PROJET: REGRESSION LINEAIRE

Traoré Abou Dramane, Kra Kouamé Gérard, Togbé Joseph Marie

2023-11-11

EXERCICE 1

L'entreprise INFORMATEX se specialise dans l'analyse de systemes et la programmation sur ordinateur de problemes techniques et de gestion. Elle veut utiliser la regression dans une etude sur le temps requis, par ses analystes-programmeurs, pour programmer des projets complexes. Cette etude pourrait permettre a la firme d'etablir des normes quant au temps requis pour programmer certains projets et d'assurer eventuellement une meilleure planification des ressources humaines.

1)Si nous voulons expliquer les fluctuations dans le temps requis pour programmer les projets quelle variable devons-nous identifier comme variable dependante ? Comme variable explicative ?

Pour expliquer les fluctuations dans le temps pour programmer les projets nous allons choisir :

- *comme variable dépendante* ***Temps total en heure*** *, car c'est la variable que nous cherchons à prédire.*
- *comme variables explicatives ou indépendantes* ***Nombres d'intructions,*** *car il peut avoir une relation entre le nombre d'instructions dans un projet et le temps total nécessaire pour le programmer.*

2) Qu'est-ce qui peut renseigner l'entreprise sur la forme de liaison statistique?

Pour répondre à cette question, nous allons examiner le nuage de point et le coefficient de corrélation linéaire

```
# creation de vecteurs X (nombres d'instructions)
x=c(60,82,100,142,190,220,285,354,400,425,440,500,530,640)

#creation du vecteur Y (temps total en heures)
y=c(40,55,62,58,82,94,120,134,128,140,152,174,167,218)
# Affichage des vecteurs
x
```

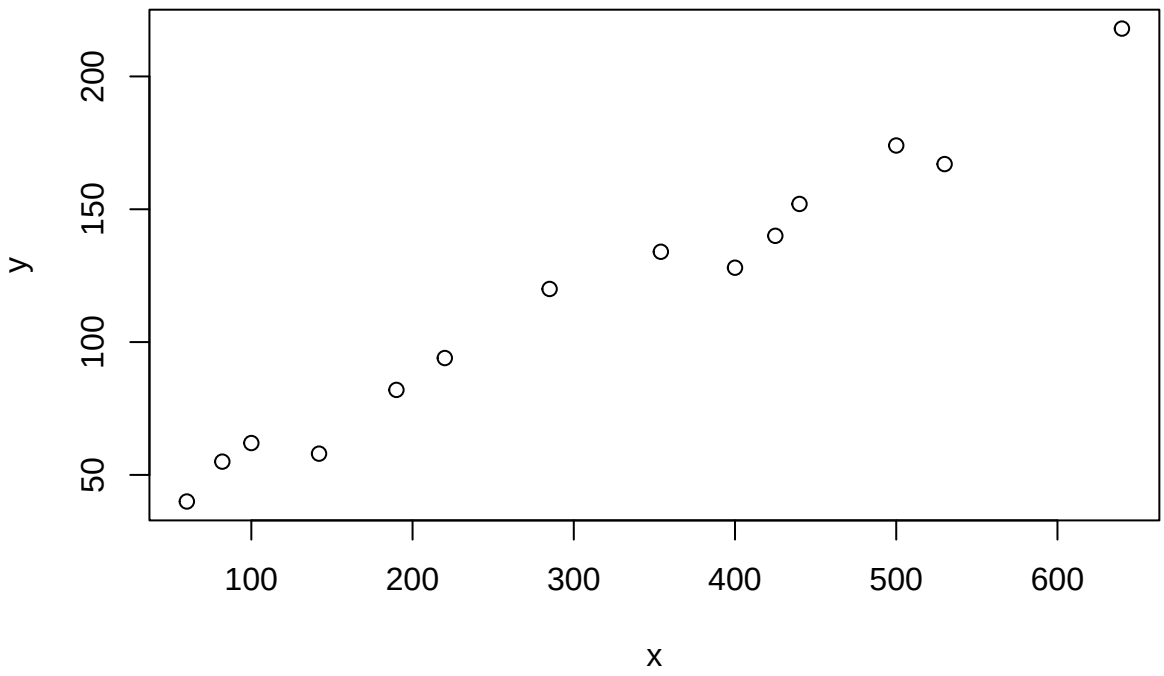
```
##  [1]  60  82 100 142 190 220 285 354 400 425 440 500 530 640
```

```
y
```

```
##  [1]  40  55  62  58  82  94 120 134 128 140 152 174 167 218
```

Le modèle de régression linéaire simple de Y sur X s'écrit: $Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$, $\forall_i = 1, \ldots, 14$.

```
#Nuage des points
plot(x,y)
```




```
# coefficient de corrélation lineaire
cor(x,y)
```

```
## [1] 0.9886826
```

D'après le nuage des points nous observons une tendance linéaire, cela indique une liaison linéaire entre les variables.

D'autre part, le coéfficient de corrélation linéaire ($r = 0.9886826$) *étant très proche de* 1 *, nous permet de conlure également qu'il y a une liaison linéaire entre ces deux variables.*

3) Méthode d'ajustement linéaire à utiliser.

Pour obtenir les estimateurs des coefficients de la droite de regression, nous allons utiliser la méthode des moindres carrées ordinaires.

4) Estimons $\hat{\beta}_0$ et $\hat{\beta}_1$ par la méthode des moindres carrés

```
# Mise en place du modèle de regression linéaire simple
mls=lm(y~x)
# estimation des coefficients de le droite de regression
coef(mls)
```

```
## (Intercept)           x
##  27.9354498   0.2822582
```

On en déduit que $\hat{\beta}_0 = 27.93545$ et $\hat{\beta}_1 = 0.28226$.

5) Equation de la droite des moindres carrés .

D'après la question n°4) on en déduit que la droite des moindres carrés s'écrit: $(D) : y = \hat{\beta}_0 + \hat{\beta}_1 x$ où $\hat{\beta}_0 = 27.93545$ et $\hat{\beta}_1 = 0.28226$.

Donc $(D) : y = 27.93545 + 0.28226x$.

6) Estimation à prendre dans le cas où la variable X n'est pas prise en compte.

Si nous ne tenons pas compte du nombre d'instruction, c'est-à-dire si nous ne tenons pas compte de la variable X, alors, on a:

```
# Calcul de la moyenne de la variable Y
y_moyenne=mean(y)
y_moyenne
```

```
## [1] 116
```

Ainsi, le temps moyen de programmation des projets serait estimé à environ $116 heurs$.

7) Correction à apporter à l'estimation obtenue en 6.

```
b0=27.93545
b1=0.28226
corige=b0+b1*x
mean(corige)
```

```
## [1] 116.0006
```

Par conséquent, une correction de la moyenne obtenue en 6 donne une moyenne de 116.006.

8)D'apres la droite des moindres carres, a quelle augmentation du temps deprogrammation pouvons-nous nous attendre lorsque le nombre d'instructions augmente de 50 ?

```
# Nombre d'instruction augmenté de 50
new_x=x+50
# Prédiction du temps de programmation
pred_y=predict(mls,newdata=data.frame(x=new_x))
# Affichons la valeur augmentée
aug=pred_y[2]-pred_y[1]
aug
```

```
##       2
## 6.20968
```

Ainsi, nous pouvons nous attendre à une augmentation d'environ 6heure de temps.

9)Estimation du temps de programmation a l'aide de la droite des moindres carrées pour chaque donnée.

```
x=c(100,220,440)
newdata=data.frame(x)
ypred=predict.lm(mls,newdata)
ypred
```

```
##         1         2         3
##  56.16127  90.03225 152.12905
```

Ainsi,

- *Pour* $X = 100$*, le temps de programation estimé est* $Ypred = 56.16127$*.*
- *Pour* $X = 220$*, le temps de programation estimé est* $Ypred = 90.03225$*.*
- *Pour* $X = 440$*, le temps de programation estimé est* $Ypred = 152.12905$*.*

10) Les ecarts de prevision de l'equation des moindres carres pour le nombre d'instruction en 9, selon les résultats observés.

Pour obtenir les écarts de prévision de l'équation des moindres carrées pour le nombre d'instruction en 9), nous allons d'abord définir le vecteur des instructions que nous appellons x, puis nous allons ajuster l'équation des moindres carrée et calculer en fin les écarts de prévision pour les données de la question 9).

On obtient donc:

```r
x=c(100,220,440)
# Ajustement de l'equation des moindres carrées
rg=lm(ypred~x)
# Calcul des résidus(écarts de prévision)
residuals(rg)
```

```
##             1             2             3
## -1.480388e-14  2.287872e-14 -8.074842e-15
```

- *Pour* $X = 100$, *on a* $\hat{\varepsilon_1} = -1,480388.10^{-14}$.
- *Pour* $X = 220$, *on a* $\hat{\varepsilon_2} = 2,287872.10^{-14}$.
- *Pour* $X = 440$, *on a* $\hat{\varepsilon_3} = -8,074842.10^{-15}$.

11) Calcul des écarts en prenant la valeur estimée obtenue en 6).

```r
# Temps estimé en 6)
varrpred=116
# Nouvelle prédiction avec cette valeur
Res_new=varrpred-ypred
# Ajustement de l'equation des moindres carrées
rgg=lm(Res_new~x)
# Calcul des résidus(écarts de prévision)
residuals(rgg)
```

```
##             1             2             3
##  1.480388e-14 -2.287872e-14  8.074842e-15
```

Ainsi on aurait:

- *Pour* $X = 100$, *on a* $\hat{\varepsilon_1} = 1,480388.10^{-14}$.
- *Pour* $X = 220$, *on a* $\hat{\varepsilon_2} = -2,287872.10^{-14}$.
- *Pour* $X = 440$, *on a* $\hat{\varepsilon_3} = 8,074842.10^{-15}$.

Remarque:

les valeurs obtenues représentent exactement l'opposées de celles obtenues en 10).

12)Vérifions l'égalité suivantes $y_i - \overline{y} = (\hat{y} - \overline{y}) + (y_i - \hat{y})$ pour tout x_i spécifié dans 9).

```r
#Estimation des coefficients de la droite de regression
b0=27.93545
b1=0.28226

#Donnée de 9)
N=c(100,220,440)
#Moyenne de la variable Y
moy_y=mean(y)
#Y prédit
N_y=predict(mls,newdata = data.frame(x=N))
for (i in 1:length(N)){
  yi=b0+b1*N[i]
  yi_chapo=N_y[i]

  R1 = yi - moy_y
  R2 = (yi_chapo - moy_y) + (yi - yi_chapo)
  # Affichage du résultat pour chaque valeur xi
  cat("Pour x =", N[i], ":\n")
  cat("Gauche :", R1, "\n")
  cat("Droite :", R2, "\n")
  cat("\n")
}
```

```
## Pour x = 100 :
## Gauche : -59.83855
## Droite : -59.83855
##
## Pour x = 220 :
## Gauche : -25.96735
## Droite : -25.96735
##
## Pour x = 440 :
## Gauche : 36.12985
## Droite : 36.12985
```

Nous remarquons effectivement qu'il y a une égalité entre ces deux quantités.

13)Calcul de SCT, SCE et SCR.

```r
# Calcul de la moyenne de Y
moy_Y=mean(y)
# Prédiction des valeurs de Y
pred_Y=predict(mls)

# Variation totale (SCT)
SCT=sum((y - moy_Y)^2)

# Variation expliquée (SCE)
SCE=sum((pred_Y - moy_Y)^2)
```

```r
# Variation résiduelle (SCR)
SCR=sum(mls$residuals^2)

# Affichage des résultats
SCT
```

```
## [1] 36142
```

```r
SCE
```

```
## [1] 35328.56
```

```r
SCR
```

```
## [1] 813.438
```

On en déduit que la Variation total est: $SCT = 36142$*, la Variation expliquée est* $SCE = 35328.56$ *et la Variation résiduelle est* $SCR = 813.438$*.*

14)Determinons la proportion expliquée par la droite des moindres carrées.

```r
#Proportion de variation expliquée
R_carr=SCE/SCT
R_carr
```

```
## [1] 0.9774933
```

Ainsi, $R^2 = 0.9775$*, cela voudrait dire que la droite des moindres carrées explique* $97,75\%$ *de la variation totale du temps de programmation.*

Cependant, $2,25\%$ *de la variation totale du temps de programmation demeure inexpliquée par la droite des moindres carrées.*

15)

Si nous avions fixé la valeur de R^2 *à* $0,90$*, compte tenu du resultat précédent (N°14), nous allions utiliser la droite des moindres carrées comme outil de prediction, car le* R^2 *était d'environ* $0,9775$ *ce qui est supérieur à* $0,90$*.*

Par conséquent, la droite des moindres carrées demeure un bon outil de prédiction pour ce modèle.

EXERCICE 2

1) Régressons Y sur X_1.

Avant de faire la régression linéaire pour cet exercice, supposons que les hypothèses suivantes sont vérifiées:

- H_0 : *les variables* X_i sont non aléatoires.
- H_1 : $E(\varepsilon_i) = 0$, $\forall_i = 1, \ldots, 22$.
- H_3 : $\varepsilon_1, \ldots, \varepsilon_i$ *sont indépendantes et* $\varepsilon_i \hookrightarrow N(0, \sigma^2)$, $\forall_i = 1, \ldots, 22$.

En effet, sous les hypothèses ci-dessus, nous pourrions répondre facilement aux questions de tests statistiques et d'intervalles de confiances un peu plus loin dans cet exercice.

Création des vecteurs

```
#Creation du vecteur x1(telephone par millier d'habitants).
x1=c(124,49,181,4,22,152,75,54,43,41,17,22,16,10,63,170,125,12,221,171,97,254)

#Creation du vecteur x2(calories grasses en pourcentage ddu total des calories).
x2=c(33,31,38,17,20,39,30,29,35,31,23,21,8,23,37,40,38,25,39,33,38,39)

#Creation du vecteur x3(calories provenant des proteines animales en pourcentage du total des calories)
x3=c(8,6,8,2,4,6,7,7,6,5,4,3,3,3,6,8,6,4,7,7,6,8)

#Creation du vecteur y.
y=c(81,55,80,24,71,52,88,45,50,69,66,45,24,43,38,72,41,38,52,52,66,89)

#Affichage des vecteurs.
x1
```

```
##  [1] 124  49 181   4  22 152  75  54  43  41  17  22  16  10  63 170 125  12 221
## [20] 171  97 254
```

```
x2
```

```
##  [1] 33 31 38 17 20 39 30 29 35 31 23 21  8 23 37 40 38 25 39 33 38 39
```

```
x3
```

```
##  [1] 8 6 8 2 4 6 7 7 6 5 4 3 3 3 6 8 6 4 7 7 6 8
```

```
y
```

```
##  [1] 81 55 80 24 71 52 88 45 50 69 66 45 24 43 38 72 41 38 52 52 66 89
```

mise en place du modèle linéaire simple de Y sur X_1.

Le modèle de régression linéaire simple de Y *sur* X_1 *s'écrit:* $Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$, $\forall_i = 1, \ldots, 22$.

```r
# Modèle linéaire simple de Y sur x1.
mls=lm(y~x1)
# Test de significativité.
summary(mls)
```

```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -23.566 -14.173  -2.109  11.991  33.128
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 45.58436    5.59695   8.145 8.83e-08 ***
## x1           0.12384    0.04892   2.532   0.0198 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.94 on 20 degrees of freedom
## Multiple R-squared:  0.2427, Adjusted R-squared:  0.2048
## F-statistic: 6.409 on 1 and 20 DF,  p-value: 0.01985
```

Ainsi, nous avons: $p - value = 0.01985$.

Et comme la \$p-value< \$ seuil de signification $\alpha = 0,05$,*alors la relation entre x1 et Y est* ***statistiquement significative.***

2)Equation de la regression linéaire multiple de Y sur X_1 et X_2.

Le modèle de régression linéaire multiple de Y *sur* X_1 *et* X_2 *s'écrit:* $Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \varepsilon_i, \forall_i = 1, \ldots, 22$.

```r
#Mise en place du modèle linéaire multiple entre x1, x2 et Y.
mlm_1=lm(y~x1+x2)

#Estimation des coefficients de la droite de régression.
coefficients=coef(mlm_1)
coefficients
```

```
## (Intercept)          x1          x2
## 33.81275961  0.07857841  0.51876013
```

On en déduit que $\hat{\beta}_0 = 33.81276$, $\hat{\beta}_1 = 0.07858$ et $\hat{\beta}_2 = 0.51876$.

Donc l'équation de la regression linéaire multiple s'écrit :

$(D) : y = \hat{\beta}_0 + \hat{\beta}_1 x1 + \hat{\beta}_2 x2$

où $\hat{\beta}_0 = 33.81276$, $\hat{\beta}_1 = 0.07858$ *et* $\hat{\beta}_2 = 0.51876$.

Donc $(D) : y = 33.81276 + 0.07858x1 + 0.51876x2$

3) Effectuons un test conjoint d'hypothese nulle $H_0 : \beta_1 = \beta_2 = 0$.

Il s'agira de verifier si toutes les variables explicatives incluses dans le modèle (x1 et x2) sont conjointement significatives. Et nous allons utiliser la fonction ***anova()***

```
Test=anova(mlm_1)
#la statistique de test
f_value=Test$F[2]
#la valeur de p associée au test
p_value=Test$'Pr(>F)'[2]
#affichage des resultats
f_value
```

```
## [1] 0.6182931
```

```
p_value
```

```
## [1] 0.4413797
```

On remarque que la $p - value = 0.4414 >$ *seuil de signification* $(\alpha = 0,05)$

Par conséquent, nous acceptons l'hypothèse nulle $H_0 : \beta_0 = \beta_1 = 0$.

4) Tester si l'adjonction de la variable X_2 à l'équation trouvée à la question 2 a significativement ameliorer l'estimation.

```
# Régression de Y sur X1.
mls=lm(y~x1)

# Régression de Y sur X1 et X2.
mlm_1=lm(y~x1+x2)

# Calcul du R-carré ajusté.
R_carr_x1=summary(mls)$adj.r.squared
R_carr_x1_x2=summary(mlm_1)$adj.r.squared
#Affichage.
R_carr_x1
```

```
## [1] 0.2048071
```

```
R_carr_x1_x2
```

```
## [1] 0.1893353
```

```
# Comparaison des modèles.
if (R_carr_x1_x2 > R_carr_x1) {
  Evolution=R_carr_x1_x2 - R_carr_x1
  message("L'ajout de la variable X2 a significativement amélioré l'estimation du modèle de régression
} else {
  message("L'ajout de la variable X2 n'a pas significativement amélioré l'estimation du modèle de régres
}
```

```
## L'ajout de la variable X2 n'a pas significativement amélioré l'estimation du modèle de régression.
```

5) Régression linéaire multiple de Y sur x1, x2 et x3

Le modèle de régression linéaire multiple de Y sur x1, x2 et x3 s'écrit: $Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \varepsilon_i$, $\forall_i = 1, \ldots, 22$.

```
# Mise en place du modèle de régression linéaire multiple.
mlm=lm(y~x1+x2+x3 )

#Estimation des coefficients de la droite de régression.
coefficients=coef(mlm)
coefficients
```

```
##  (Intercept)           x1           x2           x3
## 22.353115212 -0.005842359 -0.423987726  8.413436565
```

On en déduit que $\hat{\beta}_0 = 22.35316$, $\hat{\beta}_1 = -0.00584$, $\hat{\beta}_2 = -0.42398$ *et* $\hat{\beta}_3 = 8.41344$.

Donc l'équation de la regression linéaire multiple s'écrit :

$(D) : y = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_3$

où $\hat{\beta}_0 = 22.35316$, $\hat{\beta}_1 = -0.00584$, $\hat{\beta}_2 = -0.42398$ *et* $\hat{\beta}_3 = 8.41344$.

Donc $(D) : y = 22.35316 - 0.00584x_1 - 0.42398x_2 + 8.41344x_3$

6) les limites de l'intervalle de confiance à 95% pour β_3 dans cette equation.

```
# Calcul des intervalles de confiance à 95%.
IC=confint(mlm)["x3", ]
# Affichage des limites des intervalles de confiance.
IC
```

```
##      2.5 %     97.5 %
##  0.6618697 16.1650034
```

On en déduit que pour β_3, *l'intervalle de confiance à* 95% *est* $IC = [0.6618697; 16.1650034]$.

Ainsi, la borne inférieur de cet intervalle de confiance est 0.6618697 *et sa borne supérieur est* 16.1650034 .

7)Les limites de l'intervalle de confiance à 95% pour $\hat{Y}$ aux points $X1 = 221$ et $X2 = 39$ et $X3 = 7$.

```
# Prédiction de Y avec intervalle de confiance à 95%.
Y_pred=predict(mlm,data.frame(x1=221,x2=39,x3=7), interval = "confidence")
# Affichage des limites de l'intervalle de confiance.
IC=Y_pred[, c("lwr", "upr")]
IC
```

```
##      lwr      upr
## 46.88115 79.95982
```

On en déduit que pour $\hat{Y}$, *l'intervalle de confiance à* 95% *est* $IC = [46.88115; 79.95982]$.

Ainsi, la borne inférieur de cet intervalle de confiance est 46.88115 *et sa borne supérieur est* 79.95982 .

8) Testons si X_1 et X_2 ensemble apportent quelque chose à la régression linéaire simple de Y sur X_1.

```
# Calcul du coefficient de détermination ajusté.
R_carr_aj_mls= summary(mls)$adj.r.squared
R_carr_aj_mlm= summary(mlm)$adj.r.squared

# Comparaison des coefficients de détermination ajustés.
if (R_carr_aj_mlm > R_carr_aj_mls) {
  Evolution=R_carr_aj_mlm - R_carr_aj_mls
  message("L'ajout des variables x2 et x3 ensemble a significativement amélioré le modèle de régression
} else {
  message("L'ajout des variables x2 et x3 ensemble n'a pas significativement amélioré le modèle de régr
}
```

```
## L'ajout des variables x2 et x3 ensemble a significativement amélioré le modèle de régression (coeffi
```

9) Régressons X_1 sur X_2 et X_3 .

mise en place du modèle de régression linéaire de X_1 sur X_2 et X_3.

Le modèle de régression linéaire multiple de X_1 sur X_2 et X_3 s'écrit: $X_{1i} = \beta_0 + \beta_1 x_{2i} + \beta_2 x_{3i} + \varepsilon_i$, $\forall_i = 1, \ldots, 22$.

```
# Régression de x1 sur x2 et x3.
mlm_new=lm(x1~x2+x3)
#Estimation des coefficients de la droite de régression.
coefficients=coef(mlm_new)
coefficients
```

```
## (Intercept)          x2          x3
## -117.910264    2.596946   22.458574
```

On en déduit que $\hat{\beta}_0 = -117.910264$, $\hat{\beta}_1 = 2.596946$ et $\hat{\beta}_2 = 22.458574$.

Donc l'équation de la regression linéaire multiple s'écrit :

$(D) : y = \hat{\beta}_0 + \hat{\beta}_1 x_2 + \hat{\beta}_2 x_3$

où $\hat{\beta}_0 = -117.910264$, $\hat{\beta}_1 = 2.596946$ *et* $\hat{\beta}_2 = 22.458574$.

Donc $(D) : y = -117.910264 + 2.596946 x_2 + 22.458574 x_3$.

10)Les limites de l'intervalle de confiance à 95% pour les coefficients de la régression de X_1 sur X_3.

Le modèle de régression linéaire simple de X_1 sur X_3 s'écrit: $X_{1i} = \beta_0 + \beta_1 x_{3i} + \varepsilon_i$, $\forall_i = 1, \ldots, 22$.

```r
# Régression linéaire de x1 sur x3
mls_new=lm(x1~x3)

# Calcul des intervalles de confiance à 95% pour les coefficients
IC=confint(mls_new)

# Affichage des limites des intervalles de confiance
IC
```

```
##                  2.5 %    97.5 %
## (Intercept) -162.54946 -28.96795
## x3            21.22245  43.77258
```

Ains,

- *Pour* $\hat{\beta}_0$, $IC = [-162.54946; -28.96795]$ *et donc la borne inférieur de cet intervalle de confiance est* -162.54946 *et sa borne supérieur est* -28.96795 .
- *Pour* $\hat{\beta}_1$, $IC = [21.22245; 43.77258]$ *et donc la borne inférieur de cet intervalle de confiance est* 21.22245 *et sa borne supérieur est* 43.77258 .