UNIVERSITÉ NATIONALE DE HO CHI MINH VILLE
UNIVERSITÉ DES SCIENCES NATURELLES
FACULTÉ DES TECHNOLOGIES DE L'INFORMATION

PROJET

(TRAITER DES CAS CONCRETS D'ANALYSE DE DONÉES)

Groupe 10

Nom et Prénom	Code d'étudiant
Phạm Quang Huy	20126016
Nguyễn Huỳnh Mẫn	20126041
Thiều Vĩnh Trung	20126062

ANALYSE STATISTIQUE MULTIVARIÉE

Ho Chi Minh ville – 2022

UNIVERSITÉ NATIONALE DE HO CHI MINH VILLE
UNIVERSITÉ DES SCIENCES NATURELLES
FACULTÉ DES TECHNOLOGIES DE L'INFORMATION

PROJET

(TRAITER DES CAS CONCRETS D'ANALYSE DE DONÉES)

| Instructeurs |

Mme. Nguyễn Thị Mộng Ngọc
M. Nguyễn Văn Thìn

ANALYSE STATISTIQUE MULTIVARIÉE

Ho Chi Minh ville – 2022

INTRODUIRE

Une bonne recherche, un rapport scientifique acceptable sur le plan académique, nécessite une bonne méthodologie, l'application d'outils techniques pour fournir des informations authentiques. Surtout dans les questions socio-économiques et lors de l'étude de grands nombres, nous devons prêter attention aux outils techniques tels que les statistiques.

La statistique est un domaine assez large, aussi dans le cadre de cette matière, il est souhaité de doter les apprenants de connaissances de base en analyse statistique afin de pouvoir exploiter efficacement les informations collectées au service de la recherche scientifique des sciences socio-économiques .

Le concept d'analyse statistique : C'est une combinaison de statistiques, de réflexion et de compréhension des problèmes économiques.

Exigences : Pour être en mesure de maîtriser les connaissances de ce sujet, les apprenants doivent avoir une connaissance approfondie des statistiques, de l'économie ainsi qu'une compréhension pratique du problème de recherche. De plus, une connaissance de l'informatique et d'autres outils quantitatifs est requise pour être intégrée à la recherche.

Dans ce projet, notre équipe souhaite présenter les méthodes de test du modèle, comment améliorer le modèle et analyser les données. Dans ce projet, il y aura 3 problèmes à résoudre pour nous aider à mieux comprendre le processus de résolution de problèmes ainsi qu'à acquérir une compréhension plus approfondie du modèle de régression linéaire, du test de Fisher et de l'analyse ANOVA.

TABLE DES MATIÈRES

Modèle de régression linéaire multiple (sujet de votre choix)

1. Présentation du sujet

Construire une modele qui prédit le poids des poissons vendus sur le marché.

Avec l'ensemble de données fourni comme suit :

- Weight: poids (gram) => weight of fish in Gram g
- Length1: longueur verticale (cm) => vertical length in cm
- Length2: longueur diagonale (cm) => diagonal length in cm
- Length3: longueur de croix (cm) => cross length in cm
- Height: taille (cm) => height in cm
- Width: largeur diagonale (cm) => diagonal width in cm

2. Afficher la modèle d'origine

```
data_acc <- read.csv('Fish.CSV', header=TRUE, sep=",")
data_acc
# afficher les donnes

summary(lm(..Weight ~ ., data = new_data_acc))
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -499.587     29.572 -16.894  < 2e-16 ***
Length1       62.355     40.209   1.551  0.12302
Length2       -6.527     41.759  -0.156  0.87601
Length3      -29.026     17.353  -1.673  0.09643 .
Height        28.297      8.729   3.242  0.00146 **
Width         22.473     20.372   1.103  0.27169
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 123.2 on 153 degrees of freedom
Multiple R-squared:  0.8853,    Adjusted R-squared:  0.8815
F-statistic: 236.2 on 5 and 153 DF,  p-value: < 2.2e-16
```

Nous choisissons le modèle en fonction des données d'entrée.

- Après avoir utilisé la commande summary(), nous commentons le modèle d'origine, il semble que le modèle n'ait aucun sens.

Nous pouvons voir que, bien que la valeur de R-carré ajusté : 0,8815 soit assez élevée, il reste des colonnes qui ne sont pas marquées d'un astérisque (*), cela signifie que

la p-valu de ces colonnes est supérieure à 0,1 => cela signifie que notre modèle à 5 variables n'est pas significatif pour prédire le poids .quantité.

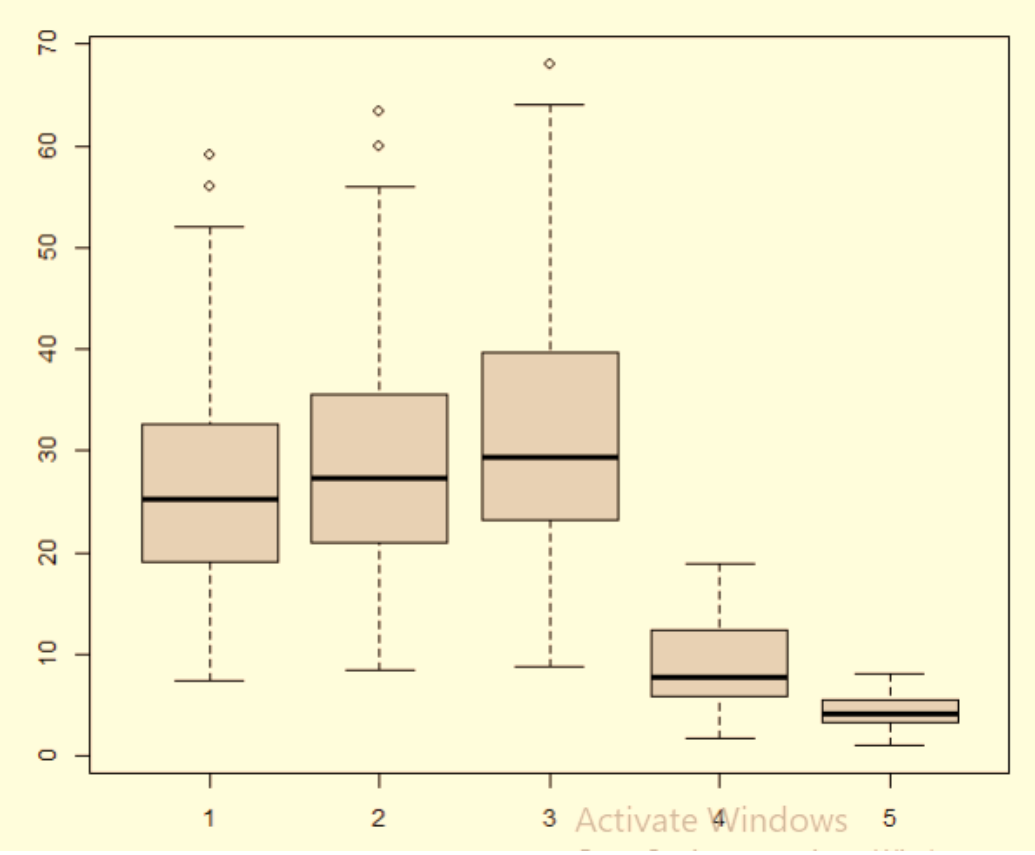
→ Il est possible qu'il y ait des défauts dans les données d'entrée d'origine. Nous devons vérifier cela.

3. Visualisation des données

La première chose que nous pouvons faire est de voir si nos données ont des valeurs aberrantes.

Nous devons boxplot() sur les 5 variables de nos données pour avoir un aperçu de notre jeu de données.

```
op<-par(mfrow = c(1,1))
boxplot(data_acc$Length1,data_acc$Length2,data_acc$Length3,data_acc$Height,data_acc$Width)
```

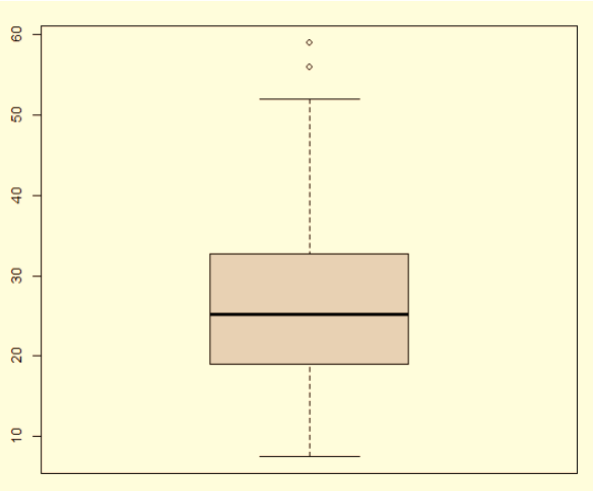


Selon le graphique ci-dessus, nous pouvons voir que nos colonnes ont la présence de valeurs aberrantes. Exactement comme nous l'espérions.

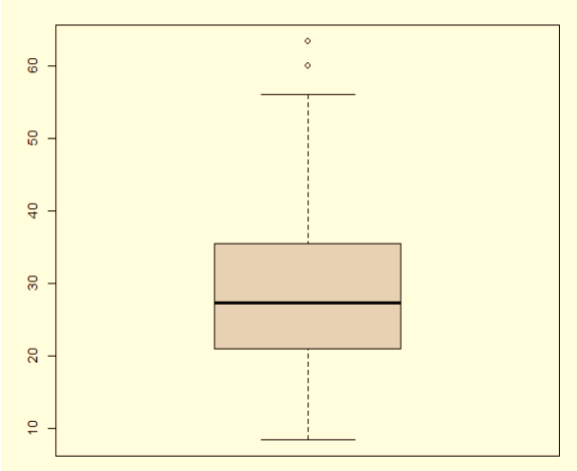
Besoin d'une vue plus claire des colonnes des données. Le but est de localiser les valeurs aberrantes dans les données et de les supprimer.

Utilisez boxplot() sur chaque variable des données.

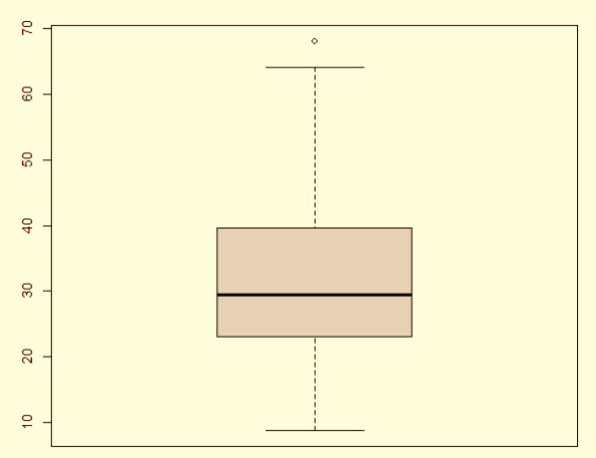
```
# visualiser le jeu de donnees
boxplot(data_acc$Length1)
which(data_acc$Length1>55)
#[1] 143 144 145

boxplot(data_acc$Length2)
which(data_acc$Length2>59) #[1] 143 144 145


boxplot(data_acc$Length3)
which(data_acc$Length3>65) #[1] 145

boxplot(data_acc$Height) # La Height colonne n'a pas de valeurs aberrantes
boxplot(data_acc$Width)  # La Width colonne n'a pas de valeurs aberrantes
```



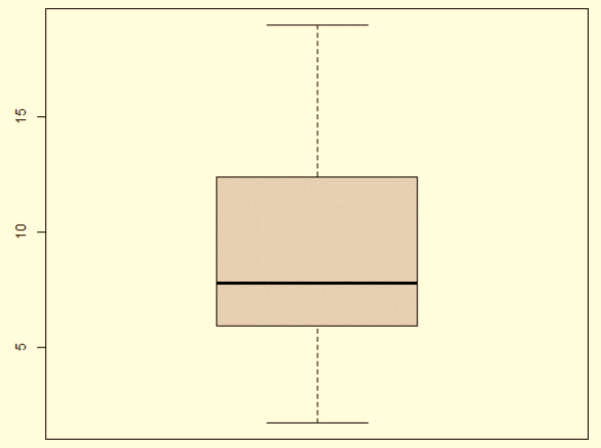
Colonne de valeur Longueur1



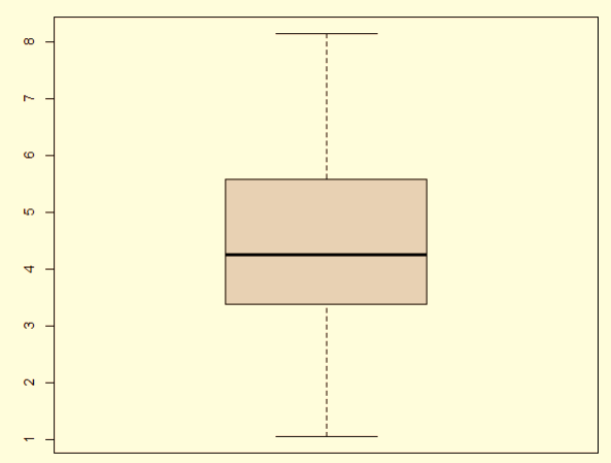
Colonne de valeur Longueur2



Colonne de valeur Length3



Colonne de valeur Hauteur



Colonne de valeur Largeur

4. Éliminer les valeurs aberrantes

Une fois que nous connaissons l'emplacement des valeurs aberrantes, nous procédons à leur suppression de nos données.

```
new_data_acc <- data_acc[-c(143,144,145),]
#supprimer les valeurs aberrantes dans les lignes 143,144,145
#recharger le tableau des statistiques pour verifier si le modele est toujours bon
new_data_acc
dim(new_data_acc)
#verifier le sous-trait et le nombre d'observations dans les donnees
```

Vérifiez à nouveau si ces valeurs ont été supprimées.

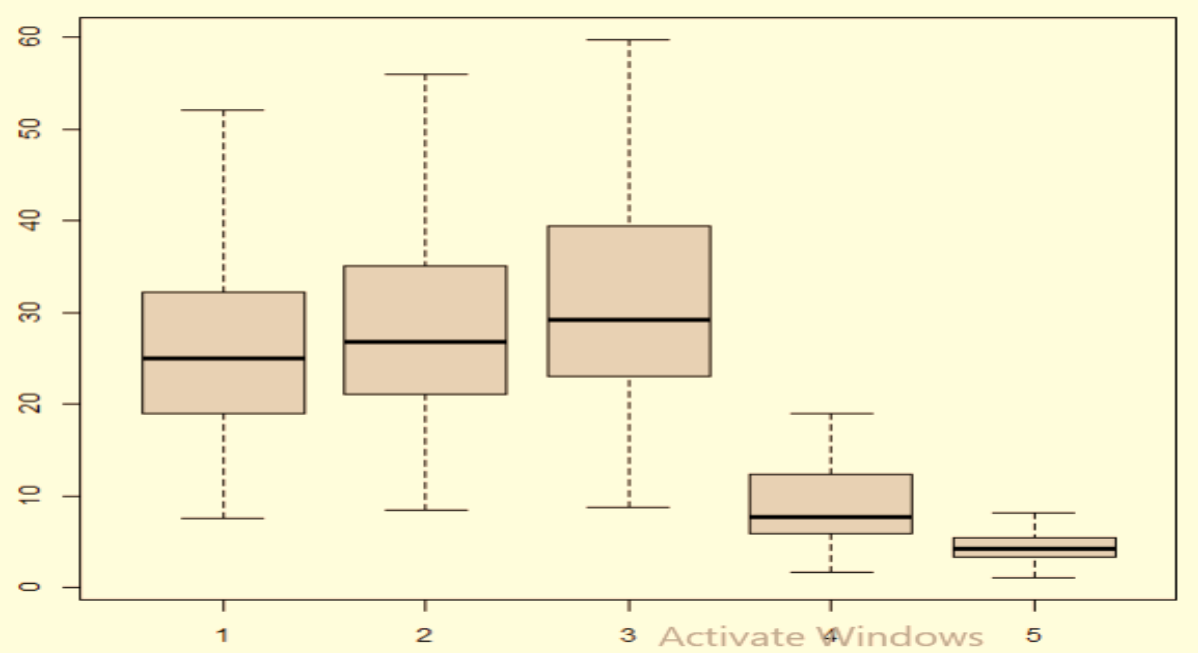
```
> dim(new_data_acc)
[1] 156   6
> dim(data_acc)
[1] 159   6
>
```

On confirme que nous avons éliminé les valeurs aberrantes que nous venons de trouver.

5. Deuxième vérification des valeurs aberrantes

Étant donné que le nombre de valeurs aberrantes supprimées est inférieur à 5 % de nos observations totales, le modèle ne sera pas affecté. Pour être plus approfondi, vérifions s'il y a des valeurs aberrantes dans notre ensemble de données.

```
boxplot(new_data_acc$Length1,new_data_acc$Length2,new_data_acc$Length3,new_data_acc$Height,new_data_acc$Width)
# visualiser le jeu de donnees encore
```




```
boxplot(new_data_acc$Length1)
boxplot(new_data_acc$Length2)
boxplot(new_data_acc$Length3)
boxplot(new_data_acc$Height)
boxplot(new_data_acc$Width)
#toutes les valeurs aberrantes ont ete supprimees et le nombre est inferieur a 5% du total des observations de la base de donn◆es
```

Toutes les valeurs aberrantes ont été complètement supprimées du jeu de données.

6. Modélisation des ensembles de données nettoyés

Une fois les données nettoyées, nous utilisons la régression linéaire pour modéliser le nouvel ensemble de données.

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -426.943     25.576 -16.693  < 2e-16 ***
Length1      102.558     33.218   3.087  0.00241 **
Length2      -45.076     34.413  -1.310  0.19224
Length3      -37.148     14.293  -2.599  0.01028 *
Height        36.847      7.254   5.079 1.11e-06 ***
Width         52.402     17.113   3.062  0.00261 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 100.7 on 150 degrees of freedom
Multiple R-squared:  0.9038,    Adjusted R-squared:  0.9006
F-statistic:   282 on 5 and 150 DF,  p-value: < 2.2e-16
```

```
view <- lm(◆..Weight ~ ., data = new_data_acc) # charger le modele lineaire
summary(view)# afficher le tableau statistique
```

On peut commenter, le modèle a été amélioré. Maintenant, la valeur R-carré ajusté : 0,9006. Cela signifie que le modèle est meilleur que l'original.

Mais je ne peux toujours pas dire que c'est un bon modèle car la colonne Length2 a une valeur de 0,19224 > 0,5. La plupart des p-values des colonnes sont inférieures à 0,01. Cela peut signifier que la colonne n'a aucune signification dans notre modèle.

Pour tester cela, nous utilisons la méthode AIC pour estimer la qualité du modèle => à travers laquelle nous pouvons choisir le meilleur modèle.

7. Méthode AIC

Utiliser la méthode AIC pour sélectionner le meilleur modèle à partir de l'ensemble de données traitées.

```
#AIC
# Choisissez le meilleur modele
step(view, direction = "backward")
```

```
Step:  AIC=1444.72
ï..Weight ~ Length1 + Length3 + Height + Width

          Df Sum of Sq     RSS    AIC
<none>                 1539189 1444.7
- Width    1     78784 1617973 1450.5
- Length3  1    116398 1655588 1454.1
- Length1  1    186191 1725380 1460.5
- Height   1    277501 1816690 1468.6

Call:
lm(formula = ï..Weight ~ Length1 + Length3 + Height + Width,
    data = new_data_acc)

Coefficients:
(Intercept)      Length1      Length3       Height        Width
    -431.98        63.64       -44.51        37.76        45.06
```

A partir de là, on choisit le meilleur modèle avec 4 variables : Longueur1, Longueur3, Hauteur, Largeur.

Cela confirme que l'hypothèse selon laquelle la colonne Length2 ne sera pas significative dans le modèle est vraie.

Après avoir exécuté step() sur l'ensemble de données traité, nous choisissons le meilleur modèle de régression linéaire avec la formule :Weight ~ Length1 + Length3 + Height + Width.

Régression linéaire => y = -431.98 + 63.64(Length1) - 44.51(Length3)

+37.76(Height) + 45.06(Width).

8. Méthode BIC

Utilisez la méthode BIC pour confirmer si le modèle sélectionné ci-dessus est le meilleur modèle ou non.

Parce que la probabilité de choisir le bon modèle de cette méthode est supérieure à celle de la méthode AIC. Alors peut-être qu'avec cette méthode, nous pouvons avoir un meilleur modèle plus petit que le modèle sélectionné dans la méthode AIC.

```
#BIC
# Choisissez le meilleur modele
n = length(resid(view))
step(view,direction = "backward",k = log(n))
```

```
Step:  AIC=1459.97
i..Weight ~ Length1 + Length3 + Height + Width

          Df Sum of Sq     RSS    AIC
<none>                 1539189 1460.0
- Width    1     78784 1617973 1462.7
- Length3  1    116398 1655588 1466.3
- Length1  1    186191 1725380 1472.7
- Height   1    277501 1816690 1480.8

Call:
lm(formula = i..Weight ~ Length1 + Length3 + Height + Width,
    data = new_data_acc)

Coefficients:
(Intercept)      Length1      Length3       Height        Width
    -431.98        63.64       -44.51        37.76        45.06
```

Après avoir sélectionné le nombre d'observations de l'ensemble de données traité et exécuté step() selon la méthode BIC sur l'ensemble de données traité, nous obtenons toujours le meilleur modèle de régression linéaire avec la formule: Weight ~ Length1 + Length3 + Height + Width.

Régression linéaire => y = -431.98 + 63.64(Length1) - 44.51(Length3)

+37.76(Height) + 45.06(Width).

À partir de là, nous pouvons confirmer que la formule ci-dessus avec un nombre variable de 4 (Longueur1, Longueur3, Hauteur, Largeur) est notre meilleur modèle.

9. Meilleure modélisation d'ensemble de données

```
view2 <- lm(..Weight ~ Length1 + Length3 + Height + Width, data = new_data_acc) # charger le meilleur modele lineaire
summary(view2)# afficher le tableau statistique
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-199.14  -66.46  -30.64   63.20  311.66

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -431.981     25.345 -17.044  < 2e-16 ***
Length1       63.641     14.891   4.274 3.39e-05 ***
Length3      -44.511     13.172  -3.379 0.000925 ***
Height        37.763      7.238   5.218 5.89e-07 ***
Width         45.056     16.207   2.780 0.006126 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 101 on 151 degrees of freedom
Multiple R-squared:  0.9027,    Adjusted R-squared:  0.9002
F-statistic: 350.4 on 4 and 151 DF,  p-value: < 2.2e-16
```

En observant le tableau de statistiques traité et amélioré ci-dessus, nous pouvons confirmer qu'il s'agit de notre meilleur modèle pour expliquer le poids, toutes les colonnes sont significatives, toutes les valeurs sont inférieures à 0,001.

La valeur du R-carré ajusté : 0,9002, est proche de 1. Cette valeur s'est considérablement améliorée par rapport à l'ensemble de données d'origine fourni.

10. Calculez la valeur résiduelle et voyez si le nouveau modèle suit une distribution normale

```
shapiro.test(residuals(view2))
# Le modele ne suit pas une distribution normale
```

```
        Shapiro-Wilk normality test

data:  residuals(view2)
W = 0.93753, p-value = 2.298e-06
```

Utilisez shapeiro.test() pour voir si les résidus du nouveau modèle suivent une distribution normale.

Nous obtenons le p-valu = 2.298e-06 < 0.5 => Cela signifie que le modèle ne suit pas une distribution normale.

11. Conclusion

Au départ, on nous donne une donnée. Après avoir modélisé les données, nous avons remarqué que le modèle des données d'origine n'était pas bon.

Faites le pas pour voir si l'ensemble de données a des valeurs aberrantes, afin de les supprimer, afin d'amener les observations à un certain modèle.

Modélisez à nouveau l'ensemble de données pour la deuxième fois. Le modèle s'est considérablement amélioré, mais il reste encore de la place pour d'autres améliorations. Nous remarquons que l'une des cinq colonnes du jeu de données (Length2) peut ne pas être significative pour ce modèle. Passez à l'étape suivante du test.

Adoptez la méthode AIC et BIC. Nous pouvons confirmer que la prédiction ci-dessus est correcte. Continuez à supprimer cette colonne non significative (Length2) du modèle et nous avons le meilleur modèle.

Modélisez le meilleur ensemble de données. Nous considérons ce modèle comme notre meilleur modèle. Et il peut prédire pour notre colonne Poids. Passez à l'étape suivante.

Vérifiez si le modèle que nous avons construit suit une distribution normale. Nous constatons que ce modèle ne suit pas une distribution normale. Malheureusement, à ce stade, nous ne pouvons pas améliorer davantage le modèle.

Nous concluons qu'il s'agit du meilleur modèle de prédiction du poids des poissons vendus sur le marché que nous puissions construire.

Deux cas études

I. Choix du modèle Exercice: Taux d'accidents

1. Afficher la modèle d'origine

```
data_acc <- read.csv('tauxaccidents.CSV', header=TRUE, sep=",")

modele_1 <- lm(y_i~.,data=data_acc)
```

Nous choisissons le modèle en fonction des données d'entrée.

- Après utiliser la commande summary(), on retrouve le modèle d'origine, les variables semblent n'avoir aucune signification.

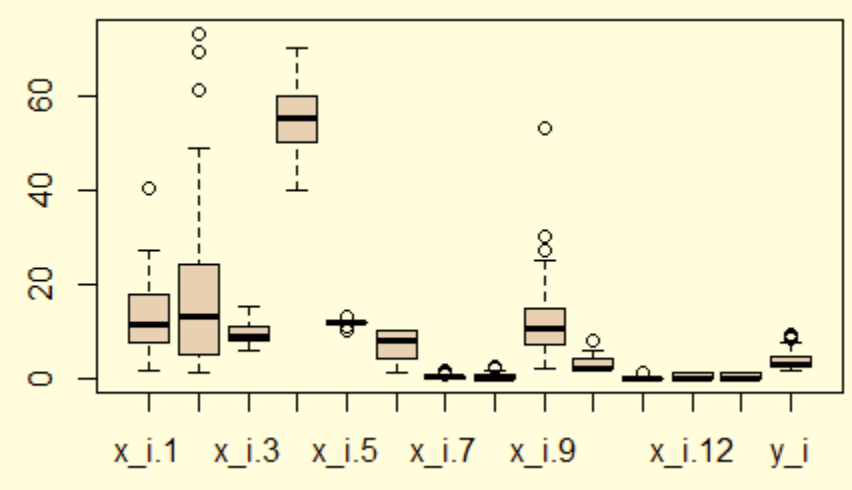
→ Il peut y avoir des valeurs aberrantes dans les données d'entrée d'origine.

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 13.7129031  6.9126865   1.984   0.0584 .
x_i.1       -0.0589293  0.0314673  -1.873   0.0728 .
x_i.2       -0.0054182  0.0337952  -0.160   0.8739
x_i.3       -0.1106588  0.1134557  -0.975   0.3387
x_i.4       -0.1266860  0.0817554  -1.550   0.1338
x_i.5       -0.1196817  0.5985717  -0.200   0.8431
x_i.6        0.0183357  0.1628311   0.113   0.9112
x_i.7       -0.3882033  1.1811637  -0.329   0.7451
x_i.8        0.7087845  0.5245588   1.351   0.1887
x_i.9        0.0654378  0.0427391   1.531   0.1383
x_i.10       0.0006672  0.2864299   0.002   0.9982
x_i.11       0.5033821  1.7304348   0.291   0.7735
x_i.12      -0.9602033  1.1124585  -0.863   0.3963
x_i.13      -0.5605308  0.9784518  -0.573   0.5718
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.202 on 25 degrees of freedom
Multiple R-squared:  0.7589,    Adjusted R-squared:  0.6335
F-statistic: 6.053 on 13 and 25 DF,  p-value: 6.176e-05
```

Nous pouvons voir que la valeur du R au carré ajusté : 0,6335 est encore faible.

Utilisation de boxplot() sur 13 variables (de x_i.1 à x_i.13).

```
# --- Ameliorer ---
boxplot(data_acc$x_i.1)  #--> il a des outliers
outliers_1 <- boxplot.stats(data_acc$x_i.1)$out
which(data_acc$x_i.1 %in% c(outliers_1))            # => row 21

boxplot(data_acc$x_i.2)  #--> il a des outliers
outliers_2 <- boxplot.stats(data_acc$x_i.2)$out
which(data_acc$x_i.2 %in% c(outliers_2))            # => row 1 2 4

.....................................
```



=> Il y a 17 valeurs outliers sur 39 observations dans les lignes 1, 2, 3, 4, 5, 6, 9, 11, 19, 21, 25, 27, 29, 31, 32, 34, 37.

=> Les valeurs aberrantes s'élèvent à 43,59 % du total des observations.

-> Il n'est pas possible de supprimer toutes les lignes avec des valeurs aberrantes, car elles représentent plus de 5 % du total des observations.

=> Par conséquent, nous pouvons remplacer les valeurs aberrantes par la moyenne de la colonne correspondante pour réduire les valeurs aberrantes.

2. Remplacer la colonne de données en position médiane

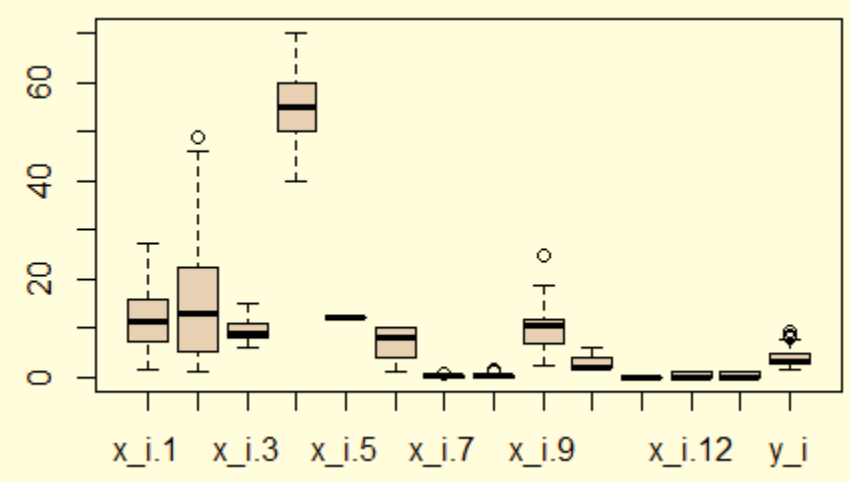
- Faites une copie des données d'origine (au cas où les modifications directes des données d'origine ne fonctionneraient pas →Difficile de revenir en arrière).

```
temp_data <- data_acc
```

- Ensuite, nous remplacerons les colonnes aberrantes par la médiane de la colonne correspondante.

```
temp_data$x_i.1[which(temp_data$x_i.1 %in% c(outliers_1))] <- median(data_acc$x_i.1,na.rm = TRUE)
temp_data$x_i.2[which(temp_data$x_i.2 %in% c(outliers_2))] <- median(data_acc$x_i.2,na.rm = TRUE)
temp_data$x_i.5[which(temp_data$x_i.5 %in% c(outliers_5))] <- median(data_acc$x_i.5,na.rm = TRUE)
temp_data$x_i.7[which(temp_data$x_i.7 %in% c(outliers_7))] <- median(data_acc$x_i.7,na.rm = TRUE)
temp_data$x_i.8[which(temp_data$x_i.8 %in% c(outliers_8))] <- median(data_acc$x_i.8,na.rm = TRUE)
temp_data$x_i.9[which(temp_data$x_i.9 %in% c(outliers_9))] <- median(data_acc$x_i.9,na.rm = TRUE)
temp_data$x_i.10[which(temp_data$x_i.10 %in% c(outliers_10))] <- median(data_acc$x_i.10,na.rm = TRUE)
temp_data$x_i.11[which(temp_data$x_i.11 %in% c(outliers_11))] <- median(data_acc$x_i.11,na.rm = TRUE)
```

- En utilisant boxplot() pour observer, nous voyons.



=> Le nombre de valeurs outliers après remplacement par la médiane est encore beaucoup (aux lignes 3 5 6 9 10 12 16 20 26) par rapport au nombre total d'observations.

=> Donc, remplacer par la médiane n'est toujours pas vraiment bon.

==> Remplacer par Q3.

3. Remplacez les données de la colonne par Q3

- Faire une copie des anciennes données.

```
new_data <- data_acc
```

- Remplacer les colonnes avec des valeurs aberrantes par Q3.

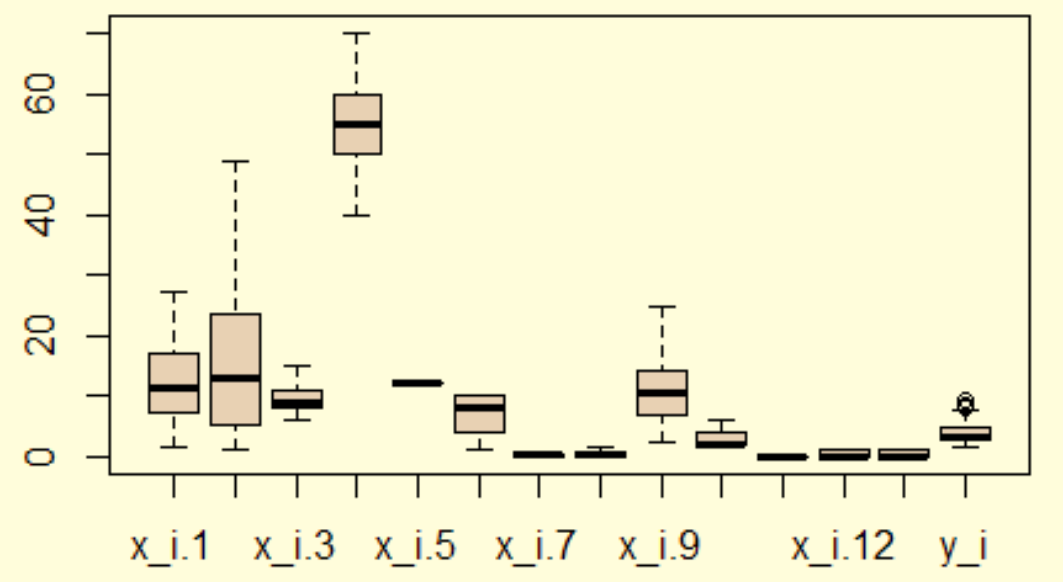
```
new_data$x_i.1[which(new_data$x_i.1 %in% c(outliers_1))] <- quantile(data_acc$x_i.1,0.75,na.rm = TRUE)
new_data$x_i.2[which(new_data$x_i.2 %in% c(outliers_2))] <- quantile(data_acc$x_i.2,0.75,na.rm = TRUE)
new_data$x_i.5[which(new_data$x_i.5 %in% c(outliers_5))] <- quantile(data_acc$x_i.5,0.75,na.rm = TRUE)
new_data$x_i.7[which(new_data$x_i.7 %in% c(outliers_7))] <- quantile(data_acc$x_i.7,0.75,na.rm = TRUE)
new_data$x_i.8[which(new_data$x_i.8 %in% c(outliers_8))] <- quantile(data_acc$x_i.8,0.75,na.rm = TRUE)
new_data$x_i.9[which(new_data$x_i.9 %in% c(outliers_9))] <- quantile(data_acc$x_i.9,0.75,na.rm = TRUE)
new_data$x_i.10[which(new_data$x_i.10 %in% c(outliers_10))] <- quantile(data_acc$x_i.10,0.75,na.rm = TRUE)
new_data$x_i.11[which(new_data$x_i.11 %in% c(outliers_11))] <- quantile(data_acc$x_i.11,0.75,na.rm = TRUE)
```

- Utilisez boxplot(), pour afficher les données sur les nouvelles données.

```
boxplot(new_data)
```

Nous pouvons voir qu'en remplaçant les colonnes de valeurs aberrantes par la valeur de Q3, les données n'ont pas de valeurs aberrantes.

On voit que, lorsque nous le remplaçons par Q3, le nombre de valeurs outliers est bien inférieur.



- Après traitement des valeurs aberrantes, nous choisirons un nouveau modèle basé sur les nouvelles données.

4. La nouveau modèle basé sur les données filtrées du T3 (modele_2)

```
modele_2 <- lm(y_i~.,data=new_data)
summary(modele_2)
```

```
Coefficients: (2 not defined because of singularities)
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 14.34702    3.29374   4.356 0.000172 ***
x_i.1       -0.09520    0.03630  -2.622 0.014175 *
x_i.2       -0.03771    0.02916  -1.293 0.206865
x_i.3       -0.14044    0.09487  -1.480 0.150338
x_i.4       -0.15648    0.06094  -2.568 0.016090 *
x_i.5             NA         NA      NA       NA
x_i.6        0.10462    0.11695   0.895 0.378939
x_i.7        2.47452    1.46309   1.691 0.102292
x_i.8        1.97516    0.73907   2.672 0.012611 *
x_i.9        0.08367    0.05104   1.639 0.112745
x_i.10      -0.12315    0.30003  -0.410 0.684702
x_i.11            NA         NA      NA       NA
x_i.12      -1.78549    0.63187  -2.826 0.008768 **
x_i.13      -0.32789    0.77886  -0.421 0.677095
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.142 on 27 degrees of freedom
Multiple R-squared:  0.7649,    Adjusted R-squared:  0.6692
F-statistic: 7.988 on 11 and 27 DF,  p-value: 5.67e-06
```

- Après avoir sélectionné le nouveau modèle à partir des nouvelles données, nous procédons au choix du meilleur modèle.

```
step(modele_2)
```

➔ Modele Choisi: **y_i = x_i.1 + x_i.3 + x_i.4 + x_i.8 + x_i.9 + x_i.12**
-> Équation linéaire: y_i = 12.08859 - 0.09550(x_i.1) - 0.14805(x_i.3) - 0.11327(x_i.4) + 1.34911(x_i.8) + 0.08108(x_i.9) - 1.27857(x_i.12).

- Nous choisissons le nouveau modèle:

```
modele_3 <- lm(y_i~x_i.1+x_i.3+x_i.4+x_i.8+x_i.9+x_i.12, data=new_data)
summary(modele_3)
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 12.08859    2.73669   4.417 0.000107 ***
x_i.1       -0.09550    0.03149  -3.033 0.004775 **
x_i.3       -0.14805    0.08898  -1.664 0.105883
x_i.4       -0.11327    0.04255  -2.662 0.012055 *
x_i.8        1.34911    0.56651   2.381 0.023362 *
x_i.9        0.08108    0.04524   1.792 0.082579 .
x_i.12      -1.27857    0.41928  -3.049 0.004577 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.119 on 32 degrees of freedom
Multiple R-squared:  0.7329,    Adjusted R-squared:  0.6828
F-statistic: 14.63 on 6 and 32 DF,  p-value: 5.768e-08
```

→ Avec R-carré ajusté : 0,6828 de ce nouveau modèle est meilleur que le modèle original (0,6828 > 0,6335).

5. Calculez les résidus et vérifiez si le nouveau modèle suit une distribution normale

```
res_modele_3 <- residuals(modele_3)
shapiro.test(res_modele_3)
```

- Nous calculons la valeur résiduelle puis utilisons shapero.test pour calculer...
- On obtient p-value : 0.9863 => Ne suivre pas la distribution normale.

II. ANOVA à deux facteurs Exercice: Agents toxiques

Modèle linéaire général : Durée en fonction de Poison; Traitement.

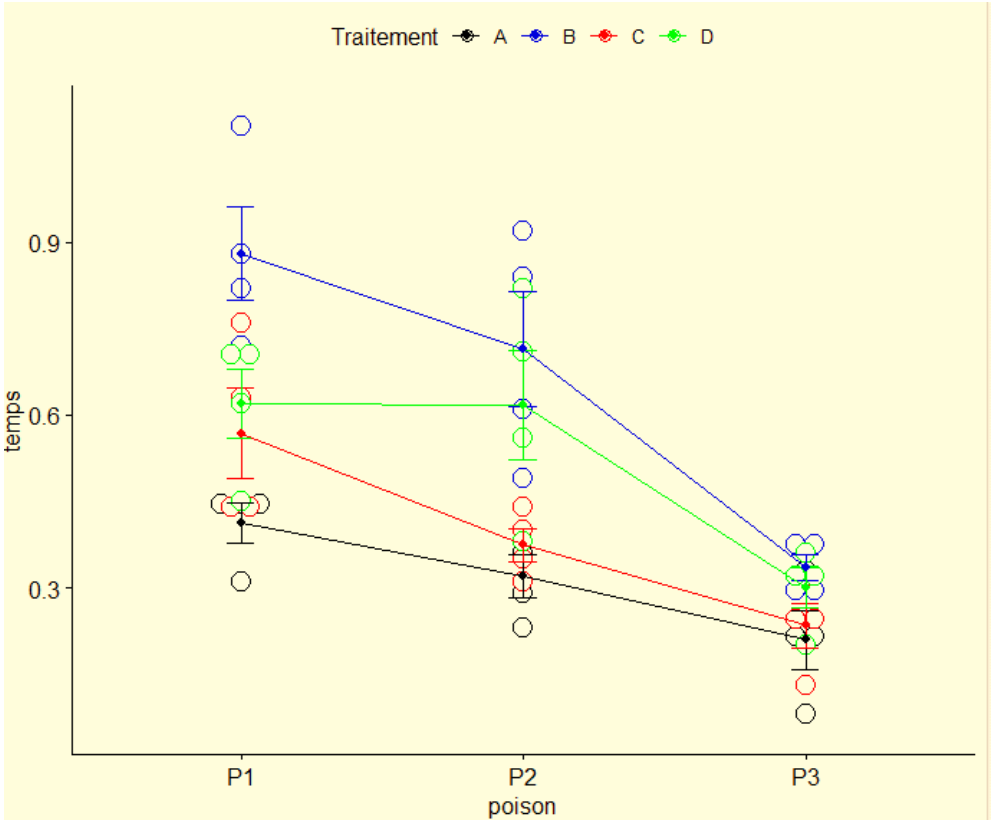
Facteur type niveaux Valeurs.

Poison fixe 3 I; II; III.

Traitement fixe 4 A; B; C; D.

Visualisation de données.

```
plot <- ggline(data_project, x = "poison", y = "temps", color = "Traitement",
               add = c("mean_se", "dotplot"),
               palette = c("black", "blue", "red", "green"))
plot
```



L'axe Y montre les temps de survie de quatre traitements A, B, C et D.

L'axe X montre les classes de poison de quatre traitements A, B, C et D.

On peut voir ça:

- ⇨ Le traitement B a le plus de temps, puis le traitement D, puis le traitement C et enfin le traitement A.
- ⇨ Le temps de survie du poison P3 est le plus court, suivi du poison P2 et enfin du poison P1.

1. Shapiro-Wilk normalité test

```
> shapiro.test(aov_residuals)

        Shapiro-Wilk normality test

data:  aov_residuals
W = 0.9817, p-value = 0.6507
```

Nous pouvons voir le p-valu = 0.6507 > 0,05.

Par conséquent, nous n'avons pas suffisamment de raisons de rejeter l'hypothèse H0 ou que la variable aov_residuals obéit à la distribution normale.

```
> anova <- aov(temps ~ poison*Traitement, data = data_project)
> summary(anova)
                 Df Sum Sq Mean Sq F value   Pr(>F)
poison            2 1.0208  0.5104  34.319 4.56e-09 ***
Traitement        3 0.7448  0.2483  16.692 5.82e-07 ***
poison:Traitement 6 0.1801  0.0300   2.019   0.0885 .
Residuals        36 0.5354  0.0149
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2. Test Fisher hypothesis avec a = 5%:

a. Type de Poison

H0: il N'a pas effect "type de Poison ".

H1: il y a effect "type de Poison ".

P-valu ~= 0.000 < a = 0.05 → rejeter H0 avec le seuil a=0.05

⇨ il y a effect "type de Poison " dans la modele.

b. Type de Traitement

H0: il n'a pas effect "type de Traitement "

H1: il y a effect "type de Traitement "

P-valu ~= 0.000 < a = 0.05 → rejeter H0 avec le seuil a=0.05

Source de variation	SC	ddl	MC	Fos	P-value
Poison	1,0208	2	0,5104	34,32	0,000
Traitement	0,7448	3	0,2483	16,69	0,000
Interaction Poison*Traitement	0,1801	6	0,0300	2,02	0.08
Intérieur	0,5354	36	0,0149		
Total	2,4811	47			

⇨ il y a effect "type de Traitement " dans la modele.

c. Type de Poison * Traitement

H0: il N'a pas effect "type de Poison * Traitement "

H1: il y a effect "type de Poison * Traitement "

P-valu = 0.08 > a = 0.05 → pas assez de base pour rejeter H0 avec le seuil a=0.05

- ⇨ il y n'a pas effect "type de Poison * Traitement " dans la modele.
- ⇨ il n'y a pas d'interaction entre les 2 Poison et Traitement.

```
> bartlett.test(temps ~ interaction(poison,Traitement), data = data_project)

        Bartlett test of homogeneity of variances

data:  temps by interaction(poison, Traitement)
Bartlett's K-squared = 13.35, df = 11, p-value = 0.2711
```

#H0: La variance entre chaque groupe est égale.

#H1: Au moins un groupe a une variance qui n'est pas égale aux autres.

P-valu = 0.2711 > a = 0.05 → pas assez de base pour rejeter H0 avec le seuil a=0.05

- ⇨ La variance entre chaque groupe est égale.

```
> leveneTest(temps ~ poison * Traitement, data = data_project)
Levene's Test for Homogeneity of Variance (center = median)
      Df F value  Pr(>F)
group 11  1.9648 0.06283 .
      36
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

#H0 : les variances entre les échantillons sont égales.

#H1 : au moins un échantillon a une variance différente.

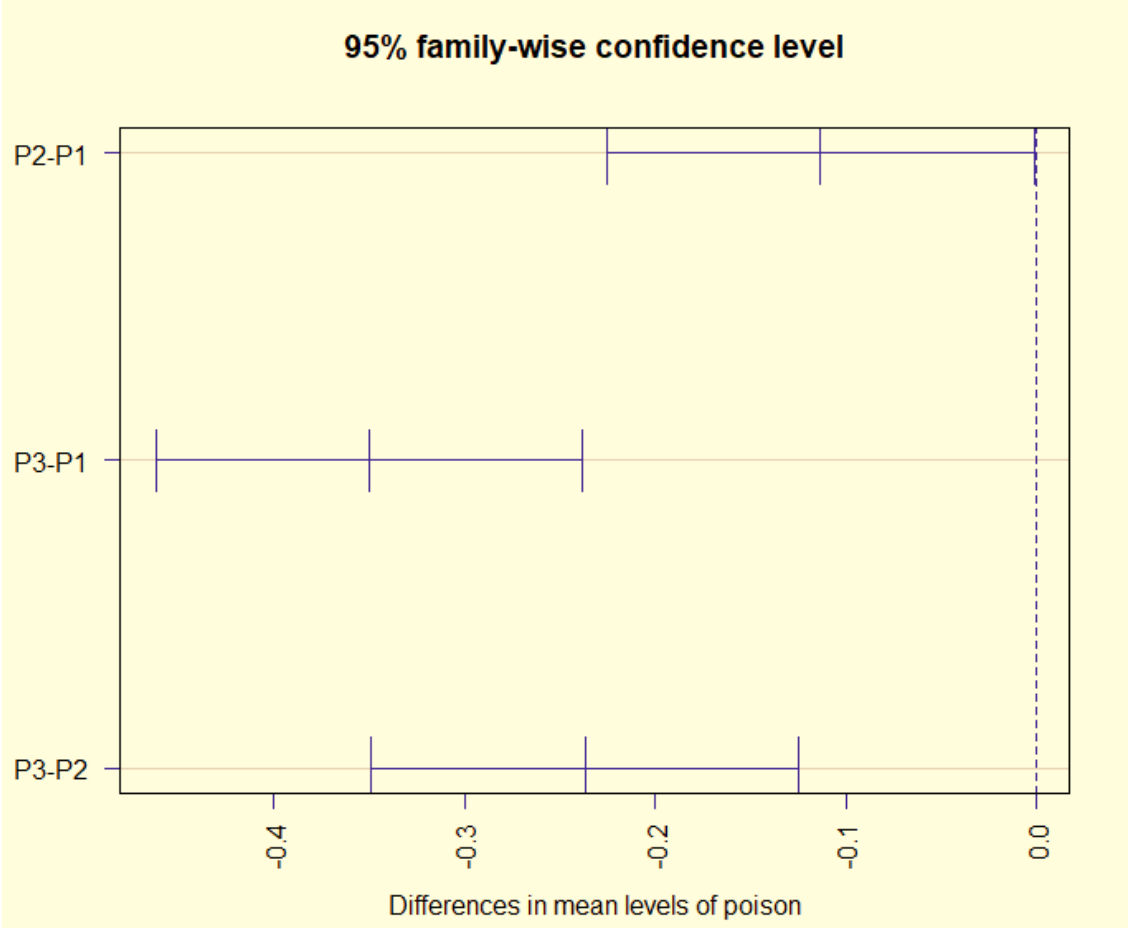
Levene's Test identique au test de Bartlett:

P-valu = 0.06283 > a = 0.05

- ⇨ Nous ne pouvons pas rejeter H0 selon laquelle la variance est égale dans tous les échantillons au seuil de signification de 0,05.
- ⇨ Les variances entre les échantillons sont égales.

3. TurkeyHSD Test

```
> TukeyHSD(anova2, which = "poison", data = data_project)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = temps ~ poison + Traitement, data = data_project)

$poison
           diff        lwr          upr     p adj
P2-P1 -0.113125 -0.2252423 -0.001007743 0.0475653
P3-P1 -0.350000 -0.4621173 -0.237882743 0.0000000
P3-P2 -0.236875 -0.3489923 -0.124757743 0.0000203
```



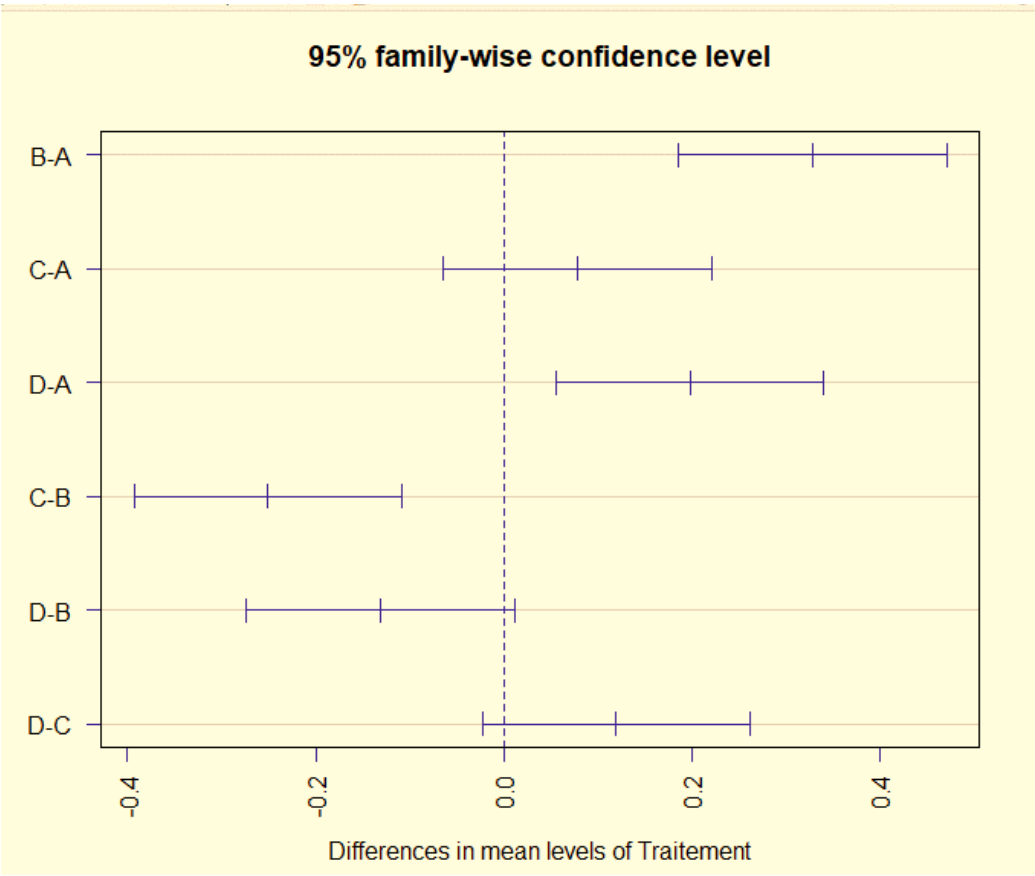
Intervalles de confiance simultanés de Tukey = 95,0 %

On a p-adj P1-P2 = 0.04 < a = 0.05 → Il existe une différence entre les poisons P1 et P2 et est statistiquement significative.

Semblable à p-adj P1-P3 = 0.0000 < a = 0.05 → Il existe une différence entre les poisons P1 et P3 et est statistiquement significative.

Semblable à p-adj P2-P3 = 0.0000 < a = 0.05 → Il existe une différence entre les poisons P2 et P3 et est statistiquement significative.

```
> TukeyHSD(anova2, which = "Traitement", data = data_project)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = temps ~ poison + Traitement, data = data_project)

$Traitement
           diff         lwr         upr     p adj
B-A  0.32916667  0.18662465  0.47170868 0.0000013
C-A  0.07833333 -0.06420868  0.22087535 0.4642520
D-A  0.19833333  0.05579132  0.34087535 0.0031454
C-B -0.25083333 -0.39337535 -0.10829132 0.0001562
D-B -0.13083333 -0.27337535  0.01170868 0.0822055
D-C  0.12000000 -0.02254201  0.26254201 0.1260802
```



Intervalles de confiance simultanés de Tukey = 95,0 %.

On a p-adj B-A = 0.0000 < a = 0.05 → Il y a une différence entre le traitement B et le traitement A et elle est statistiquement significative.

Semblable à p-adj D-A = 0.003 < a = 0.05 → Il y a une différence entre le traitement D et le traitement A et elle est statistiquement significative.

Semblable à p-adj C-B = 0.000 < a = 0.05 → Il y a une différence entre le traitement C et le traitement B et elle est statistiquement significative.

On a p-adj C-A = 0.46 > a = 0.05 → Il n'y a pas de différence statistiquement significative entre le traitement C et le traitement A.

Semblable à p-adj D-B = 0.08 > a = 0.05 → Il n'y a pas de différence statistiquement significative entre le traitement D et le traitement B.

Semblable à p-adj D-C = 0.12 > a = 0.05 → Il n'y a pas de différence statistiquement significative entre le traitement D et le traitement C.

Tableau de répartition du travail

Code d'étudiant	Nom et Prénom	Travail	Pourcentage de travail effectué
20126016	Phạm Quang Huy	ANOVA à deux facteurs Exercice : Agents toxiques	100%
20126041	Nguyễn Huỳnh Mẫn	Modèle de régression linéaire multiple (Construire une modele qui prédit le poids des poissons vendus sur le marché)	100%
20126062	Thiều Vĩnh Trung	Choix du modèle Exercice : Taux d'accidents	100%
20126041	Nguyễn Huỳnh Mẫn	Ecrire un rapport	100%
20126016	Phạm Quang Huy	Vérifier et finaliser le rapport	100%
20126062	Thiều Vĩnh Trung		100%

Les références

Consulter le guide de traitement des données et la théorie

- https://dzchilds.github.io/stats-for-bio/two-way-anova-in-r.html?fbclid=IwAR0VPgVpS4ZCF1IwKDS55SXAQM4VgWnNSVxxWldgwrO-cPNPH_7n4MCKY0w

Les sources de données

- https://www.kaggle.com/datasets/aungpyaeap/fish-market?resource=download
- https://www.kaggle.com/ (Vous pouvez rechercher de nombreux ensembles de données sur ce site. C'est le site que de nombreux analystes de données utilisent pour pratiquer)