Devoir4-MAT3775

Jonathan Domingue 300246863

2023-12-03

Question 8.3:

```r
library(MPV)

# Create a data frame from the dataset
data <- data.frame(p8.3)

# View the structure of the data frame
str(data)
```

```
## 'data.frame':    25 obs. of  3 variables:
##  $ y : num  16.7 11.5 12 14.9 13.8 ...
##  $ x1: num  7 3 3 4 6 7 2 7 30 5 ...
##  $ x2: num  560 220 340 80 150 330 110 210 1460 605 ...
```

```r
# Create a new column 'city' and assign city names based on observation ranges
data$city <- NA  # Create a new column filled with NAs

# Assign city names based on observation ranges
data$city[1:7] <- "San Diego"
data$city[8:17] <- "Boston"
data$city[18:23] <- "Austin"
data$city[c(24, 25)] <- "Minneapolis"

# View the updated data frame
#data
```

Question a) Faites un Model qui relie y avec les prédicteurs x1, x2 et la variable categorielle city

```r
model1= lm(y ~x1+x2+city , data=data)
summary(model1)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + city, data = data)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4800 -1.5922 -0.5583  1.1045  6.1611
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.036389   1.754025  -0.021  0.98366
## x1              1.770277   0.186790   9.477 1.24e-08 ***
## x2              0.010833   0.003786   2.862  0.00999 **
## cityBoston      4.190275   1.749048   2.396  0.02704 *
## cityMinneapolis 0.452636   2.687420   0.168  0.86803
## citySan Diego   2.737737   1.936269   1.414  0.17356
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.986 on 19 degrees of freedom
## Multiple R-squared:  0.9707, Adjusted R-squared:  0.963
## F-statistic: 125.9 on 5 and 19 DF,  p-value: 6.919e-14
```

Interprétation: Nous pouvons voir que la p-value du predicteur x1 et x3 sont inferieur a 0.05, et qu'ils sont donc statistiquement significatif. Pour la variable categorielle city, nous avons une p-value inferieur a 0.05 uniquement pour cityBoston et non pour cityMinneapolis et citySan Diego

Partie b)

Les coefficients associés aux différentes villes indiquent des variations dans le délai de livraison par rapport à la ville de référence (probablement la référence dans votre modèle).

La p-value associée à la variable "city" dans le cas de "cityMinneapolis" est élevée (0.86803), ce qui suggère que le coefficient pour "cityMinneapolis" n'est pas statistiquement significatif. Cela signifie qu'il n'y a pas suffisamment de preuves pour affirmer que le lieu de livraison à Minneapolis a un effet significatif sur le délai de livraison dans le modèle actuel.

Pour la ville de "San Diego" (0.17356) nous avons egalement une p-value superieur a 0.05.

En revanche, les p-values pour la villes de "Boston" (0.02704) indiquant une signification statistique pour ces deux villes.

En conclusion, il y a des indications que le lieu de livraison est une variable importante dans le modèle, mais cela dépend de la ville spécifique. Les résultats suggèrent que le délai de livraison peut varier de manière significative en fonction de la ville de livraison, en particulier pour Boston.

Cependant, il est essentiel de considérer le contexte spécifique de l'étude, la taille de l'échantillon et d'autres facteurs potentiels non inclus dans le modèle pour une interprétation complète.

Partie c)

```r
donnees_diag <- broom::augment(model1)

p1 <- ggplot(donnees_diag, aes(.fitted, .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
```

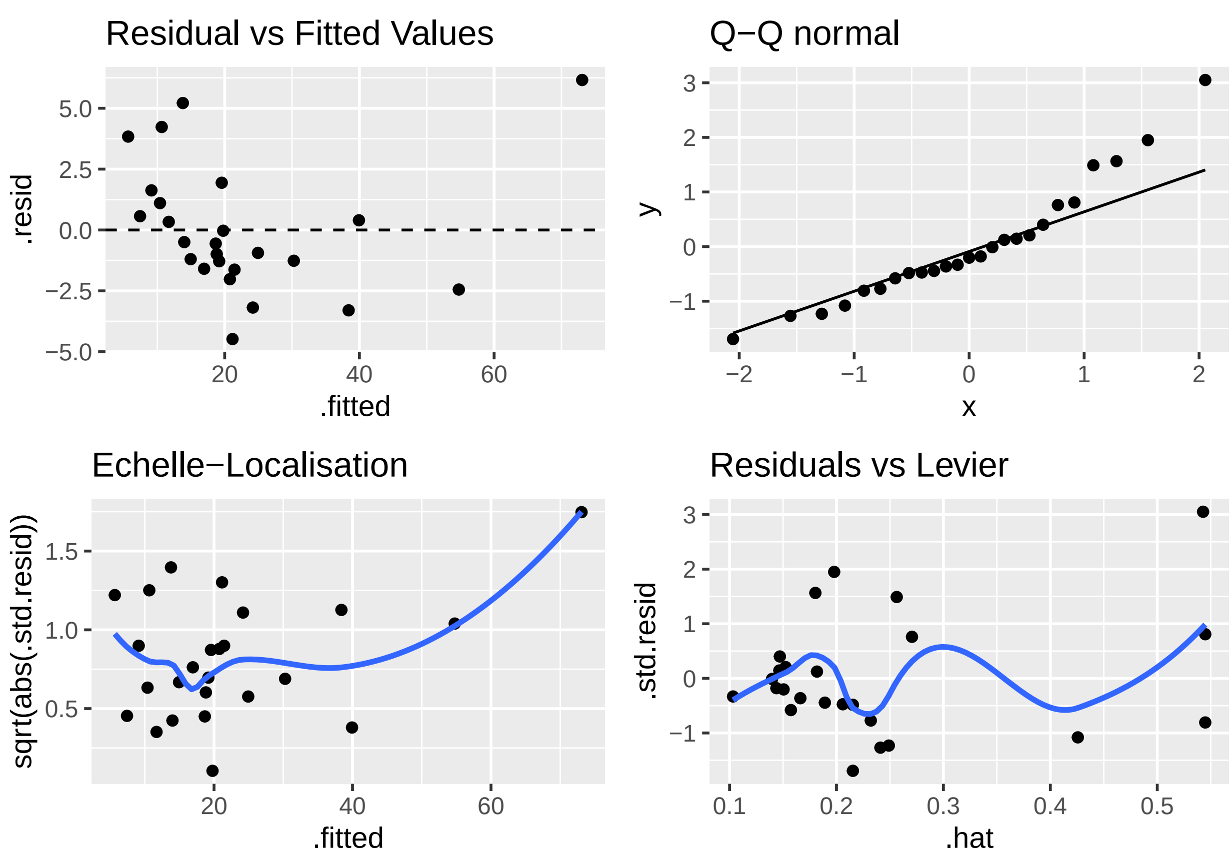
```
  labs(title = "Residual vs Fitted Values")

p2 <- ggplot(donnees_diag, aes(sample = .std.resid)) +
  stat_qq() +
  stat_qq_line() +
  labs(title = "Q-Q normal")

p3 <- ggplot(donnees_diag, aes(.fitted, sqrt(abs(.std.resid)))) +
  geom_point() +
  geom_smooth(se = FALSE) +
  labs(title = "Echelle-Localisation")

p4 <- ggplot(donnees_diag, aes(.hat, .std.resid)) +
  geom_point() +
  geom_smooth(se = FALSE) +
  labs(title = "Residuals vs Levier")

grid.arrange(p1, p2, p3, p4, ncol = 2)
```



Interprétation : Dans le graphique de residuals vs fitted values, on voit que la variance des residus n'est pas constante. Cela nous demontre que nous avons un cas de heteroscedacite.

De plus, le graphique de echelle-localisation nous montre que les residus ne sont pas repartis egalement sur les gammes de predicteur.

Conclusion: Nous voyons que nous pourrions faire une transformation de Box Cox pour obtenir un meilleur model.

Question 9.17

```r
#Loading the data
data = data.frame(cement)
data
```

```
##     X     y x1 x2 x3 x4
## 1   1  78.5  7 26  6 60
## 2   2  74.3  1 29 15 52
## 3   3 104.3 11 56  8 20
## 4   4  87.6 11 31  8 47
## 5   5  95.9  7 52  6 33
## 6   6 109.2 11 55  9 22
## 7   7 102.7  3 71 17  6
## 8   8  72.5  1 31 22 44
## 9   9  93.1  2 54 18 22
## 10 10 115.9 21 47  4 26
## 11 11  83.8  1 40 23 34
## 12 12 113.3 11 66  9 12
## 13 13 109.4 10 68  8 12
```

Partie a: Utiliser une Regression Ridge pour selectioner une valeur de k.

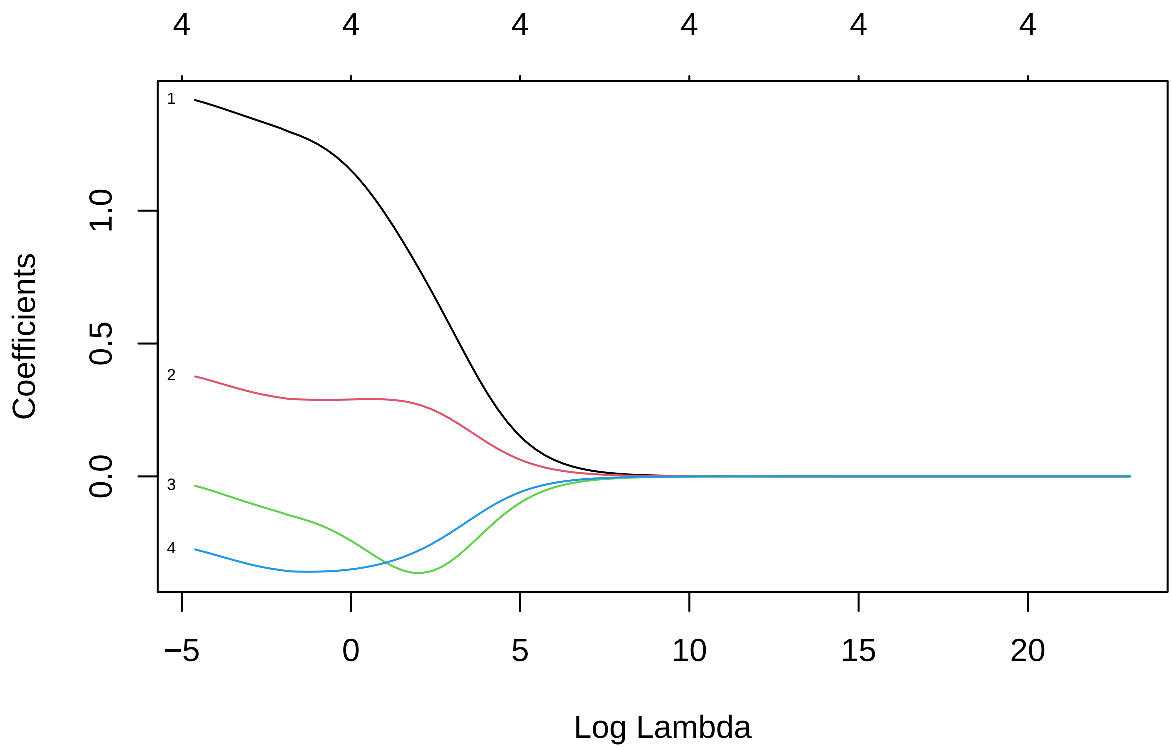
```r
# Extracting data
response_col <- 2
predictor_cols <- 3:6

x <- as.matrix(data[, predictor_cols])
y <-  data[, response_col]

# Adjust the Ridge regression model on a grid of lambda values
grid <- 10^seq(10, -2, length = 100) # grid of lambda values
ridge_mod <- glmnet(x, y, alpha = 0, lambda = grid)
```

Imprimer la Trace de Ridge:

```r
plot(ridge_mod, xvar = "lambda", label = TRUE)
```

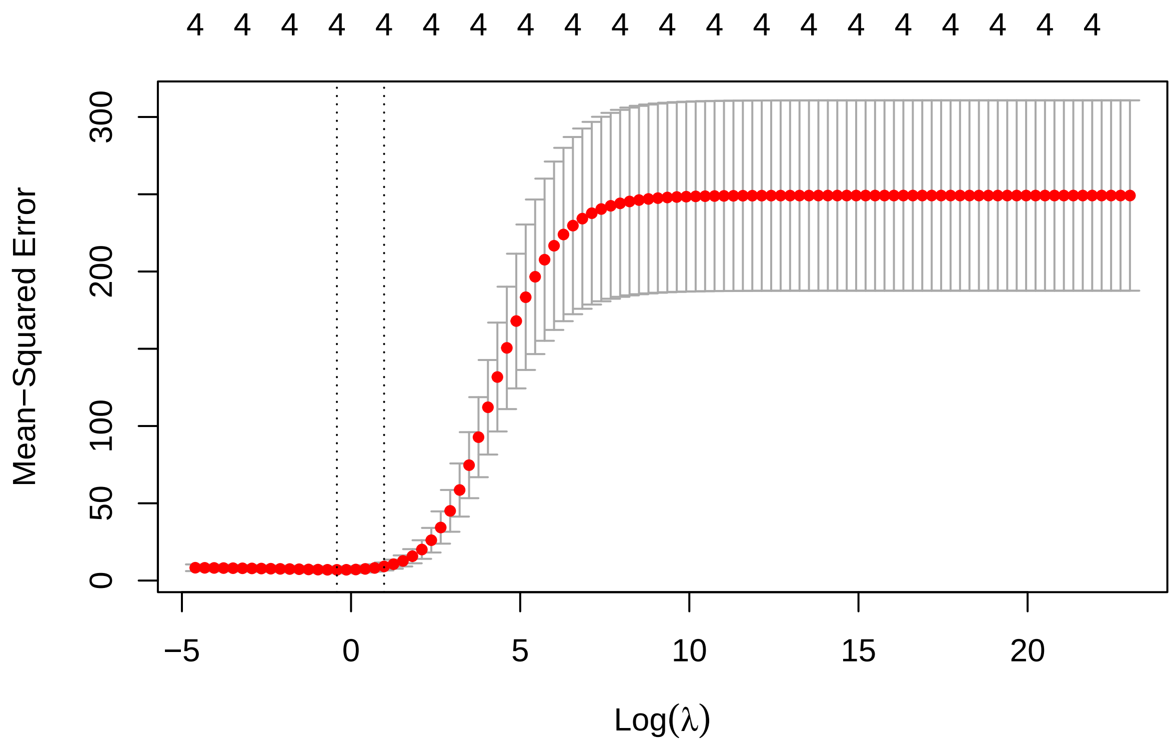


Selection du Model avec la validation croise:

```r
# Sélectionnez une valeur pour k (lambda) en fonction du tracé
# Effectuer une validation croisée
cv_ridge <- cv.glmnet(x, y, alpha = 0, lambda = grid)
```

```
## Warning: Option grouped=FALSE enforced in cv.glmnet, since < 3 observations per
## fold
```

```r
#Tracer la courbe de validation croisée
plot(cv_ridge)
```



Selection du Meilleur Lambda:

```
# Sélectionnez lambda qui minimise l'erreur de validation croisée
selected_lambda <- cv_ridge$lambda.min
selected_lambda
```

```
## [1] 0.6579332
```

Nous voyons que le Lambda qui semble minimiser le MSE est de 0.284

##Test de l'Adéquation de Notre Modele

```
# Ajuster le modèle final avec la valeur sélectionnée de lambda
final_ridge <- glmnet(x, y, alpha = 0, lambda = selected_lambda)
# Donne les valeurs estimées des coeficients
coef(final_ridge)
```

```
## 5 x 1 sparse Matrix of class "dgCMatrix"
##                     s0
## (Intercept) 85.4424616
## x1           1.2043345
## x2           0.2912842
## x3          -0.2090183
## x4          -0.3524014
```

##Comparaison des RSS

```r
# Ajuster un modèle OLS à des fins de comparaison
ols_mod <- lm(y ~ .,  data =  data)
ols_rss <- sum(resid(ols_mod)^2)
# Calculer RSS pour le modèle de régression de ridge
ridge_predictions <- predict(final_ridge, s = selected_lambda, newx = x)
ridge_rss <- sum((y - ridge_predictions)^2)
# Calculate inflation in RSS
rss_inflation <- ridge_rss-ols_rss
cat("Le RSS du OLS", ols_rss,"\n")
```

```
## Le RSS du OLS 47.85161
```

```r
cat("Le RSS du Ridge", ridge_rss,"\n")
```

```
## Le RSS du Ridge 51.29356
```

```r
cat("L'Inflation", rss_inflation,"\n")
```

```
## L'Inflation 3.441953
```

Reduction du R carre

```r
# R au carré pour le modèle OLS
ols_r2 <- summary(ols_mod)$r.squared
# R-carré pour le modèle de régression de ridge
ridge_r2 <- 1 - (ridge_rss / sum((y - mean(y))^2))
# Calculer la réduction du R-carré
r2_reduction <- ols_r2 - ridge_r2
cat("La Reduction du R carre est", r2_reduction,"\n")
```

```
## La Reduction du R carre est 0.001267398
```

Partie c: Comparaison entre Model de Regression Ridge avec le Model avec le model a deux Regresseurs impliquant x1 et x2

```r
## On va changer x  a x2 , y  a y2

predictor_cols <- 3:4
response_col <- 2

x2 <- as.matrix(data[, predictor_cols])
y2 <-  data[, response_col]

model2<- lm( y2~ data$x1 + data$x2 , data=data)

summary(model2)
```

```
##
## Call:
## lm(formula = y2 ~ data$x1 + data$x2, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -2.893 -1.574 -1.302  1.363  4.048
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 52.57735    2.28617   23.00 5.46e-10 ***
## data$x1      1.46831    0.12130   12.11 2.69e-07 ***
## data$x2      0.66225    0.04585   14.44 5.03e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.406 on 10 degrees of freedom
## Multiple R-squared:  0.9787, Adjusted R-squared:  0.9744
## F-statistic: 229.5 on 2 and 10 DF,  p-value: 4.407e-09
```

##Partie c)Comparez le modèle de régression ridge avec le modèle à deux régresseurs impliquant x1 et x2

Avec OLS avec deux regresseurs nous obtenons les coefficients suivants pour les estimateurs:

beta de x1= 1.47 beta de x2=0.66

Avec Ridge nous avions obtenu les coefficients des estimateurs suivants:

beta de x1=1.2793015 beta de x2=0.2997523

On voit que les valeurs n'ont pas autant ete diminue par la penalite. cela demontre etre robuste et qui nous fait savoir que les deux regresseurs sont important pour expliquer le model. x1 et x2 met en resilience les coefficients de x1 et x2 face a la regularisation.

C'est une indication que ces variables ont une contribution significative et robuste à la relation avec la variable dépendante.

Question 10.4: Examination des donnees de l'energie solaire thermique

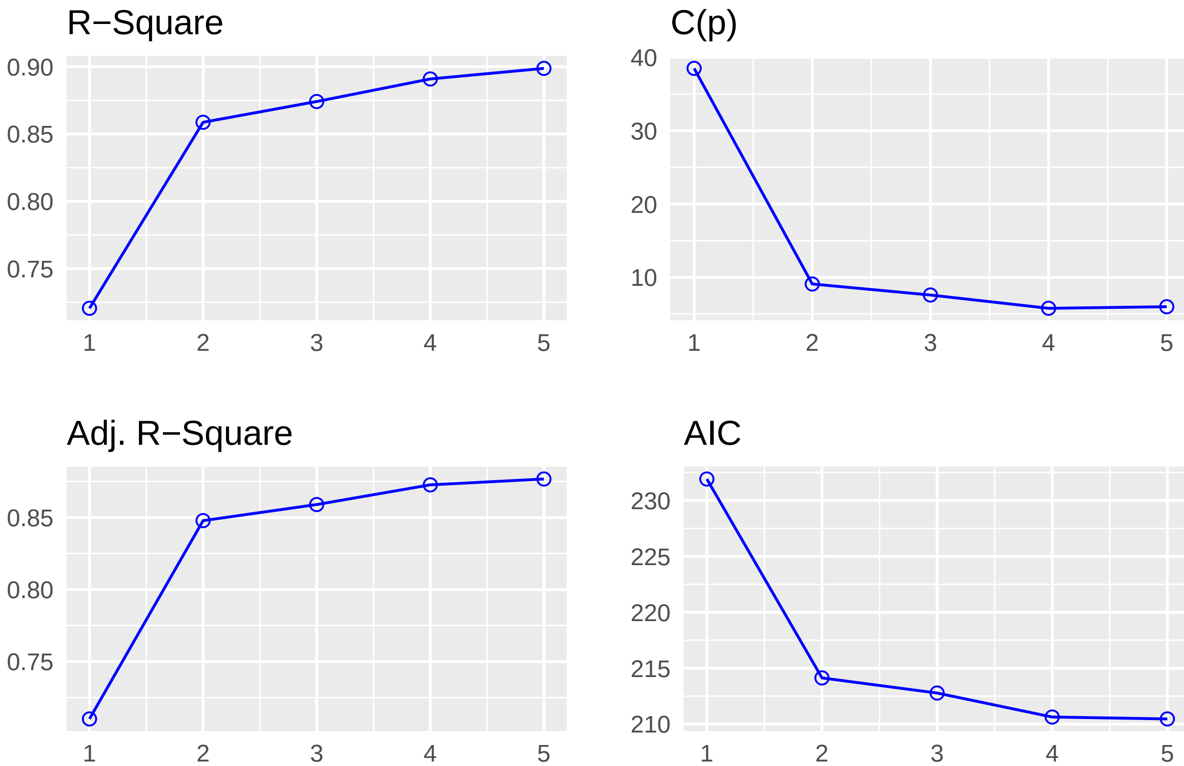
```r
data <- data.frame(table.b2)
data
```

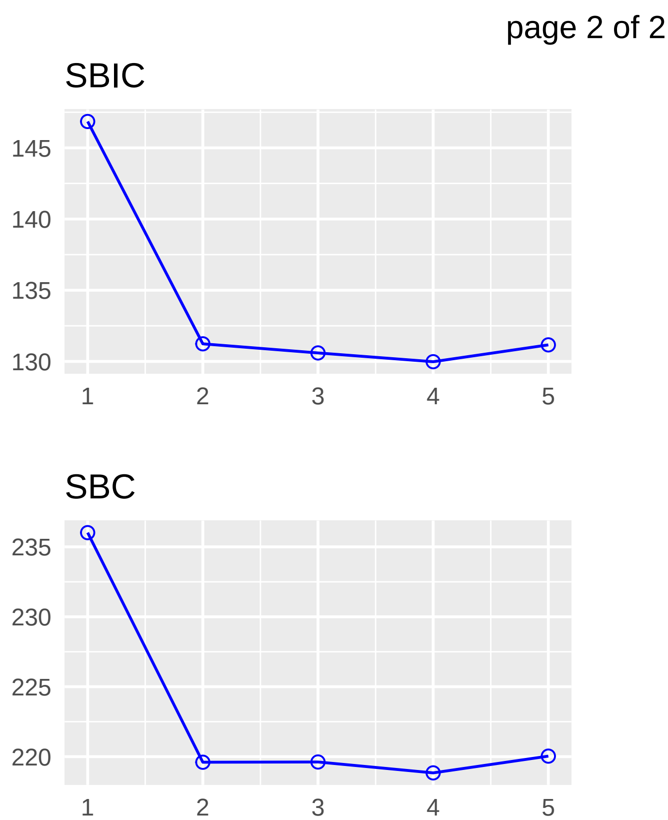
```
##         y     x1    x2    x3    x4    x5
## 1  271.8 783.35 33.53 40.55 16.66 13.20
## 2  264.0 748.45 36.50 36.19 16.46 14.11
## 3  238.8 684.45 34.66 37.31 17.66 15.68
## 4  230.7 827.80 33.13 32.52 17.50 10.53
## 5  251.6 860.45 35.75 33.71 16.40 11.00
## 6  257.9 875.15 34.46 34.14 16.28 11.31
## 7  263.9 909.45 34.60 34.85 16.06 11.96
## 8  266.5 905.55 35.38 35.89 15.93 12.58
## 9  229.1 756.00 35.85 33.53 16.60 10.66
```

```
## 10 239.3 769.35 35.68 33.79 16.41 10.85
## 11 258.0 793.50 35.35 34.72 16.17 11.41
## 12 257.6 801.65 35.04 35.22 15.92 11.91
## 13 267.3 819.65 34.07 36.50 16.04 12.85
## 14 267.0 808.55 32.20 37.60 16.19 13.58
## 15 259.6 774.95 34.32 37.89 16.62 14.21
## 16 240.4 711.85 31.08 37.71 17.37 15.56
## 17 227.2 694.85 35.73 37.00 18.12 15.83
## 18 196.0 638.10 34.11 36.76 18.53 16.41
## 19 278.7 774.55 34.79 34.62 15.54 13.10
## 20 272.3 757.90 35.77 35.40 15.70 13.63
## 21 267.4 753.35 36.44 35.96 16.45 14.51
## 22 254.5 704.70 37.82 36.26 17.62 15.38
## 23 224.7 666.80 35.07 36.34 18.12 16.10
## 24 181.5 568.55 35.26 35.90 19.05 16.73
## 25 227.5 653.10 35.56 31.84 16.51 10.58
## 26 253.6 704.05 35.73 33.16 16.02 11.28
## 27 263.0 709.60 36.46 33.83 15.89 11.91
## 28 265.8 726.90 36.26 34.89 15.83 12.65
## 29 263.8 697.15 37.20 36.27 16.71 14.06
```

Selection vers l'avant pour model de regression sous-ensemble

```r
model <- lm(y ~ . , data = data)
k <- ols_step_forward_p(model)
plot(k)
```


R−Square
0.90
0.85
0.80
0.75
1
2
3
4
5
C(p)
40
30
20
10
1
2
3
4
5
Adj. R−Square
0.85
0.80
0.75
1
2
3
4
5
AIC
230
225
220
215
210
1
2
3
4
5




```r
ols_step_forward_p(model,details = FALSE)
```

```
##
##                              Selection Summary
## ---------------------------------------------------------------------------
##         Variable                  Adj.
## Step    Entered     R-Square    R-Square     C(p)         AIC        RMSE
## ---------------------------------------------------------------------------
##    1    x4            0.7205      0.7102    38.4923    231.9133    12.3277
##    2    x3            0.8587      0.8478     9.0975    214.1313     8.9321
##    3    x2            0.8741      0.8590     7.5963    212.7817     8.5978
##    4    x1            0.8909      0.8727     5.7873    210.6363     8.1698
##    5    x5            0.8988      0.8768     6.0000    210.4660     8.0390
## ---------------------------------------------------------------------------
```

Nous pouvons observer que tous les predicteurs ont ete choisis lorsque nous avons fais la selection vers l'avant.

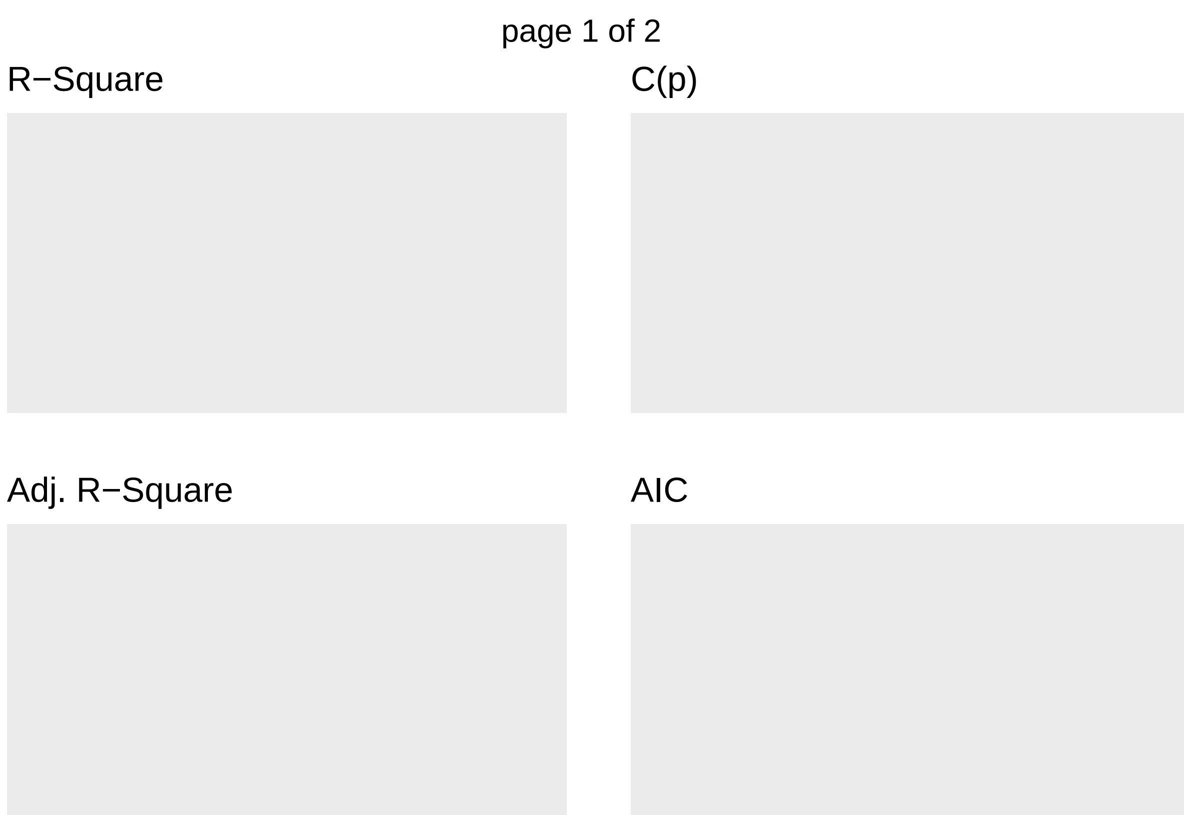
```r
summary(model)
```

```
##
## Call:
## lm(formula = y ~ ., data = data)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -13.6848  -2.7688   0.6273   3.9166  17.3962
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 325.43612   96.12721   3.385  0.00255 **
## x1            0.06753    0.02899   2.329  0.02900 *
## x2            2.55198    1.24824   2.044  0.05252 .
## x3            3.80019    1.46114   2.601  0.01598 *
## x4          -22.94947    2.70360  -8.488 1.53e-08 ***
## x5            2.41748    1.80829   1.337  0.19433
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.039 on 23 degrees of freedom
## Multiple R-squared:  0.8988, Adjusted R-squared:  0.8768
## F-statistic: 40.84 on 5 and 23 DF,  p-value: 1.077e-10
```

Partie b: Selection vers l'arriere

```r
k2=ols_step_backward_p(model)
plot(k2)
```



page 2 of 2

SBIC

SBC

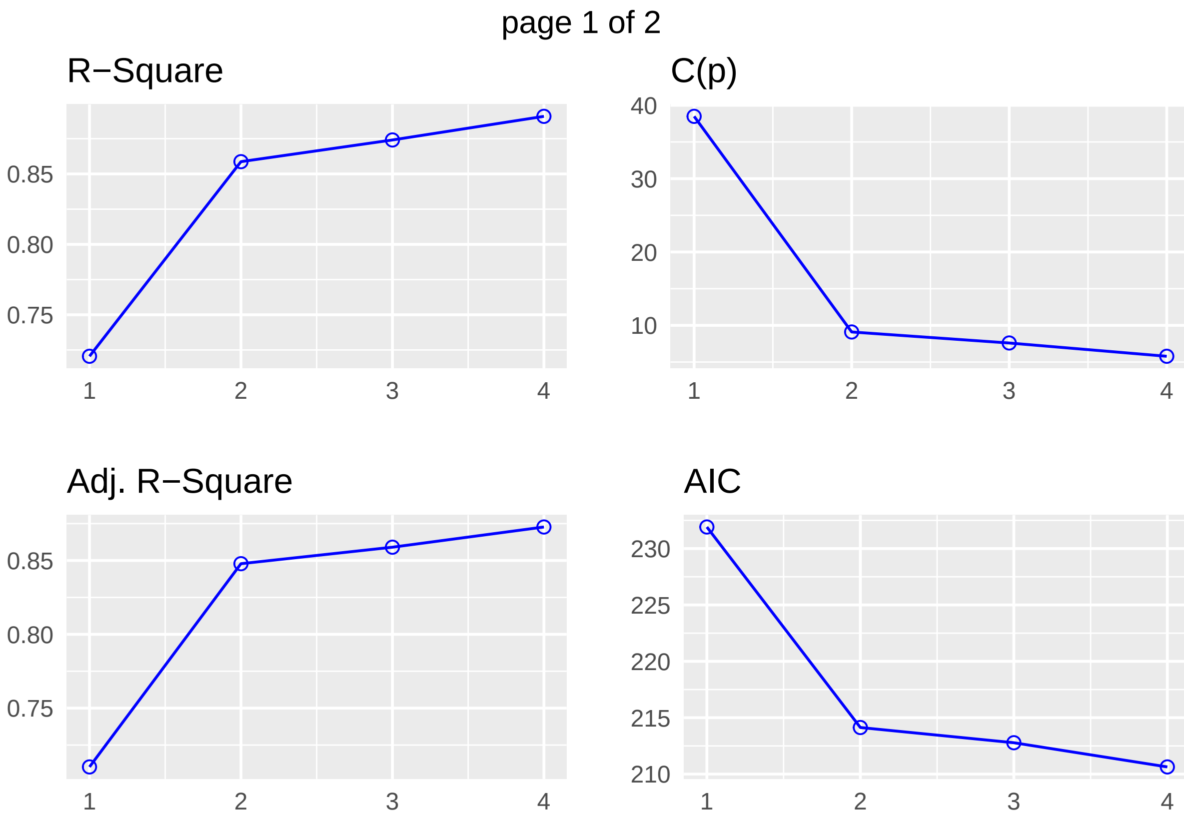

```r
ols_step_backward_p(model, details=FALSE)
```

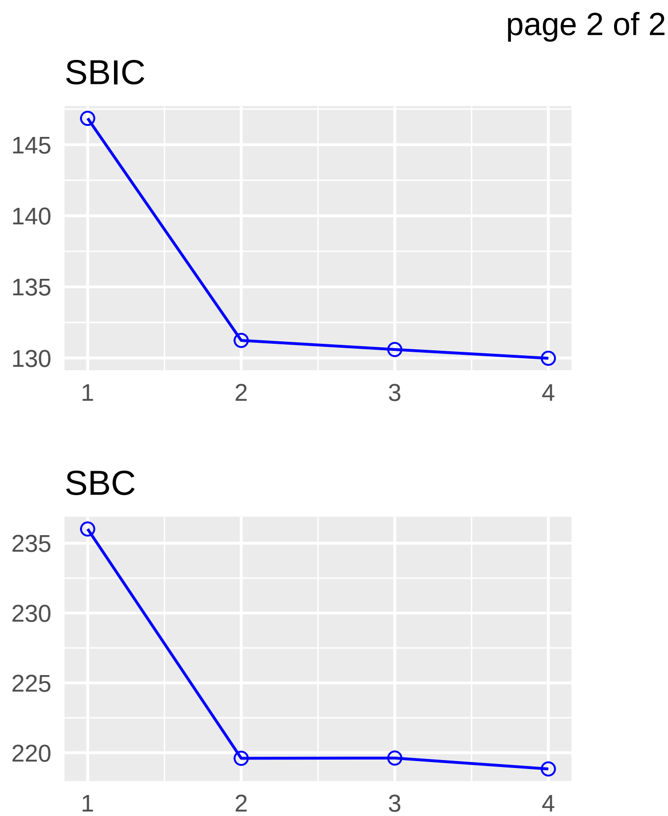
```
## [1] "No variables have been removed from the model."
```

Nous pouvons constater que la step backward approach n'a pas enleve aucun des predicteurs non plus

Partie c: Selection par Etapes

```r
k3=ols_step_both_p(model)
plot(k3)
```


page 1 of 2
R−Square
C(p)
Adj. R−Square
AIC




```r
ols_step_both_p(model, details = FALSE)
```

```
##
##                               Stepwise Selection Summary
## -------------------------------------------------------------------------------------
##                      Added/                  Adj.
## Step    Variable    Removed     R-Square    R-Square     C(p)        AIC        RMSE
## -------------------------------------------------------------------------------------
##    1       x4       addition       0.721       0.710    38.4920    231.9133    12.3277
##    2       x3       addition       0.859       0.848     9.0980    214.1313     8.9321
##    3       x2       addition       0.874       0.859     7.5960    212.7817     8.5978
##    4       x1       addition       0.891       0.873     5.7870    210.6363     8.1698
## -------------------------------------------------------------------------------------
```

Nous avons les predicteurs/regresseurs x1, x2 , x3 et x4 qui ont ete selectionnes pour faire le Model.Le predicteur x5 a ete enleve lors de la selection par etapes.

Nous pouvons faire un summary lors que nous avons les regresseurs x1,x2,x3 et x4

```r
model2= lm(y~x1+ x2+ x3+ x4 , data=data)
summary(model2)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4, data = data)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.322  -2.639   0.025   4.786  16.003
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 270.21013   88.21060   3.063  0.00534 **
## x1            0.05156    0.02685   1.920  0.06676 .
## x2            2.95141    1.23167   2.396  0.02471 *
## x3            5.33861    0.91506   5.834 5.13e-06 ***
## x4          -21.11940    2.36936  -8.914 4.42e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.17 on 24 degrees of freedom
## Multiple R-squared:  0.8909, Adjusted R-squared:  0.8727
## F-statistic: 48.99 on 4 and 24 DF,  p-value: 3.327e-11
```

Partie c: Tous les Models Possible

```r
ols_step_all_possible(model)
```

```
##    Index N     Predictors   R-Square Adj. R-Square Mallow's Cp
## 4      1 1             x4 0.72052420    0.71017324   38.492277
## 1      2 1             x1 0.39393873    0.37149202  112.687090
## 5      3 1             x5 0.12328631    0.09081543  174.174842
## 3      4 1             x3 0.01256447   -0.02400721  199.329010
## 2      5 1             x2 0.01047594   -0.02617310  199.803490
## 13     6 2          x3 x4 0.85871542    0.84784738    9.097518
## 15     7 2          x4 x5 0.82020641    0.80637613   17.846129
## 8      8 2          x1 x4 0.73386067    0.71338841   37.462451
## 11     9 2          x2 x4 0.72053206    0.69903453   40.490491
## 6     10 2          x1 x2 0.44922541    0.40685814  102.126871
## 7     11 2          x1 x3 0.42636430    0.38223847  107.320539
## 9     12 2          x1 x5 0.39429011    0.34769704  114.607262
## 14    13 2          x3 x5 0.37034618    0.32191127  120.046927
## 12    14 2          x2 x5 0.12963616    0.06268509  174.732260
## 10    15 2          x2 x3 0.03427915   -0.04000706  196.395794
## 22    16 3       x2 x3 x4 0.87412678    0.85902200    7.596312
## 19    17 3       x1 x3 x4 0.86478901    0.84856370    9.717698
## 25    18 3       x3 x4 x5 0.86376234    0.84741382    9.950942
## 21    19 3       x1 x4 x5 0.86288638    0.84643274   10.149946
## 24    20 3       x2 x4 x5 0.82022133    0.79864789   19.842738
## 17    21 3       x1 x2 x4 0.73615340    0.70449181   38.941581
## 16    22 3       x1 x2 x3 0.52969885    0.47326272   85.844637
## 20    23 3       x1 x3 x5 0.46570209    0.40158634  100.383642
## 23    24 3       x2 x3 x5 0.46085816    0.39616114  101.484104
## 18    25 3       x1 x2 x5 0.45487844    0.38946385  102.842597
## 26    26 4    x1 x2 x3 x4 0.89089317    0.87270870    5.787266
## 29    27 4    x1 x3 x4 x5 0.88036169    0.86042197    8.179844
## 30    28 4    x2 x3 x4 x5 0.87488338    0.85403061    9.424426
```

```
## 28    29 4    x1 x2 x4 x5 0.86898542    0.84714966   10.764344
## 27    30 4    x1 x2 x3 x5 0.58159740    0.51186363   76.054147
## 31    31 5 x1 x2 x3 x4 x5 0.89876023    0.87675159    6.000000
```

On peut clairement voire que ce serait mieux d'inclure tous les predicteurs

Parte e: Comparaison des Analyses faites en Partie a et c

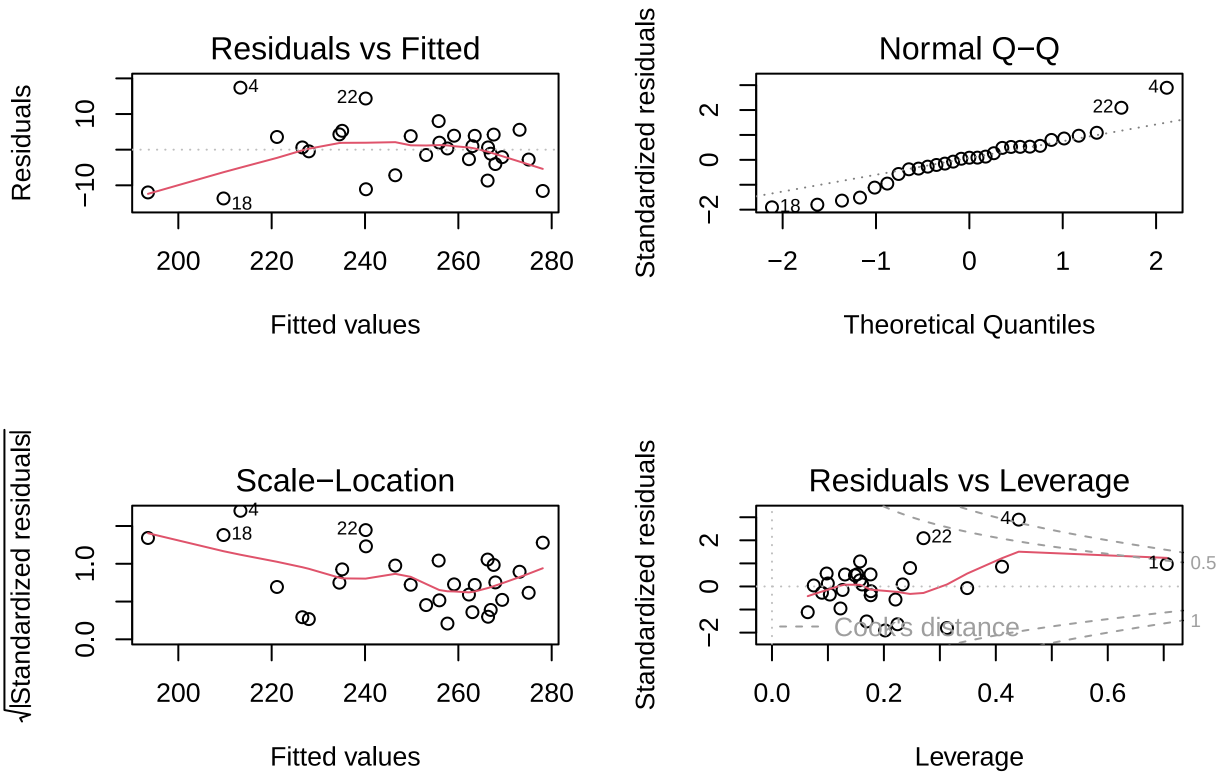
Analyse du Premier Model:

```r
summary(model)
```

```
##
## Call:
## lm(formula = y ~ ., data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.6848  -2.7688   0.6273   3.9166  17.3962
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 325.43612   96.12721   3.385  0.00255 **
## x1            0.06753    0.02899   2.329  0.02900 *
## x2            2.55198    1.24824   2.044  0.05252 .
## x3            3.80019    1.46114   2.601  0.01598 *
## x4          -22.94947    2.70360  -8.488 1.53e-08 ***
## x5            2.41748    1.80829   1.337  0.19433
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.039 on 23 degrees of freedom
## Multiple R-squared:  0.8988, Adjusted R-squared:  0.8768
## F-statistic: 40.84 on 5 and 23 DF,  p-value: 1.077e-10
```

Graphique pour analyser comportement des Residus:

```r
par(mfrow=c(2,2))
plot(model)
```



Diagnostique de Multicollinearite:

```
ols_coll_diag(model)
```

```
## Tolerance and Variance Inflation Factor
## ---------------------------------------
##   Variables Tolerance      VIF
## 1        x1 0.4312139 2.319035
## 2        x2 0.7377635 1.355448
## 3        x3 0.3148644 3.175970
## 4        x4 0.3828387 2.612066
## 5        x5 0.1862177 5.370059
##
##
## Eigenvalue and Condition Index
## ------------------------------
##     Eigenvalue Condition Index    intercept           x1           x2
## 1 5.9668908760         1.00000 6.767793e-06 0.0001277963 3.247665e-05
## 2 0.0254639495        15.30774 1.421387e-04 0.0781510580 6.706206e-04
## 3 0.0048957271        34.91125 3.114582e-03 0.2626084537 7.714844e-02
## 4 0.0015969065        61.12717 5.777198e-04 0.0020179978 2.721602e-01
## 5 0.0009778935        78.11389 8.735064e-03 0.5618754807 2.816480e-02
## 6 0.0001746475       184.83871 9.874237e-01 0.0952192134 6.218234e-01
##             x3           x4           x5
## 1 2.313969e-05 0.0000305478 0.0001056611
## 2 7.705420e-05 0.0007613186 0.0776507788
```

```
## 3 1.065040e-02 0.0260365246 0.1182825202
## 4 4.500398e-04 0.3935994846 0.1026553078
## 5 5.716067e-01 0.0384692395 0.4264493421
## 6 4.171927e-01 0.5411028849 0.2748563901
```

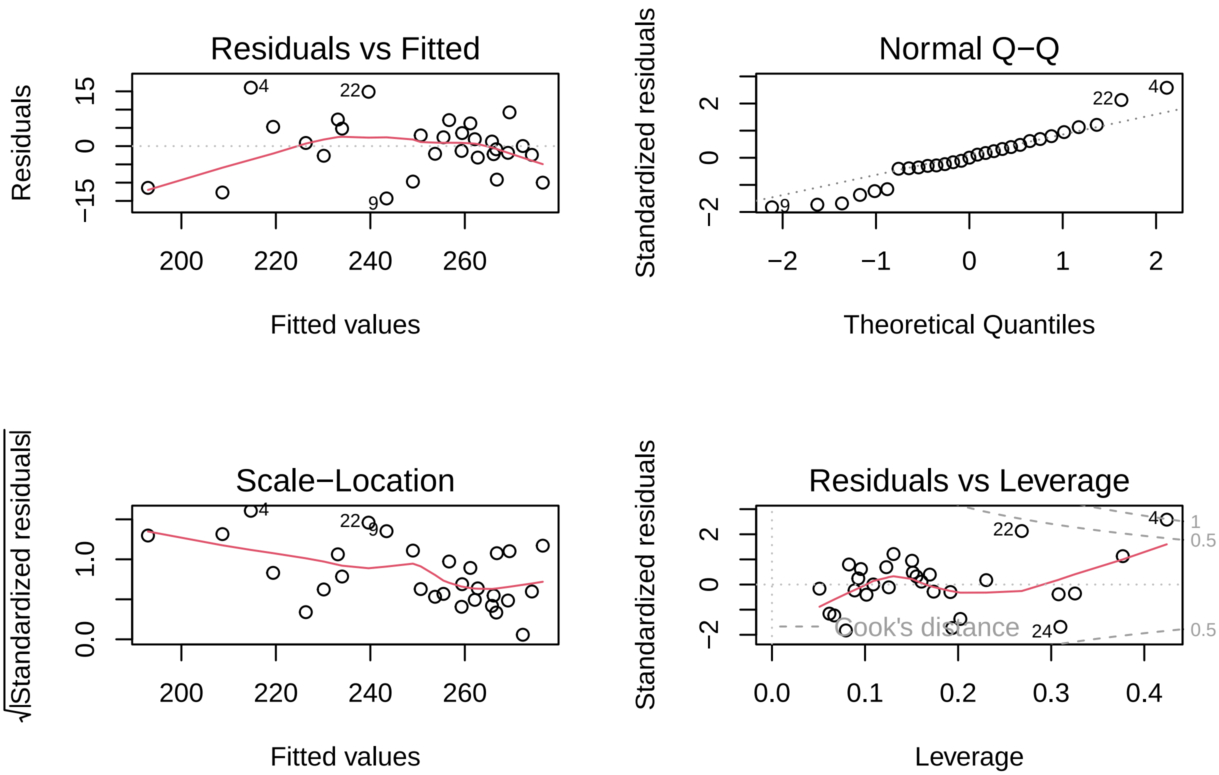
Maintenant on va faire la meme analyse pour notre model2, celui avec tout les predicteurs sauf x5

Analyse:

```r
summary(model2)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.322  -2.639   0.025   4.786  16.003
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 270.21013   88.21060   3.063  0.00534 **
## x1            0.05156    0.02685   1.920  0.06676 .
## x2            2.95141    1.23167   2.396  0.02471 *
## x3            5.33861    0.91506   5.834 5.13e-06 ***
## x4          -21.11940    2.36936  -8.914 4.42e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.17 on 24 degrees of freedom
## Multiple R-squared:  0.8909, Adjusted R-squared:  0.8727
## F-statistic: 48.99 on 4 and 24 DF,  p-value: 3.327e-11
```

```r
par(mfrow=c(2,2)) #Residuals
plot(model2)
```




```r
ols_coll_diag(model2)
```

```
## Tolerance and Variance Inflation Factor
## ---------------------------------------
##   Variables Tolerance      VIF
## 1        x1 0.5192672 1.925791
## 2        x2 0.7825999 1.277792
## 3        x3 0.8291484 1.206057
## 4        x4 0.5148215 1.942421
##
##
## Eigenvalue and Condition Index
## ------------------------------
##     Eigenvalue Condition Index    intercept           x1           x2
## 1 4.9838045250         1.00000 1.190499e-05 0.0002214391 4.941464e-05
## 2 0.0117219692        20.61960 1.565581e-04 0.3478225865 2.004681e-03
## 3 0.0028524425        41.79959 2.043824e-03 0.0132812661 2.301331e-01
## 4 0.0013918322        59.83938 5.964702e-05 0.1576982611 1.097120e-01
## 5 0.0002292311       147.44964 9.977281e-01 0.4809764471 6.581008e-01
##             x3           x4
## 1 8.725762e-05 5.881296e-05
## 2 6.236713e-03 2.577810e-02
## 3 3.196264e-01 1.020974e-02
## 4 5.140050e-01 5.333921e-01
## 5 1.600446e-01 4.305613e-01
```

Nous obtenons plus ou moins les memes resultats avec les deux differents models On remarque aussi que dans les deux cas nous n'avons pas la normalite des residus

Dans le diagramme quantile quantile aussi on voit que tous les points ne sont pas plus ou moins sur la ligne. Il nya donc pas la normalite.

VIF et Tolerance:

Dans les deux modèles, les variables ont des tolérances inférieures à 1, indiquant une multicollinéarité potentielle. En général, des tolérances plus faibles et des valeurs de VIF plus élevées peuvent suggérer une multicollinéarité plus élevée.

Valeurs Propres: Les deux modèles ont des valeurs propres indiquant un certain degré de multicollinéarité

Comparaison: L'inclusion de x5 dans la partie d affecte les diagnostics de la multicollinéarité, avec un VIF plus élevé pour x5.

Question 13.3

```
df =data.frame(p13.3)
df
```

```
##       x   n  r
## 1  2500  50 10
## 2  2700  70 17
## 3  2900 100 30
## 4  3100  60 21
## 5  3300  40 18
## 6  3500  85 43
## 7  3700  90 54
## 8  3900  50 33
## 9  4100  80 60
## 10 4300  65 51
```

Partie a:

```
x_load = df$x
n_sampleSize = df$n
rNumberFailing = df$r
fail_per_sample = rNumberFailing / n_sampleSize
logistic_model = glm(fail_per_sample ~ x_load, family = "binomial", weights = n_sampleSize)
logistic_model
```

```
##
## Call:  glm(formula = fail_per_sample ~ x_load, family = "binomial",
##     weights = n_sampleSize)
##
## Coefficients:
## (Intercept)       x_load
##   -5.339712     0.001548
##
```

```
## Degrees of Freedom: 9 Total (i.e. Null);  8 Residual
## Null Deviance:       112.8
## Residual Deviance: 0.3719    AIC: 49.09
```

Partie b: Le model est-il adequat

Oui le model est adequat. dans la question precedente on avait obtenu 0.3719 comme valeur pour le residual deviance qui est tres bas.

Partie c:

```
quad_term = x_load^2
quadratic_logistic_model = glm(fail_per_sample ~ x_load + quad_term, family = "binomial", weights = n_sa
quadratic_logistic_model
```

```
##
## Call:  glm(formula = fail_per_sample ~ x_load + quad_term, family = "binomial",
##     weights = n_sampleSize)
##
## Coefficients:
## (Intercept)       x_load    quad_term
##  -4.269e+00    9.059e-04    9.408e-08
##
## Degrees of Freedom: 9 Total (i.e. Null);  7 Residual
## Null Deviance:       112.8
## Residual Deviance: 0.2837    AIC: 51
```

Partie d

```
summary(quadratic_logistic_model)
```

```
##
## Call:
## glm(formula = fail_per_sample ~ x_load + quad_term, family = "binomial",
##     weights = n_sampleSize)
##
## Deviance Residuals:
##       Min         1Q     Median         3Q        Max
## -0.241575  -0.133579   0.006556   0.140825   0.279195
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.269e+00  3.643e+00  -1.172    0.241
## x_load       9.059e-04  2.168e-03   0.418    0.676
## quad_term    9.408e-08  3.167e-07   0.297    0.766
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 112.83207  on 9  degrees of freedom
## Residual deviance:   0.28371  on 7  degrees of freedom
## AIC: 51
##
## Number of Fisher Scoring iterations: 3
```

beta0 beta1 et beta2 ne sont pas statistiquement significatif pour la statistique de Wald

Partie e

```
confint(quadratic_logistic_model)
```

```
## Waiting for profiling to be done...
```

```
##                     2.5 %       97.5 %
## (Intercept) -1.147650e+01 2.825783e+00
## x_load      -3.332064e-03 5.177857e-03
## quad_term   -5.277476e-07 7.155743e-07
```