Éléments de correction du TD numéro 8

Exercice 1

Question 1

```
R Console

> Exo1<-read.table("td8.txt",dec=",")
> Exo1
     V1   V2
1  1.73 2.32
2  1.73 2.31
3  1.83 2.40
4  1.84 2.40
5  1.84 2.40
6  1.78 2.33
7  1.84 2.37
8  1.85 2.37
9  1.85 2.37
10 1.86 2.37
11 1.85 2.36
12 1.91 2.41
13 1.87 2.36
14 1.96 2.45
15 1.88 2.36
16 1.94 2.39
17 1.85 2.28
18 2.00 2.42
19 1.94 2.35
20 2.01 2.40
> droite<-lm(V2~V1,data=Exo1)
> droite

Call:
lm(formula = V2 ~ V1, data = Exo1)

Coefficients:
(Intercept)           V1
     1.7312       0.3425
```

Les estimations par la méthode des MCO des deux coefficients sont égales à
$\beta 0=1.7312$
$\beta 1=0.3424$

Question 2

Source de variation	Degrés de liberté	Somme des carrés	Moyenne des carrés	Fobs
Régression	1	0.0128225	0.0128225	12.711
Résiduelle	18	0.0181575	0.0010088	
Totale	19	0.0309800		

Question 3

Le pourcentage de la variation totale des performances est expliqué par 41,39% de la variable taille. Ce résultat n’est pas très satisfaisant. En tant que chargé d’étude, il faudrait chercher à introduire d’autres variables explicatives, comme le poids ou l’âge.

<u>Exercice 2</u>

Question 1

R Console

```
> Exo2<-read.table("xy.csv",sep=";",header=T)
> Exo2
   xi yi
1   7  5
2   9  4
3   9  6
4  10  4
5  13  1
6  17  2
7  19  0
8  20  1
9  21  1
10 25  0
> Droite2<-lm(yi~xi,data=Exo2)
> Droite2

Call:
lm(formula = yi ~ xi, data = Exo2)

Coefficients:
(Intercept)           xi
     7.0387      -0.3092
```

Les estimations par la méthode des MCO des deux coefficients sont égales à
$\beta 0$=7.0387
$\beta 1$=–0.3092

Question 2

En utilisant la commande « confint » et en indiquant que nous souhaitons un « level=0.99 », nous obtenons :

```
> confint(model,level=0.99)
                0.5 %      99.5 %
(Intercept)  3.903305 10.1741520
x_i         -0.503849 -0.1146481
```

Donc un intervalle de confiance à 99%, pour beta_0 est égal à] 3.903305; 10.1741520[.
Donc un intervalle de confiance à 99%, pour beta_1 est égal à] -0.503849;-0.1146481 [

Question 3

En faisant à la suite les deux lignes de commande « sum(exo2TD8$y_i) » puis « sum(fitted(model)) », nous trouvons bien à chaque fois la somme égale à 24.

Question 4

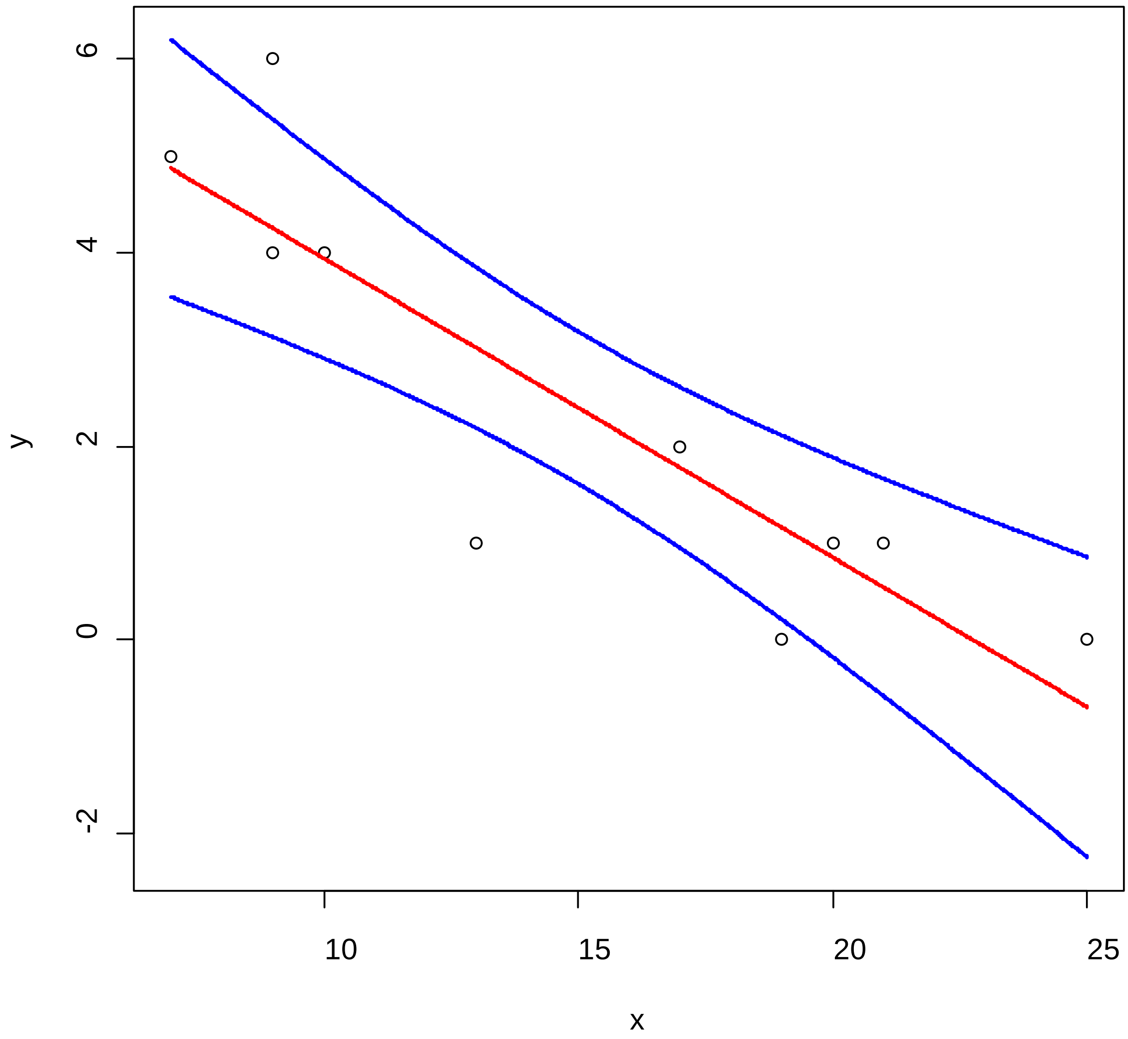
Pour répondre à cette question, il faut utiliser la commande « predict) dans laquelle il faut rajouter l'option « interval= "confidence" ».

	fit	lwr	upr
1	4.8739884	3.5459832	6.201994
2	4.2554913	3.1317570	5.379226
3	4.2554913	3.1317570	5.379226
4	3.9462428	2.9137561	4.978729
5	3.0184971	2.1875852	3.849409
6	1.7815029	0.9505910	2.612415
7	1.1630058	0.2116640	2.114348
8	0.8537572	-0.1787294	1.886244
9	0.5445087	-0.5792257	1.668243
10	-0.6924855	-2.2441001	0.859129

Question 5

Voir le graphique

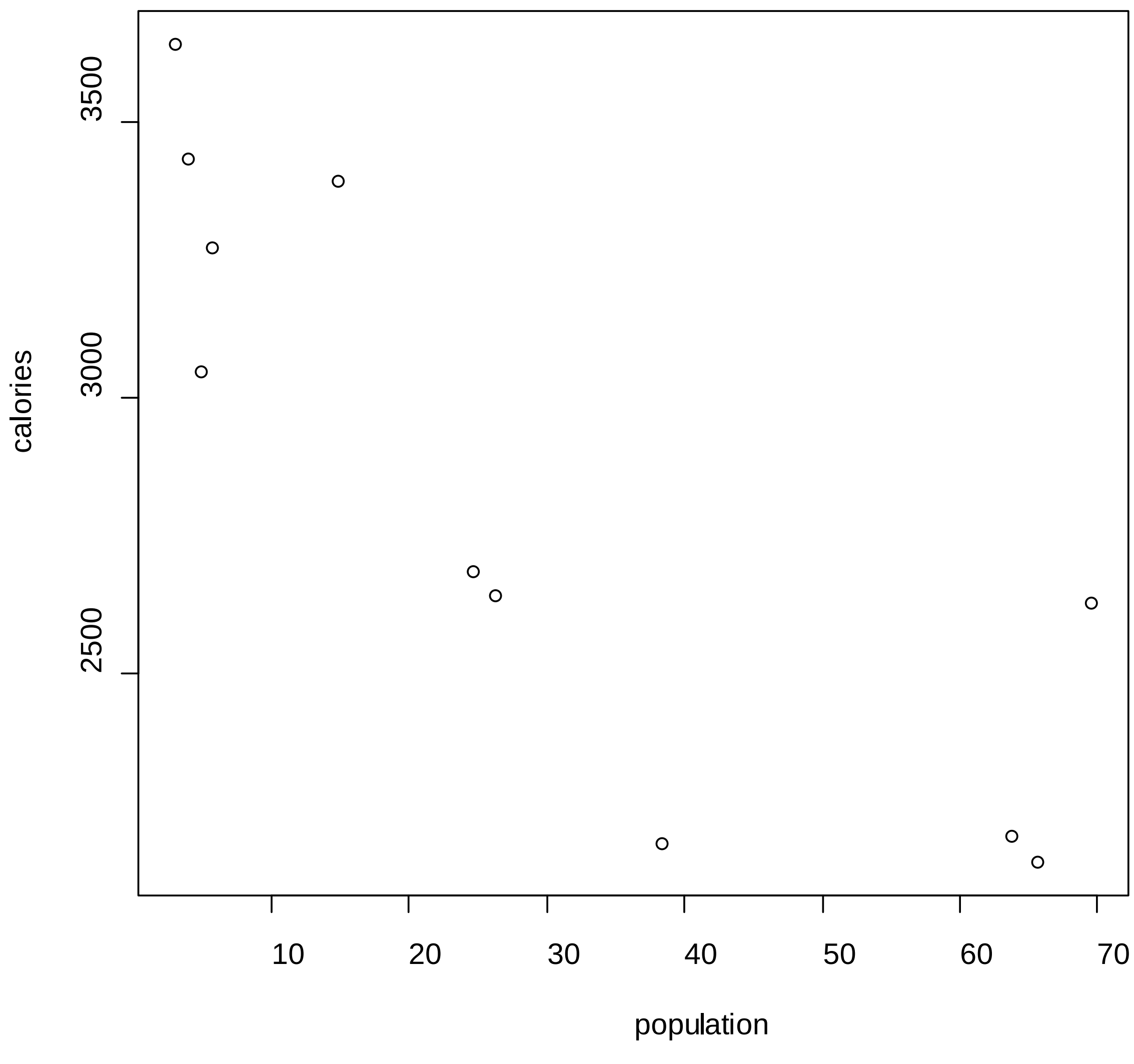
```
x_int <- seq(min(exo2TD8$x_i),max(exo2TD8),length.out=1000)
preds_int <- predict(model,list(x_i=x_int),interval="confidence")
plot(exo2TD8$x_i, exo2TD8$y_i,xlab="x",ylab="y", ylim=c(min(preds_int),
max(preds_int)))
lines(x_int, preds_int[,1],col="red",lwd=2)
lines(x_int, preds_int[,2],col="blue",lwd=2)
lines(x_int, preds_int[,3],col="blue",lwd=2)

y_i_chap <- predict(model, interval= "confidence")
for (ii in 1:(dim(y_i_chap)[1])) {
lines(c(exo2TD8$x_i[ii], exo2TD8$x_i[ii]),c(y_i_chap[ii,2], y_i_chap[ii,3]),lty=2)
}
rm(ii)
```



6
4
2
0
-2
y
10
15
20
25
x

Question 1

```
> population<-c(4,5.7,4.9,3,14.8,69.6,63.8,26.2,38.3,24.7,65.7)
> population
 [1]  4.0  5.7  4.9  3.0 14.8 69.6 63.8 26.2 38.3 24.7 65.7
> calories<-c(3432,3273,3049,3642,3394,2628,2204,2643,2192,2687,2159)
> calories
 [1] 3432 3273 3049 3642 3394 2628 2204 2643 2192 2687 2159
>plot(calories~population)
```



Question 2

```
> donnees<-data.frame(population, calories)
> donnees
   population calories
1         4.0     3432
2         5.7     3273
3         4.9     3049
```

```
4         3.0     3642
5        14.8     3394
6        69.6     2628
7        63.8     2204
8        26.2     2643
9        38.3     2192
10       24.7     2687
11       65.7     2159
>modele<-lm(calories~population,data=donnees)
>coef(modele)
(Intercept)     population
3346.12221   -17.16353
```

Donc

Beta_0=3346.12221

Beta_1=-17.16353

Question 3

```
> anova(modele)
Analysis of Variance Table

Response: calories
           Df  Sum Sq Mean Sq F value   Pr(>F)
population  1 2045960 2045960  20.907 0.001342 **
Residuals   9   880756      97862
Total        10 2926716
---
```

Question 4

```
>y_i_chap <- predict(modele, interval= "confidence")
>y_i_chap
        fit      lwr      upr
1  3277.468 2975.554 3579.382
2  3248.290 2956.411 3540.170
3  3262.021 2965.465 3558.577
4  3294.632 2986.651 3602.612
5  3092.102 2846.370 3337.834
6  2151.540 1747.213 2555.868
7  2251.089 1887.665 2614.512
8  2896.438 2681.598 3111.277
9  2688.759 2461.696 2915.822
10 2922.183 2705.486 3138.880
11 2218.478 1841.875 2595.081
```

Question 5

```
x_int <- seq(min(donnees$population),max(donnees$population),length.out=1000)
preds_int <- predict(modele,list(population=x_int),interval="confidence")
```

```
plot(donnees$population,donnees$calories,xlab="population",ylab="calories",ylim=c(min(preds_int), max(preds_int)))
lines(x_int, preds_int[,1],col="red",lwd=2)
lines(x_int, preds_int[,2],col="blue",lwd=2)
lines(x_int, preds_int[,3],col="blue",lwd=2)

y_i_chap <- predict(modele, interval= "confidence")
for (ii in 1:(dim(y_i_chap)[1]))
{
lines(c(donnees$population[ii], donnees$population[ii]),c(y_i_chap[ii,2],
y_i_chap[ii,3]),lty=2)
}
rm(ii)
```