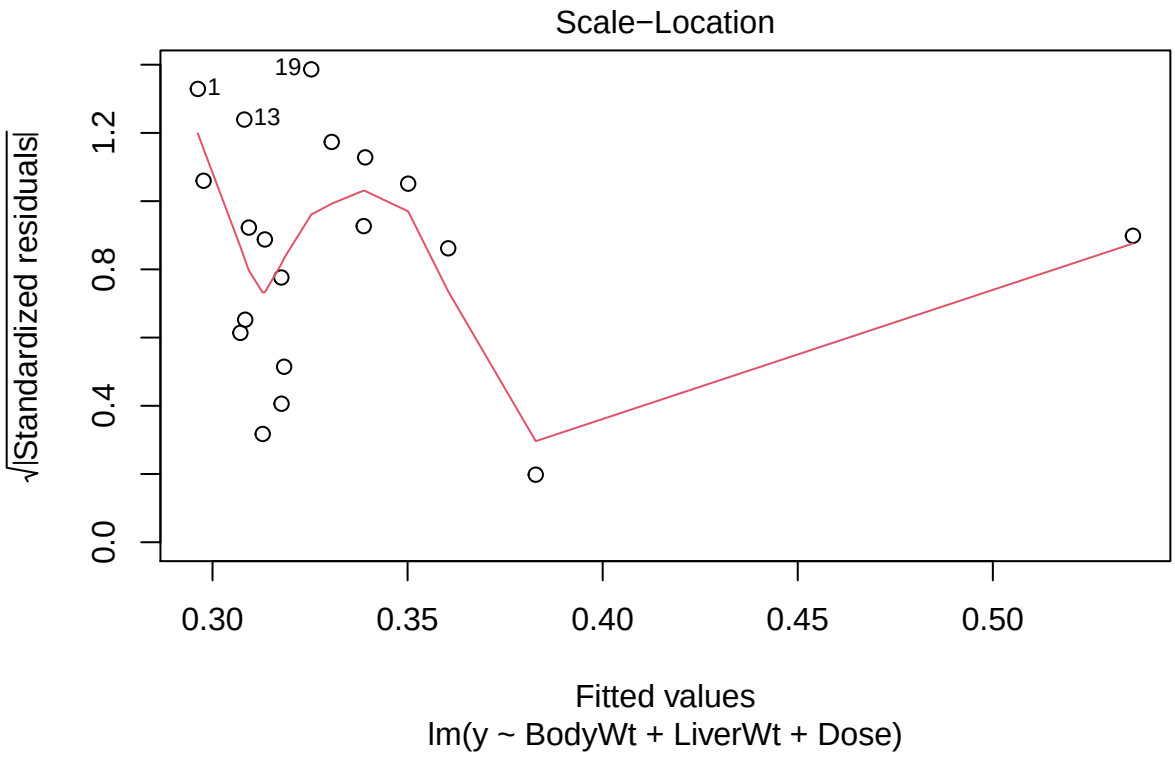
After fitting the model, the p-value is 0.07197 which is quite large, and R-squared is 0.3639 looks not very well. the p-values for each variable are 0.0177, 0.4193, 0.0151. If we choose our significant level as 0.05, then the p-value for BodyWt and Dose are less than 0.05, We can say that BodyWt have linear relationship with response variable in presence of other variables in the model. We can make the same conclusion to Dose as well.

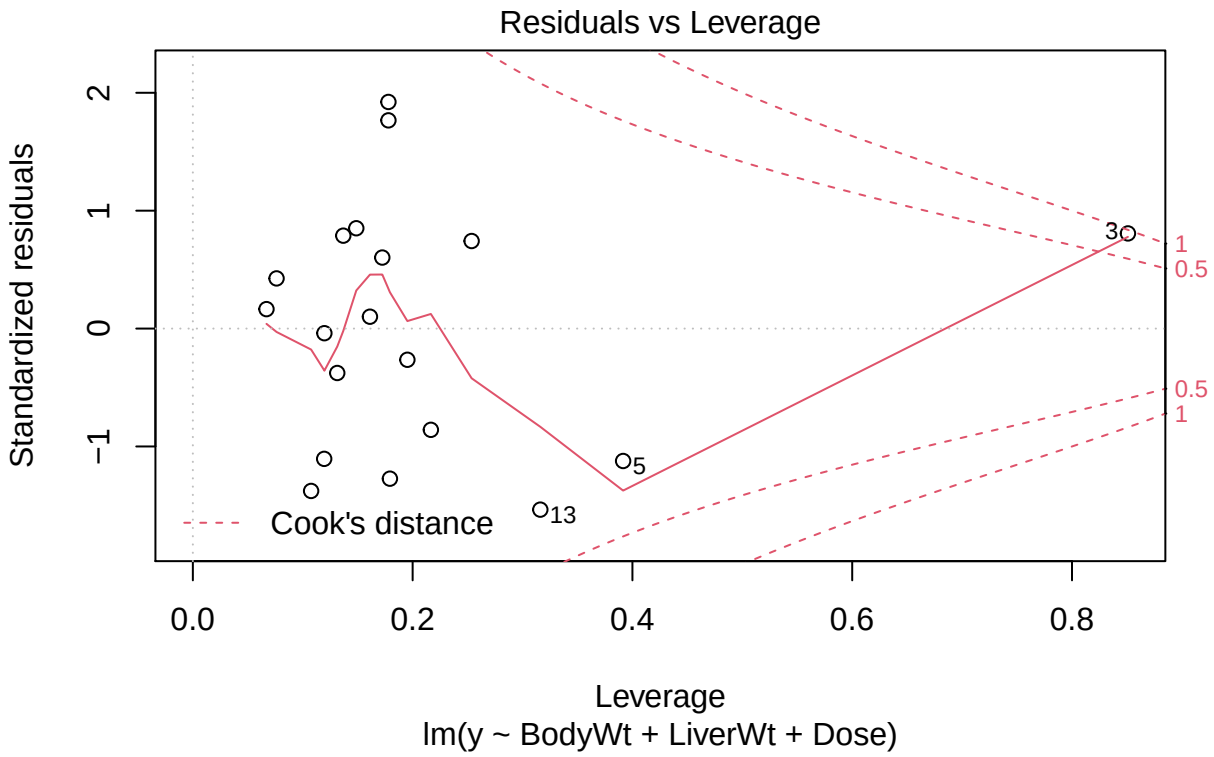
c).

```
plot(pig.lm)
```

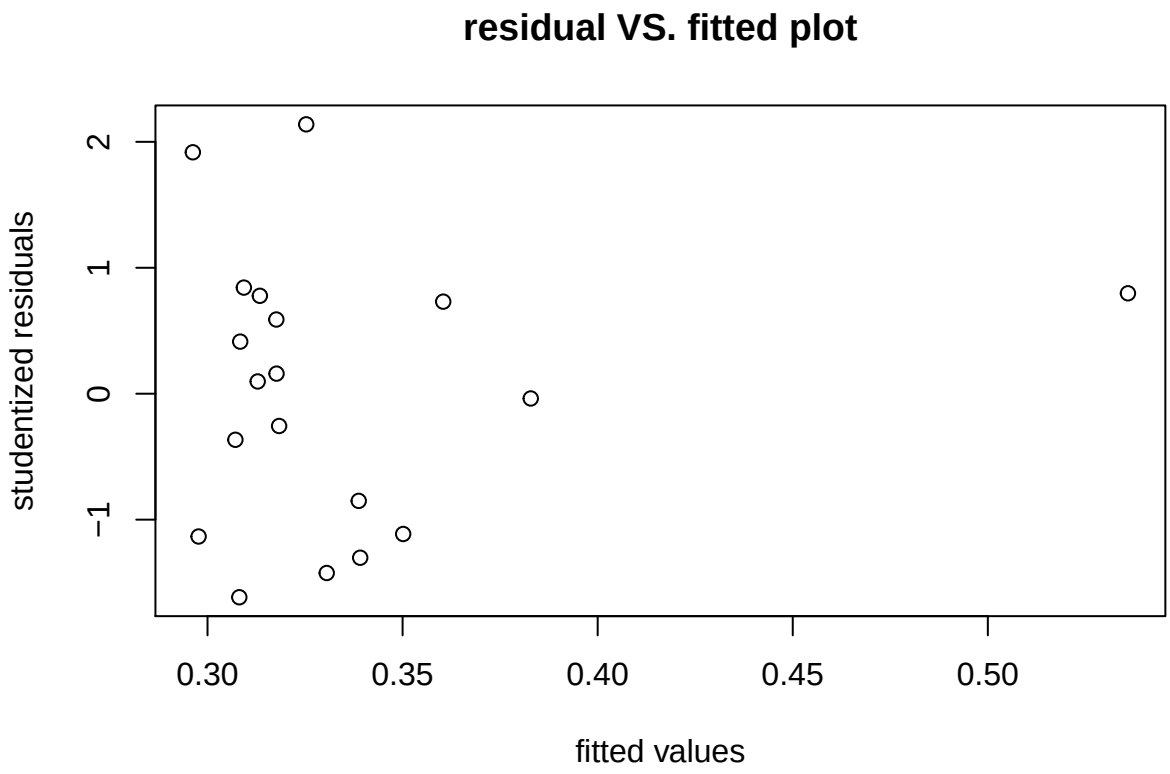

Residuals vs Fitted
Residuals
0.10
0.00
−0.10
1
19
4
0.30
0.35
0.40
0.45
0.50
Fitted values
lm(y ~ BodyWt + LiverWt + Dose)


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(y ~ BodyWt + LiverWt + Dose)
13
1
19
−2
−1
0
1
2

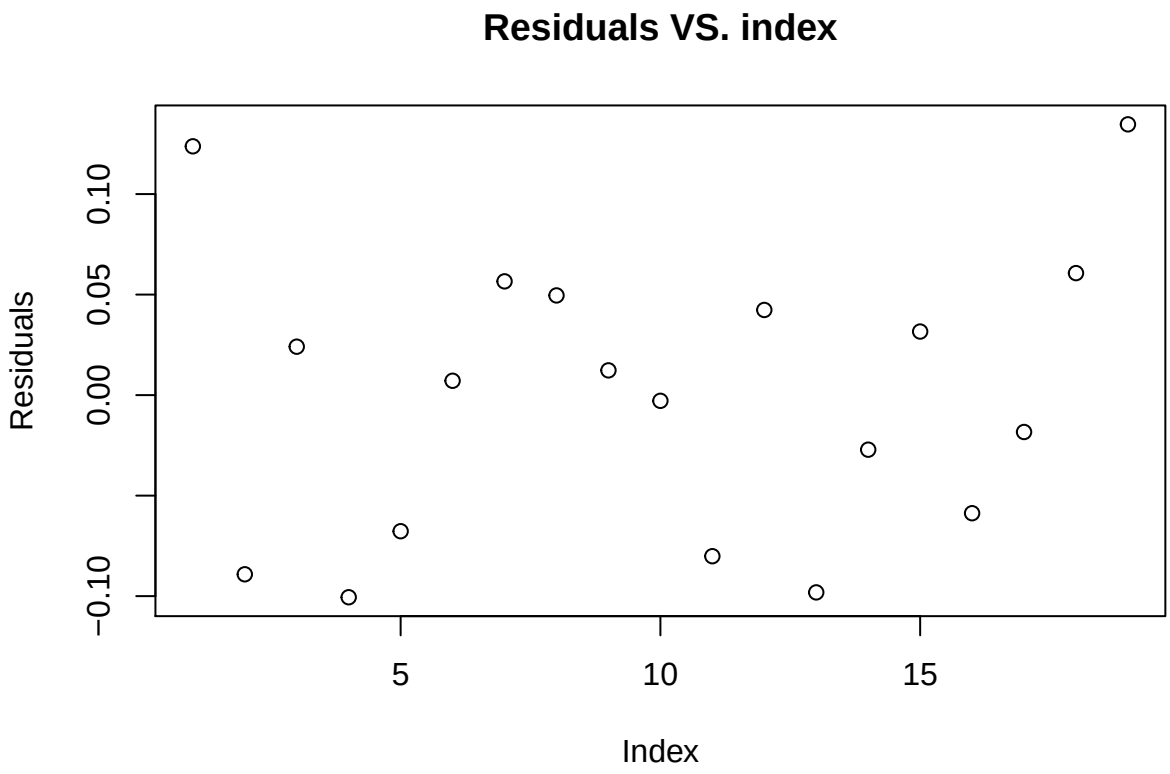

Scale−Location
√|Standardized residuals|
0.0
0.4
0.8
1.2
19
1
13
0.30
0.35
0.40
0.45
0.50
Fitted values
lm(y ~ BodyWt + LiverWt + Dose)



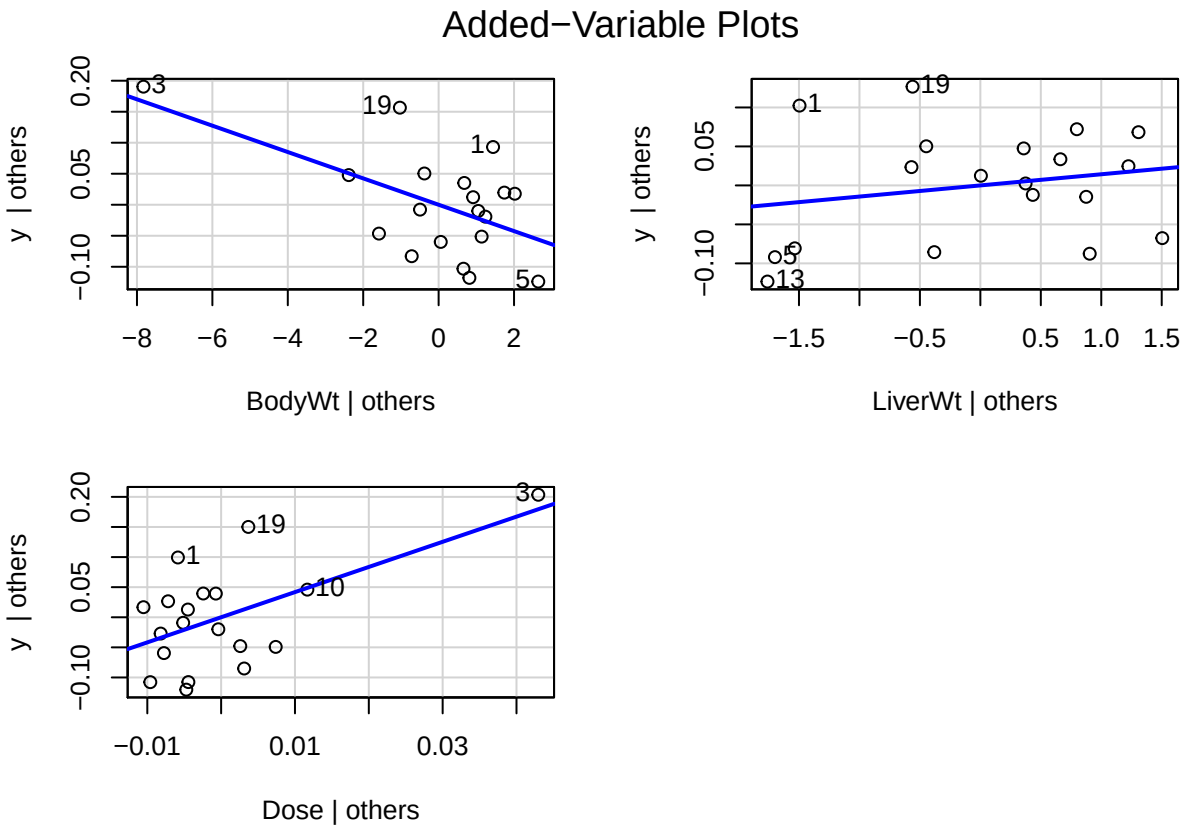

```
plot(pig.lm$fitted.values,rstudent(pig.lm),
     xlab = "fitted values",
     ylab = "studentized residuals",
     main = "residual VS. fitted plot")
```




```r
plot(pig.lm$residuals,
     ylab = "Residuals",
     main = "Residuals VS. index")
```




```
avPlots(pig.lm)
```



From residual vs fitted, and scale-location plot, the point are not distributed around 0, so the assumption of constant variance is violated. From Normal Q-Q plot, we can see most of the point are not fitted on the line, so it violates the normality assumption. From Residuals vs Leverage, we can see point 3 has a high leverage, and outside the cook's distance line which means point 3 is influential point. From partial regression plot, we can see LiverWt is not linear to y. BodyWt and Dose slightly linear to y.

d).

```
# X matrix
X <- cbind(rep(1,19), BodyWt, LiverWt, Dose)
# Hat matrix
H <- X %*% solve(t(X) %*% X) %*% t(X)
# hii
hii <- diag(H)
# Identify points of high Leverage
p<-ncol(X)
n<-nrow(X)
which(hii>2*p/n)
```

```
## [1] 3
```

```
pig_inf <- influence(pig.lm)
sort(pig_inf$hat, decreasing = TRUE)
```

```
##         3         5        13         8        16        17         2
## 0.8509146 0.3915383 0.3161834 0.2536745 0.2166146 0.1952244 0.1793410
##         1        19        12         6        18         7        14
## 0.1779827 0.1779618 0.1723960 0.1611596 0.1487222 0.1368811 0.1314070
##        10        11         4        15         9
## 0.1196867 0.1195058 0.1076158 0.0761748 0.0670158
```

We will consider the point has high leverage if its leverage large than 2*p/n (in this case 0.4210526316), point 3 has leverage higher than this value.

e).

```
pig_cook <- cooks.distance(pig.lm)
sort(pig_cook, decreasing = TRUE)
```

```
##            3           13            5           19            1            2
## 9.296160e-01 2.726019e-01 2.029162e-01 1.999403e-01 1.688268e-01 8.854024e-02
##            4           16            8           11           18            7
## 5.718456e-02 5.099189e-02 4.685795e-02 4.143644e-02 3.162543e-02 2.461564e-02
##           12           14           17           15            9            6
## 1.889847e-02 5.370022e-03 4.249284e-03 3.733265e-03 4.883028e-04 4.874208e-04
##           10
## 5.229549e-05
```

The cook's distance (di) for point 3 is relatively large. From Residuals vs Leverage plot, point 3 is outside the cook's distance line (red dot line) which indicates point 3 is influential point. From the partial regression plot, we observe that point 3 is remote point which indicates high leverage, and we proved this by the calculation of leverage.

f).

```
# VIF
vif(pig.lm)
```

```
##    BodyWt   LiverWt      Dose
## 52.101917  1.335679 51.427154
```

```
# Pairwise correlations
pigx <- cbind(BodyWt,LiverWt,Dose)
cor(pigx)
```

```
##            BodyWt   LiverWt      Dose
## BodyWt  1.0000000 0.5000101 0.9902126
## LiverWt 0.5000101 1.0000000 0.4900711
## Dose    0.9902126 0.4900711 1.0000000
```

```
# eigenvalues
xx <- t(pigx) %*% pigx
(lambda <- eigen(xx)$values)
```

```
## [1] 5.650788e+05 2.049275e+01 2.647914e-03
```

```
# Condition number
max(lambda)/min(lambda)
```

```
## [1] 213405227
```

```
max(lambda)/lambda
```

```
## [1]         1.00     27574.56 213405226.98
```

we observed a strong linear relationship between BodyWt and Dose from paris plot VIF for BodyWt and Dose are large than 10. From pairwise correlations we observed that covariance between BodyWt and Dose is almost 1 which is really high. The eigenvalue for Dose are small, and our condition number is larger than 1000 which indicates we have a multicollinearily problem. Above all, we can conclude that BodyWt and Dose are correlated.

g).

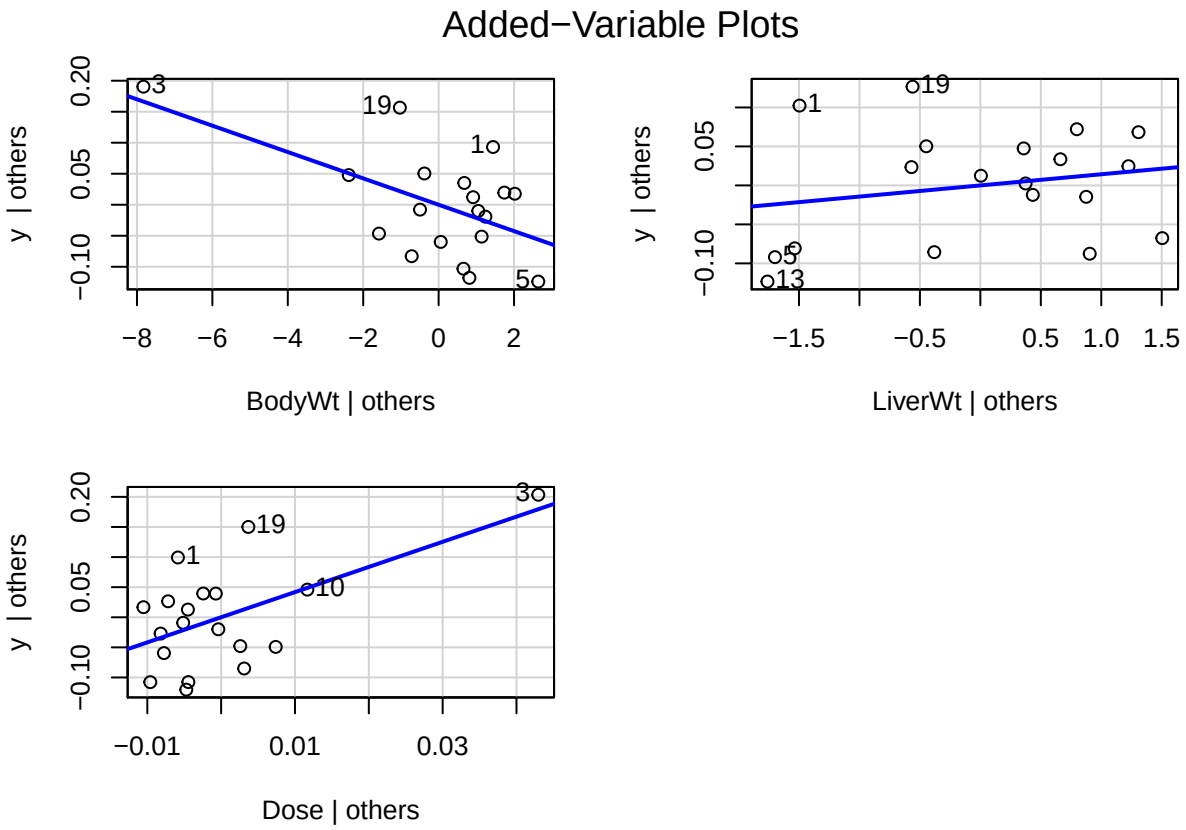
```
pig_new <- filter(pig, Gpig != 3)
pig.lm.new <- lm(data = pig_new,
                 y ~ BodyWt + LiverWt + Dose)
summary(pig.lm.new)
```

```
##
## Call:
## lm(formula = y ~ BodyWt + LiverWt + Dose, data = pig_new)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.102154 -0.056486  0.002838  0.046519  0.137059
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.311427   0.205094   1.518    0.151
## BodyWt      -0.007783   0.018717  -0.416    0.684
## LiverWt      0.008989   0.018659   0.482    0.637
## Dose         1.484877   3.713064   0.400    0.695
##
## Residual standard error: 0.07825 on 14 degrees of freedom
## Multiple R-squared:  0.02106,    Adjusted R-squared:  -0.1887
## F-statistic: 0.1004 on 3 and 14 DF,  p-value: 0.9585
```

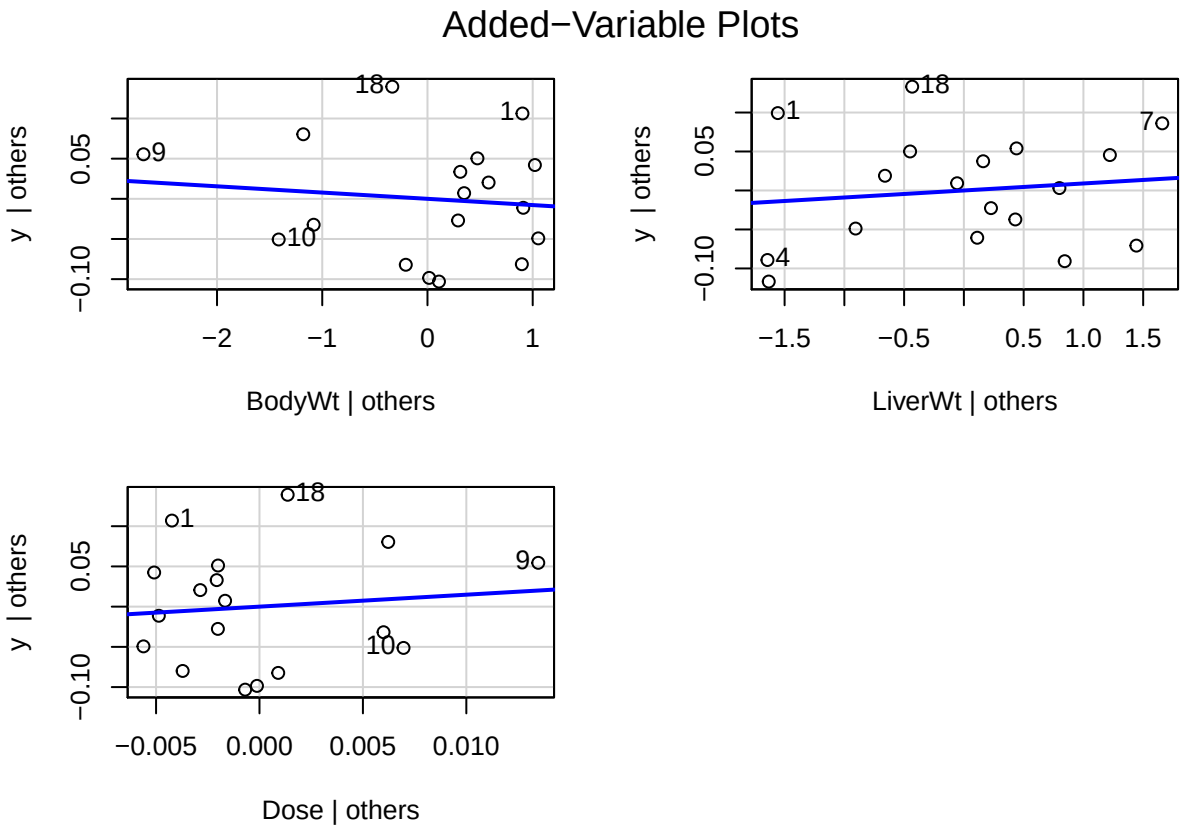
```
summary(pig.lm)
```

```
##
## Call:
## lm(formula = y ~ BodyWt + LiverWt + Dose, data = pig)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.100557 -0.063233  0.007131  0.045971  0.134691
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.265922   0.194585   1.367   0.1919
## BodyWt      -0.021246   0.007974  -2.664   0.0177 *
## LiverWt      0.014298   0.017217   0.830   0.4193
## Dose         4.178111   1.522625   2.744   0.0151 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07729 on 15 degrees of freedom
## Multiple R-squared:  0.3639, Adjusted R-squared:  0.2367
## F-statistic:  2.86 on 3 and 15 DF,  p-value: 0.07197
```

```
avPlots(pig.lm)
```




```
avPlots(pig.lm.new)
```



After refit the model without the most influential observation (point 3). There is no evidence of a significant linear relationship with the response variable, since all the p-values are large. Because point 3 has high leverage which means point 3 is remote point, and it is potential influential point may change the regression coefficient. Moreover point 3 has relatively high cook's distance which indicates point 3 is considered as influential point. We can clearly see the estimated coefficients change from the regression models, and we can observed the significant change from partial regression plot as well especially for variable BodyWt and Dose.