S. Dame GADIAGA

TP modélisation statistiques

Exercice 1 :

1)

Commande R :

```
library(faraway)
data(pima)
summary(pima)
```

Retour:

```
    pregnant         glucose       diastolic         triceps     
 Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00  
 1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00  
 Median : 3.000   Median :117.0   Median : 72.00   Median :23.00  
 Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54  
 3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00  
 Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00  
    insulin           bmi          diabetes          age       
 Min.   :  0.0   Min.   : 0.00   Min.   :0.0780   Min.   :21.00  
 1st Qu.:  0.0   1st Qu.:27.30   1st Qu.:0.2437   1st Qu.:24.00  
 Median : 30.5   Median :32.00   Median :0.3725   Median :29.00  
 Mean   : 79.8   Mean   :31.99   Mean   :0.4719   Mean   :33.24  
 3rd Qu.:127.2   3rd Qu.:36.60   3rd Qu.:0.6262   3rd Qu.:41.00  
 Max.   :846.0   Max.   :67.10   Max.   :2.4200   Max.   :81.00  
      test       
 Min.   :0.000  
 1st Qu.:0.000  
 Median :0.000  
 Mean   :0.349  
 3rd Qu.:1.000  
 Max.   :1.000  
```

le premier quartile de nos données selon l'insuline était à 0. Cela indique que plus d'un quart de nos taux d'insuline ont subis des erreurs de mesures. Cela remet en question le jeu de donnée.

2)

On supprime donc les valeurs aberrantes avec la commande R :

```
pima_corrigé = pima[-which(pima$bmi==0 | pima$diastolic==0 |
pima$triceps==0 |pima$glucose==0 | pima$insulin==0),]
```

En regardant le résumé de notre nouveau jeu de donnée grâce à la commande R :

```
summary(pima_corrigé)
```

On obtient le retour suivant qui montre que nos valeurs aberrantes ont bien été supprimé :

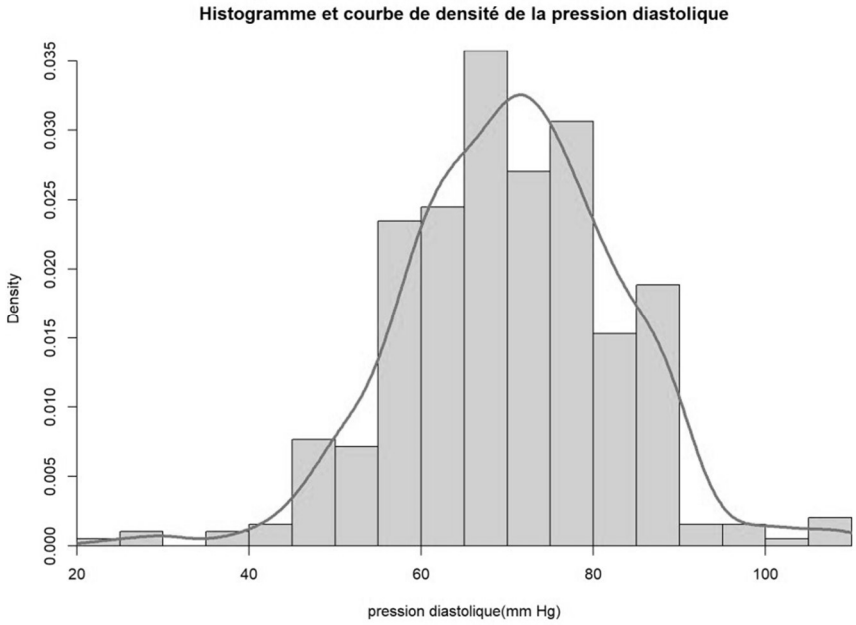
```
    pregnant         glucose        diastolic        triceps
 Min.   : 0.000   Min.   : 56.0   Min.   : 24.00   Min.   : 7.00
 1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.:21.00
 Median : 2.000   Median :119.0   Median : 70.00   Median :29.00
 Mean   : 3.301   Mean   :122.6   Mean   : 70.66   Mean   :29.15
 3rd Qu.: 5.000   3rd Qu.:143.0   3rd Qu.: 78.00   3rd Qu.:37.00
 Max.   :17.000   Max.   :198.0   Max.   :110.00   Max.   :63.00
    insulin           bmi          diabetes          age
 Min.   : 14.00   Min.   :18.20   Min.   :0.0850   Min.   :21.00
 1st Qu.: 76.75   1st Qu.:28.40   1st Qu.:0.2697   1st Qu.:23.00
 Median :125.50   Median :33.20   Median :0.4495   Median :27.00
 Mean   :156.06   Mean   :33.09   Mean   :0.5230   Mean   :30.86
 3rd Qu.:190.00   3rd Qu.:37.10   3rd Qu.:0.6870   3rd Qu.:36.00
 Max.   :846.00   Max.   :67.10   Max.   :2.4200   Max.   :81.00
      test
 Min.   :0.0000
 1st Qu.:0.0000
 Median :0.0000
 Mean   :0.3316
 3rd Qu.:1.0000
 Max.   :1.0000
```

3)

Commande R :

```
hist(pima_corrigé$diastolic,yaxs="i",xaxs="i",breaks=15,col="#F5D0A9", freq=FALSE, main="Histogramme et courbe de densité de la pression diastolique",xlab="pression diastolique(mmHg)")
lines(density(pima_corrigé$diastolic), col="red", lwd=3)
```

Retour:



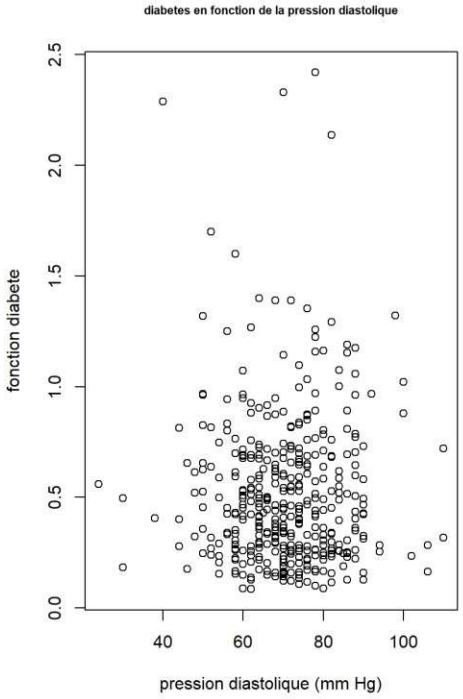
4)

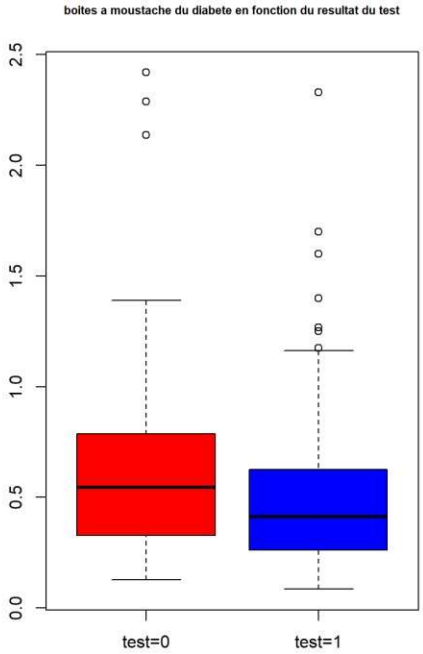
Commande R:

```
plot(pima_corrigé$diastolic,pima_corrigé$diabetes, main="diabetes en fonction de la pression diastolique",cex.main=0.7,xlab="pression
```

```
diastolique (mm Hg)" ,ylab="fonction diabete")
par(fig=c(0.5,1,0,1),new=TRUE)
boxplot(pima_corrigé[which(pima_corrigé$test==1),]$diabetes,pima_cor
rigé[which(pima_corrigé$test==0),]$diabetes,
names=c("test=0","test=1"), col=c("red", "blue"), main="boites a
moustache du diabete en fonction du resultat du test",cex.main=0.7)
```

Retour :



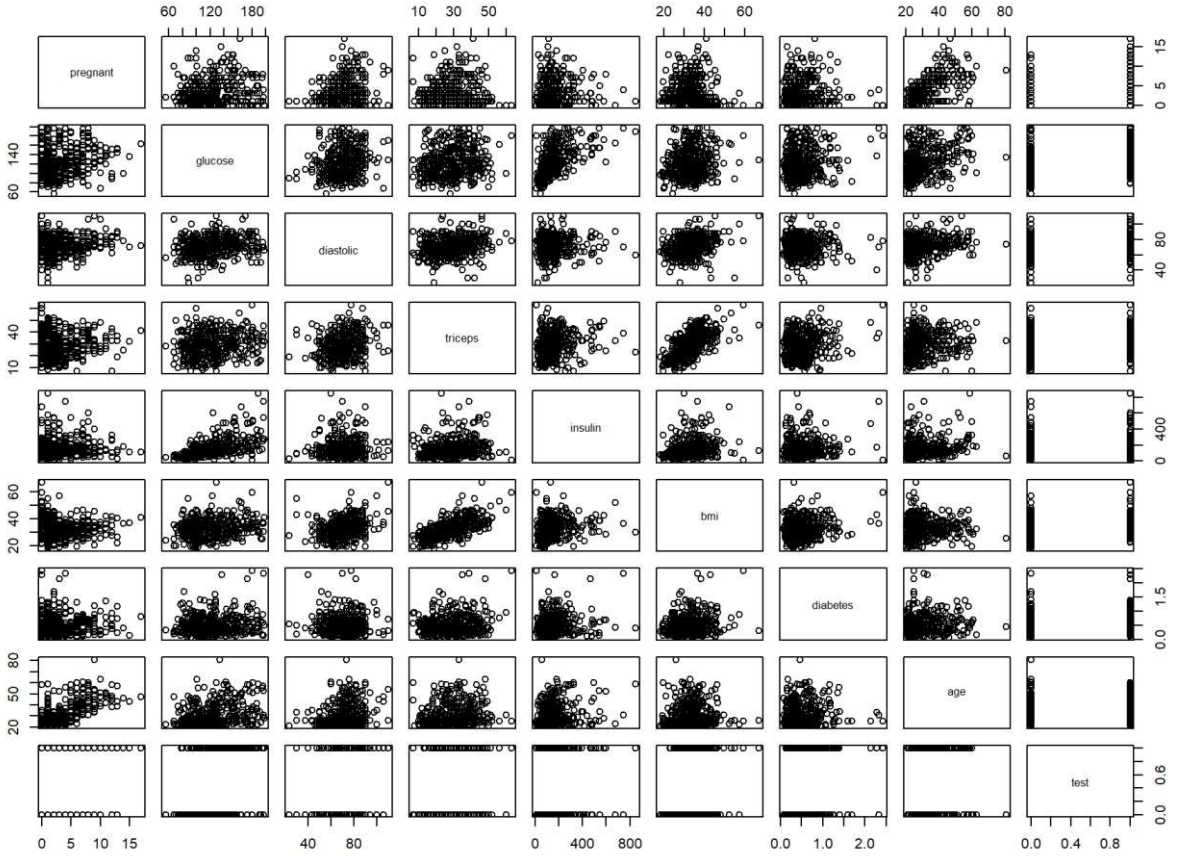


5)

Commande R :

```
pairs(pima_corrigé)
```

Retour:



Exercice 2

1)

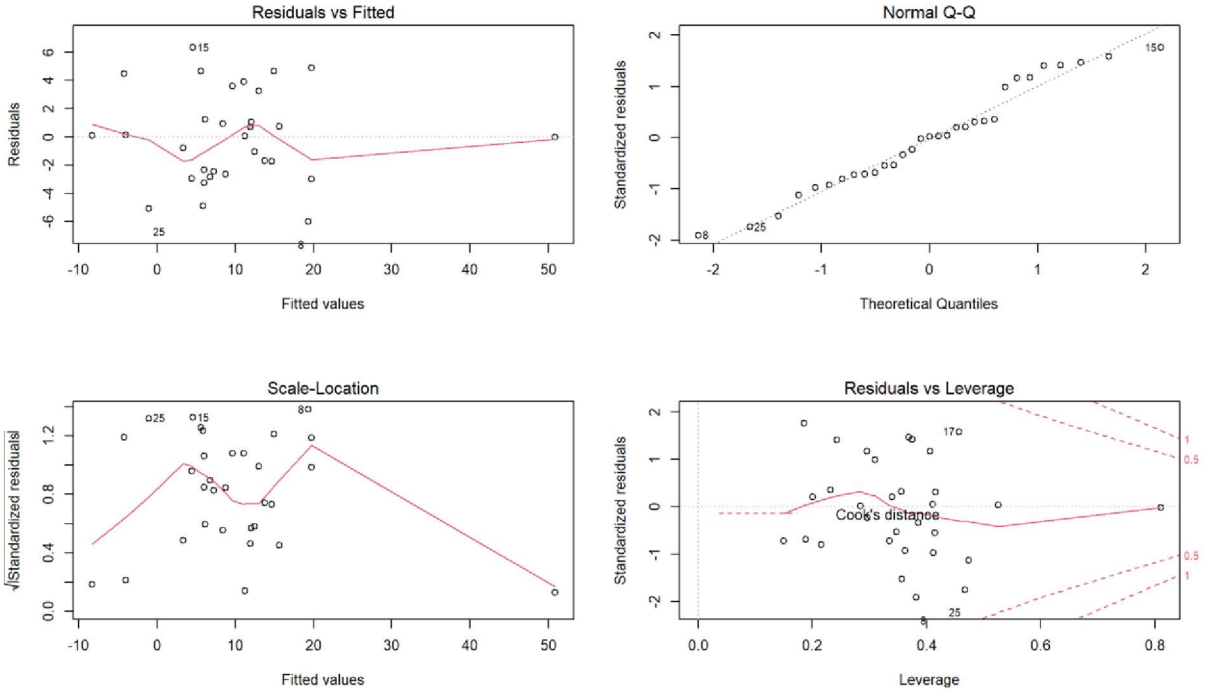
Code R:

```
donnee =
read.table("http://math.univlyon1.fr/~honore/selection.txt",header=T
)
reg=lm(Y ~ Z1+Z2+Z3+Z4+Z5+Z6+Z7+Z8+Z9+Z10, data=donnee)
```

On peut alors dans reg accéder au coefficient devant nos Zi de notre droite de regression mais aussi à notre constante.
Grâce à ces commandes :

```
par(mfrow=c(2,2)) plot(reg)
```

Nous pouvons observer que la ligne rouge du graphique Residuals vs Fitted est proche d'etre une droite ce qui est satisfaisant. Le graphique quantile-quantile nous satisfait aussi car nos quantiles approximés sont cohérents avec les quantiles théoriques.



On peut pousser notre analyse en utilisant :

```
summary(reg)
```

Retour :

```
Call:
lm(formula = Y ~ Z1 + Z2 + Z3 + Z4 + Z5 + Z6 + Z7 + Z8 + Z9 +
    Z10, data = donnee)

Residuals:
    Min      1Q  Median      3Q     Max
-5.9863 -2.5476  0.0658  2.2484  6.3257

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -10.4334    11.6906  -0.892   0.3828
Z1            1.0268     1.0604   0.968   0.3444
Z2            1.3688     0.9664   1.416   0.1721
Z3           -1.5714     0.8014  -1.961   0.0640 .
Z4           -9.3346     9.3867  -0.994   0.3319
Z5           -0.9682     1.8138  -0.534   0.5994
Z6            0.6619     0.8869   0.746   0.4642
Z7            2.5751     1.8437   1.397   0.1778
Z8           -0.3770     0.1510  -2.497   0.0214 *
Z9            1.4103     1.6817   0.839   0.4116
Z10           1.8243     1.7400   1.048   0.3069
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.988 on 20 degrees of freedom
Multiple R-squared:  0.9079,    Adjusted R-squared:  0.8618
F-statistic: 19.71 on 10 and 20 DF,  p-value: 3.113e-08
```

Nous constatons que les valeurs de t sont supérieures à 5% en valeur absolue, validant ainsi nos tests de Student et rendant nos coefficients significatifs. Avec un R^2 ajusté de 86%, notre régression semble de bonne qualité, mais cela peut masquer des problèmes d’endogénéité. En utilisant la fonction vif() de la bibliothèque "car", nous pouvons détecter la colinéarité entre les régressseurs : des valeurs de VIF proches de 1 indiquent une faible colinéarité, tandis que des valeurs supérieures à 10 signalent une multicolinéarité significative à corriger.

```
library(car)
vif(reg)
```

Retour :

```
       Z1          Z2          Z3          Z4          Z5          Z6
 5.434137    3.503790    1.324476 172.526637 291.961779    8.084059
       Z7          Z8          Z9         Z10
327.230750    1.348143 150.751455    1.889313
```

On remarque qu'il y a effectivement présence de colinéarité. Les variables Z4, Z5, Z7 et Z9 ne sont pas décorrélé des autres variables. C'est pourquoi nous pouvons améliorer notre régression en supprime une ou plusieurs de ces variables.

2)

Code R :

```
reg_amélioré=step(reg)
```

Cela retourne comme dernière étape :

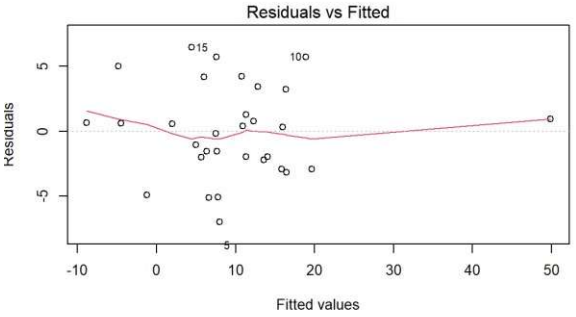
```
Step:  AIC=88.8
Y ~ Z1 + Z2 + Z3 + Z7 + Z8

       Df Sum of Sq    RSS     AIC
<none>              369.2  88.799
- Z3    1     80.72  449.9  92.929
- Z8    1    116.70  485.9  95.313
- Z2    1    215.59  584.8 101.056
- Z1    1    263.38  632.6 103.491
- Z7    1   2932.36 3301.6 154.713
```

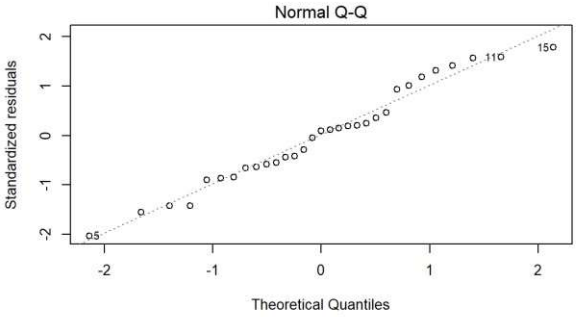
On choisit donc comme sous modèle Y selon Z1 Z2 Z3 Z7 Z8 et la constante. Ce sous modèle est plus performant car il a corrigé les deux problèmes que nous avions repéré dans le modèle complet :

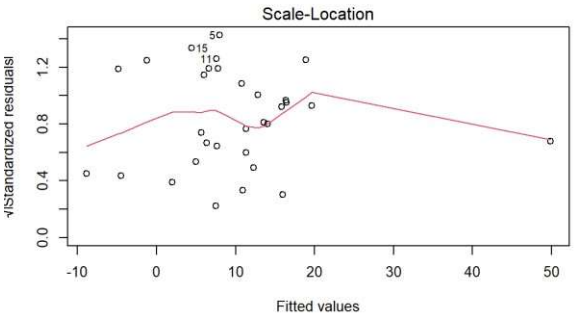
- Avec la commande :

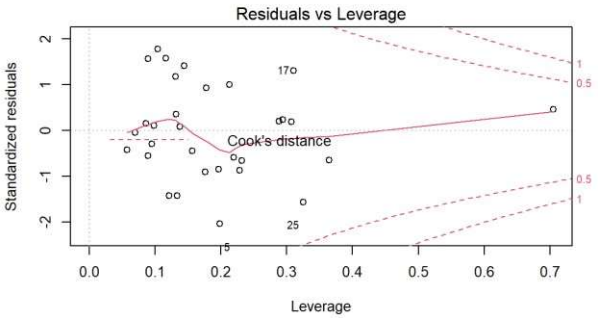
```
plot(reg_amélioré)
```

On obtient :









Sur notre troisième graphique, la courbe rouge représentant la variance des résidus est bien plus horizontal et donc bien plus proche d'être une constante ce qui est la situation idéale.

- Avec la commande : `vif(reg_amélioré)`

On obtient :

```
      z1       z2       z3       z7       z8
1.162069 1.198145 1.097965 1.213803 1.191776
```

Nos facteurs d'inflation de variance sont maintenant tous très proche de 1 ce qui montre que nos variables sont maintenant décorrélé. Il n'y a plus de problèmes de colinéarité.