Practical exam

Scéance 2: IC, test d'hypothèse et test de Student

On charge le jeu de données iris:

```
data(iris)
str(iris)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

L'interval de confiance autour de la moyenne se calcule par $IC = x \pm \frac{ts}{\sqrt{n}}$ ou t est le percentile à 95% d'une distribution de Student à n-1 ddl.

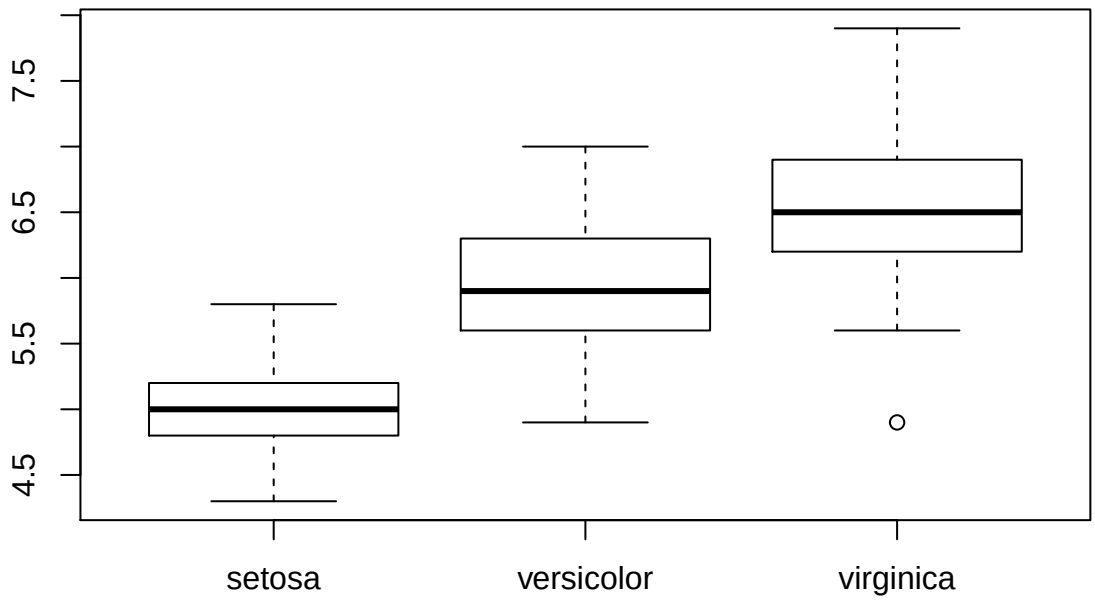
On calcule la moyenne de la longueur des sépales, écart-type de la moyenne, bornes sup et inf de l'IC de la moyenne:

```
# moyenne de la longueur des sépales
SepMn = mean(iris$Sepal.Length)
# Écart-type
SepSd = sd(iris$Sepal.Length)
# IC à 95% avec 149 ddl
myIC = qt(0.975, 149) * SepSd/(150^0.5)
# Borne supérieure et inférieure de l'IC
SepMnUpperCI = SepMn + myIC
SepMnLowerCI = SepMn - myIC
```

La moyenne SepMn diffère significativement de 0 car l'IC ne contient pas 0.

On réalise une boîte de dispersion de la longueur des sépales en fonction de l'espèce.

```
boxplot(iris$Sepal.Length~iris$Species)
```



On voit que la longueur des sépales de *setosa* diffère de celles de *versicolor* car il n'y a pas de recoupement entre les deux. Il y a donc une différence entre les deux espèces. Entre *versicolor* et *virginica* il y a moins de différences entre les limites des boites.

On construit un jeu de donnée par espèces et calcule la moyenne, l'écart-type et les bornes inférieures et supérieures de l'IC pour la longueur des sépales de *versicolor* et *setosa*.

```
# Création des différents sous-ensembles
Ver = subset(iris, Species == "versicolor")
Set = subset(iris, Species == "setosa")
Vir = subset(iris, Species == "virginica")
# moyenne de la longueur des sépales
SepMn_Ver = mean(Ver$Sepal.Length)
SepMn_Set = mean(Set$Sepal.Length)
# Écart-type
SepSd_Ver = sd(Ver$Sepal.Length)
SepSd_Set = sd(Set$Sepal.Length)
# IC à 95% avec 149 ddl
myIC_Ver = qt(0.975, 149) * SepSd_Ver/(150^0.5)
myIC_Set = qt(0.975, 149) * SepSd_Set/(150^0.5)
# Borne supérieure et inférieure de l'IC
SepMnUpperCI_Ver = SepMn_Ver + myIC_Ver
SepMnLowerCI_Ver = SepMn_Ver - myIC_Ver
SepMnUpperCI_Set = SepMn_Set + myIC_Set
SepMnLowerCI_Set = SepMn_Set - myIC_Set
```

On voit que la probabilité que les moyennes des longueurs de sépales de *setosa* et *versicolor* soient les mêmes

est faible vu que la borne supérieure de l'IC de *setosa* n'a pas de "overlap" avec la borne inférieure de l'IC de *versicolor*.

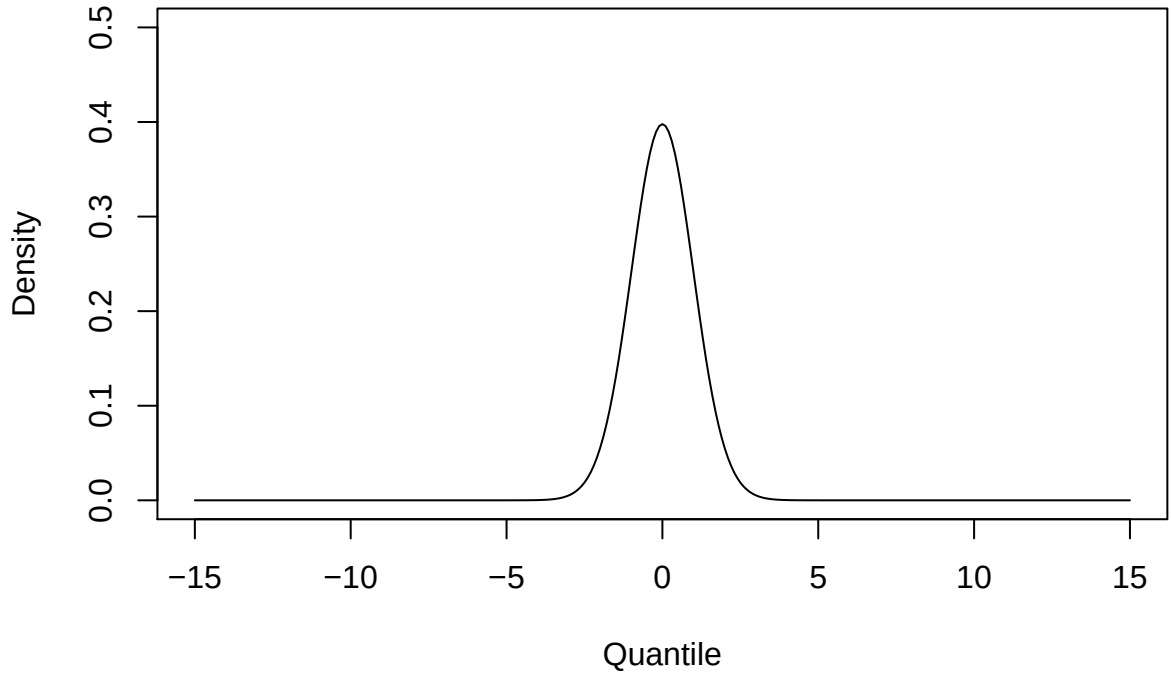
Réalisez un test t pour comparer les moyennes des longueurs de sépales de *versicolor* et *setosa* et les moyennes de la longueur des sépales de *versicolor* et *virginica*.

```
# Comparaison des moyennes des longueurs de sépales test t bilatéral
t_Ver_Set_bil = t.test(Ver$Sepal.Length, Set$Sepal.Length)
t_Ver_Vir_bil = t.test(Ver$Sepal.Length, Vir$Sepal.Length)
# Comparaison des moyennes des longueurs de sépales test t unilatéral
t_Ver_Set_uni = t.test(Ver$Sepal.Length, Set$Sepal.Length, alternative = "greater")
t_Ver_Vir_uni = t.test(Ver$Sepal.Length, Vir$Sepal.Length, alternative = "greater")
```

Les tests bilatéraux permettent de dire si il y a une différence entre les moyennes des deux types de données. Le test t unilatéral permets de voir si une moyenne diffère de l'autre. Il faut que la p-value soit > 0.05.

On représente la distribution des valeurs prises par la distribution t de Student sous H0. On peut y reporter la statistique obtenue pour le premier test (ici 10.521) et on voit donc que la probabilité d'obtenir cette statistique sous H0 est hautement improbable.

```
myseq = seq(from = -15, to = 15,by = 0.1)
plot(myseq, dt(myseq, 86.538), type = "l", ylim = c(0,0.5),
xlim = c(-15,15), ylab = "Density", xlab = "Quantile")
```



On fait ici un calcul de la puissance du test t pour une moyenne des sépales de 1 et un écart-type correspondant à la moyenne des écarts-types des différentes espèces. On voit que la puissance du test avec 50 observations est de 1 et donc la puissance est très bonne. La puissance du test avec 5 observations est de 0.79 et donc il

y a une assez grande probabilité de se tromper en disant qu'il n'y a pas d'effet sur un échantillon de petite taille.

```
# Calcul de la moyenne des écarts-types
mean_sd = mean(c(sd(Vir$Sepal.Length), sd(Ver$Sepal.Length), sd(Set$Sepal.Length)))
# Calcul de la puissance avec 50 observations par échantillon
power.t.test(50, delta=1, sd=mean_sd)
```

```
##
##      Two-sample t test power calculation
##
##               n = 50
##           delta = 1
##              sd = 0.5015135
##       sig.level = 0.05
##           power = 1
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

```
# Calcul de la puissance avec 5 observations par échantillon
power.t.test(5, delta = 1, sd=mean_sd)
```

```
##
##      Two-sample t test power calculation
##
##               n = 5
##           delta = 1
##              sd = 0.5015135
##       sig.level = 0.05
##           power = 0.7881655
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

Scéance 3: ANOVA à un et deux facteurs

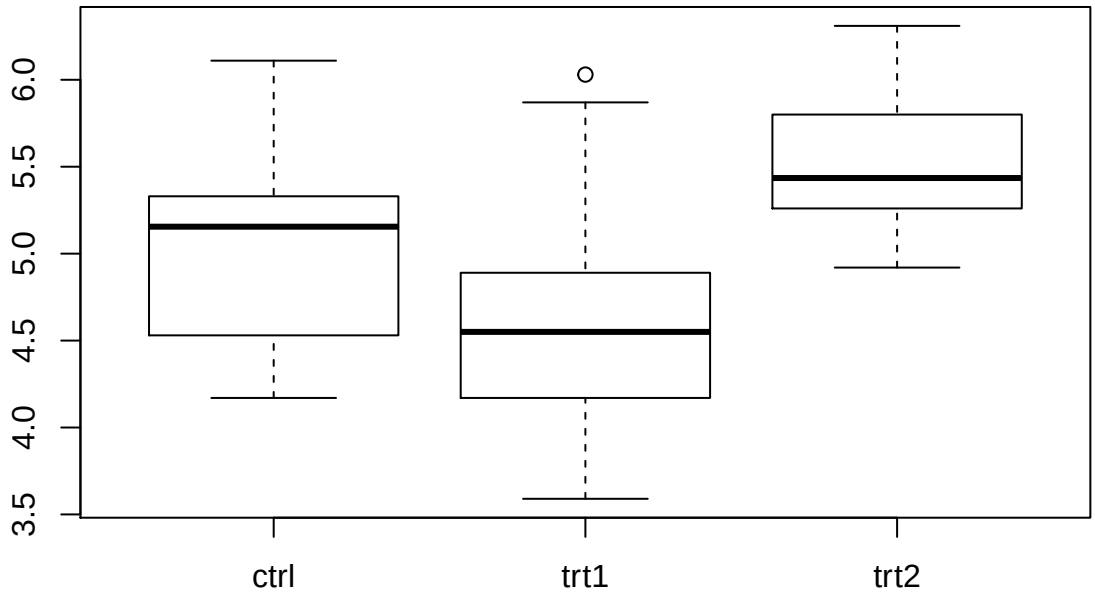
Partie 1: ANOVA

```
# Chargement du jeu de données PlantGrowth
data("PlantGrowth")
str(PlantGrowth)
```

```
## 'data.frame':    30 obs. of  2 variables:
##  $ weight: num  4.17 5.58 5.18 6.11 4.5 4.61 5.17 4.53 5.33 5.14 ...
##  $ group : Factor w/ 3 levels "ctrl","trt1",..: 1 1 1 1 1 1 1 1 1 1 ...
```

Afficher les boîtes de dispersion de la variable weight en fonction de la variable de groupement: Il semble y avoir une grande différence entre le traitement 1 et le traitement 2 vu qu'il n'y a pas de recouvrement entre les deux boites.

```r
# Graphique de boites de dispersion du poids en fonction des groupes
boxplot(PlantGrowth$weight~PlantGrowth$group)
```



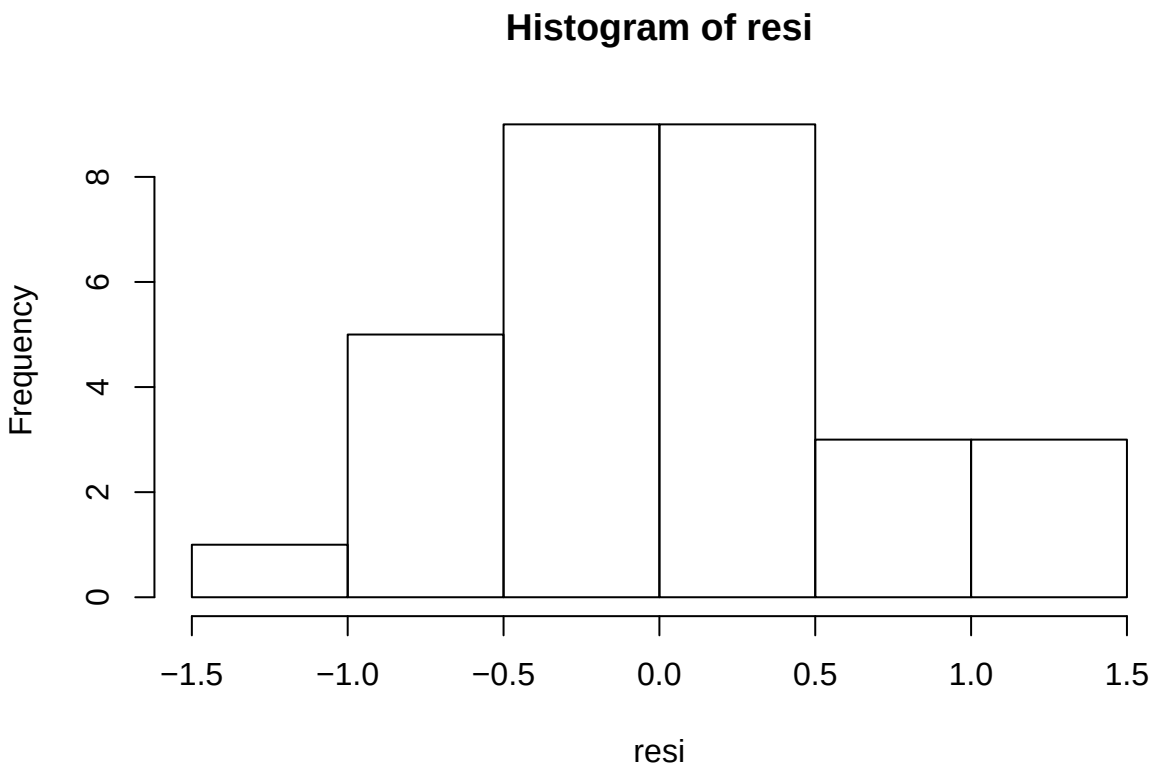
Réalisation d'une ANOVA à un facteur (poids) et 3 niveaux (3 traitements) qui compare les moyennes de poids du contrôle et des deux traitements. H0: il n'y a pas de différence entre le poid du contrôle ou des traitements. H1: il y a au moins une moyenne qui diffère des autres. Il y a une p-valeur < 0.0159 et donc il y a une probabilité de 1.59% d'avoir une statistique F plus grande que 4.846 si H0 est vraie. On peut donc rejeter H0 au seuil α de 5%.

```r
# ANOVA
myAOV = aov(weight~group, data=PlantGrowth)
summary(myAOV)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## group        2  3.766  1.8832   4.846 0.0159 *
## Residuals   27 10.492  0.3886
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

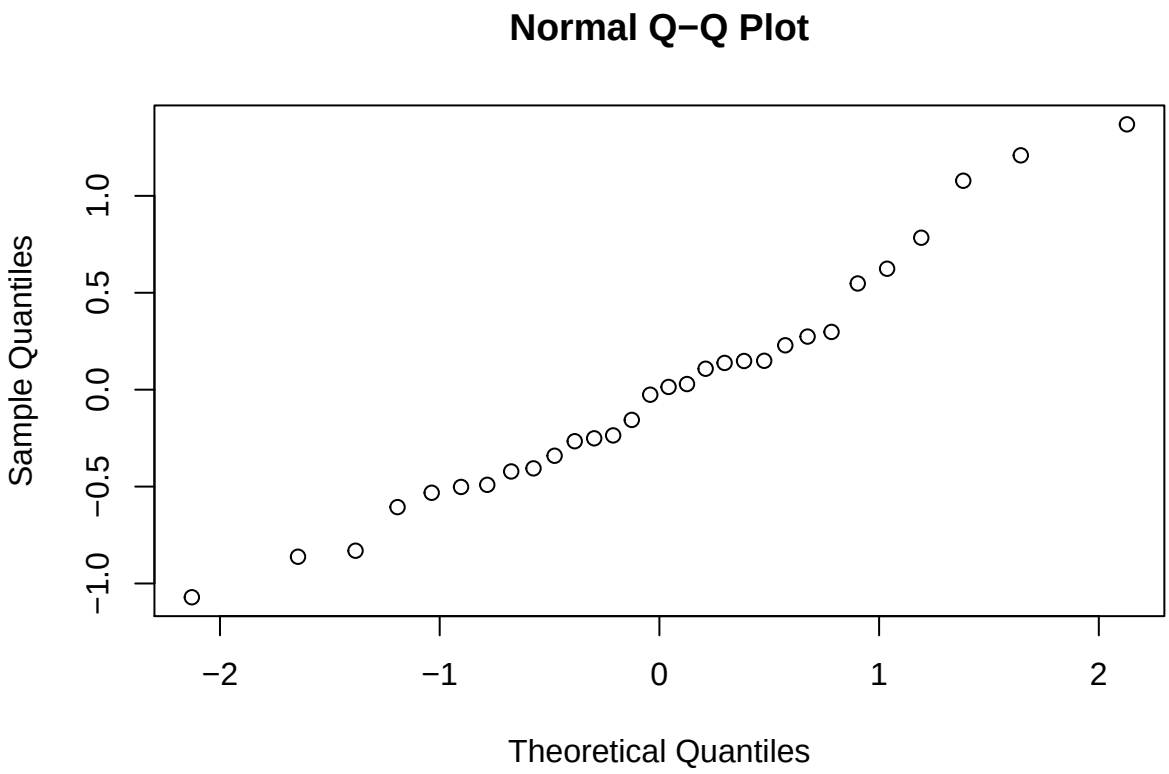
Les résidus sont la différence entre chaque observation et la moyenne du groupe. Pour appliquer une ANOVA, il faut que les résidus soient distribués normalement. Les résidus ont l'air distribués de façon presque normale

```r
# Analyse des résidus
resi = residuals(myAOV)
hist(resi)
```



Le Q-Q plot permets de comparer une distribution théorique idéale des résidus avec la distribution réelle. On veut donc avoir une droite. Ici on voit que les résidus ne sont pas parfaitement alignés sur la droite.

```
# Réalisation d'un Q-Q plot (quantile-quantile plot)
qqnorm(resi)
```



On peut réaliser un test de Shapiro pour voir si les résidus sont distribués normalement ou non. H0: les résidus suivent une loi normale -> ce qu'on veut H1: les résidus ne sont pas distribués normalement Ici l'on voit que la p-value est de 0.44 et donc on ne peut pas rejeter H0 au seuil α de 5% et donc on en conclut que les résidus sont normaux. La condition d'application de l'ANOVA (**normalité des résidus**) est donc respectée.

```
# Réalisation d'un test de Shapiro
shapiro.test(resi)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resi
## W = 0.96607, p-value = 0.4379
```

On vérifie l'homogénéité des variances grâce au test de Bartlett. H0: les variances des trois groupes sont égales H1: au moins une des variances diffère d'une des autres La p-value est de 0.237 et est donc > 0.05. On ne peut donc pas rejeter l'H0 au seuil α de 5%. Les conditions d'application de l'ANOVA (**homoscédasticité**: la variance est homogène) est respectée.

```
# Test de Bartlett pour voir l'homogénéité des variances
bartlett.test(weight~group, data=PlantGrowth)
```

```
##
##  Bartlett test of homogeneity of variances
```

```
##
## data:  weight by group
## Bartlett's K-squared = 2.8786, df = 2, p-value = 0.2371
```

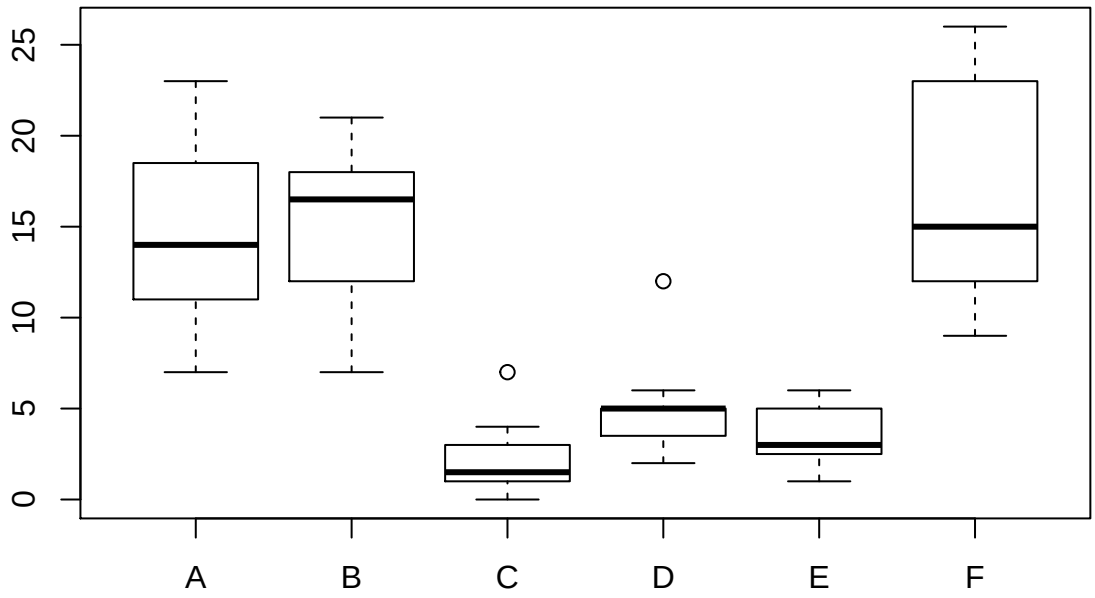
Partie 2: Transformations pour améliorer la normalité des résidus et l'homoscédasticité

```r
# Charge le jeu de données InsectSprays
data("InsectSprays")
str(InsectSprays)
```

```
## 'data.frame':    72 obs. of  2 variables:
##  $ count: num  10 7 20 14 14 12 10 23 17 20 ...
##  $ spray: Factor w/ 6 levels "A","B","C","D",..: 1 1 1 1 1 1 1 1 1 1 ...
```

En montrant les boxplots, on voit qu'il y a en effet des traitements qui semblent donner des nombres d'insectes très différents.

```r
# Graphique de dispersion du nombre d'insectes en fonction des différents sprays
boxplot(count~spray, data = InsectSprays)
```



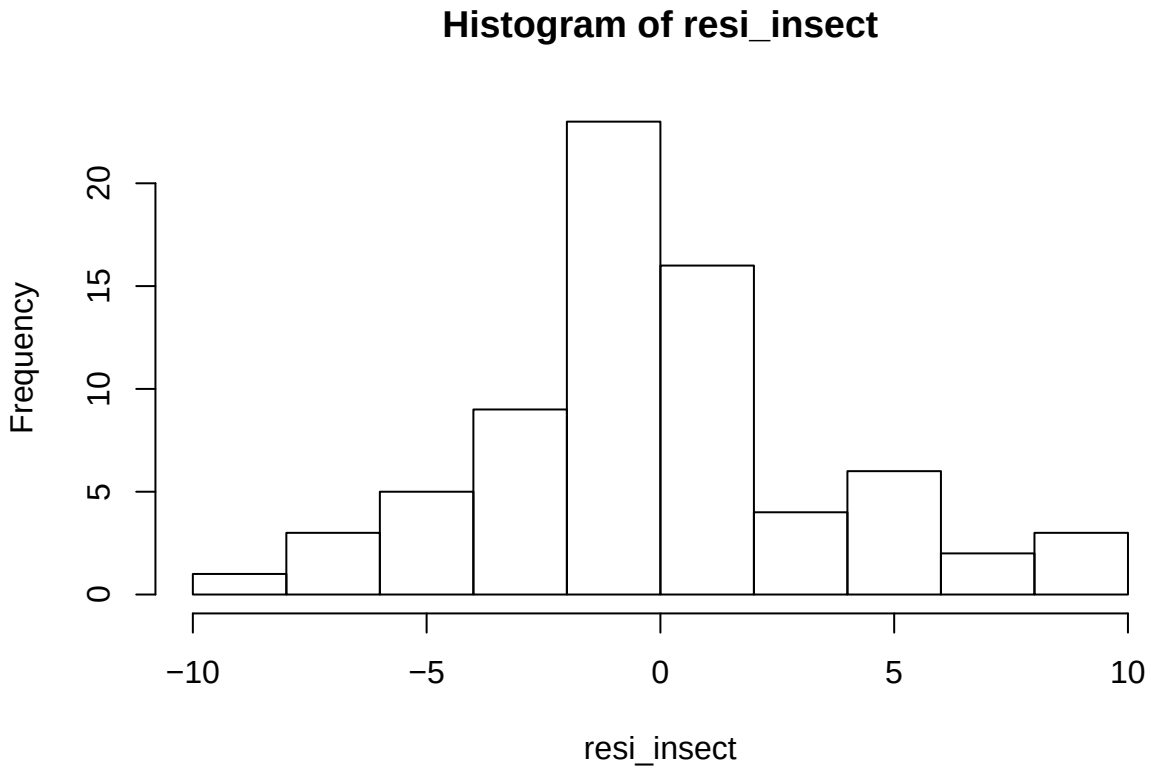
On réalise une ANOVA a un facteur (nombre d'insectes) et 6 niveaux (différents sprays). On voit en effet qu'il y a une différence entre au moins un des sprays (rejeter H0).

```r
# ANOVA
AOV_insect = aov(count~spray, data=InsectSprays)
summary(AOV_insect)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## spray        5   2669   533.8    34.7 <2e-16 ***
## Residuals   66   1015    15.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

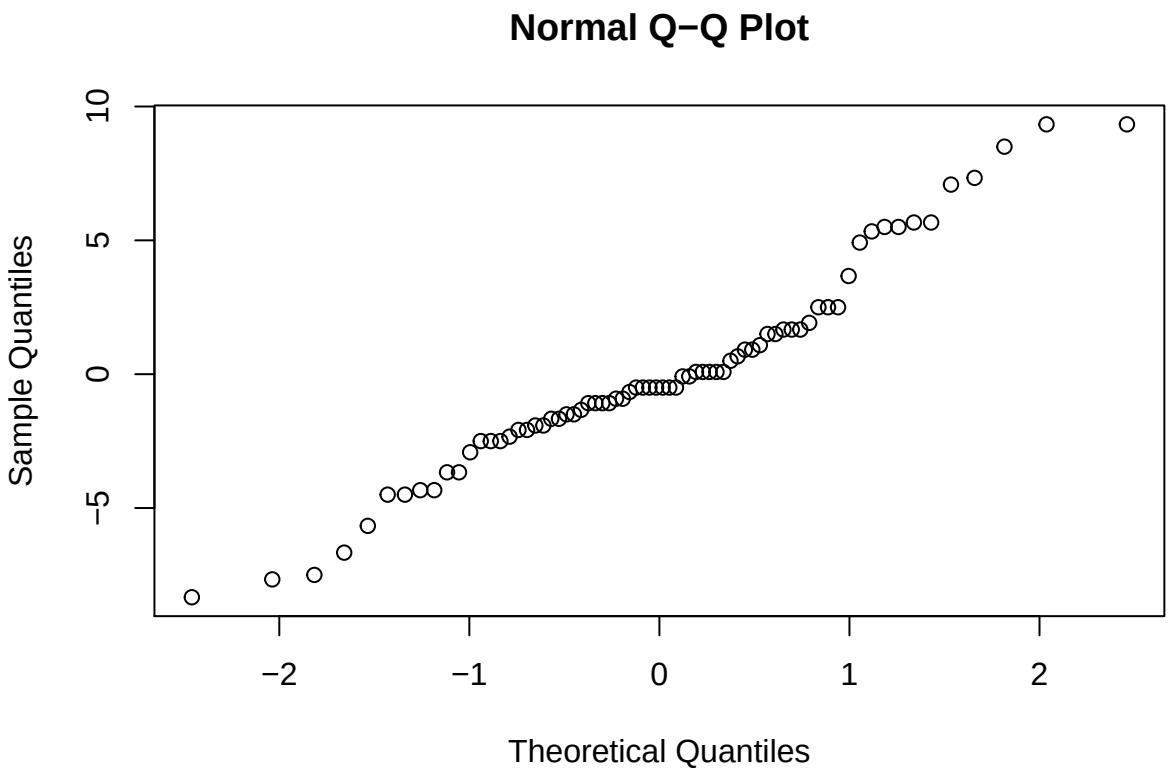
Vérification de la normalité des résidus par histogramme, Q-Q plot et test de Shapiro:

```r
# Calcul des résidus
resi_insect = residuals(AOV_insect)
# Histogramme de distribution des résidus
hist(resi_insect)
```



En faisant un Q-Q plot, on voit que les résidus ne sont pas entièrement sur la diagonale. On doit donc faire un test de Shapiro pour voir.

```r
# Q-Q plot
qqnorm(resi_insect)
```



Le test de Shapiro nous donne une p-value < 0.05 et donc on peut rejeter H0. Les résidus ne sont donc pas distribués normalement. Les conditions d'application de l'ANOVA (**normalité des résidus**) n'est pas vérifiée.

```
# Test de Shapiro
shapiro.test(resi_insect)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resi_insect
## W = 0.96006, p-value = 0.02226
```

Il faut aussi vérifier que les variances soient homogènes via un test de Bartlett. On rejette l'H0 qui dit que toutes les variances sont égales et donc la condition d'application de l'ANOVA n'est pas respectée. Au moins une variable diffère des autres (**hétéroscédasticité**).

```
# Test de Bartlett
bartlett.test(count~spray, data = InsectSprays)
```

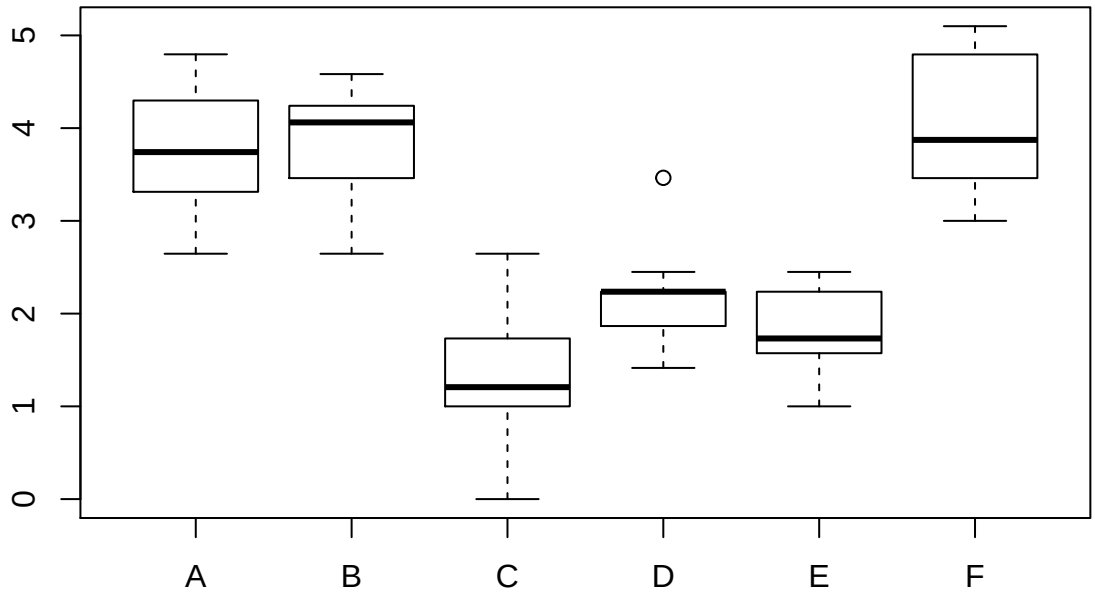
```
##
##  Bartlett test of homogeneity of variances
##
## data:  count by spray
## Bartlett's K-squared = 25.96, df = 5, p-value = 9.085e-05
```

Pour quand-même utiliser les données, on va essayer de rendre les résidus normaux et les variances égales en applicant une transformation (log et racine) sur les données.

```
# Transformation racine
InsectSprays$Sqrt = sqrt(InsectSprays$count)
```

On voit que certaines moyennes des sprays diffèrent des autres.

```
# Plot du comptage d'insectes transformé racine carrée en fonction du spray utilisé
boxplot(Sqrt~spray, InsectSprays)
```



L'ANOVA sur les données après transformation racine carrée donne une p-value < 0.05 et donc il y a en effet une différence significative entre les différent traitements.

```
# ANOVA sur les données transformés sqrt
AOV_insect_sqrt = aov(Sqrt~spray, data = InsectSprays)
summary(AOV_insect_sqrt)
```

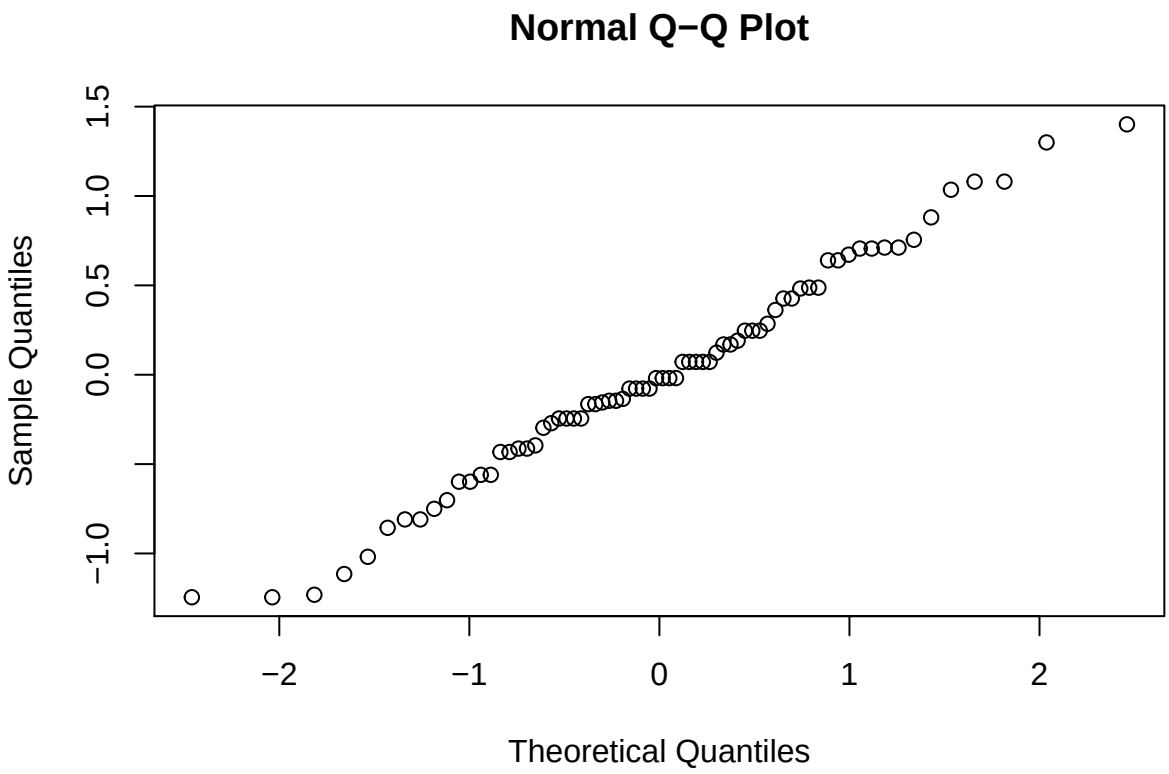
```
##             Df Sum Sq Mean Sq F value Pr(>F)
## spray        5  88.44  17.688    44.8 <2e-16 ***
## Residuals   66  26.06   0.395
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Vérification de la normalité des résidus après transformation racine carrée par un histogramme de la distribution des résidus, un Q-Q plot et un test de Shapiro:

```r
# Calcul des résidus
resi_insect_sqrt = residuals(AOV_insect_sqrt)
# Boxplot des résidus
hist(resi_insect_sqrt)
```

On voit que les quantiles ont quand-même une distribution plus proche de la diagonale que celle des résidus non-transformés.

```r
# Q-Q plot
qqnorm(resi_insect_sqrt)
```



Le test de Shapiro nous donne une p-value de 0.681 qui ne permets donc pas de rejeter H0. On a donc une distribution normale des résidus après transformation racine carrée.

```
# Test de Shapiro
shapiro.test(resi_insect_sqrt)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resi_insect_sqrt
## W = 0.98721, p-value = 0.6814
```

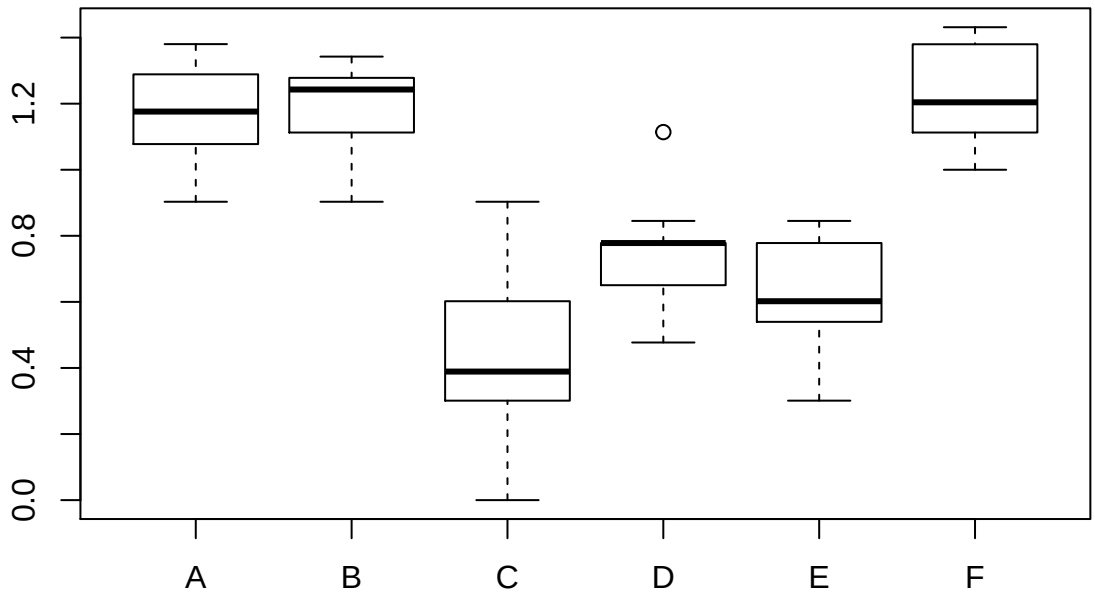
On applique un test de Bartlett pour voir si les variances sont distribuées de façon homogène. On ne peut pas rejetter H0 car il y a une p-value > 0.5856. Les variances sont donc distribuées de façon homogène.

```
# Test de Bartlett
bartlett.test(Sqrt~spray, data = InsectSprays)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  Sqrt by spray
## Bartlett's K-squared = 3.7525, df = 5, p-value = 0.5856
```

On fait la même chose pour la transformation log:

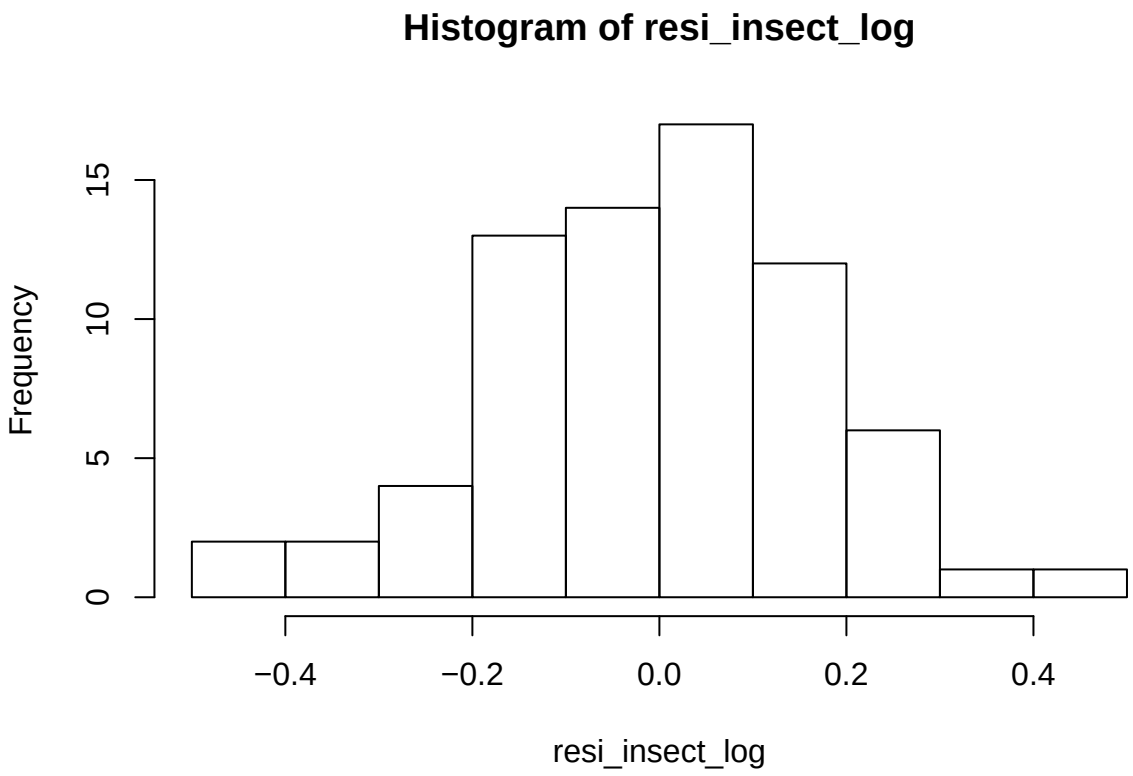
```r
# Transformation log des données
InsectSprays$Log = log10(InsectSprays$count + 1)
# Boxplot des données transformées log en fonction du spray
boxplot(Log~spray, InsectSprays)
```



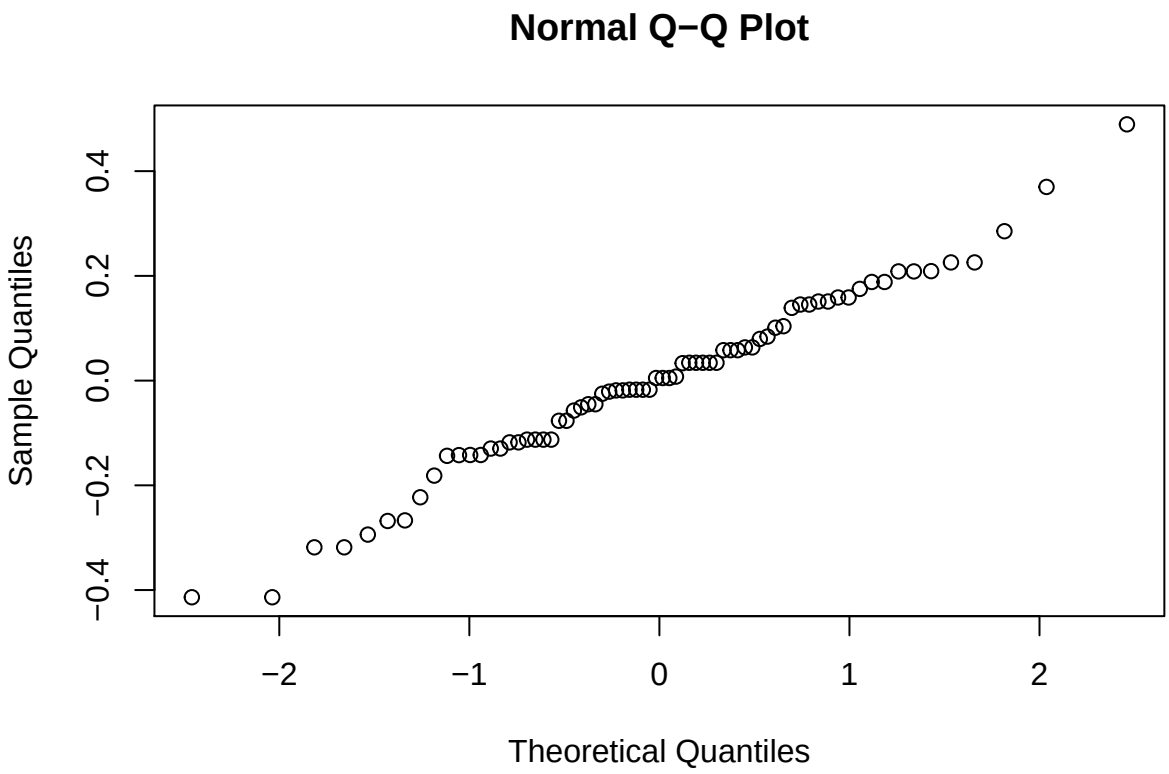

```r
# ANOVA des données transformées log
AOV_insect_log = aov(Log~spray, data=InsectSprays)
summary(AOV_insect_log)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## spray        5  7.265  1.4530   46.01 <2e-16 ***
## Residuals   66  2.084  0.0316
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Calcul des résidus après transformation log
resi_insect_log = residuals(AOV_insect_log)
# Histogramme des résidus après transformation log
hist(resi_insect_log)
```




```
# Q-Q plot des résidus
qqnorm(resi_insect_log)
```



La p-value est plus grande que 0.05 et donc on ne peut pas rejeter H0. Les résidus sont donc distribués normalement après transformation log.

```
# Test de Shapiro
shapiro.test(resi_insect_log)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resi_insect_log
## W = 0.98475, p-value = 0.5348
```

Le test de Bartlett nous donne une p-value > 0.05 et donc on ne rejette pas H0. Les variances sont donc distribuées de façon homogène.

```
# Test de Bartlett
bartlett.test(Log~spray, data=InsectSprays)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  Log by spray
## Bartlett's K-squared = 8.7705, df = 5, p-value = 0.1186
```

On voit donc que les transformations racine carrée et log permettent de normaliser les résidus et d'améliorer l'homoscédasticité des variances. On peut alors appliquer une ANOVA sur les données transformées.

Partie 3: ANOVA à plusieurs facteurs

Dans le data frame cabbages, on étudie l'influence du cultivar (2 types) et de la date (3 dates de plantation) sur la teneur en vitamine C et le poids du choux.

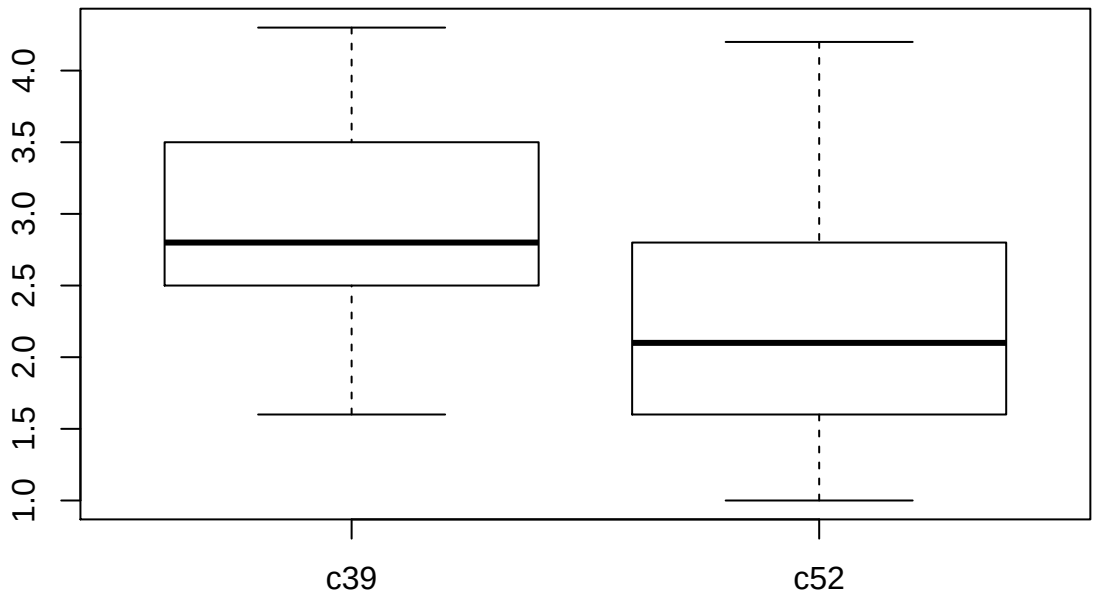
```
# Charger le pachage MASS
library(MASS)
# Charger le jeu de données "cabbages"
data("cabbages")
str(cabbages)
```

```
## 'data.frame':    60 obs. of  4 variables:
##  $ Cult  : Factor w/ 2 levels "c39","c52": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Date  : Factor w/ 3 levels "d16","d20","d21": 1 1 1 1 1 1 1 1 1 1 ...
##  $ HeadWt: num  2.5 2.2 3.1 4.3 2.5 4.3 3.8 4.3 1.7 3.1 ...
##  $ VitC  : int  51 55 45 42 53 50 50 52 56 49 ...
```

```
help("cabbages")
```

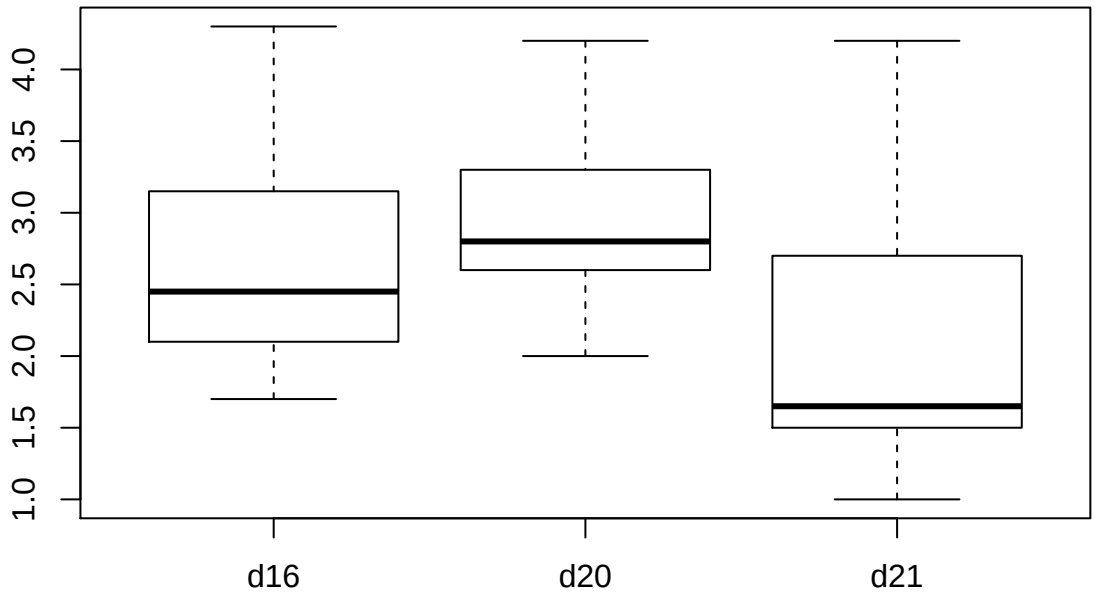
Analyse des poids des choux en fonction du cultivar. On voit que les choux du cultivar c39 ont une moyenne de poids plus haute que ceux du cultivar c52. Mais il y a tout de même un recouvrement.

```
# Boxplot des poids des choux en fonction du cultivar
boxplot(HeadWt~Cult, data=cabbages)
```



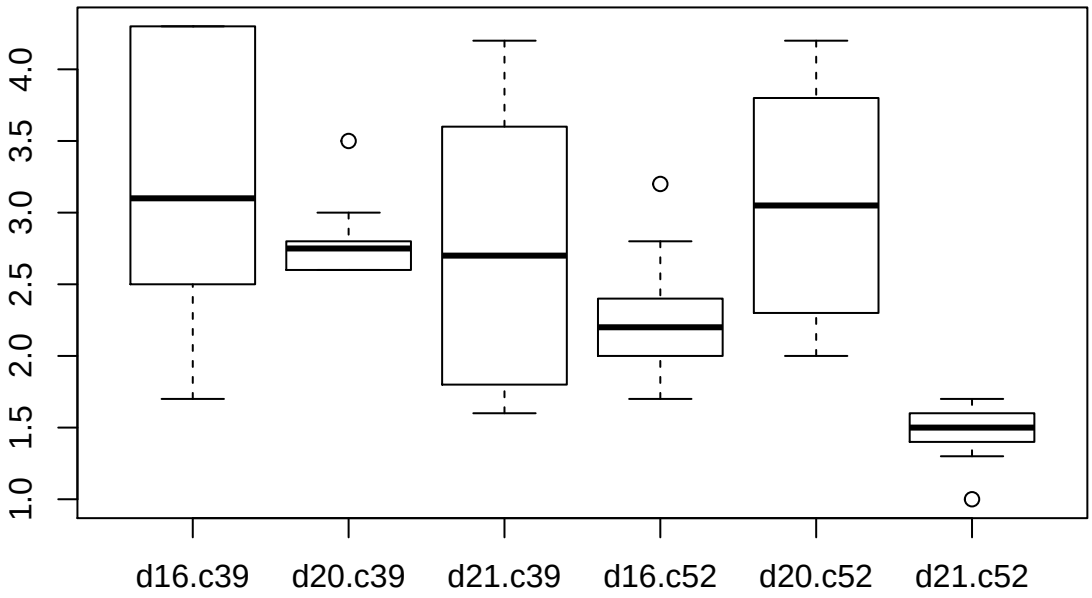
Analyse des poids des choux en fonction de la date. Il n'y a pas clairement d'effet de la date sur le poids. Les boites de dispersions se recouvrent.

```
# Boxplot des poids des choux en fonction de la date de cultivation
boxplot(HeadWt~Date, data=cabbages)
```



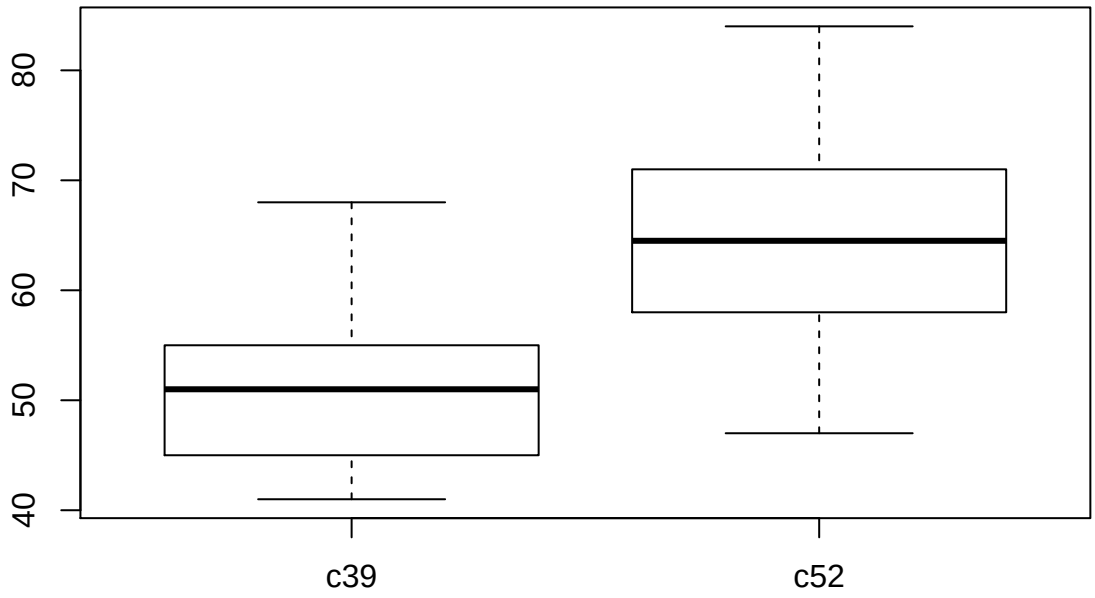
On peut analyser l'effet de la date et du cultivar sur le poids. Il semble y avoir un effet d'interaction entre le cultivar et la date.

```
# Boxplot donnant l'effet de la date et du cultivar sur le poids
boxplot(HeadWt~Date+Cult, data=cabbages)
```



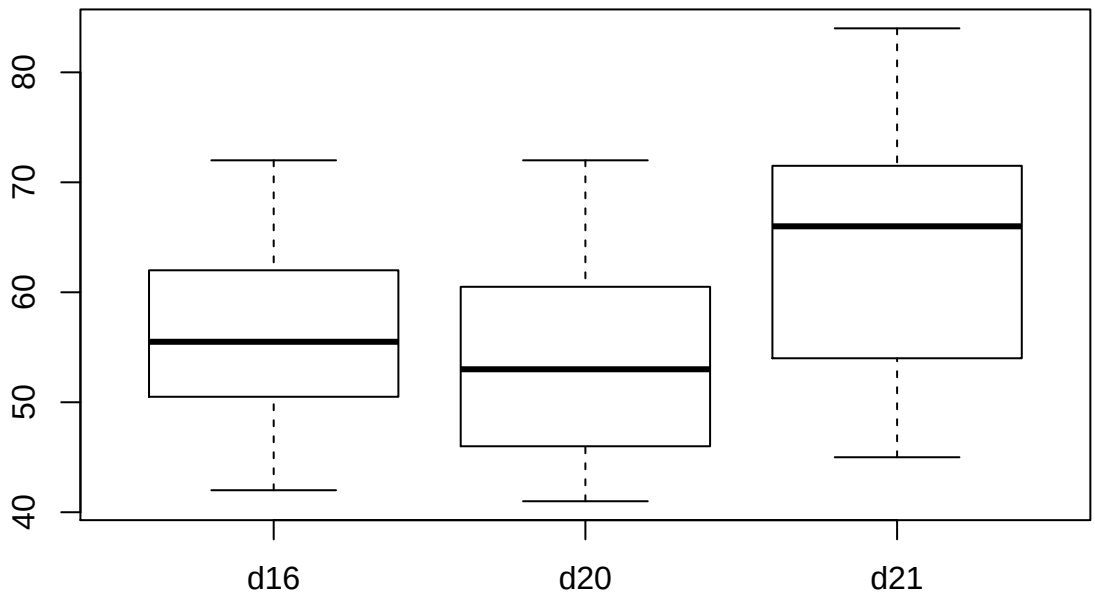
Analyse du taux de VitC en fonction du cultivar. Il y a une nette augmentation de la vitamine C dans les choux du cultivar c52.

```
# Boxplot du taux de VitC en fonction du cultivar
boxplot(VitC~Cult, data=cabbages)
```



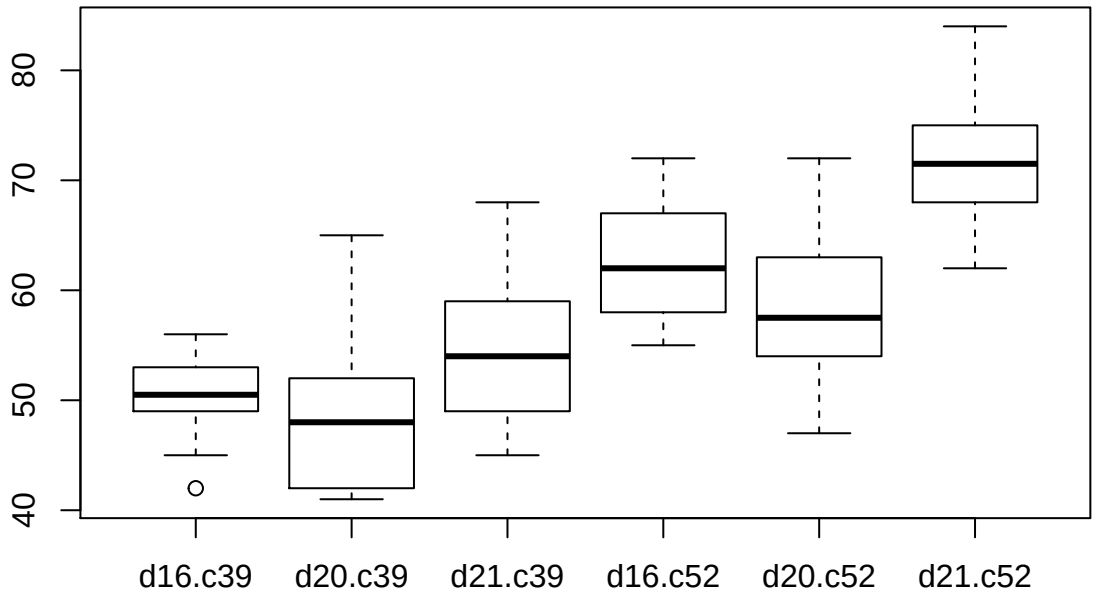
Analyse du taux de VitC en fonction de la date de plantation. Il ne semble pas y avoir d'effet clair de la date de plantation sur le taux en VitC.

```
# Boxplot du taux de VitC en fonction de la date de plantation
boxplot(VitC~Date, data=cabbages)
```



Analyse du taux de VitC en fonction de la date de plantation et du cultivar. Il semble y avoir à la fois un effet du cultivar et de la date sur le taux de VitC.

```
# Boxplot du taux de VitC en fonction de la date et du cultivar
boxplot(VitC~Date+Cult, data=cabbages)
```



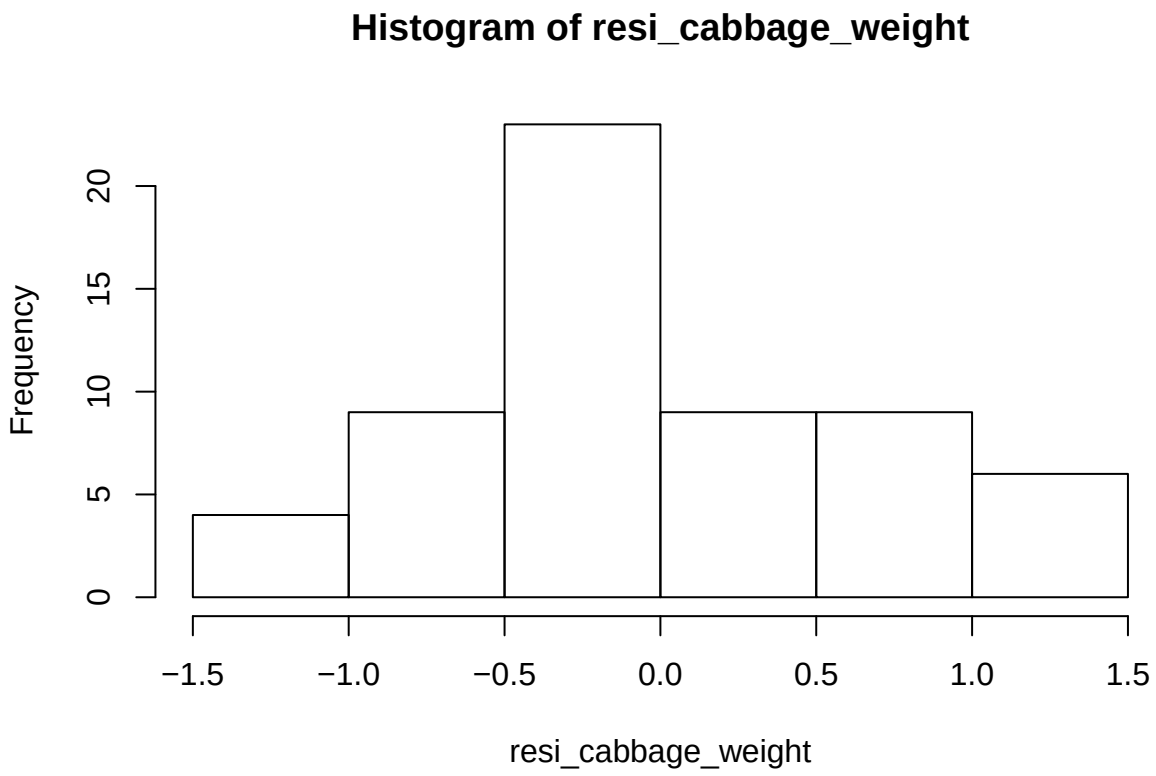
On réalise une ANOVA pour comparer les moyennes de poids en fonction du cultivar et de la date. On voit que toutes les p-values sont < 0.05 et donc tous les facteurs semblent significatifs. Il y a aussi un effet d'interaction entre la date et le cultivar. Il semble donc y avoir des dates ou l'effet du cultivar est différent.

```
# Réalisation d'une ANOVA
AOV_cabbage_weight = aov(HeadWt~Date + Cult + Date:Cult, data=cabbages)
summary(AOV_cabbage_weight)
```

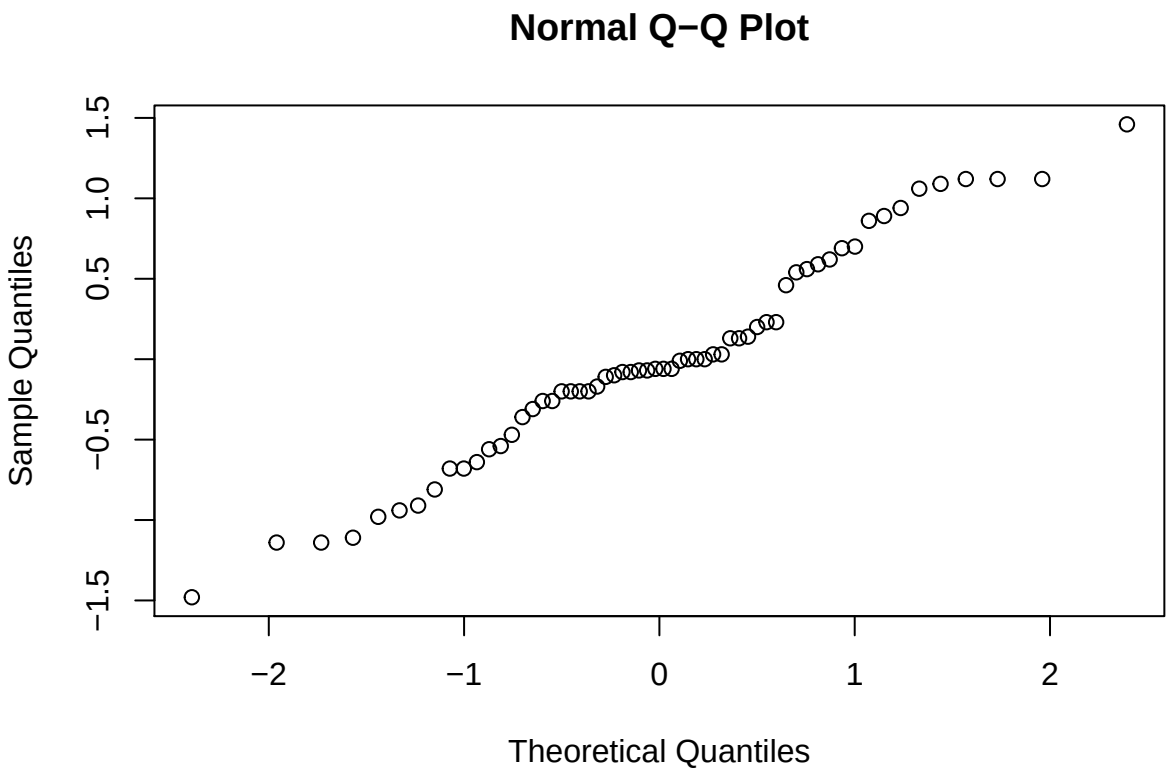
```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## Date         2  7.706   3.853   8.174 0.000792 ***
## Cult         1  5.891   5.891  12.497 0.000845 ***
## Date:Cult    2  6.886   3.443   7.305 0.001557 **
## Residuals   54 25.454   0.471
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

On vérifie si les conditions d'applications de l'ANOVA sont respectées:

```
# Calcul des résidus
resi_cabbage_weight = residuals(AOV_cabbage_weight)
# Histogramme des résidus
hist(resi_cabbage_weight)
```




```
# Q-Q plot
qqnorm(resi_cabbage_weight)
```



Le test de Shapiro montre que les résidus sont distribués normalement.

```
# Test de Shapiro
shapiro.test(resi_cabbage_weight)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resi_cabbage_weight
## W = 0.9748, p-value = 0.2488
```

Le test de Bartlett montre que la date ainsi que le cultivar ont des variances distribuées de façon homogène.

```
# Test de Bartlett
bartlett.test(HeadWt~Date, cabbages)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  HeadWt by Date
## Bartlett's K-squared = 4.0242, df = 2, p-value = 0.1337
```

```
bartlett.test(HeadWt~Cult, cabbages)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  HeadWt by Cult
## Bartlett's K-squared = 0.11037, df = 1, p-value = 0.7397
```

On voit donc que les conditions d'application de l'ANOVA sont respectées.

On peut faire un test de comparaison multiple sur l'analyse de variance = Test de Tukey. Dates: il n'y a pas de différence de poids entre d20 et d16 mais il y a des différences entre d21-d16 et d21-d20. Cultivars: il y a une différence hautement significative de poids entre les deux cultivars. Combinaisons: les combinaisons suivantes présentent une différence de poids significative: d16:c52-d16:c39 d21:c52-d16:c39 d21:c52-d20:c39 d21:c52-d21:c39 d21:c52-d20:c52

```
# Test de Tukey
TukeyHSD(AOV_cabbage_weight)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = HeadWt ~ Date + Cult + Date:Cult, data = cabbages)
##
## $Date
##           diff        lwr         upr     p adj
## d20-d16  0.235 -0.2882332  0.75823319 0.5290010
## d21-d16 -0.615 -1.1382332 -0.09176681 0.0175173
## d21-d20 -0.850 -1.3732332 -0.32676681 0.0007371
##
## $Cult
##               diff        lwr        upr     p adj
## c52-c39 -0.6266667 -0.9820718 -0.2712615 0.0008451
##
## $`Date:Cult`
##                   diff        lwr        upr     p adj
## d20:c39-d16:c39 -0.38 -1.2871459  0.5271459 0.8164215
## d21:c39-d16:c39 -0.44 -1.3471459  0.4671459 0.7070205
## d16:c52-d16:c39 -0.92 -1.8271459 -0.0128541 0.0450106
## d20:c52-d16:c39 -0.07 -0.9771459  0.8371459 0.9999106
## d21:c52-d16:c39 -1.71 -2.6171459 -0.8028541 0.0000119
## d21:c39-d20:c39 -0.06 -0.9671459  0.8471459 0.9999583
## d16:c52-d20:c39 -0.54 -1.4471459  0.3671459 0.5003393
## d20:c52-d20:c39  0.31 -0.5971459  1.2171459 0.9127372
## d21:c52-d20:c39 -1.33 -2.2371459 -0.4228541 0.0008772
## d16:c52-d21:c39 -0.48 -1.3871459  0.4271459 0.6256356
## d20:c52-d21:c39  0.37 -0.5371459  1.2771459 0.8324988
## d21:c52-d21:c39 -1.27 -2.1771459 -0.3628541 0.0016537
## d20:c52-d16:c52  0.85 -0.0571459  1.7571459 0.0784254
## d21:c52-d16:c52 -0.79 -1.6971459  0.1171459 0.1218079
## d21:c52-d20:c52 -1.64 -2.5471459 -0.7328541 0.0000271
```

On peut faire la même chose pour la différence en VitC. L'ANOVA de la différence en VitC selon la date et le cultivar. On voit qu'il y a un effet significatif de la date et du cultivar mais qu'il n'y a pas d'interaction significative entre la date et le cultivar.

```r
# ANOVA de VitC en fonction de la Date et Cult
AOV_cabbage_VitC = aov(VitC~Date+Cult+Date:Cult, data=cabbages)
summary(AOV_cabbage_VitC)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## Date         2  909.3   454.7   9.856 0.000225 ***
## Cult         1 2496.2  2496.2  54.109 1.09e-09 ***
## Date:Cult    2  144.3    72.2   1.564 0.218627
## Residuals   54 2491.1    46.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Le test de Shapiro a une p-value de plus de 0.05 et donc on peut dire que les résidus suivent une distribution normale.

```r
# Test de Shapiro
shapiro.test(residuals(AOV_cabbage_VitC))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(AOV_cabbage_VitC)
## W = 0.96293, p-value = 0.06549
```

Le test de Bartlett a deux fois une p-valeur > 0.05 et donc les variables sont distribuées de façon homogène.

```r
# Test de Bartlett
bartlett.test(VitC~Date, cabbages)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  VitC by Date
## Bartlett's K-squared = 1.9391, df = 2, p-value = 0.3792
```

```r
bartlett.test(VitC~Cult, cabbages)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  VitC by Cult
## Bartlett's K-squared = 0.83351, df = 1, p-value = 0.3613
```

On peut en conclure que la date et le cultivar qui maximisent le poids et le taux de VitC est: d20-c52.

Scéance 4: Régression linéaire

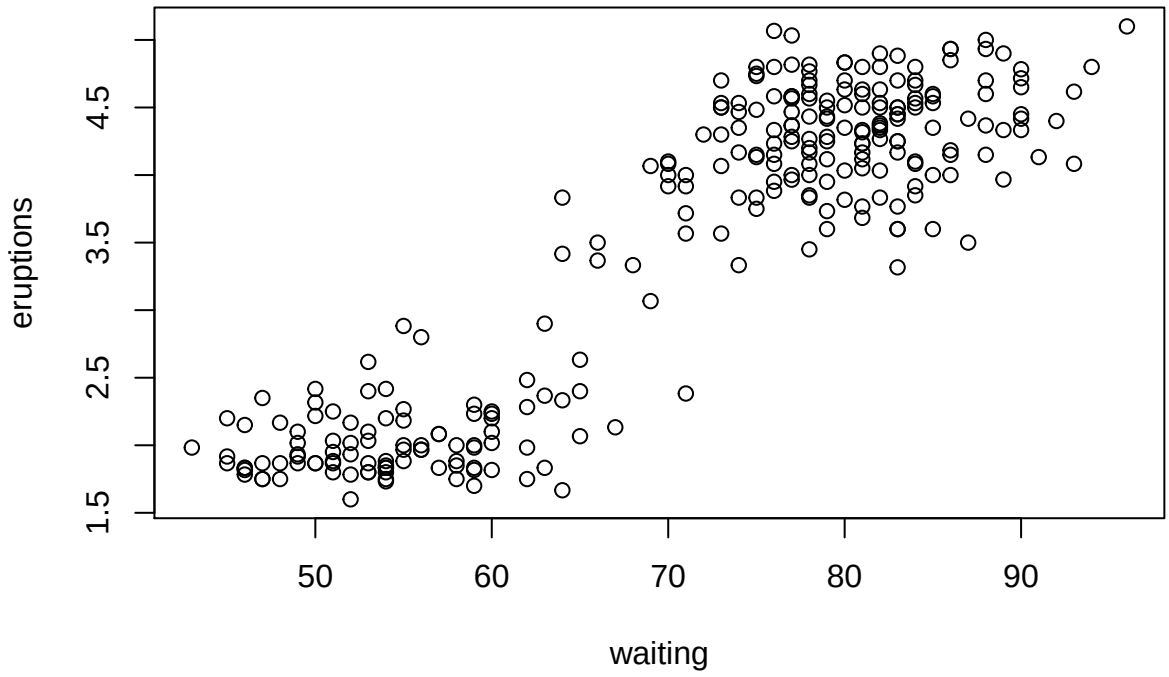
Le jeu de données faithful donne le temps d'attente entre les éruptions et la durée des éruptions d'un geyser du Yellowstone.

```r
# Charger jeu de données faithful
data("faithful")
str(faithful)
```

```
## 'data.frame':    272 obs. of  2 variables:
##  $ eruptions: num  3.6 1.8 3.33 2.28 4.53 ...
##  $ waiting  : num  79 54 74 62 85 55 88 85 51 85 ...
```

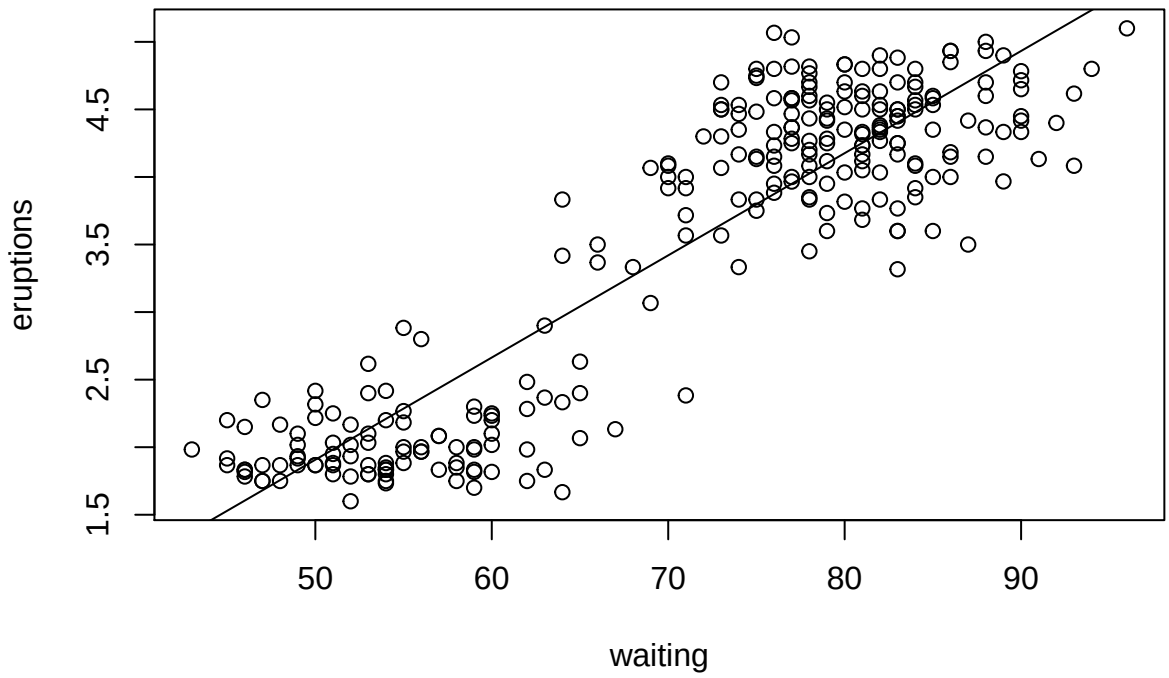
En montrant la durée des éruptions en fonction du temps d'attente, il semble en effet qu'il y ait une corrélation entre le temps d'attente et la durée des éruptions.

```r
# Graphique de durée des éruptions en fonction du temps d'attente
plot(eruptions~waiting, data=faithful)
```



On construit une régression linéaire

```r
# Graphique de durée des éruptions en fonction du temps d'attente
plot(eruptions~waiting, data=faithful)
# Construction d'une régression linéaire
Reg = lm(eruptions~waiting, data=faithful)
abline(Reg)
```



L'ordonnée à l'origine est donnée par intercept (-1.87), la pente de la droite est de (0.0756). Les statistiques t et les p-values nous donnent la probabilité d'avoir respectivement une ordonnée à l'origine et pente plus grande que la statistique sous H0 (ordonnée à l'origine et pente = 0). On peut ici rejeter dans les deux cas H0. On voit également le coefficient de détermination R^2 (donne le pourcentage de variabilité expliquée par le modèle). Ici 81.15% donc assez bon. On voit aussi que la régression est significative vu que la p-value totale est de $2.2 \cdot 10^{-16}$. Il y a donc $2.2 \cdot 10^{-16}$ chances d'avoir une statistique F plus haute que celle qu'on a (1162) sous H0.

```
# Discussion de la régression
summary(Reg)
```

```
##
## Call:
## lm(formula = eruptions ~ waiting, data = faithful)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.29917 -0.37689  0.03508  0.34909  1.19329
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.874016   0.160143  -11.70   <2e-16 ***
## waiting      0.075628   0.002219   34.09   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

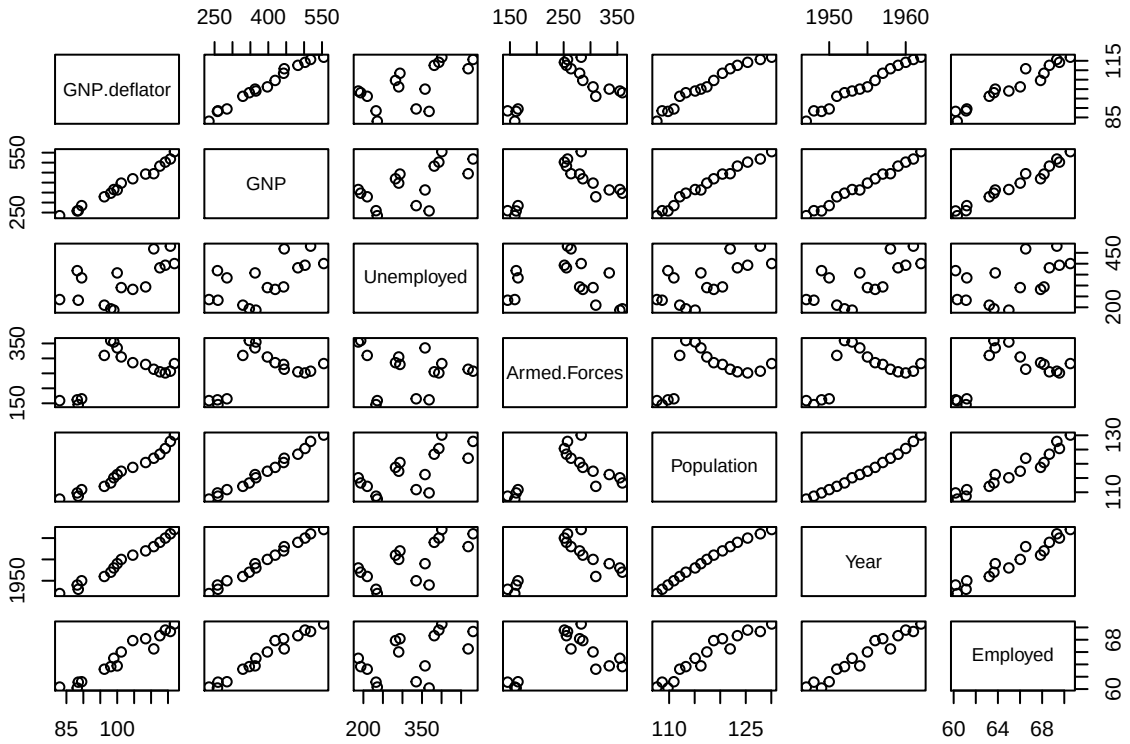
```
## Residual standard error: 0.4965 on 270 degrees of freedom
## Multiple R-squared:  0.8115, Adjusted R-squared:  0.8108
## F-statistic:  1162 on 1 and 270 DF,  p-value: < 2.2e-16
```

```r
# Charger le jeu de données longley
data("longley")
str(longley)
```

```
## 'data.frame':    16 obs. of  7 variables:
##  $ GNP.deflator: num  83 88.5 88.2 89.5 96.2 ...
##  $ GNP         : num  234 259 258 285 329 ...
##  $ Unemployed  : num  236 232 368 335 210 ...
##  $ Armed.Forces: num  159 146 162 165 310 ...
##  $ Population  : num  108 109 110 111 112 ...
##  $ Year        : int  1947 1948 1949 1950 1951 1952 1953 1954 1955 1956 ...
##  $ Employed    : num  60.3 61.1 60.2 61.2 63.2 ...
```

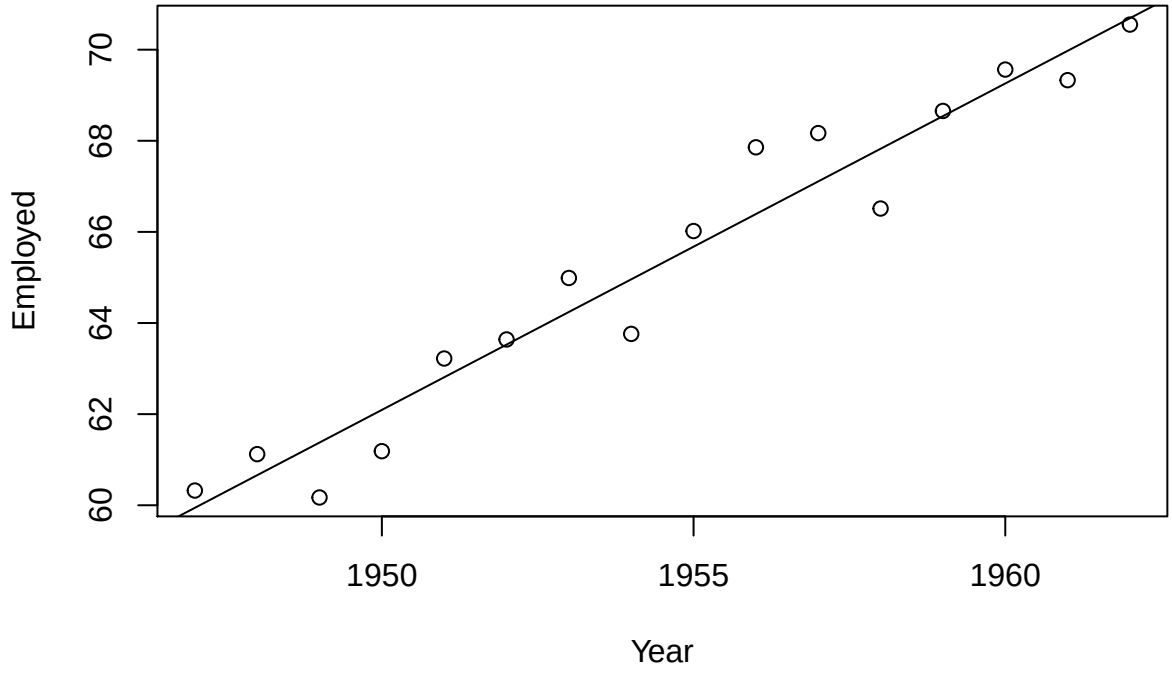
On peut analyser les relations entre les variables en faisant un graphique ou chaque variable est montrée en fonction des autres variables. Une corrélation entre deux variables est montrée par une droite diagonale. Ici l'on veut analyser les facteurs influençant la variable "Employed". On voit donc que les variables GNP.deflator, GNP, population et year sont corrélées avec Employed.

```r
# Examiner relations bi-variées entre les variables
pairs(longley)
```



On réalise deux régressions: une régression montrant Employed en fonction de Year et l'autre montrant Employed en fonction de GNP (Produit Intérieur Brut). On voit que les deux régressions sont hautement significatives (p-value<0.05) et les pentes des droites de régressions sont non-nulles (p-value du test t sur le coefficient <0.05). Le pourcentage de variabilité expliqué par cette variable est donné par le R^2 qui est ici pour les deux >90%. La régression de Employed en fonction de GNP explique plus de variabilité et est donc à privilégier.

```
# Régression linéaire de Employed en fonction de l'Année
plot(Employed~Year, data=longley)
Reg_Em_Ye = lm(Employed~Year, data=longley)
abline(Reg_Em_Ye)
```

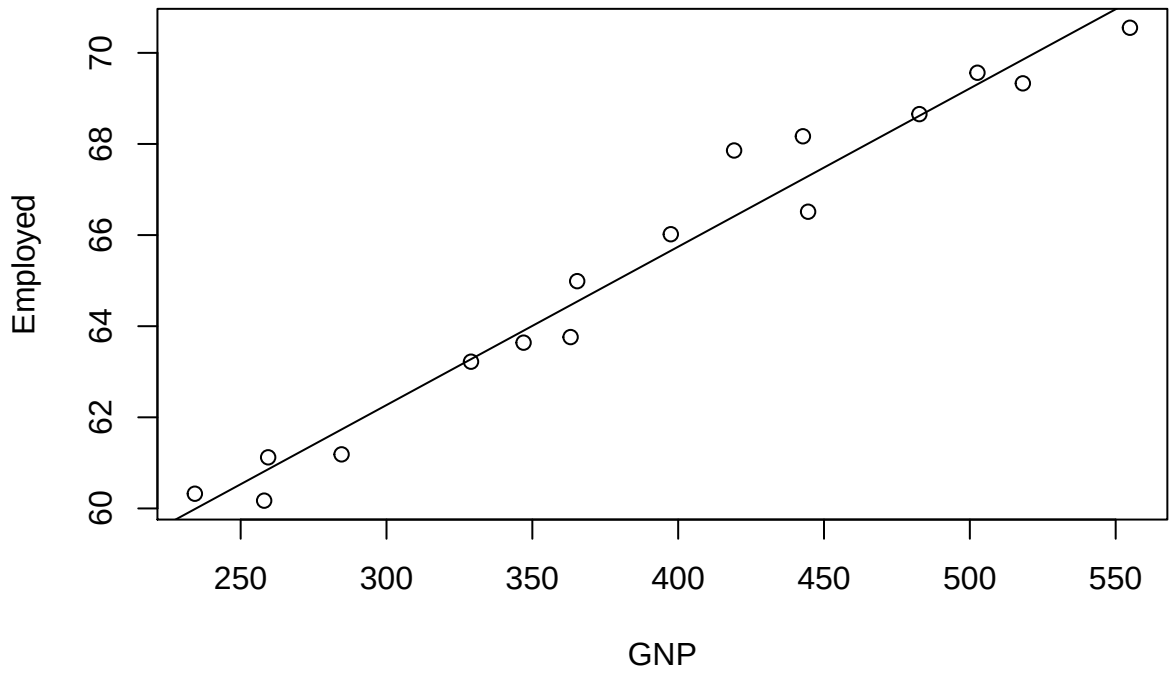



```
summary(Reg_Em_Ye)
```

```
##
## Call:
## lm(formula = Employed ~ Year, data = longley)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.3118 -0.7089  0.2099  0.4244  1.4652
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.335e+03  9.161e+01  -14.57 7.44e-10 ***
```

```
## Year         7.165e-01  4.687e-02   15.29 3.96e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8642 on 14 degrees of freedom
## Multiple R-squared:  0.9435, Adjusted R-squared:  0.9394
## F-statistic: 233.7 on 1 and 14 DF,  p-value: 3.958e-10
```

```
# Régression linéaire de Employed en fonction du GNP
plot(Employed~GNP, data=longley)
Reg_Em_GNP = lm(Employed~GNP, data=longley)
abline(Reg_Em_GNP)
```




```
summary(Reg_Em_GNP)
```

```
##
## Call:
## lm(formula = Employed ~ GNP, data = longley)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.77958 -0.55440 -0.00944  0.34361  1.44594
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 51.843590   0.681372   76.09  < 2e-16 ***
## GNP          0.034752   0.001706   20.37 8.36e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6566 on 14 degrees of freedom
## Multiple R-squared:  0.9674, Adjusted R-squared:  0.965
## F-statistic: 415.1 on 1 and 14 DF,  p-value: 8.363e-12
```

On ajoute la variable Armed Forces au modèle. Cependant, la concerver n'a pas beaucoup de sens vu que le coefficient de significativité de la variable Armed Forces dans cette régression n'est pas différent de 0 (p-value>0.05).

```
# Ajout de Armed Forces au modèle Employed~Year
Reg_Em_Ye_Ar = lm(Employed~Year+Armed.Forces, data=longley)
summary(Reg_Em_Ye_Ar)
```

```
##
## Call:
## lm(formula = Employed ~ Year + Armed.Forces, data = longley)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4437 -0.5519  0.2003  0.4711  1.4147
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.298e+03  1.011e+02 -12.834  9.3e-09 ***
## Year          6.971e-01  5.194e-02  13.420  5.4e-09 ***
## Armed.Forces  3.179e-03  3.554e-03   0.895    0.387
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8705 on 13 degrees of freedom
## Multiple R-squared:  0.9468, Adjusted R-squared:  0.9386
## F-statistic: 115.6 on 2 and 13 DF,  p-value: 5.256e-09
```

```
# Ajout de Armed Forces au modèle Employed~GNP
Reg_Em_Ye_GNP = lm(Employed~GNP+Armed.Forces, data=longley)
summary(Reg_Em_Ye_GNP)
```

```
##
## Call:
## lm(formula = Employed ~ GNP + Armed.Forces, data = longley)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.79572 -0.49110 -0.02805  0.33583  1.42851
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  51.683469   0.804341  64.256  < 2e-16 ***
## GNP           0.034393   0.001966  17.498 2.04e-10 ***
```

```
## Armed.Forces  0.001148   0.002807   0.409    0.689
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6771 on 13 degrees of freedom
## Multiple R-squared:  0.9678, Adjusted R-squared:  0.9628
## F-statistic: 195.3 on 2 and 13 DF,  p-value: 2.005e-10
```

On ajoute la variable Population au modèle. Pour le modèle Employed~Year, la variable Population n'a pas une pente significativement différente de 0 (p-value>0.05). On ne peut dons pas l'ajouter à ce modèle. Au contraire pour le modèle Employed~GNP, la variable Population a une pente qui diffère de 0. De plus, l'ajout de la variable Population permets de légèrement augmenter le R^2 et donc d'améliorer le modèle.

```
# Ajout de Population au modèle Employed~Year
Reg_Em_Ye_Pop = lm(Employed~Year+Population, data=longley)
summary(Reg_Em_Ye_Pop)
```

```
##
## Call:
## lm(formula = Employed ~ Year + Population, data = longley)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4299 -0.5496  0.2083  0.6017  1.2811
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1912.1802   814.5818  -2.347   0.0354 *
## Year            1.0245     0.4345   2.358   0.0347 *
## Population     -0.2121     0.2974  -0.713   0.4884
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8798 on 13 degrees of freedom
## Multiple R-squared:  0.9456, Adjusted R-squared:  0.9372
## F-statistic:   113 on 2 and 13 DF,  p-value: 6.039e-09
```

```
# Ajout de Population au modèle Employed~GNP
Reg_Em_GNP_Pop = lm(Employed~GNP+Population, data=longley)
summary(Reg_Em_GNP_Pop)
```

```
##
## Call:
## lm(formula = Employed ~ GNP + Population, data = longley)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.80899 -0.33282 -0.02329  0.25895  1.08800
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 88.93880   13.78503   6.452 2.16e-05 ***
## GNP          0.06317    0.01065   5.933 4.96e-05 ***
```

```
## Population  -0.40974    0.15214  -2.693   0.0184 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5459 on 13 degrees of freedom
## Multiple R-squared:  0.9791, Adjusted R-squared:  0.9758
## F-statistic: 303.9 on 2 and 13 DF,  p-value: 1.221e-11
```

On regarde si le taux d'emploi peut être prédit par les variables Population, Forces Armées, PIB et Années. On voit que toutes les variables sont significatives sauf la variable Population. Pour simplifier le modèle, on peut uniquement enlever la variable Population. Le modèle final permets d'expliquer 93% de variabilité.

```
# Calcul du taux d'emploi
longley$EmplRate = longley$Employed / (longley$Employed + longley$Unemployed)
Reg_EmplRate = lm(EmplRate~Armed.Forces+GNP+Year+Population, data=longley)
summary(Reg_EmplRate)
```

```
##
## Call:
## lm(formula = EmplRate ~ Armed.Forces + GNP + Year + Population,
##     data = longley)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.019361 -0.005860  0.000048  0.005640  0.024398
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.011e+02  1.546e+01   6.540 4.19e-05 ***
## Armed.Forces  2.103e-04  6.468e-05   3.251  0.00772 **
## GNP           2.803e-03  3.715e-04   7.544 1.14e-05 ***
## Year         -5.170e-02  8.107e-03  -6.378 5.24e-05 ***
## Population   -8.411e-03  5.086e-03  -1.654  0.12641
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01171 on 11 degrees of freedom
## Multiple R-squared:  0.9432, Adjusted R-squared:  0.9225
## F-statistic: 45.63 on 4 and 11 DF,  p-value: 8.75e-07
```

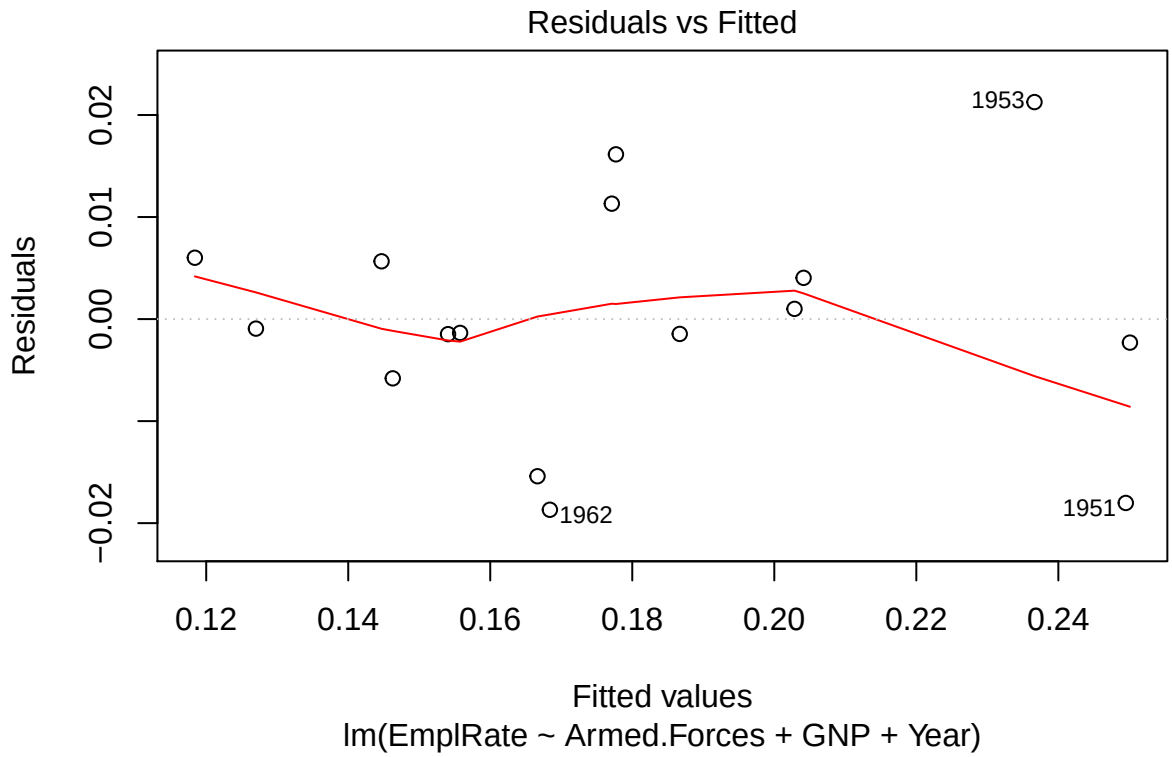
```
# Simplification du modèle
Reg_EmplRate_Simp = lm(EmplRate~Armed.Forces+GNP+Year, data=longley)
summary(Reg_EmplRate_Simp)
```

```
##
## Call:
## lm(formula = EmplRate ~ Armed.Forces + GNP + Year, data = longley)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.018687 -0.003181 -0.001144  0.005750  0.021267
##
## Coefficients:
```

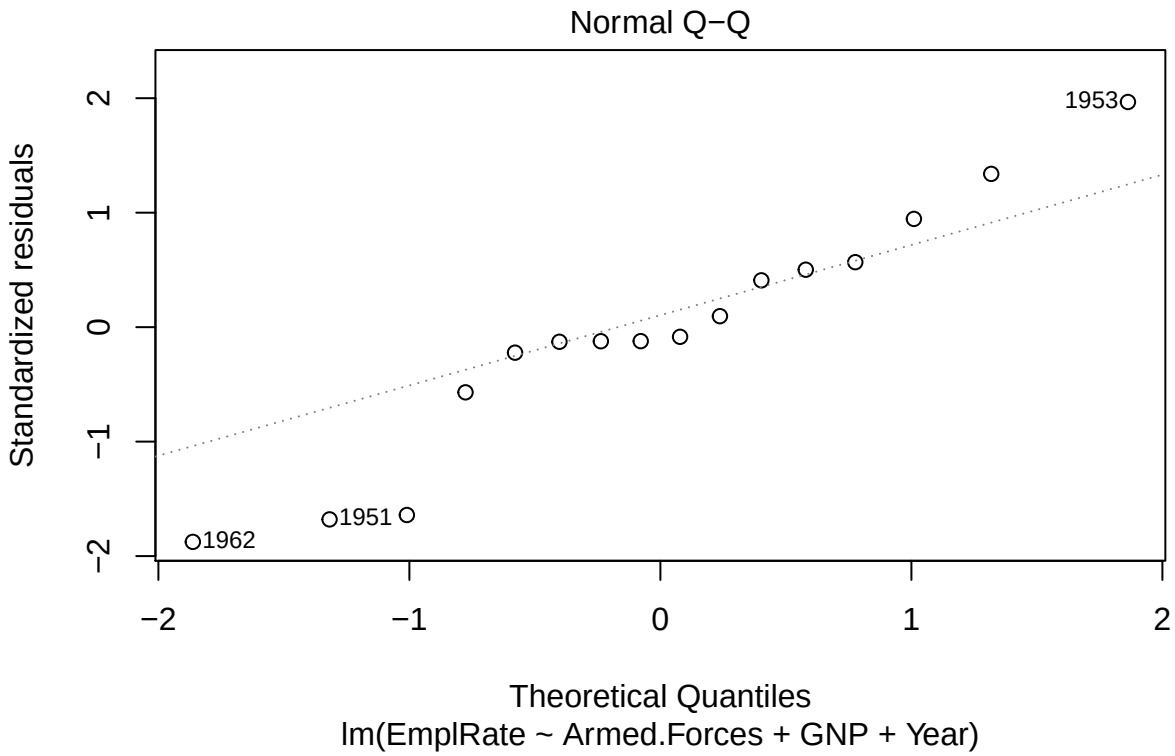
```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.140e+02  1.426e+01   7.997 3.78e-06 ***
## Armed.Forces  2.758e-04  5.467e-05   5.045 0.000287 ***
## GNP           2.537e-03  3.582e-04   7.081 1.28e-05 ***
## Year         -5.879e-02  7.364e-03  -7.983 3.84e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01253 on 12 degrees of freedom
## Multiple R-squared:  0.929,  Adjusted R-squared:  0.9113
## F-statistic: 52.36 on 3 and 12 DF,  p-value: 3.631e-07
```

On peut vérifier les conditions d'applications de la régression linéaire. On réalise d'abord un graphique des résidus pour voir si ils sont distribués de façon normale. On voit que les résidus semblent être distribués normalement mais il y a peu de points. On ne peut donc pas réellement en tirer de conclusions. Le Q-Q plot permets de voir si la distribution des quantiles corresponds à une distribution théorique des quantiles. Ici on voit effectivement que c'est le cas vu qu'ils semblent proche de la diagonale. Le test de Shapiro permets de voir si les résidus sont distribués normalement. La distribution normale étant l'hypothèse nulle (donc ce qu'on veut est une p-value>0.05). Ceci est le cas ici et donc les résidus sont distribués normalement. Les conditions d'applications du modèle sont donc appliquées.

```r
# Vérification des conditions d'applications par normalité des résidus
plot(Reg_EmplRate_Simp, which = 1)
```



```r
# Vérification des conditions d'application par un graphique quantile-quantile
plot(Reg_EmplRate_Simp, which = 2)
```




```r
# Test de Shapiro
shapiro.test(residuals(Reg_EmplRate_Simp))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(Reg_EmplRate_Simp)
## W = 0.95118, p-value = 0.5086
```

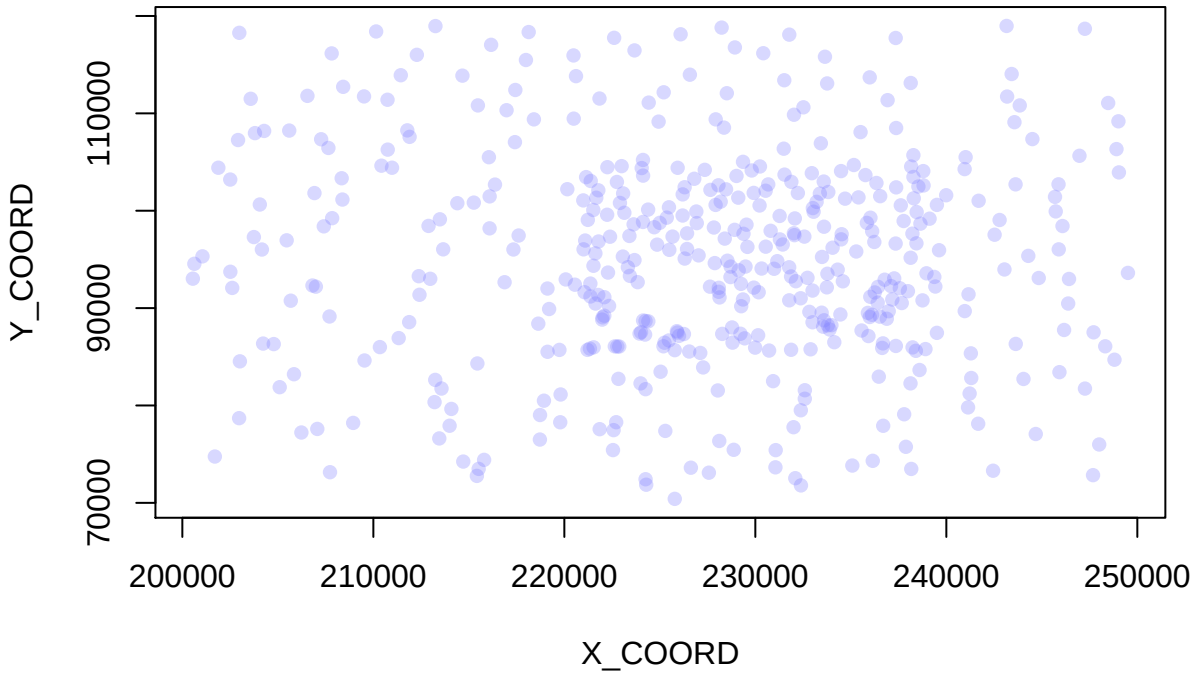
Scéance 5: Régression multiple

```r
# Enlever les données vides du data frame
file_data = read.table("~/Bureau/Ecole/MA1/Q1/Acquisition_et_analyse_de_donnees/donnees_ips.txt", heade
myD = na.omit(file_data)
# Construction de la variable LGIT correspondant au log10 de myD$AVIT
myD$LOGIT = log10(myD$AVIT + 1)
myD$SUP = as.factor(myD$SUP)
str(myD)
```

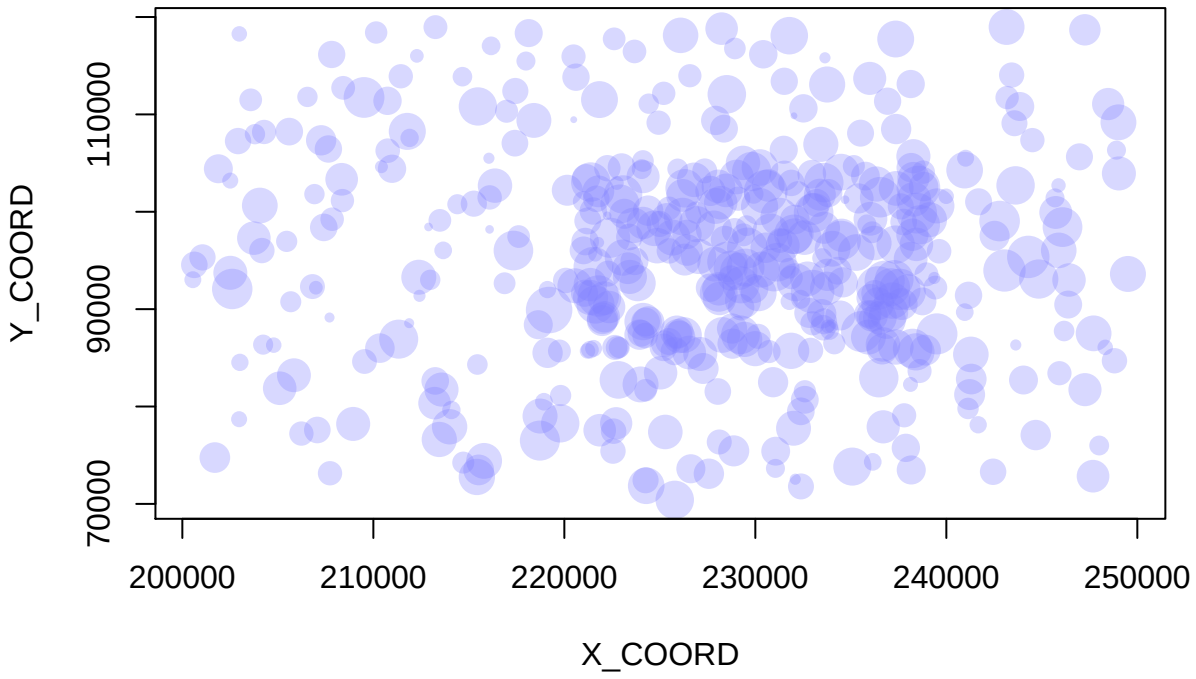
```
## 'data.frame':    459 obs. of  14 variables:
##  $ X_COORD: num  221139 222253 222999 224050 224123 ...
##  $ Y_COORD: num  103450 104473 104580 104382 103604 ...
##  $ AVIT   : int  91 67 85 23 196 8 542 142 27 162 ...
##  $ SUP    : Factor w/ 3 levels "1","2","3": 1 1 2 1 1 2 2 2 2 1 ...
##  $ ACCQT  : num  8.38 22.26 49.13 16.71 35.19 ...
##  $ CLCC   : num  0.0306 0.0377 0.0245 0.0403 0.0627 ...
##  $ CLCM   : num  0.183 0.156 0.176 0.212 0.286 ...
##  $ CLCF   : num  0.0245 0.0984 0.1198 0.1198 0.1025 ...
##  $ OUVQL  : int  1 3 1 2 3 1 1 1 0 1 ...
##  $ EPIQLP : int  1 2 0 0 0 0 0 0 1 0 ...
##  $ EPIQLD : int  2 2 2 2 2 2 1 1 1 2 ...
##  $ MARR04 : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ MARR05 : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ LOGIT  : num  1.96 1.83 1.93 1.38 2.29 ...
##  - attr(*, "na.action")=Class 'omit'  Named int 72
##   .. ..- attr(*, "names")= chr "72"
```

On réalise une cartographie des sites de capture grâce aux coordonnées et on représente les points avec une taille proportionelle à la variable LOGIT.

```
# Cartographie des sites de capture
plot(Y_COORD ~ X_COORD, myD, pch = 16, col = rgb(0.5,0.5,1,0.3))
```



```r
plot(Y_COORD ~ X_COORD, myD, pch = 16, col = rgb(0.5,0.5,1,0.3)
, cex = myD$LOGIT)
```



Construction d'une régression multiple prédisant le nombre d'ips (AVIT) en fonction de toutes les variables prédictives. Il ne semble y avoir que trois variables (CLCM, EPIQLP et MARR05) qui soient significatives. Le modèle contient donc de nombreuses variables non-significatives et donc le modèle n'est pas très bon.

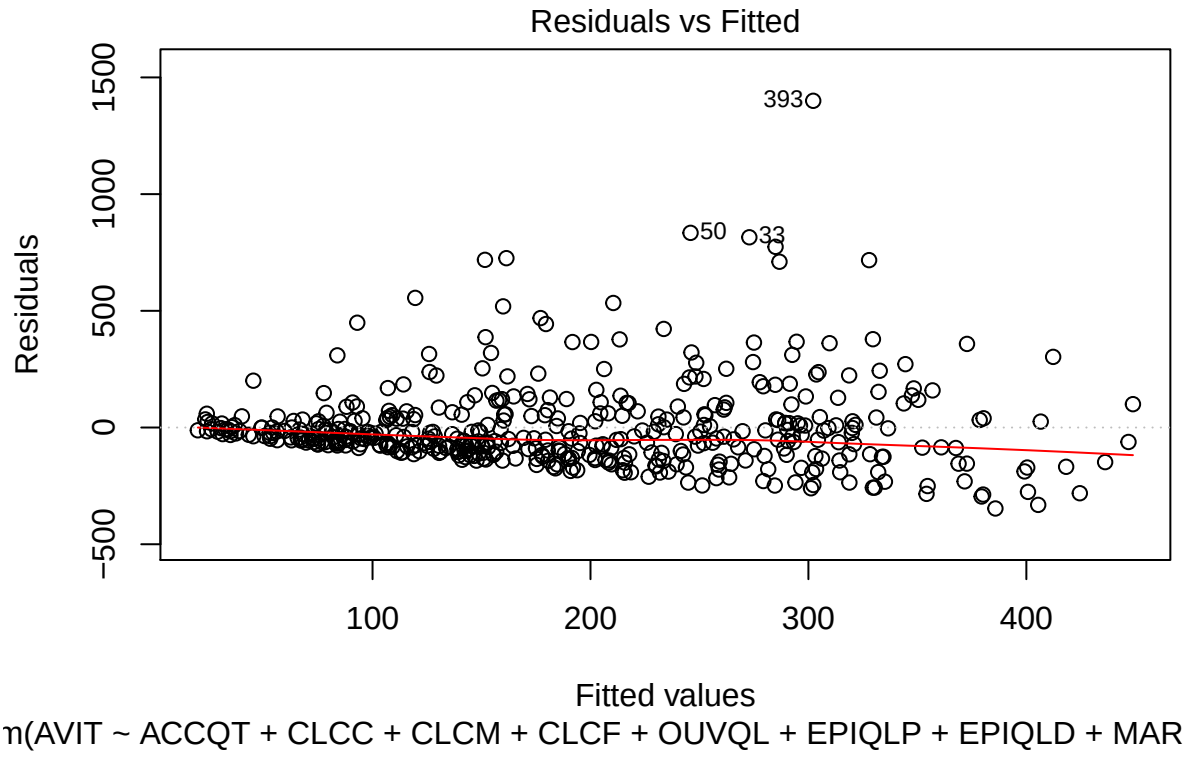
```r
# Régression multiple: prédiction de AVIT
myFullReg = lm(AVIT~ACCQT+CLCC+CLCM+CLCF+OUVQL+EPIQLP+EPIQLD+MARR04+MARR05, data=myD)
summary(myFullReg)
```

```
##
## Call:
## lm(formula = AVIT ~ ACCQT + CLCC + CLCM + CLCF + OUVQL + EPIQLP +
##     EPIQLD + MARR04 + MARR05, data = myD)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -346.79 -100.39  -37.29   44.02 1399.80
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  27.5723    38.5914   0.714  0.47531
## ACCQT        -0.1088     0.7420  -0.147  0.88353
## CLCC          5.0538    89.0844   0.057  0.95479
```

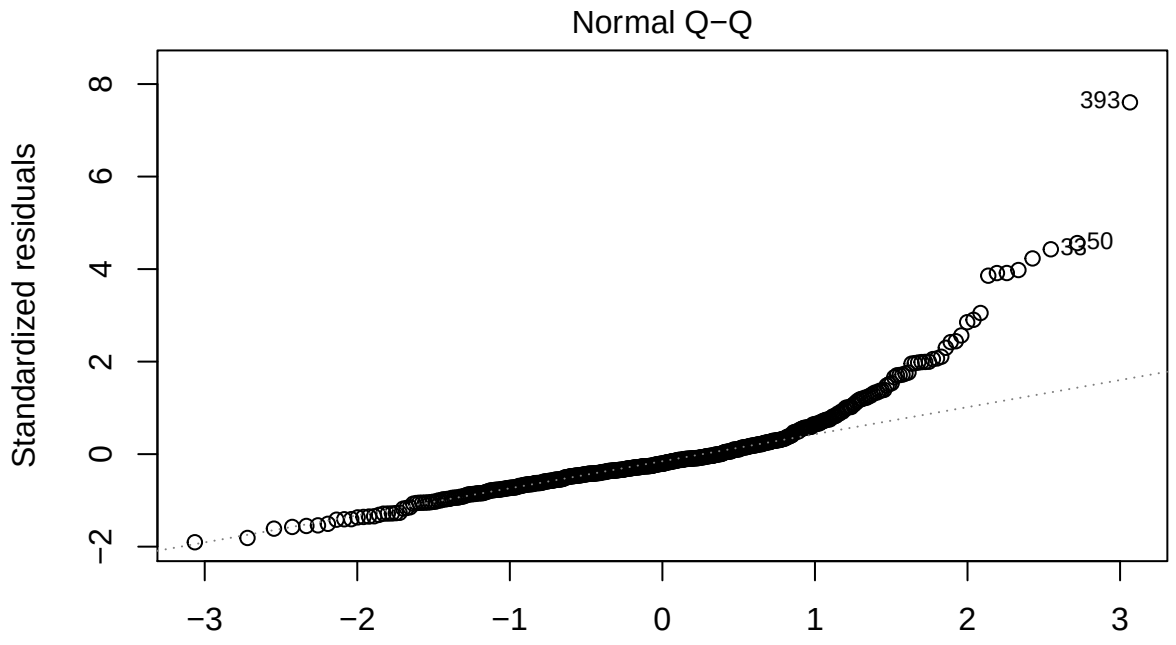
```
## CLCM        230.5298    71.4175   3.228  0.00134 **
## CLCF        -76.7279    71.0304  -1.080  0.28063
## OUVQL        17.5593     9.7338   1.804  0.07191 .
## EPIQLP       48.0425     9.2741   5.180 3.35e-07 ***
## EPIQLD        3.1069    11.0368   0.282  0.77845
## MARR04      -52.6417   205.5829  -0.256  0.79802
## MARR05      819.8148   406.1932   2.018  0.04416 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 184.9 on 449 degrees of freedom
## Multiple R-squared:  0.2218, Adjusted R-squared:  0.2062
## F-statistic: 14.22 on 9 and 449 DF,  p-value: < 2.2e-16
```

On analyse les conditions d'applications du modèle en analysant la normalité des résidus (graphique des résidus et Q-Q plot). On voit que les résidus ne sont pas distribués de façon normale vu qu'ils ne s'allignent pas le long de la ligne. De plus, la variabilité des résidus augmente quand la valeur prédite est grande. Il n'y a donc pas d'homoscédasticité (homogénéité des variances).

```
# Analyse de la normalité des résidus
plot(myFullReg, which = 1)
```




```
# Analyse de la normalité des résidus par Q-Q plot
plot(myFullReg, which = 2)
```



ɳ(AVIT ~ ACCQT + CLCC + CLCM + CLCF + OUVQL + EPIQLP + EPIQLD + MARR04

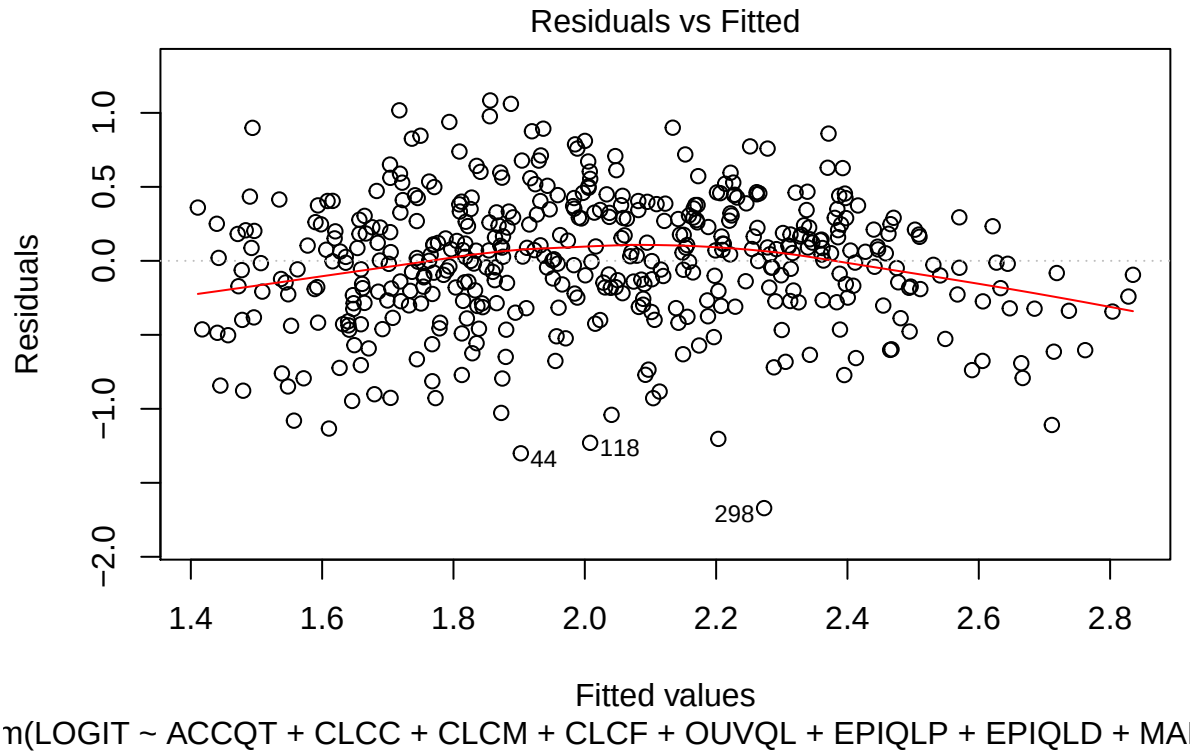
On refait le même modèle en utilisant cette fois-ci le logarithme des captures (LOGIT). On teste ensuite les conditions d'applications de ce modèle. On voit que la transformation logarithmique améliore l'homoscédasticité des résidus et la normalité des données. Le graphique des résidus montre une distribution plus homogène surtout en fonction de la taille des valeurs prédites.

```
# Régression multiple: prédiction de AVIT
myFullReg_Log = lm(LOGIT~ACCQT+CLCC+CLCM+CLCF+OUVQL+EPIQLP+EPIQLD+MARR04+MARR05, data=myD)
summary(myFullReg_Log)
```

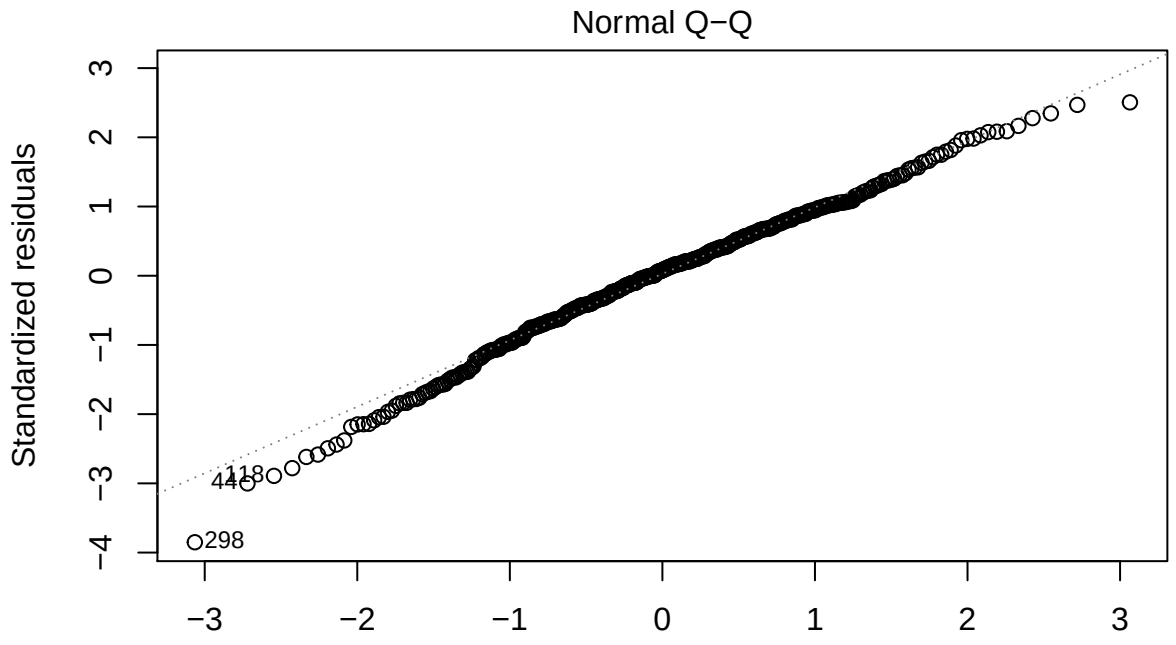
```
##
## Call:
## lm(formula = LOGIT ~ ACCQT + CLCC + CLCM + CLCF + OUVQL + EPIQLP +
##     EPIQLD + MARR04 + MARR05, data = myD)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.67102 -0.26778  0.03479  0.29066  1.08401
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.493974   0.090975  16.422  < 2e-16 ***
## ACCQT       -0.002065   0.001749  -1.180 0.238466
## CLCC         0.137196   0.210006   0.653 0.513900
## CLCM         0.679977   0.168358   4.039 6.32e-05 ***
## CLCF        -0.257390   0.167446  -1.537 0.124960
## OUVQL        0.088392   0.022946   3.852 0.000134 ***
```

```
## EPIQLP       0.103284   0.021863   4.724 3.09e-06 ***
## EPIQLD       0.041812   0.026018   1.607 0.108749
## MARR04      -0.110525   0.484638  -0.228 0.819705
## MARR05       1.806010   0.957553   1.886 0.059931 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4358 on 449 degrees of freedom
## Multiple R-squared:  0.3445, Adjusted R-squared:  0.3313
## F-statistic: 26.22 on 9 and 449 DF,  p-value: < 2.2e-16
```

```r
# Analyse de la normalité des résidus
plot(myFullReg_Log, which = 1)
```




```r
# Analyse de la normalité des résidus par Q-Q plot
plot(myFullReg_Log, which = 2)
```



ɳ(LOGIT ~ ACCQT + CLCC + CLCM + CLCF + OUVQL + EPIQLP + EPIQLD + MARR0

On essaye maintenant de simplifier le modèle en enlevant les variables une à une. Ceci nous permets d'avoir un modèle hautement significatif contenant uniquement des variables significatives. Le problème est que l'on explique très peu de variabilité avec ce modèle (seulement 33%).

```
# Enlève les variables une à une en commençant par celles ayant les moins bonnes p-values
myFullReg = lm(LOGIT ~ ACCQT + CLCC + CLCM + CLCF + OUVQL + EPIQLP + EPIQLD + MARR04 + MARR05, data = my
summary(myFullReg)
```

```
##
## Call:
## lm(formula = LOGIT ~ ACCQT + CLCC + CLCM + CLCF + OUVQL + EPIQLP +
##     EPIQLD + MARR04 + MARR05, data = myD)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.67102 -0.26778  0.03479  0.29066  1.08401
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.493974   0.090975  16.422  < 2e-16 ***
## ACCQT       -0.002065   0.001749  -1.180 0.238466
## CLCC         0.137196   0.210006   0.653 0.513900
## CLCM         0.679977   0.168358   4.039 6.32e-05 ***
## CLCF        -0.257390   0.167446  -1.537 0.124960
## OUVQL        0.088392   0.022946   3.852 0.000134 ***
## EPIQLP       0.103284   0.021863   4.724 3.09e-06 ***
```

```
## EPIQLD       0.041812   0.026018   1.607 0.108749
## MARR04      -0.110525   0.484638  -0.228 0.819705
## MARR05       1.806010   0.957553   1.886 0.059931 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4358 on 449 degrees of freedom
## Multiple R-squared:  0.3445, Adjusted R-squared:  0.3313
## F-statistic: 26.22 on 9 and 449 DF,  p-value: < 2.2e-16
```

```
myReg = lm(LOGIT ~ ACCQT + CLCC + CLCM + CLCF + OUVQL + EPIQLP + EPIQLD + MARR05, data = myD)
summary(myReg)
```

```
##
## Call:
## lm(formula = LOGIT ~ ACCQT + CLCC + CLCM + CLCF + OUVQL + EPIQLP +
##     EPIQLD + MARR05, data = myD)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.66899 -0.26906  0.03679  0.29244  1.08264
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.492476   0.090642  16.466  < 2e-16 ***
## ACCQT       -0.002093   0.001743  -1.201 0.230289
## CLCC         0.131882   0.208489   0.633 0.527343
## CLCM         0.677680   0.167880   4.037 6.37e-05 ***
## CLCF        -0.253320   0.166316  -1.523 0.128431
## OUVQL        0.088520   0.022915   3.863 0.000129 ***
## EPIQLP       0.103052   0.021816   4.724 3.10e-06 ***
## EPIQLD       0.041985   0.025979   1.616 0.106776
## MARR05       1.755948   0.931070   1.886 0.059946 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4354 on 450 degrees of freedom
## Multiple R-squared:  0.3444, Adjusted R-squared:  0.3327
## F-statistic: 29.55 on 8 and 450 DF,  p-value: < 2.2e-16
```

```
myReg = lm(LOGIT ~ ACCQT + CLCM + CLCF + OUVQL + EPIQLP + EPIQLD + MARR05, data = myD)
summary(myReg)
```

```
##
## Call:
## lm(formula = LOGIT ~ ACCQT + CLCM + CLCF + OUVQL + EPIQLP + EPIQLD +
##     MARR05, data = myD)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.67777 -0.26647  0.03242  0.28335  1.08076
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.498413   0.090094  16.632  < 2e-16 ***
## ACCQT       -0.002147   0.001740  -1.234   0.2179
## CLCM         0.660725   0.165616   3.990 7.72e-05 ***
## CLCF        -0.240376   0.164943  -1.457   0.1457
## OUVQL        0.090243   0.022738   3.969 8.40e-05 ***
## EPIQLP       0.103616   0.021783   4.757 2.65e-06 ***
## EPIQLD       0.045652   0.025308   1.804   0.0719 .
## MARR05       1.765240   0.930335   1.897   0.0584 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4351 on 451 degrees of freedom
## Multiple R-squared:  0.3438, Adjusted R-squared:  0.3336
## F-statistic: 33.76 on 7 and 451 DF,  p-value: < 2.2e-16
```

```
myReg = lm(LOGIT ~ CLCM + CLCF + OUVQL + EPIQLP + EPIQLD + MARR05, data = myD)
summary(myReg)
```

```
##
## Call:
## lm(formula = LOGIT ~ CLCM + CLCF + OUVQL + EPIQLP + EPIQLD +
##     MARR05, data = myD)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.66731 -0.26418  0.04185  0.29200  1.08839
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.42279    0.06607  21.533  < 2e-16 ***
## CLCM         0.65319    0.16560   3.944 9.27e-05 ***
## CLCF        -0.24689    0.16495  -1.497   0.1352
## OUVQL        0.10547    0.01911   5.519 5.77e-08 ***
## EPIQLP       0.10570    0.02173   4.864 1.59e-06 ***
## EPIQLD       0.04474    0.02531   1.768   0.0778 .
## MARR05       1.92812    0.92145   2.092   0.0370 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4353 on 452 degrees of freedom
## Multiple R-squared:  0.3416, Adjusted R-squared:  0.3329
## F-statistic: 39.09 on 6 and 452 DF,  p-value: < 2.2e-16
```

```
myReg = lm(LOGIT ~ CLCM + OUVQL + EPIQLP + EPIQLD + MARR05, data = myD)
summary(myReg)
```

```
##
## Call:
## lm(formula = LOGIT ~ CLCM + OUVQL + EPIQLP + EPIQLD + MARR05,
##     data = myD)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -1.65915 -0.26288  0.04049  0.29430  1.08620
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.38889    0.06216  22.345  < 2e-16 ***
## CLCM         0.68907    0.16408   4.200 3.22e-05 ***
## OUVQL        0.09535    0.01790   5.327 1.58e-07 ***
## EPIQLP       0.10912    0.02164   5.043 6.65e-07 ***
## EPIQLD       0.04795    0.02526   1.899   0.0582 .
## MARR05       1.91091    0.92264   2.071   0.0389 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4359 on 453 degrees of freedom
## Multiple R-squared:  0.3383, Adjusted R-squared:  0.331
## F-statistic: 46.33 on 5 and 453 DF,  p-value: < 2.2e-16
```

```
myReg = lm(LOGIT ~ CLCM + OUVQL + EPIQLP + MARR05, data = myD)
summary(myReg)
```

```
##
## Call:
## lm(formula = LOGIT ~ CLCM + OUVQL + EPIQLP + MARR05, data = myD)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.63059 -0.26037  0.03689  0.28676  1.09021
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.45686    0.05096  28.589  < 2e-16 ***
## CLCM         0.73152    0.16301   4.487 9.15e-06 ***
## OUVQL        0.08963    0.01770   5.065 5.94e-07 ***
## EPIQLP       0.13445    0.01708   7.870 2.63e-14 ***
## MARR05       2.07474    0.92123   2.252   0.0248 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4372 on 454 degrees of freedom
## Multiple R-squared:  0.3331, Adjusted R-squared:  0.3272
## F-statistic: 56.68 on 4 and 454 DF,  p-value: < 2.2e-16
```

On peut automatiser le processus de sélection en utilisant la fonction step(). On voit que cette fonction enlève moins de variables que la réduction manuelle. On obtient un modèle avec des variables proches du seuil de significativité.

```
# Automatisation de la sélection des variables via la fonction step
myStepReg = step(myFullReg_Log, direction = 'both')
```

```
## Start:  AIC=-752.51
## LOGIT ~ ACCQT + CLCC + CLCM + CLCF + OUVQL + EPIQLP + EPIQLD +
##     MARR04 + MARR05
```

```
##
##          Df Sum of Sq    RSS     AIC
## - MARR04  1    0.0099 85.296 -754.46
## - CLCC    1    0.0811 85.367 -754.08
## - ACCQT   1    0.2647 85.551 -753.09
## <none>                85.286 -752.51
## - CLCF    1    0.4488 85.735 -752.10
## - EPIQLD  1    0.4906 85.777 -751.88
## - MARR05  1    0.6757 85.962 -750.89
## - OUVQL   1    2.8186 88.105 -739.59
## - CLCM    1    3.0985 88.385 -738.13
## - EPIQLP  1    4.2393 89.526 -732.25
##
## Step:  AIC=-754.46
## LOGIT ~ ACCQT + CLCC + CLCM + CLCF + OUVQL + EPIQLP + EPIQLD +
##     MARR05
##
##          Df Sum of Sq    RSS     AIC
## - CLCC    1    0.0758 85.372 -756.05
## - ACCQT   1    0.2735 85.570 -754.99
## <none>                85.296 -754.46
## - CLCF    1    0.4397 85.736 -754.10
## - EPIQLD  1    0.4950 85.791 -753.80
## - MARR05  1    0.6742 85.970 -752.85
## + MARR04  1    0.0099 85.286 -752.51
## - OUVQL   1    2.8284 88.125 -741.49
## - CLCM    1    3.0887 88.385 -740.13
## - EPIQLP  1    4.2294 89.526 -734.25
##
## Step:  AIC=-756.05
## LOGIT ~ ACCQT + CLCM + CLCF + OUVQL + EPIQLP + EPIQLD + MARR05
##
##          Df Sum of Sq    RSS     AIC
## - ACCQT   1    0.2882 85.660 -756.51
## <none>                85.372 -756.05
## - CLCF    1    0.4020 85.774 -755.90
## - EPIQLD  1    0.6160 85.988 -754.75
## + CLCC    1    0.0758 85.296 -754.46
## - MARR05  1    0.6815 86.053 -754.40
## + MARR04  1    0.0047 85.367 -754.08
## - OUVQL   1    2.9818 88.354 -742.29
## - CLCM    1    3.0128 88.385 -742.13
## - EPIQLP  1    4.2830 89.655 -735.58
##
## Step:  AIC=-756.51
## LOGIT ~ CLCM + CLCF + OUVQL + EPIQLP + EPIQLD + MARR05
##
##          Df Sum of Sq    RSS     AIC
## <none>                85.660 -756.51
## - CLCF    1    0.4245 86.085 -756.24
## + ACCQT   1    0.2882 85.372 -756.05
## - EPIQLD  1    0.5922 86.252 -755.34
## + CLCC    1    0.0905 85.570 -754.99
## + MARR04  1    0.0108 85.649 -754.56
```

```
## - MARR05  1    0.8298 86.490 -754.08
## - CLCM    1    2.9486 88.609 -742.97
## - EPIQLP  1    4.4836 90.144 -735.09
## - OUVQL   1    5.7717 91.432 -728.58
```

```r
summary(myStepReg)
```

```
##
## Call:
## lm(formula = LOGIT ~ CLCM + CLCF + OUVQL + EPIQLP + EPIQLD +
##     MARR05, data = myD)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.66731 -0.26418  0.04185  0.29200  1.08839
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.42279    0.06607  21.533  < 2e-16 ***
## CLCM         0.65319    0.16560   3.944 9.27e-05 ***
## CLCF        -0.24689    0.16495  -1.497   0.1352
## OUVQL        0.10547    0.01911   5.519 5.77e-08 ***
## EPIQLP       0.10570    0.02173   4.864 1.59e-06 ***
## EPIQLD       0.04474    0.02531   1.768   0.0778 .
## MARR05       1.92812    0.92145   2.092   0.0370 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4353 on 452 degrees of freedom
## Multiple R-squared:  0.3416, Adjusted R-squared:  0.3329
## F-statistic: 39.09 on 6 and 452 DF,  p-value: < 2.2e-16
```

On reprend le modèle myFullReg en y ajoutant la variable SUP. La variable SUP est qualitative (3 niveaux) et a donc été incorporée comme deux variables quantitatives binaires. Ces variables binaires décrivent si l'on a un support de type 2 (SUP2), ou de type 3 (SUP3), sachant que si l'on a ni SUP2, ni SUP3, on a forcément SUP1.

```r
# Ajout de la variable SUP au modèle
myFinalReg = lm(LOGIT ~ CLCM + OUVQL + EPIQLP + MARR05 + SUP, data = myD)
summary(myFinalReg)
```

```
##
## Call:
## lm(formula = LOGIT ~ CLCM + OUVQL + EPIQLP + MARR05 + SUP, data = myD)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.62802 -0.26941  0.03082  0.28187  1.11132
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.53836    0.05972  25.758  < 2e-16 ***
## CLCM         0.67821    0.16149   4.200 3.22e-05 ***
```

```
## OUVQL        0.06319    0.01986   3.181 0.001566 **
## EPIQLP       0.14074    0.01700   8.278 1.42e-15 ***
## MARR05       2.21518    0.92242   2.401 0.016732 *
## SUP2        -0.11413    0.05849  -1.951 0.051639 .
## SUP3        -0.69552    0.20068  -3.466 0.000579 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4314 on 452 degrees of freedom
## Multiple R-squared:  0.3536, Adjusted R-squared:  0.345
## F-statistic:  41.2 on 6 and 452 DF,  p-value: < 2.2e-16
```

On réalise une analyse de variance sur la régression. Ceci nous montre que la variable SUP est significative dans ce modèle.

```
# Analyse de variance sur la régression
summary(aov(myFinalReg))
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)
## CLCM          1  15.53  15.527  83.445  < 2e-16 ***
## OUVQL         1  15.10  15.102  81.161  < 2e-16 ***
## EPIQLP        1  11.74  11.736  63.070  1.6e-14 ***
## MARR05        1   0.97   0.969   5.210 0.022924 *
## SUP           2   2.67   1.333   7.161 0.000867 ***
## Residuals   452  84.10   0.186
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

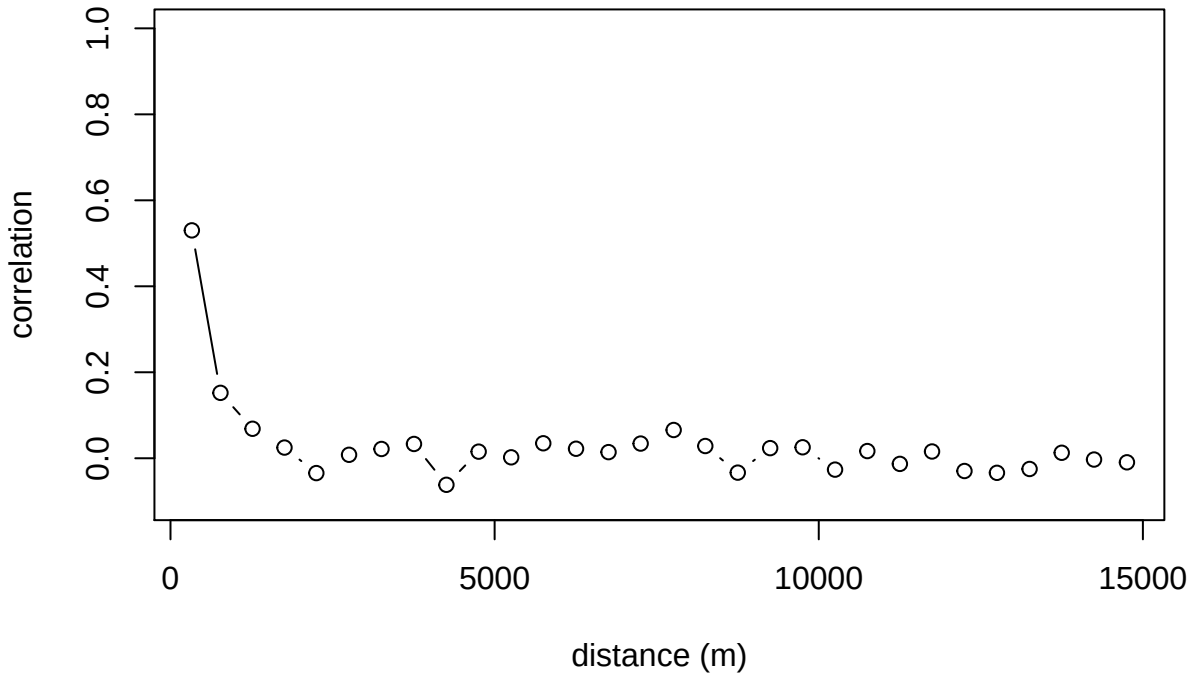
On réalise le corrélogramme des résidus en fonction des coördonnées. Ceci nous permets de voir si il y a de l'autocorrélation spatiale entre les variables. On voit en effet que l'autocorrélation spatiale diminue en fonction de la distance séparant les points. Les points séparés par une faible distance ne sont pas indépendants (corrélation de 0.5). On ne doit donc concerver uniquement les observations séparés par plus de 2500m car elles ont une autocorrélation spatiale proche de 0. L'autre possibilité est de construire un modèle prennant en complte cette autocorélation spatiale (sort du cadre de ce cours).

```
# Installation et chargement de la librairie ncf
install.packages('ncf')
```

```
## Installing package into '/home/charlotte/R/x86_64-pc-linux-gnu-library/3.4'
## (as 'lib' is unspecified)
```

```
library('ncf')
# Ajout des résidus de la régression dans le data frame
myD$res = residuals(myFinalReg)
```

```
# Affichage du corrélogramme des résidus de la régression
myCorr <- correlog(myD$X_COORD, myD$Y_COORD, myD$res,na.rm=T, increment=500,
resamp=0, latlon = F)
plot(myCorr$mean.of.class[1:30],myCorr$correlation[1:30], ylim = c(-0.1,1)
, type = "b", xlab = "distance (m)", ylab = "correlation")
```



Scéance 6: Tests non-paramétriques

Exercice 1: Étude des groupes sanguins

On veut savoir si la fréquence des groupes sanguins sont identiques dans trois états des États-Unis.

Pour ce faire, on réalise un test de χ^2 et on obtient une p-value >0.05 et donc on ne peut pas rejeter l'hypothèse nulle. On peut donc considérer que la distribution des groupes sanguins est homogène aux USA.

```
# Stockage des données dans une matrice
myMObs = matrix(c(122,117,19,244,1781, 1351, 289,
                  3301,353, 269, 60, 713), 4,3)
# On calcule les fréquences attendues
myMAtt = myMObs
for (i in 1:3){
for (j in 1:4){
myMAtt[j,i] = sum(myMObs[j,])*sum(myMObs[,i])/sum(myMObs)
}
}
# On calcule ensuite le Chi2
myChi = sum(((myMObs-myMAtt)^2)/myMAtt)
myddl = (3-1)*(4-1)
pchisq(myChi, myddl, lower.tail = F)
```

```
## [1] 0.4633774
```

```r
# Test du Chi2 en utilisant la fonction de R
chisq.test(myMObs)
```

```
##
##  Pearson's Chi-squared test
##
## data:  myMObs
## X-squared = 5.6512, df = 6, p-value = 0.4634
```

Exercice 2: Taux de dopamine de rats exposés au toluène

Le dosage de la dopamine indique le niveau de stress. On dose le taux de dopamine dans des cerveau de rats exposés ou non à du toluène et on regarde si il y a une différence significative entre les deux.

On utilise un test de Wilcoxon non-apparié car les mesures ont été effectuées sur des individus différents. On peut rejeter l'hypothèse nulle au seuil alpha de 5% et on obtient une statistique W=32 associée à une p-value de 0.01299. On peut donc dire que les deux populations ont une différence de taux de dopamine.

```r
# Réalisation d'une matrice à partir des données
tol = c(3.420,2.314,1.911,2.464,2.781,2.803)
tem = c(1.820,1.843,1.397,1.803,2.539,1.990)
# Réalisation du test de Wilcoxon
wilcox.test(c(3.420,2.314,1.911,2.464,2.781,2.803)
            ,c(1.820,1.843,1.397,1.803,2.539,1.990)
            , alternative = "greater")
```

```
##
##  Wilcoxon rank sum test
##
## data:  c(3.42, 2.314, 1.911, 2.464, 2.781, 2.803) and c(1.82, 1.843, 1.397, 1.803, 2.539, 1.99)
## W = 32, p-value = 0.01299
## alternative hypothesis: true location shift is greater than 0
```

Exercice 3: Étude d'un coupe-faim

On étudie la différence de poids entre un coupe-faim et un placébo, sachant que les mêmes personnes ont pris les deux traitements.

On utilise pour ce faire un test de Wilcoxon apparié vu que les observations ont été réalisées sur les mêmes individus. La p-value est plus grande que 0.05 et donc on ne peut pas rejeter l'hypothèse nulle. On ne peut donc pas conclure à un effet entre les deux traitements.

```r
# Création d'une matrice de données
wilcox.test(c(0.0,-1.1,-1.6,-0.3,-1.1,-0.9,-0.5,0.7,-1.2),
            c(-1.1,0.5,0.5,0.0,-0.5,1.3,-1.4,0.0,-0.8)
            , paired = T)
```

```
##
##  Wilcoxon signed rank test
##
## data:  c(0, -1.1, -1.6, -0.3, -1.1, -0.9, -0.5, 0.7, -1.2) and c(-1.1, 0.5, 0.5, 0, -0.5, 1.3, -1.4,
## V = 15, p-value = 0.4258
## alternative hypothesis: true location shift is not equal to 0
```