# TP- Aide

## Aide TP

### Test environnement travail

```r
#test package irene
villes <- read.csv('./DonneesGPSvilles.csv',header=TRUE,dec='.',sep=';',quote="\"")
coord <- cbind(villes$longitude,villes$latitude)
dist <- distanceGPS(coord)
voisins <- TSPnearest(dist)
print(voisins)
```

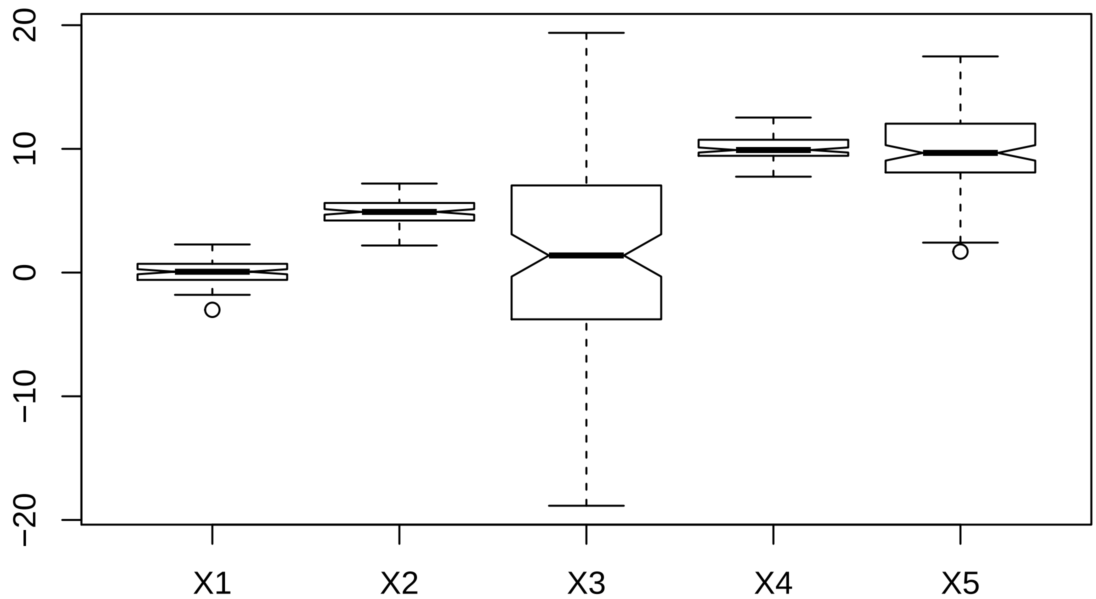
```
## $longueur
## [1] 4303.568
##
## $chemin
##  [1]  1  8 11 18 15 19  6 20  3 10 17 16  7 13 21  4  9 14 12  2 22  5
```

En théorie vous devez avoir obtenu cette sortie. Si non, vous ne pourrez pas faire le TP.

### Question

#### 1/ SI ensemble de 5 vecteur comment faire matrice avec 5 colonne pour boxplot

```r
X1 <- rnorm(100)
X2 <- rnorm(100,5)
X3 <- rnorm(100,0,8)
X4 <- rnorm(100,10)
X5 <- rnorm(100,10,3)


mat <- cbind(X1,X2,X3,X4,X5)
par(mfrow=c(1,1))
boxplot(mat,notch=TRUE)
```




**2/ Somme de loi normale (pour le t-test branch nearest voir TD 2, exo 3, q2)**

Si $X_1 \sim \mathcal{N}(m_1, \sigma_1^2)$ et $X_2 \sim \mathcal{N}(m_2, \sigma_2^2)$ alors

$$Yplus = X_1 + X_2 \sim N(m_1 + m_2, sqrt(\sigma_1^2 + \sigma_2^2)$$

et

$$Yminus = X_1 + X_2 \sim N(m_1 - m_2, sqrt(\sigma_1^2 + \sigma_2^2)$$

```
X1 <- rnorm(1000,mean=1,sd= 1)
X2 <- rnorm(1000,mean=2,sd= 0.5)

t.test(X1-X2)
```

```
##
##   One Sample t-test
##
## data:  X1 - X2
## t = -29.434, df = 999, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -1.0825527 -0.9472271
## sample estimates:
## mean of x
##  -1.01489
```

```
t.test(X1,X2,paired=TRUE)
```

```
##
##   Paired t-test
##
## data:  X1 and X2
## t = -29.434, df = 999, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.0825527 -0.9472271
## sample estimates:
## mean of the differences
##                -1.01489
```

Le test pairé est le même que celui fait sur la différence.

**3/ microBenchmark**

Tout les membres d'un meme groupe n'ont pas de différence significative pour leurs moyenne et les groupes $\{a, b, c, d, \dots\}$ sont rangés de manière croisante.

Exemple - si variables X et variable Y sont dans le groupe $a$ alors $m_X \simeq m_Y$ où plutot qu'il n'a pas pu être mis en évidence une différence significative entre les deux. - si variables X et variable Y sont dans le groupe $a$ et **b** alors $m_X \neq m_Y$ significativement. Et comme $\{a, b, c, d, \dots\}$ sont rangés de manière croisante alors $m_a < m_b$ donc $m_X < m_Y$

## Regression

### Modèle linéaire univarié

$$Y = aX + b + \epsilon$$

où $\epsilon \sim \mathcal{N}(0, \sigma^2)$'

```
  varNoise <- 2
```

- Exemple $Y = 15X + 1 + \epsilon$ où $\epsilon \sim \mathcal{N}(0, 2)$

```
n <- 100
 X <- runif(n)
a <- 15
b <- 1
noi <- rnorm(n,0,varNoise)
yobs <- a*X +b + noi
```

Ici X suit une loi uniforme. En effet aucun a priori sur X n'est nécessaire mais Y et X doivent être linéairement corrélés.

```
  mod <- lm(yobs~X) # estimation modèle linéaire par regression des moindres carrés
  mod
```

```
##
## Call:
## lm(formula = yobs ~ X)
##
## Coefficients:
## (Intercept)            X
##      0.7408      15.7217
```

```
  estB <- mod$coefficients[1]  #  b soit non nul significativement (h0 pas d'intercept)
  estA <- mod$coefficients[2] #  a soit non nul significativement
  #(H0 : modéle linaire -> qualité modele)
```

**lm**($y \sim X$) applique régression sur Y expliqué par X. L'intercept correspond à **b**. Le coefficient à **a**. (Si on est en multivarié il y a plusieurs coefficients)

```
# residu : bruit du modele  + partie non expliquée de yobs
# pour que modele soit bon residu = bruit (ou presque)
yhat <- estA * X + estB
res <- yobs - yhat # doivent suivre un loi gaussienne
```

Les résidus $r = y - (\hat{a}X + \hat{b}) = y - \hat{y}$.

**TEST DU MODELE**

test du modèle : - tester significativité de a (pertinence du modele)

$$(H_0) : a = 0 \text{ contre } (H_1) : a \neq 0$$

- tester significativité de b (besoin d'un intercept) : Optionnel car ne renseigune pas sur la pertinence du modèle mais simplement pour savoir si intercept utile ou non

$$(H_0) : b = 0 \text{ contre } (H_1) : b \neq 0$$

- tester residus gaussien (modele a bien fitté ou non). En théorie ne reste que résidus gaussien ou presque.

$$(H_0) : \text{Résidus suivent loi normale contre } (H_1) : \text{Résidus ne suivent pas loi normale}$$

* Test Linéarité

```
summary(mod)
```

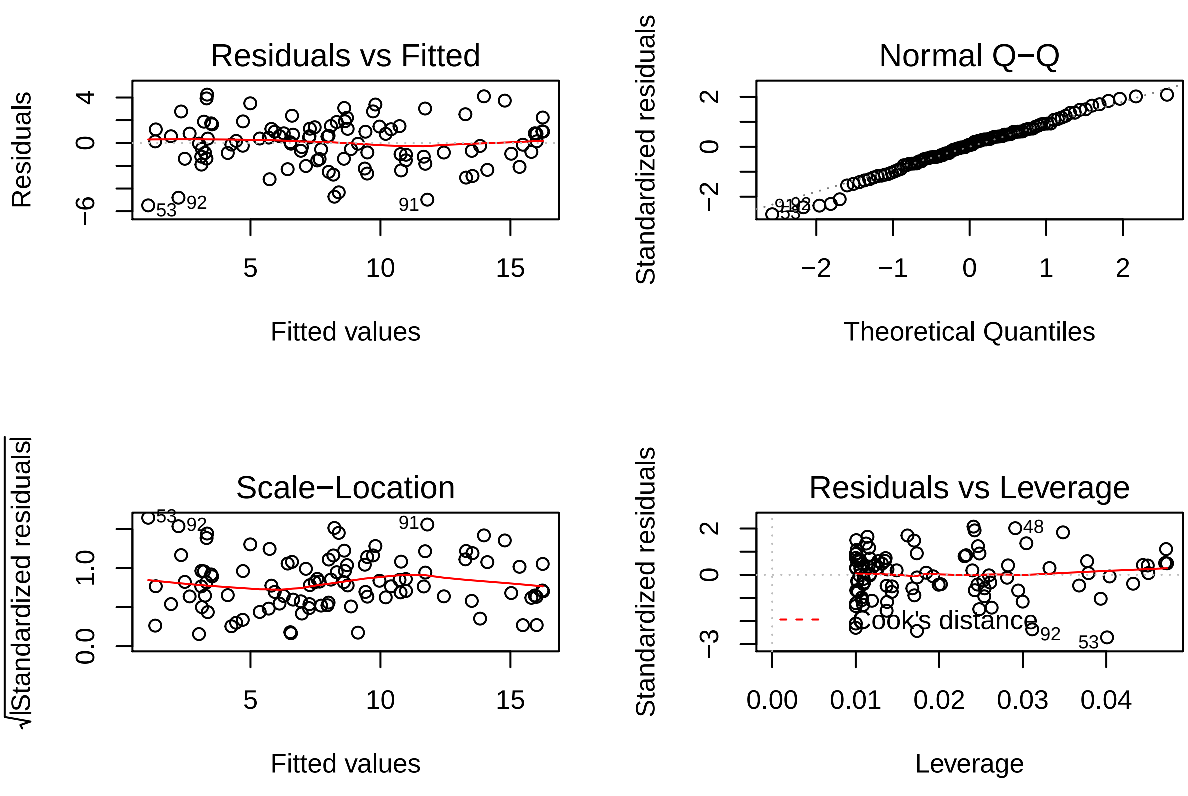
```
##
## Call:
## lm(formula = yobs ~ X)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.483 -1.243  0.145  1.251  4.254
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.7408     0.4285   1.729    0.087 .
## X            15.7217     0.7851  20.026   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 2.069 on 98 degrees of freedom
## Multiple R-squared:  0.8036, Adjusted R-squared:  0.8016
## F-statistic:   401 on 1 and 98 DF,  p-value: < 2.2e-16
```

- $Pr(>|t|)$ :p-value pour test sur **b** pour **(Intercept)**
- $Pr(>|t|)$ :p-value pour test sur **coefficients** pour **X** et cie. . . . Pour chaque coefficient du modèle (ici un seul qui est $a$) on a la p-value
- $p - value$ : p-value pour modèle complet (ensemble des coefficients). Si modèle univarié (comme dans exemple) $p - value = Pr(>|t|)$ pour **X**
- Test Résidus

** Graphiquement

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




- Residuals vs Fitted : Si horizontal et homogene alors linearité et pas d'effet d'echelle
- Normal Q-Q : Compare distribution des residus par rapport a distribution normale. Un point correspond a un rapport des valeurs des memes quantile obtenus pour les deux distribution. Par exemple le point central fait le ratio entre les quantiles $q_{res}$ et $q_{norm}$ tel que $P(X_{res} > q_{res}) = P(X_{norm} > q_{norm}) = q_i$ où $q_i = 50\%$. Si les distribution sont identiques ou presque alors l'ensemble des points sont sur la diagonale. Sinon on observera la plupart du temps des deviation aux extremité ce qui sous-entend que les queues de distribution sont différentes.

- Scale location : Idem qur Residuals vs Fitted mais avec résidus normalisés.
- Residuals vs Leverage : Montre l'influence des echantillons (plus un point est a droite et plus il en a). Si un point est un outliers il apparaitra trés éloigné des autres et en dehors des bornes par rapport à la distance de Cook.

** test sur résidus (shapiro)

```
#permet de voir graphiquement si ok
shapiro.test(residuals(mod)) # test bruit gaussien : H0 suit loi normale
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(mod)
## W = 0.98557, p-value = 0.349
```

Pour que le modèle soit OK - coefficients significativment non nul - résidus gaussien

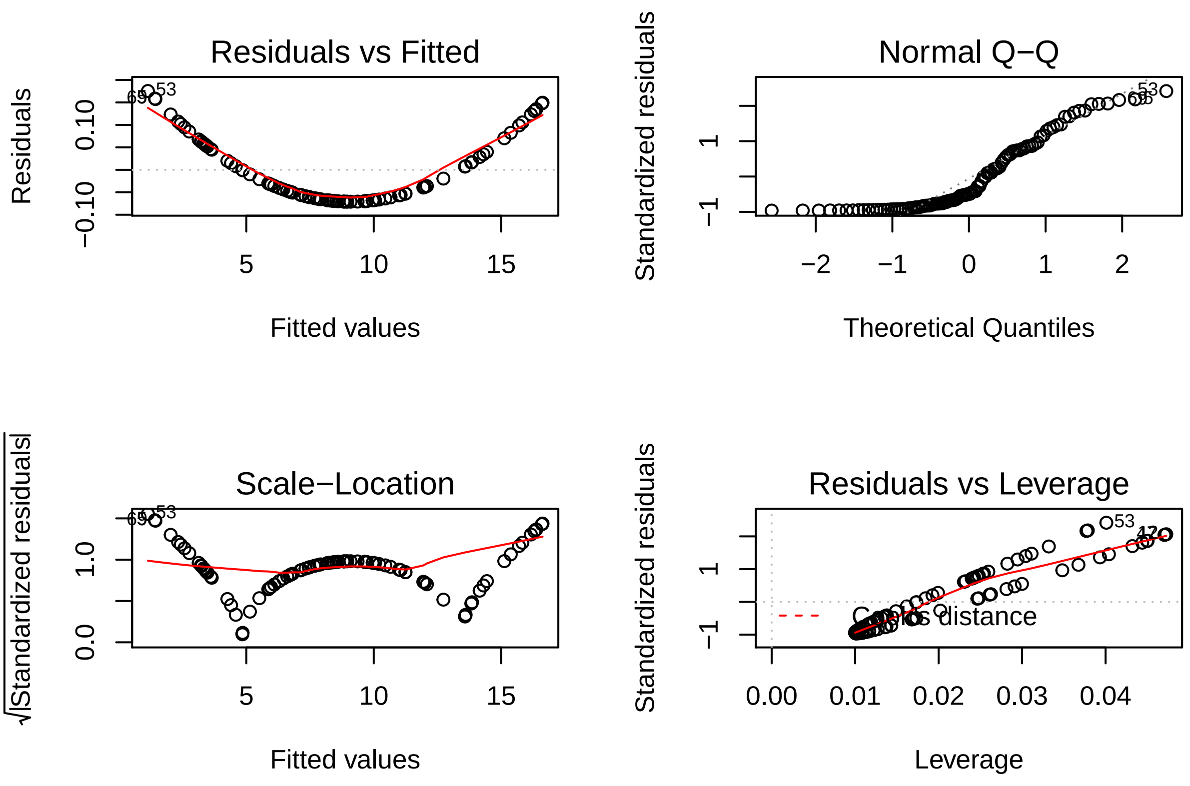
## Regression NOK

```
noi2 <- X^2
yobs2 <- a*X +b + noi2
mod2 <- lm(yobs2~X)
estB <- mod2$coefficients[1]  #  b soit non nul significativement (h0 pas d'intercept)
estA <- mod2$coefficients[2] #  a soit non nul significativement
#(H0 : modéle linaire -> qualité modele)



# residu : bruit du modele  + partie non expliquée de yobs
# pour que modele soit bon residu = bruit (ou presque)
yhat2 <- estA * X + estB
res2 <- yobs2 - yhat2 # doivent suivre un loi gaussienne


#tester significativité de a (pertinence du modele) ->
#H0  a = 0 ; permet de dire si corrélation lineaire de X avec Y
#tester significativité de b (besoin d'un intercept) ->
#H0 b=0; informatif pour interet de l'intercept
#tester residus gaussien (modele a bien fitté ou non) ->
#H0 bruit gaussien; necessaire pour savoir si
# le modele predit bien y (ne reste que résidus gaussien ou presque)

summary(mod2)
```

```
##
## Call:
## lm(formula = yobs2 ~ X)
##
## Residuals:
```

```
##       Min       1Q   Median       3Q      Max
## -0.07101 -0.06392 -0.03614  0.05496  0.17552
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.80345    0.01535   52.33   <2e-16 ***
## X           16.03451    0.02813  570.01   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07414 on 98 degrees of freedom
## Multiple R-squared:  0.9997,  Adjusted R-squared:  0.9997
## F-statistic: 3.249e+05 on 1 and 98 DF,  p-value: < 2.2e-16
```

```r
par(mfrow=c(2,2))
plot(mod2)
```




```r
#permet de voir graphiquement si ok


shapiro.test(residuals(mod2)) # test bruit gaussien : H0 suit loi normale
```

```
##
##  Shapiro-Wilk normality test
##
```

```
## data:  residuals(mod2)
## W = 0.84963, p-value = 1.149e-08
```

## Regression Multivarié

### Modèle linéaire univarié

$$Y = \beta X + b + \epsilon$$

où $\epsilon \sim \mathcal{N}(0, \sigma^2)$‘

```
varNoise <- 2
```

- Exemple $Y = 15X_1 + 3X_2 + 9X_3 + +0\epsilon$ où $\epsilon \sim \mathcal{N}(0, 2)$

```
n <- 100
 X1 <- runif(n)
 X2 <- rnorm(n,5)
 X3 <- rnorm(n,0,8)
 X4 <- rnorm(n,50,8)
 X = cbind(X1,X2,X3,X4)
a <- 15
b <- 10
c <- 9
d <- 0
varNoise <-0.5
noi <- rnorm(n,0,varNoise)
yobsM <- a*X[,1]+b*X[,2]+c*X[,3]+d*X[,4] + noi
```

```
modM <- lm(yobsM~X) # estimation modèle linéaire par regression des moindres carrés
modM
```

```
##
## Call:
## lm(formula = yobsM ~ X)
##
## Coefficients:
## (Intercept)          XX1          XX2          XX3          XX4
##   -0.494426    14.872983    10.056307     8.999995     0.005105
```

```
summary(modM)
```

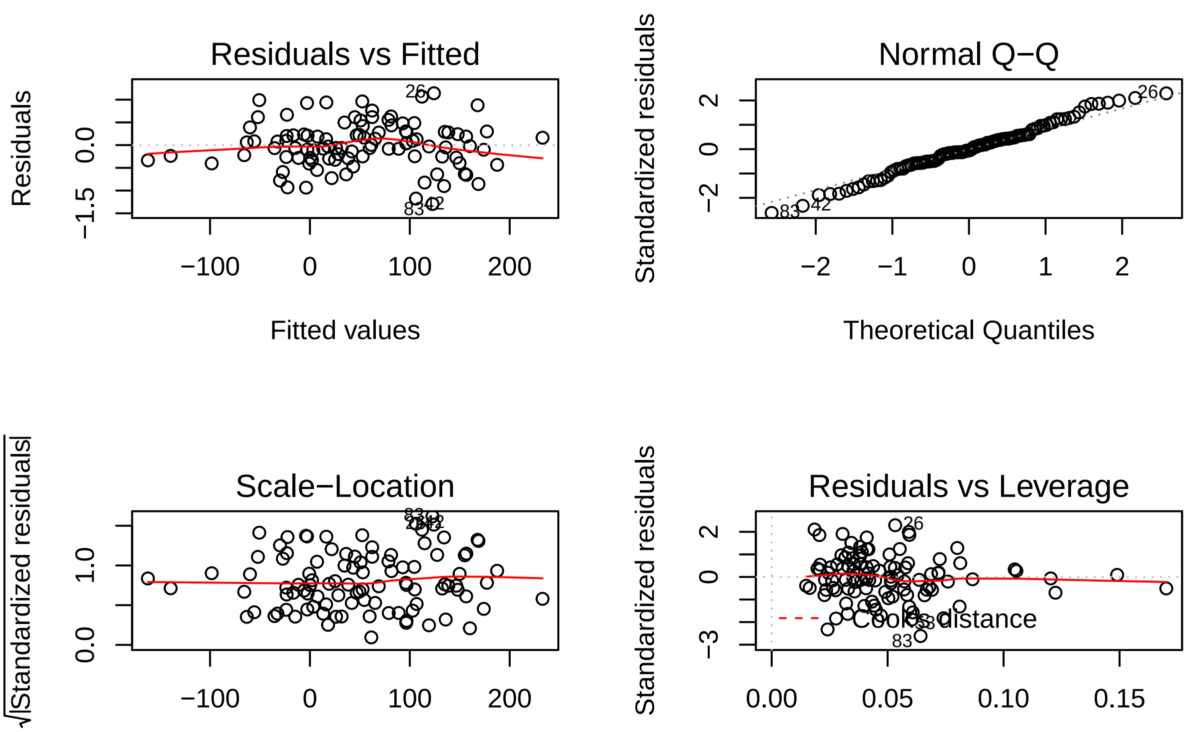
```
##
## Call:
## lm(formula = yobsM ~ X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.29570 -0.28443 -0.02622  0.28105  1.14091
##
## Coefficients:
```

```
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept) -0.494426   0.487359   -1.015    0.313
## XX1         14.872983   0.173052   85.945   <2e-16 ***
## XX2         10.056307   0.060347  166.643   <2e-16 ***
## XX3          8.999995   0.006389 1408.616   <2e-16 ***
## XX4          0.005105   0.006787    0.752    0.454
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5118 on 95 degrees of freedom
## Multiple R-squared:      1,	Adjusted R-squared:      1
## F-statistic: 5.341e+05 on 4 and 95 DF,  p-value: < 2.2e-16
```

```
  #(HO : modéle linaire -> qualité modele)
```

**lm**($y \sim X$) applique régression sur Y expliqué par X. L'intercept correspond à **b**. On est en multivarié il y a plusieurs coefficients. **Ici X4 n'est pas significative pour le modèle.**

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




```
shapiro.test(residuals(modM))
```

```
##
```

```
##  Shapiro-Wilk normality test
##
## data:  residuals(modM)
## W = 0.99065, p-value = 0.7169
```

#permet de voir graphiquement si ok