MSI : TP3 Régression Linéaire

Robin ANDRE & Robin LECONTE

Partie 1 : Puissance d'un test

Question 1

La puissance d'un test est la probabilité complémentaire de la probabilité d'accepter une hypothèse fausse si on se fie au résultat du test.

En calculant la puissance d'un test, on sait donc si un test est fiable ou non. Ainsi, selon le cadre d'utilisation du test (pharmaceutique, construction, transport etc…), on pourra en fonction des risques encourus dans le cas d'une erreur, décider si le test répond aux besoins ou non. Les tests puissants étant bien évidemment plus couteux que les tests moins puissants.

Afin de construire un test de manière à le rendre puissant, on peut utiliser des échantillons de grande taille.

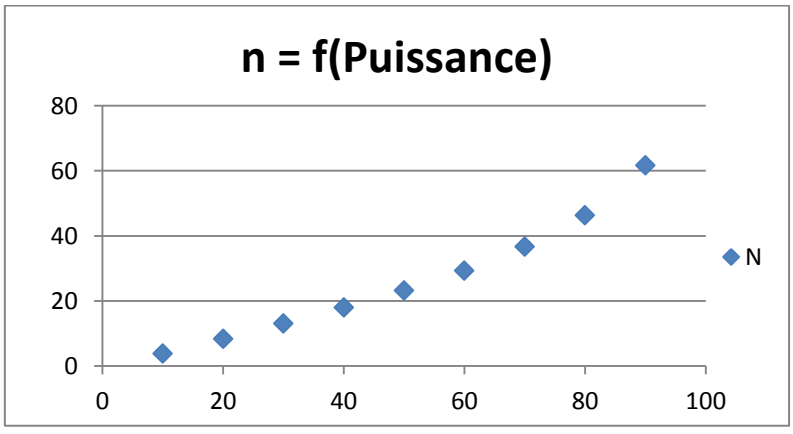
Question 2

Pour atteindre une puissance de 80% il faut au moins 47 vaches.

```
> power.t.test(delta = 1, sd = 1.7, sig.level = 0.05, power = 0.8)

     Two-sample t test power calculation

              n = 46.34074
          delta = 1
             sd = 1.7
      sig.level = 0.05
          power = 0.8
    alternative = two.sided

NOTE: n is number in each group
```

Question 3

Représentons le nombre d'individus à comparer en fonction de la puissance du test voulue :



Plus la puissance souhaitée est grande (et donc le niveau de risque fixé est faible), plus la taille de l'échantillon est grande.

- Quand l'échantillon correspond à la population toute entière (ce qui est la taille maximale pour un échantillon) alors, la puissance du test vaut 1.
- Plus on prend en compte d'individus, plus il est difficile de se tromper. La suite Un des puissances pour n individus est donc croissante.

Ces deux points expliquent l'allure de la courbe ci-dessus.

Question 4

La puissance du test pour un échantillon de 20 vaches vaut 44 %.

```
> power.t.test(n = 20, delta = 1, sd = 1.7, sig.level = 0.05)

     Two-sample t test power calculation

              n = 20
          delta = 1
             sd = 1.7
      sig.level = 0.05
          power = 0.4416743
    alternative = two.sided

 NOTE: n is number in *each* group
```

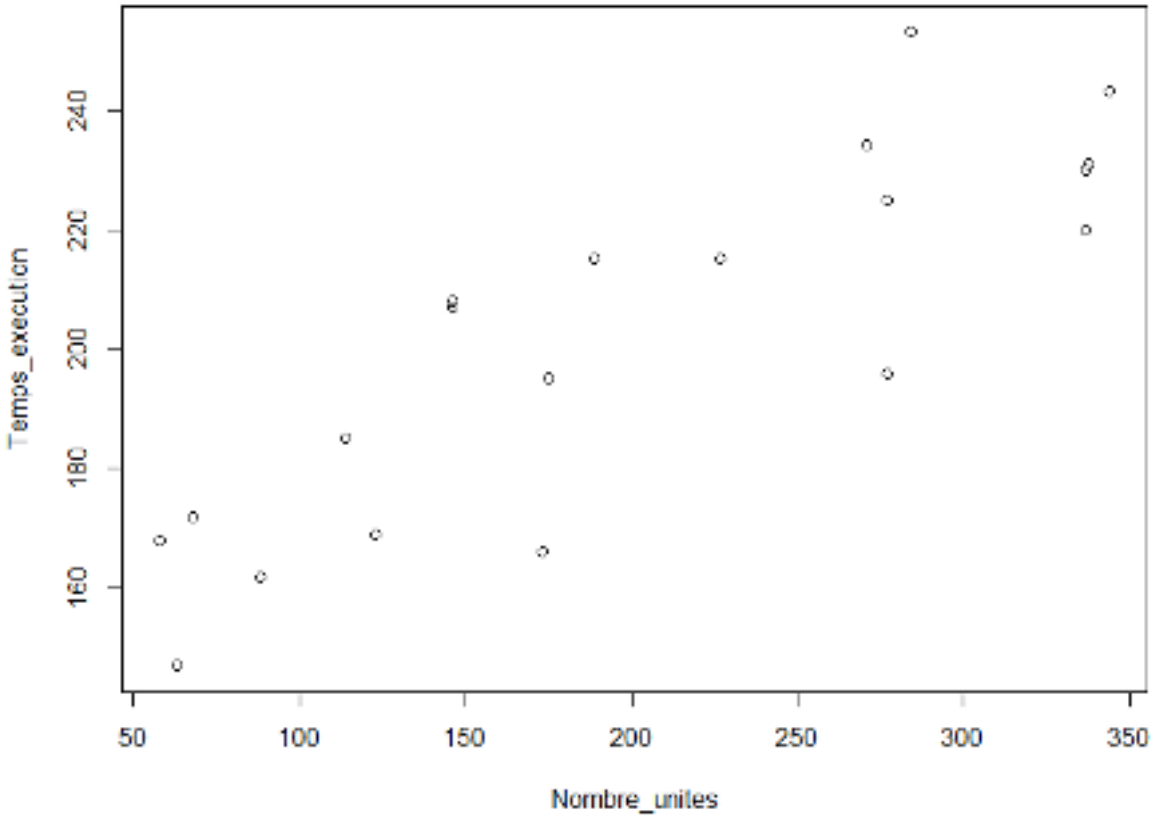
Question 5

La différence de moyennes détectable à 80 % avec 20 individus par groupe vaut 1.55 g/kg.

```
> power.t.test(n = 20, sd = 1.7, sig.level = 0.05, power = 0.8)

     Two-sample t test power calculation

              n = 20
          delta = 1.545522
             sd = 1.7
      sig.level = 0.05
          power = 0.8
    alternative = two.sided

 NOTE: n is number in *each* group
```

Partie 2 : Régression linéaire

Question 1



Au regard de ce graphique, il est légitime de penser qu'une relation linéaire existe entre les deux variables du jeu de données.

```
> cor(TP3$temps_exec,TP3$unites)
[1] 0.8545206
```

La corrélation linéaire au sens de Bravais-Pearson vaut 0.85. Autrement dit, les variables sont fortement corrélées puisque cette valeur est proche de 1.

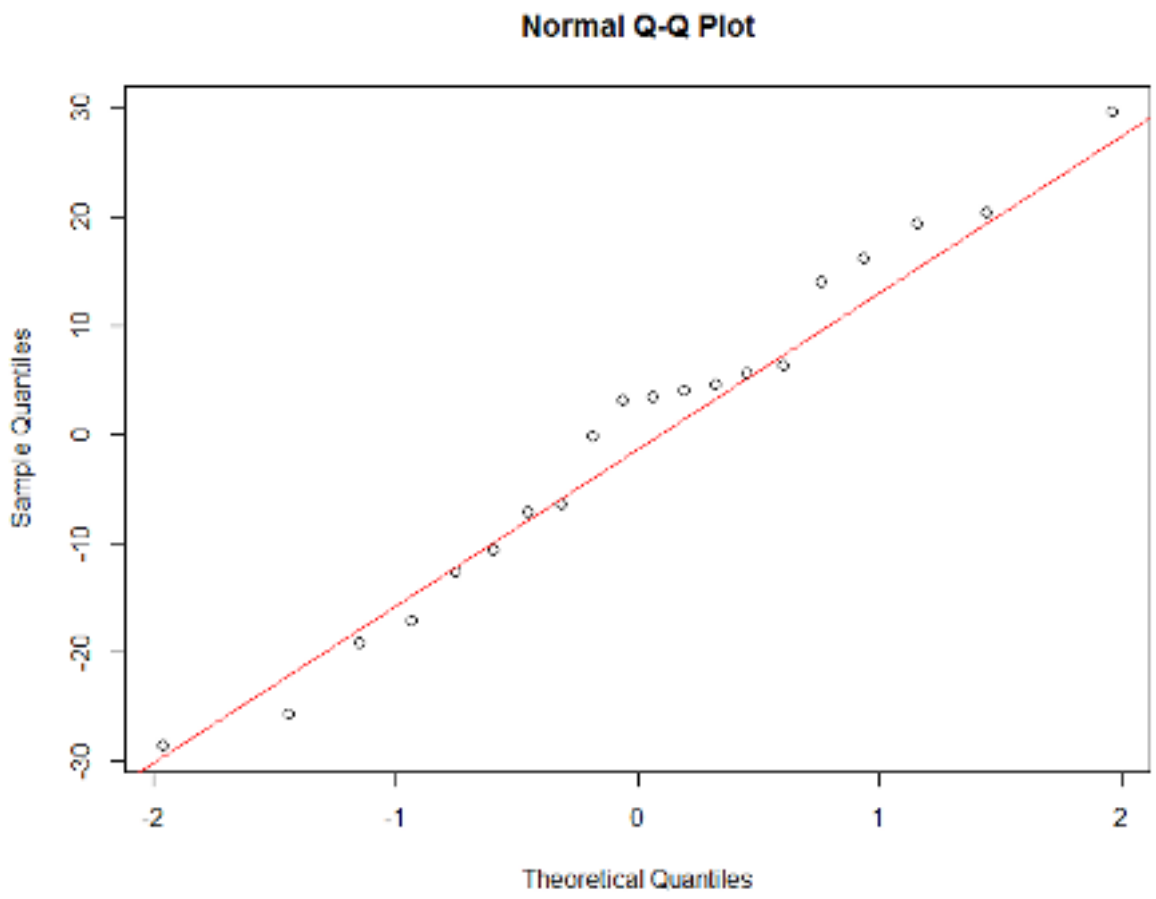
De plus, le coefficient étant positif, on sait que le temps de production est fonction croissante du nombre d'unités produites.

```
> cor(TP3$temps_exec,TP3$unites)
[1] 0.8545206
> regression = lm(temps_exec ~ unites, data = TP3)
> regression

Call:
lm(formula = temps_exec ~ unites, data = TP3)

Coefficients:
(Intercept)       unites
   149.7477       0.2592
```

Les coefficients, calculés à l'aide de la fonction *lm* donnent l'équation :

Temps_execution = 149.7 x Unités_produites + 0.26

Question 2



On observe graphiquement que la qqline est proche de la distribution des résidus. Ce qui nous permet de déduire que la répartition des résidus suit presque une loi normale.

```
> qqnorm(tableau$residus)
> parnew=TRUE
> qqline(residus)
> qqnorm(tableau$residus)
> parnew=TRUE
> qqline(residus,col='red')
```

Code du tracé

On réalise un test de Shapiro (Hypothèse : les résidus sont répartis selon une loi normale) :

```
> shapiro.test(residus)

        Shapiro-Wilk normality test

data:  residus
W = 0.9771, p-value = 0.8917
```

La p-value vaut 0.89 ce qui signifie qu'il est fort probable que les résidus soient répartis selon une loi normale. On conserve donc l'hypothèse.

Question 3

Code préliminaire :

```
> prevision = data.frame(unites = seq(50,500, by = 10))
> bandeconfiance = predict(regression,int = "c", newdata = prevision)
> bandeprecision=predict(regression, int = 'p', newdata = prevision)
```

```
> plot(bandeconfiance)
> plot(bandeconfiance$fit,prevision)
Error in bandeconfiance$fit : $ operator is invalid for atomic vectors
> plot(bandeconfiance$fit,prevision$unites)
Error in bandeconfiance$fit : $ operator is invalid for atomic vectors
> bandeconfiance$fit
Error in bandeconfiance$fit : $ operator is invalid for atomic vectors
> attach(bandeconfiance)
Error in attach(bandeconfiance) :
  'attach' only works for lists, data frames and environments
> matline(bandeconfiance)
Error: could not find function "matline"
> matlines(bandeconfiance)
> matlines(bandeconfiance$fit)
Error in bandeconfiance$fit : $ operator is invalid for atomic vectors
>
```

Erreurs lors du tracé des courbes. Bandeconfiance donne 3 valeurs : prédite, bornes inférieures et supérieures de l'intervalle de confiance – en fonction de différentes valeurs du nombre d'unités produites (de 50 à 500).