Plan

Plan

Test de comparaison de 2 moyennes

Question : Les poids des poulpes mâles et femelles sont-ils égaux ?

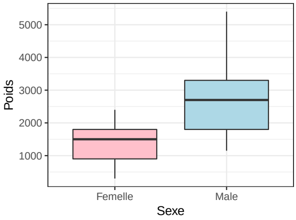
Importons et visualisons les données:

```
poulpe <- read.table("https://r-stat-sc-donnees.github.io/poulpe.csv", header=TRUE, sep=";")
summary(poulpe)
```

```
##      Poids            Sexe
##  Min.   : 300   Femelle:13
##  1st Qu.:1480   Male   :15
##  Median :1800
##  Mean   :2099
##  3rd Qu.:2750
##  Max.   :5400
```

Visualisation des données

```
library(ggplot2)
poulpe %>% ggplot() + aes(x=Sexe,y=Poids) + geom_boxplot(fill=c("pink","lightblue"))
```



Pour un graphe interactif en html:

```
library(plotly)
poulpe %>% ggplot() + aes(x=Sexe,y=Poids) + geom_boxplot(fill=c("pink","lightblue"))
ggplotly()
```

Avec les lignes de code R:

```
boxplot(Poids ~ Sexe, col=c("pink","lightblue"), data=poulpe)
```

Ou pour faire des graphes interactifs :

```
library(rAmCharts)
amBoxplot(Poids ~ Sexe, col=c("pink","lightblue"), data=poulpe)
```

Comparaison de 2 moyennes: test de la normalité

A-t-on bien la normalité des poids pour les mâles et femelles ?

```
by(poulpe$Poids, poulpe$Sexe, shapiro.test)
```

```
## poulpe$Sexe: Femelle
##
##  Shapiro-Wilk normality test
##
## data:  dd[x, ]
## W = 0.97109, p-value = 0.9069
##
## ------------------------------------------------------
## poulpe$Sexe: Male
##
##  Shapiro-Wilk normality test
##
## data:  dd[x, ]
## W = 0.93501, p-value = 0.3238
```

On accepte l'hypothèse de normalité des poids pour les femelles, et pour les mâles

Comparaison de 2 moyennes : test d'égalite des variances

Quel test utiliser ? Celui avec variances égales ou inégales ?

```
var.test(Poids ~ Sexe, conf.level=.95, data=poulpe)
```

```
##
##  F test to compare two variances
##
## data:  Poids by Sexe
## F = 0.28833, num df = 12, denom df = 14, p-value = 0.03713
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.09452959 0.92444666
## sample estimates:
## ratio of variances
##          0.2883299
```

On rejette l'hypothèse d'égalité des variances $\Longrightarrow$ on considère que les variances ne sont pas égales

Test de comparaison de 2 moyennes (suite et fin)

```
res <- t.test(Poids~Sexe, alternative="two.sided", conf.level=.95,
              var.equal=FALSE, data=poulpe)
res
```

```
##
##  Welch Two Sample t-test
##
## data:  Poids by Sexe
## t = -3.7496, df = 22.021, p-value = 0.001107
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2010.624  -578.607
## sample estimates:
## mean in group Femelle    mean in group Male
##              1405.385              2700.000
```

On considère que les poids moyennes des mâles et femelles sont différents

Les mâles sont plus lourds (2700) que les femelles (1405.4)

Plan

Problématique et données

Question : Peut-on prévoir le maximum d'ozone en fonction de données climatiques (température, nébulosité, vitesse du vent, max d'ozone de la veille) ?

Importons et visualisons les données:

```
ozone <- read.table("https://r-stat-sc-donnees.github.io/ozone.txt",header=TRUE)
library(tidyverse)
ozone.m <- ozone %>% select(1:11)
ozone.m %>% select(1:4) %>% summary()
```

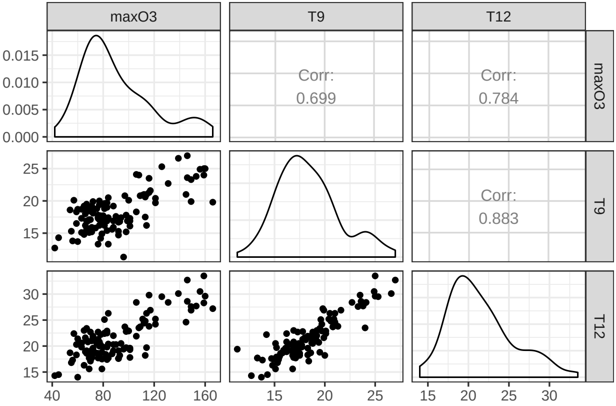
```
##      max03             T9              T12             T15
##  Min.   : 42.00   Min.   :11.30   Min.   :14.00   Min.   :14.90
##  1st Qu.: 70.75   1st Qu.:16.20   1st Qu.:18.60   1st Qu.:19.27
##  Median : 81.50   Median :17.80   Median :20.55   Median :22.05
##  Mean   : 90.30   Mean   :18.36   Mean   :21.53   Mean   :22.63
##  3rd Qu.:106.00   3rd Qu.:19.93   3rd Qu.:23.55   3rd Qu.:25.40
##  Max.   :166.00   Max.   :27.00   Max.   :33.50   Max.   :35.50
```

Avec les lignes de code R:

```
ozone <- read.table("https://r-stat-sc-donnees.github.io/ozone.txt",header=TRUE)
ozone.m <- ozone[,1:11]
summary(ozone.m[,1:4])
```

Visualisation des liaisons par paires de variables

```
library(GGally)
ozone.m %>% select(1:3) %>% ggpairs()
```



Avec les lignes de code R :

```
pairs(ozone.m[,1:3])
```

Construction du modèle complet

```r
reg.mul <- lm(maxO3~., data=ozone.m)
summary(reg.mul)
```

```
## Call:
## lm(formula = maxO3 ~ ., data = ozone.m)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.24442   13.47190   0.909   0.3656
## T9          -0.01901    1.12515  -0.017   0.9866
## T12          2.22115    1.43294   1.550   0.1243
## T15          0.55853    1.14464   0.488   0.6266
## Ne9         -2.18909    0.93824  -2.333   0.0216 *
## Ne12        -0.42102    1.36766  -0.308   0.7588
## Ne15         0.18373    1.00279   0.183   0.8550
## Vx9          0.94791    0.91228   1.039   0.3013
## Vx12         0.03120    1.05523   0.030   0.9765
## Vx15         0.41859    0.91568   0.457   0.6486
## maxO3v       0.35198    0.06289   5.597 1.88e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.36 on 101 degrees of freedom
## Multiple R-squared:  0.7638, Adjusted R-squared:  0.7405
## F-statistic: 32.67 on 10 and 101 DF,  p-value: < 2.2e-16
```

Sélection de variables

```
library(FactoMineR)
select <- RegBest(ozone.m$maxO3, ozone.m[,2:11])
select$summary ; select$best
```

```
##                                  R2       Pvalue
## Model with 1 variable   0.6150674 1.512025e-24
## Model with 2 variables  0.7012408 2.541031e-29
## Model with 3 variables  0.7519764 1.457692e-32
## Model with 4 variables  0.7622198 1.763434e-32
## Model with 5 variables  0.7630603 1.449905e-31
## Model with 6 variables  0.7635768 1.130263e-30
## Model with 7 variables  0.7637610 8.556709e-30
## Model with 8 variables  0.7638390 6.076804e-29
## Model with 9 variables  0.7638407 4.066941e-28
## Model with 10 variables 0.7638413 2.545665e-27
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.76225   11.10038   0.879    0.381
## T12          2.85308    0.48052   5.937 3.57e-08 ***
## Ne9         -3.02423    0.64342  -4.700 7.71e-06 ***
## maxO3v       0.37571    0.05801   6.477 2.85e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.23 on 108 degrees of freedom
## Multiple R-squared:  0.752,  Adjusted R-squared:  0.7451
## F-statistic: 109.1 on 3 and 108 DF,  p-value: < 2.2e-16
```

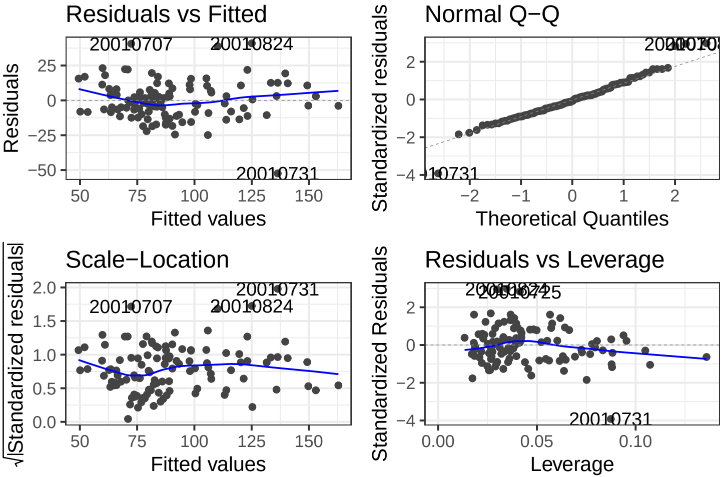
Construction du modèle final

```
reg.fin <- lm(maxO3~T12+Ne9+Vx9+maxO3v, data=ozone.m)
summary(reg.fin)
```

```
##
## Call:
## lm(formula = maxO3 ~ T12 + Ne9 + Vx9 + maxO3v, data = ozone.m)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -52.396  -8.377  -1.086   7.951  40.933
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.63131   11.00088   1.148 0.253443
## T12          2.76409    0.47450   5.825 6.07e-08 ***
## Ne9         -2.51540    0.67585  -3.722 0.000317 ***
## Vx9          1.29286    0.60218   2.147 0.034055 *
## maxO3v       0.35483    0.05789   6.130 1.50e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14 on 107 degrees of freedom
## Multiple R-squared:  0.7622, Adjusted R-squared:  0.7533
## F-statistic: 85.75 on 4 and 107 DF,  p-value: < 2.2e-16
```

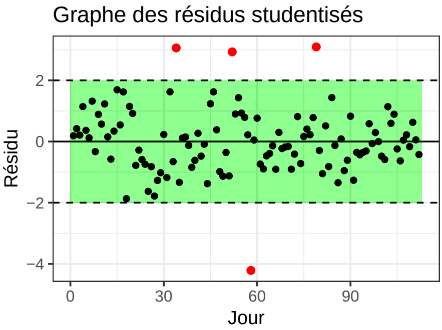
Analyser les résidus

```
library(ggfortify)
autoplot(reg.fin)
```



Analyser les résidus (suite)

```
residutib <- tibble(jour = 1:112, residu = rstudent(reg.fin))
residutib %>% ggplot() + aes(x=jour, y=residu) + geom_point() +
  labs(x="Jour", y="Résidu", title = "Graphe des résidus studentisés") +
  geom_abline(slope=0, intercept=c(-2,0,2), linetype=c(2,1,2)) +
  geom_rect(aes(xmin=0, xmax=113, ymin=-2, ymax=2), alpha=0.002,fill="green") +
  geom_point(data = residutib %>% filter(abs(residu)>2), cex=2, col="red")
```



Avec les lignes de code R :

```
plot(residu,pch=15,cex=.5,ylab="Résidus",main="Graphe des résidus studentisés",ylim=c(-3,3))
abline(h=c(-2,0,2),lty=c(2,1,2))
```

Prévoir une nouvelle valeur

Et comment prédire le maximum d'ozone pour de nouvelles valeurs ?

```
xnew <- matrix(c(19,8,2.05,70),nrow=1)
colnames(xnew) <- c("T12","Ne9","Vx9","maxO3v")
xnew <- as.data.frame(xnew)
predict(reg.fin,xnew,interval="pred")
```

```
##        fit      lwr      upr
## 1 72.51437 43.80638 101.2224
```

Plan

5 Statistique inférentielle

Problématique et données

Question : Y a-t-il un effet de la pluie et du vent sur le maximum d'ozone ? Y a-t-il un effet de l'interaction de ces deux facteurs ?

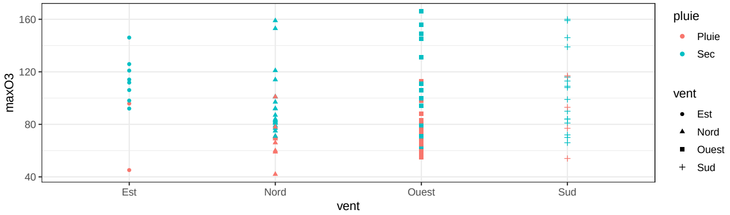
Importation des données:

```
ozone <- read.table("https://r-stat-sc-donnees.github.io/ozone.txt",header=TRUE)
summary(ozone[,c("maxO3","vent","pluie")])
```

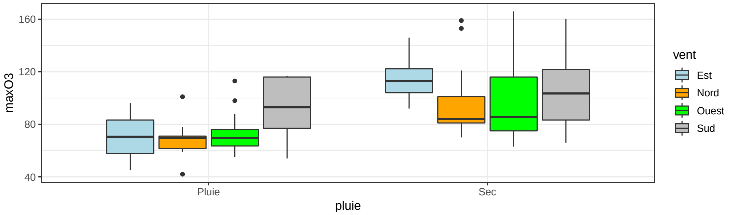
```
##      maxO3           vent      pluie
##  Min.   : 42.00   Est  :10   Pluie:43
##  1st Qu.: 70.75   Nord :31   Sec  :69
##  Median : 81.50   Ouest:50
##  Mean   : 90.30   Sud  :21
##  3rd Qu.:106.00
##  Max.   :166.00
```

Visualisation des données avec ggplot2

```
library(ggplot2)
ozone %>% ggplot() + aes(y=maxO3, x=vent) + geom_point(aes(col=pluie, shape=vent))
```

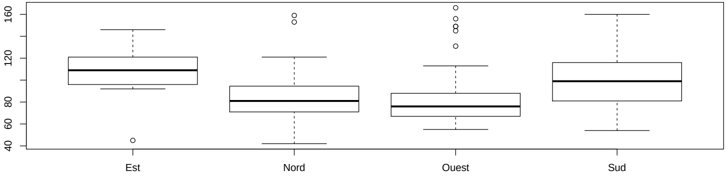



```
ozone %>% ggplot() + aes(pluie, maxO3) + geom_boxplot(aes(fill=vent)) +
  scale_fill_manual(values=c("lightblue","orange","green","grey"))
```

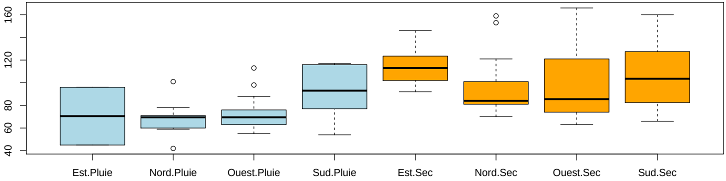


Visualisation des données en R

```
boxplot(maxO3~vent, data = ozone)
```

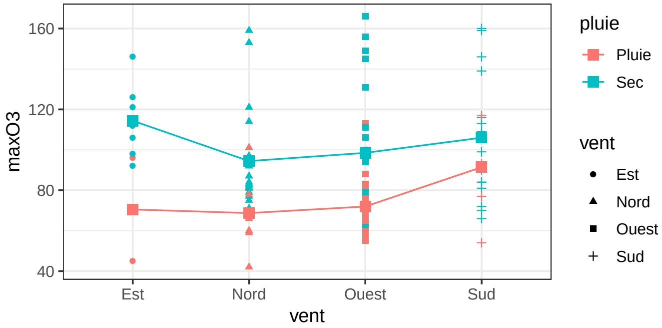



```
boxplot(maxO3~vent*pluie, data = ozone, col=c(rep("Lightblue",4),rep("orange",4)))
```



Visualisation de l'interaction

```
ozone %>% ggplot() + aes(x = vent, y = maxO3, group = pluie) +
  geom_point(aes(color = pluie, shape=vent)) +
  stat_summary(fun.y = mean, geom = "point", size=3, shape=15,aes(color = pluie)) +
  stat_summary(fun.y = mean, geom = "line", aes(color = pluie))
```

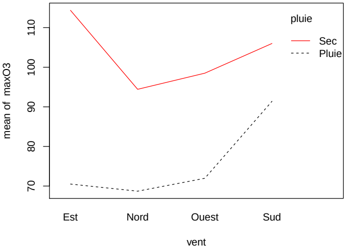


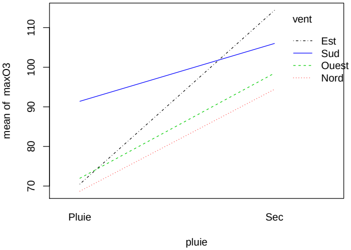
Visualiser l'autre graphe d'interaction (une ligne brisée par direction du vent) et conserver le graphe le plus explicite

```
ozone %>%  ggplot() + aes(x = pluie, y = maxO3, group = vent, color = vent, shape=pluie) +
  geom_point(alpha=0.5) + stat_summary(fun.y = mean, geom = "point", size=3, shape=15) +
  stat_summary(fun.y = mean, geom = "line")
```

Graphe : visualisation de l'interaction

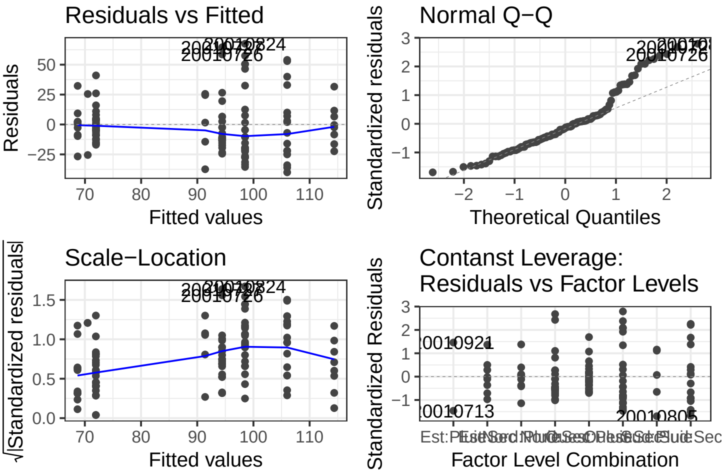
```
with(ozone,interaction.plot(vent,pluie,maxO3,col=1:nlevels(pluie)))
with(ozone,interaction.plot(pluie,vent,maxO3,col=1:nlevels(vent)))
```





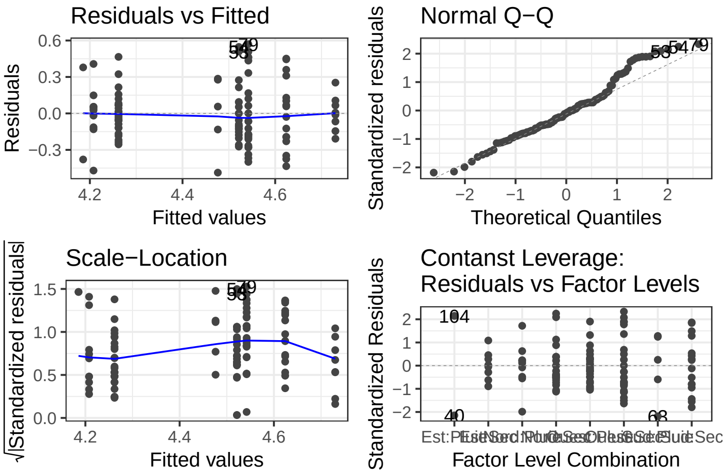
Validité du modèle

```
library(ggfortify)
mod.interaction <- lm(maxO3 ~ vent + pluie + vent:pluie, data=ozone)
autoplot(mod.interaction)
```



Validité du modèle

```
library(ggfortify)
ozone %>% mutate(log_max03 = log(max03)) -> ozone
mod.interaction <- lm(log_max03 ~ vent + pluie + vent:pluie, data=ozone)
autoplot(mod.interaction)
```



Test du modèle complet

```
mod.interaction <- lm(log_maxO3 ~ vent + pluie + vent:pluie, data=ozone)
mod.0 <- lm(log_maxO3 ~ 1, data=ozone)
anova(mod.0, mod.interaction)
```

```
## Analysis of Variance Table
##
## Model 1: log_maxO3 ~ 1
## Model 2: log_maxO3 ~ vent + pluie + vent:pluie
##   Res.Df    RSS Df Sum of Sq     F    Pr(>F)
## 1    111 9.5368
## 2    104 6.4740  7    3.0629 7.029 7.355e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

On rejette l'hypothèse qu'il n'existe aucun effet car la probabilité critique (0) est inférieure à 5%

Construction du modèle avec interaction

```
anova(mod.interaction)
```

```
## Analysis of Variance Table
##
## Response: max03
##             Df Sum Sq Mean Sq F value    Pr(>F)
## vent         3   7586  2528.7  4.1454   0.00809 **
## pluie        1  16159 16159.4 26.4910 1.257e-06 ***
## vent:pluie   3   1006   335.5  0.5500   0.64929
## Residuals  104  63440   610.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Anova(mod.interaction)
```

```
## Anova Table (Type II tests)
##
## Response: max03
##            Sum Sq  Df F value    Pr(>F)
## vent         3791   3  2.0718    0.1085
## pluie       16159   1 26.4910 1.257e-06 ***
## vent:pluie   1006   3  0.5500    0.6493
## Residuals   63440 104
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

On accepte l'hypothèse qu'il n'y a pas d'interaction car la probabilité critique (0.399) est supérieure à 5%

Choix d'un sous-modèle

```
modele_12 <- lm(log_max03 ~ vent + pluie, data = ozone)
anova(modele_12)
```

```
## Analysis of Variance Table
##
## Response: log_max03
##            Df Sum Sq Mean Sq F value    Pr(>F)
## vent        3 0.8588 0.28626  4.5994  0.004555 **
## pluie       1 2.0187 2.01866 32.4346 1.094e-07 ***
## Residuals 107 6.6594 0.06224
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Anova(modele_12)
```

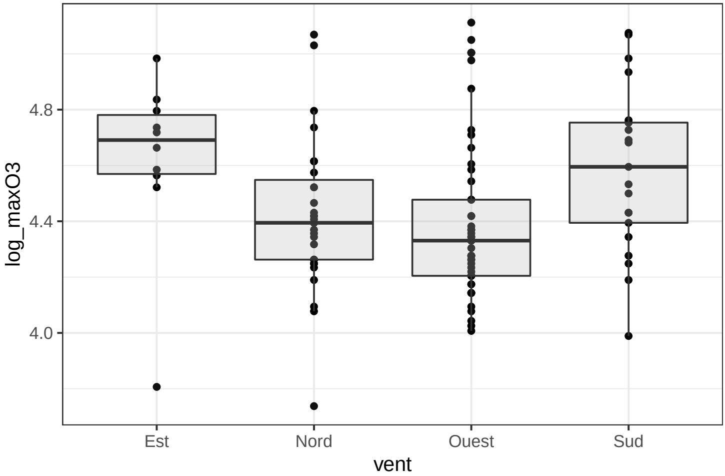
```
## Anova Table (Type II tests)
##
## Response: log_max03
##           Sum Sq  Df F value    Pr(>F)
## vent      0.3982   3  2.1329    0.1004
## pluie     2.0187   1 32.4346 1.094e-07 ***
## Residuals 6.6594 107
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Quelle définition pour l'effet du vent ?

Qu'est ce que l'effet du vent ??

Visualisation des différences de vent après ajustement à la pluie

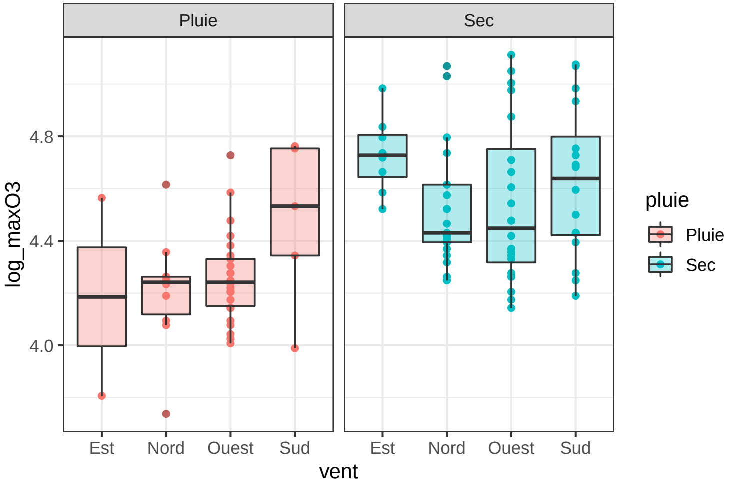
```
ozone %>%   ggplot() +
  geom_point( mapping = aes(x=vent, y=log_maxO3))+
  geom_boxplot( mapping = aes(x=vent, y=log_maxO3), alpha=0.3, fill='gray')
```



Qu'est ce que l'effet du vent ??

Visualisation des différences de vent après ajustement à la pluie

```
ozone %>%  ggplot() +  facet_wrap(~pluie)+
  geom_point( mapping = aes(x=vent, y=log_maxO3, col = pluie)) +
  geom_boxplot( mapping = aes(x=vent, y=log_maxO3, fill = pluie), alpha=0.3)
```



Estimation des coefficients

Attention à l'interprétation

Dans le modèle complet

```
summary(mod.interaction)
```

```
##
## Call:
## lm(formula = maxO3 ~ vent + pluie + vent:pluie, data = ozone)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -40.000 -15.971  -3.462   7.635  67.500
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)          70.500     17.464   4.037 0.000104 ***
## ventNord             -1.800     19.131  -0.094 0.925221
## ventOuest             1.462     18.123   0.081 0.935881
## ventSud              20.900     20.664   1.011 0.314161
## pluieSec             43.875     19.526   2.247 0.026749 *
## ventNord:pluieSec   -18.146     21.709  -0.836 0.405138
## ventOuest:pluieSec  -17.337     20.739  -0.836 0.405117
## ventSud:pluieSec    -29.275     23.267  -1.258 0.211138
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.7 on 104 degrees of freedom
## Multiple R-squared:  0.2807, Adjusted R-squared:  0.2322
```

Comparaison de moyennes ajustées

```
library('emmeans')
emmeans(modele_12,pairwise~pluie,adjust="hochberg")
```

```
## $emmeans
##  pluie    emmean       SE  df lower.CL  upper.CL
##  Pluie  77.33679 4.337116 107 68.73896  85.93462
##  Sec   102.93347 3.166613 107 96.65603 109.21092
##
## Results are averaged over the levels of: vent
## Confidence level used: 0.95
##
## $contrasts
##  contrast     estimate       SE  df t.ratio p.value
##  Pluie - Sec -25.59668 4.941713 107   -5.18  <.0001
##
## Results are averaged over the levels of: vent
```

```
emmeans(modele_12,pairwise~vent,adjust="hochberg")
```

```
## $emmeans
##  vent    emmean       SE  df lower.CL  upper.CL
##  Est   97.92099 7.901135 107 82.25792 113.58407
##  Nord  81.58769 4.494192 107 72.67847  90.49690
##  Ouest 85.21193 3.472144 107 78.32881  92.09505
##  Sud   95.81992 5.509637 107 84.89770 106.74213
##
## Results are averaged over the levels of: pluie
## Confidence level used: 0.95
##
```

Plan

Plan

Problématique et données

Importation des données:

```
decath <- read.table("https://r-stat-sc-donnees.github.io/decathlon.csv",
                     sep=";",dec=".", header=TRUE, row.names=1, check.names=FALSE)
```

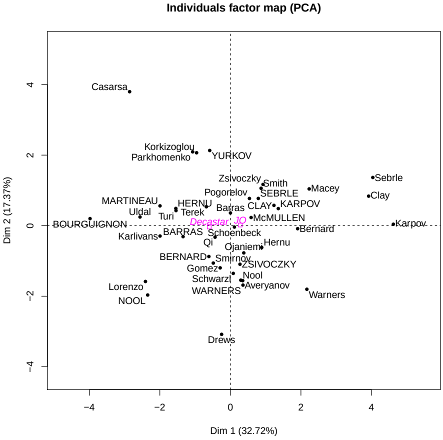
ACP par les lignes de commande :

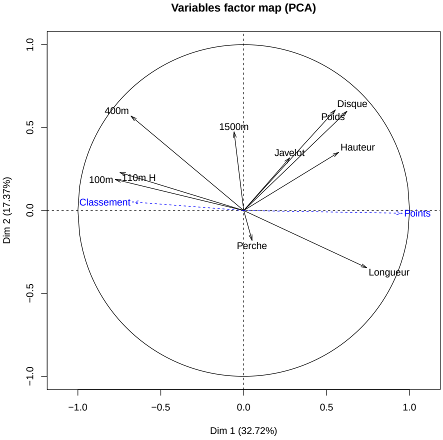
```
library(FactoMineR)
res.pca <- PCA(decath, quanti.sup=11:12, quali.sup=13)
```

ACP par un menu déroulant et pour des graphes interactifs :

```
library(Factoshiny)
res <- PCAshiny(decath)
```

Graphes des individus et des variables

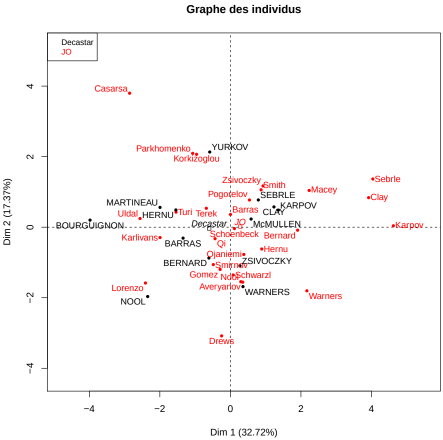


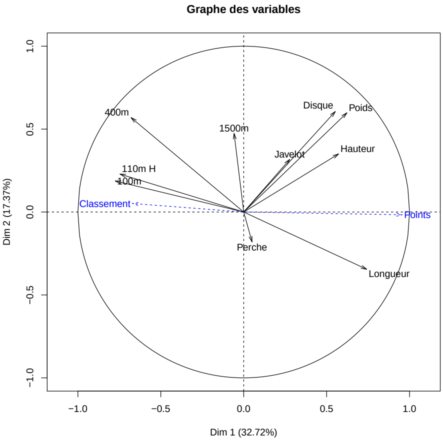


Graphes des individus et des variables

Possibilité de colorier les individus en fonction d'une variable qualitative :

```
plot(res.pca,habillage=13, cex=0.9, title="Graphe des individus")
plot(res.pca,choix="var", title="Graphe des variables")
```





Résultats

```r
summary(res.pca, ncp=2)
```

```
## Call:
## PCA(X = decath, quanti.sup = 11:12, quali.sup = 13, graph = FALSE)
##
##
## Eigenvalues
##                        Dim.1   Dim.2   Dim.3   Dim.4   Dim.5   Dim.6
## Variance               3.272   1.737   1.405   1.057   0.685   0.599
## % of var.             32.719  17.371  14.049  10.569   6.848   5.993
## Cumulative % of var.  32.719  50.090  64.140  74.708  81.556  87.548
##                        Dim.7   Dim.8   Dim.9  Dim.10
## Variance               0.451   0.397   0.215   0.182
## % of var.              4.512   3.969   2.148   1.822
## Cumulative % of var.  92.061  96.030  98.178 100.000
##
## Individuals (the 10 first)
##                Dist    Dim.1    ctr   cos2    Dim.2    ctr   cos2
## Sebrle     |  4.843 |  4.038 12.158  0.695 |  1.366  2.619  0.080 |
## Clay       |  4.647 |  3.919 11.451  0.711 |  0.837  0.984  0.032 |
## Karpov     |  5.006 |  4.620 15.911  0.852 |  0.040  0.002  0.000 |
## Macey      |  3.434 |  2.233  3.719  0.423 |  1.042  1.524  0.092 |
## Warners    |  2.979 |  2.168  3.505  0.530 | -1.803  4.565  0.366 |
## Zsivoczky  |  2.566 |  0.925  0.638  0.130 |  1.169  1.918  0.207 |
## Hernu      |  1.824 |  0.889  0.589  0.238 | -0.618  0.537  0.115 |
## Nool       |  3.098 |  0.295  0.065  0.009 | -1.546  3.354  0.249 |
## Bernard    |  2.827 |  1.906  2.709  0.455 | -0.086  0.010  0.001 |
## Schwarzl   |  1.971 |  0.081  0.005  0.002 | -1.353  2.572  0.472 |
```

Résultats (suite)

```
summary(res.pca, ncp=2)
```

```
## Variables
##                Dim.1    ctr   cos2    Dim.2    ctr   cos2  
## 100m         | -0.775 18.344  0.600 |  0.187  2.016  0.035 |
## Longueur     |  0.742 16.822  0.550 | -0.345  6.869  0.119 |
## Poids        |  0.623 11.844  0.388 |  0.598 20.607  0.358 |
## Hauteur      |  0.572  9.998  0.327 |  0.350  7.064  0.123 |
## 400m         | -0.680 14.116  0.462 |  0.569 18.666  0.324 |
## 110m H       | -0.746 17.020  0.557 |  0.229  3.013  0.052 |
## Disque       |  0.552  9.328  0.305 |  0.606 21.162  0.368 |
## Perche       |  0.050  0.077  0.003 | -0.180  1.873  0.033 |
## Javelot      |  0.277  2.347  0.077 |  0.317  5.784  0.100 |
## 1500m        | -0.058  0.103  0.003 |  0.474 12.946  0.225 |
##
## Supplementary continuous variables
##                Dim.1   cos2    Dim.2   cos2  
## Classement   | -0.671  0.450 |  0.051  0.003 |
## Points       |  0.956  0.914 | -0.017  0.000 |
##
## Supplementary categories
##                 Dist    Dim.1   cos2 v.test    Dim.2   cos2 v.test  
## Decastar     |  0.946 | -0.600  0.403 -1.430 | -0.038  0.002 -0.123 |
## JO           |  0.439 |  0.279  0.403  1.430 |  0.017  0.002  0.123 |
```

Description des dimensions

```
dimdesc(res.pca, axes=1:2)
```

```
## $Dim.1
## $Dim.1$quanti
##            correlation      p.value
## Points       0.9561543 2.099191e-22
## Longueur     0.7418997 2.849886e-08
## Poids        0.6225026 1.388321e-05
## Hauteur      0.5719453 9.362285e-05
## Disque       0.5524665 1.802220e-04
## Classement  -0.6705104 1.616348e-06
## 400m        -0.6796099 1.028175e-06
## 110m H      -0.7462453 2.136962e-08
## 100m        -0.7747198 2.778467e-09
##
##
## $Dim.2
## $Dim.2$quanti
##          correlation      p.value
## Disque     0.6063134 2.650745e-05
## Poids      0.5983033 3.603567e-05
## 400m       0.5694378 1.020941e-04
## 1500m      0.4742238 1.734405e-03
## Hauteur    0.3502936 2.475025e-02
## Javelot    0.3169891 4.344974e-02
## Longueur  -0.3454213 2.696969e-02
```

Plan

Description des données

Le jeu de données croise 700 relevés décrits par les pollens de 31 espèces d'arbres. Des variables climatiques ont été mesurées : température moyenne du mois le plus froid (mtco, mean temperature of the coldest month); température moyenne du mois le plus chaud (mtwa, mean temperature of the warmest month); the growing degree-days (gdd5, the sum of daily temperatures) above 5°C; the ratio of actual evapotranspiration to potential evapotranspiration (e_pe); précipitation annuelle (pann); température moyenne annuelle (tann).

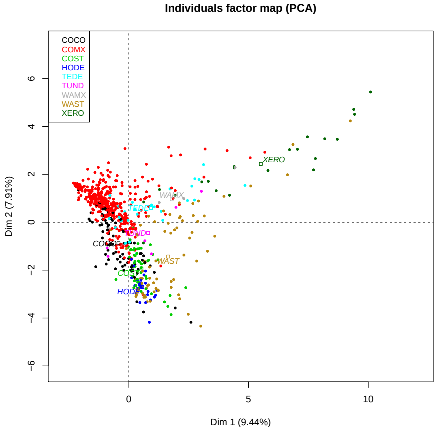
Les 700 relevés proviennent de 9 biomes différents : COCO (cool conifer forest), COMX (cool mixed forest), COST (cool steppes), HODE (hot desert), TEDE (temperate deciduous forest), TUND (tundra), WAMX (warm mixed broad-leaved forest), WAST (warm steppes), XERO (xerophytic scrubs)

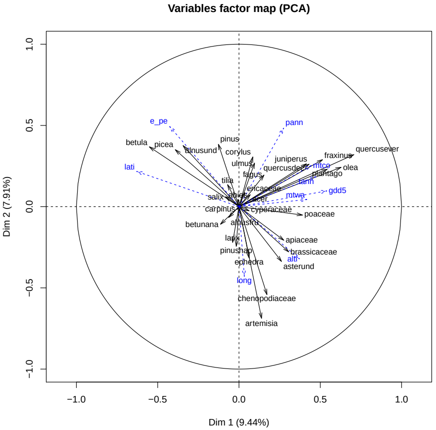
- Visualiser les 700 échantillons en fonction des concentrations de pollens (par ACP)
- Prédire la température annuelle (tann) en fonction des concentrations de pollens
- Etudier la relation entre biome et température annuelle

```
ss700 <- read.table("https://husson.github.io/img/ss700.csv", header=TRUE,
                    sep=";", row.names=1)
```

ACP

```
library(FactoMineR)
res.pca <- PCA(ss700,quanti.sup=32:40, quali.sup=41,graph=FALSE)
plot(res.pca,hab=41,label="quali",cex=0.8)
plot(res.pca,choix="var",cex=0.8)
```





ACP : description des dimensions

```
dimdesc(res.pca)
```

```
## $Dim.1
## $Dim.1$quanti
##                correlation       p.value
## quercusever     0.70653170 6.559582e-107
## olea            0.62830800  3.721671e-78
## plantago        0.54289440  6.588354e-55
## gdd5            0.54004875  3.033049e-54
## fraxinus        0.51411998  1.744448e-48
## tann            0.47292243  2.703518e-40
## mtco            0.44433104  3.120251e-35
## juniperus       0.43142594  4.229579e-33
## mtwa            0.41702844  7.922163e-31
## quercusdec      0.41445975  1.962711e-30
## poaceae         0.39041209  6.602788e-27
## alti            0.37156885  2.437317e-24
## brassicaceae    0.30576165  1.293179e-16
## apiaceae        0.27572212  1.116715e-13
## pann            0.27475690  1.369452e-13
## asterund        0.26135995  2.140036e-12
## chenopodiaceae  0.17207707  4.676703e-06
## fagus           0.15400914  4.279399e-05
## artemisia       0.13896336  2.260575e-04
## ulmus           0.09591008  1.112136e-02
## corylus         0.08510370  2.434160e-02
## salix          -0.08283928  2.841042e-02
## betunana       -0.11398173  2.526650e-03
## pinus          -0.12664636  7.842611e-04
```

Régression multiple

```
library(FactoMineR)
mod <- RegBest(ss700[,"tann"],ss700[,1:31])
mod$summary
```

```
##                                R2        Pvalue
## Model with 1 variable   0.1337244  1.428198e-23
## Model with 2 variables  0.2407250  2.084893e-42
## Model with 3 variables  0.3114470  4.720331e-56
## Model with 4 variables  0.3813431  4.512025e-71
## Model with 5 variables  0.4332160  3.824672e-83
## Model with 6 variables  0.4790691  1.023203e-94
## Model with 7 variables  0.5172760 4.771272e-105
## Model with 8 variables  0.5414604 1.133191e-111
## Model with 9 variables  0.5606172 5.385370e-117
## Model with 10 variables 0.5799961 1.120032e-122
## Model with 11 variables 0.5973145 6.550118e-128
## Model with 12 variables 0.6061371 3.445850e-130
## Model with 13 variables 0.6144407 2.386004e-132
## Model with 14 variables 0.6195355 2.464602e-133
## Model with 15 variables 0.6241397 3.657097e-134
## Model with 16 variables 0.6291686 3.443937e-135
## Model with 17 variables 0.6330612 8.542180e-136
## Model with 18 variables 0.6368946 2.120644e-136
## Model with 19 variables 0.6389037 2.730759e-136
## Model with 20 variables 0.6402782 6.156617e-136
## Model with 21 variables 0.6410069 2.466417e-135
## Model with 22 variables 0.6414979 1.202941e-134
## Model with 23 variables 0.6420853 5.243144e-134
## Model with 24 variables 0.6424831 2.665793e-133
```

Régression multiple

```
mod$best
```

```
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -4.540766   0.419324 -10.829  < 2e-16 ***
## alnusfru      -0.108785   0.038608  -2.818 0.004978 **
## artemisia      0.073286   0.011420   6.417 2.60e-10 ***
## asterund       0.095178   0.029829   3.191 0.001484 **
## betunana      -0.115918   0.041513  -2.792 0.005380 **
## carpinus       0.471085   0.047250   9.970  < 2e-16 ***
## chenopodiaceae 0.116026   0.011943   9.715  < 2e-16 ***
## corylus        0.570243   0.072462   7.870 1.40e-14 ***
## fagus          0.304931   0.103779   2.938 0.003412 **
## juniperus      0.210060   0.078342   2.681 0.007511 **
## larix         -0.172249   0.036628  -4.703 3.11e-06 ***
## olea           0.510621   0.054555   9.360  < 2e-16 ***
## pinushap      -0.084922   0.015777  -5.383 1.01e-07 ***
## pinus          0.116688   0.008417  13.863  < 2e-16 ***
## plantago       0.688050   0.125109   5.500 5.39e-08 ***
## poaceae        0.078168   0.014778   5.290 1.65e-07 ***
## quercusdec     0.208004   0.028929   7.190 1.71e-12 ***
## tilia          0.178202   0.059295   3.005 0.002750 **
## ulmus          0.771381   0.220401   3.500 0.000496 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.203 on 681 degrees of freedom
## Multiple R-squared:  0.6369, Adjusted R-squared:  0.6273
## F-statistic: 66.36 on 18 and 681 DF,  p-value: < 2.2e-16
```

Analyse de variance

```
library(FactoMineR)
mod <- AovSum(tann ~ biome,data=ss700)
mod
```

```
## Ftest
##               SS  df      MS F value    Pr(>F)
## biome     12141   8 1517.61  49.976 < 2.2e-16 ***
## Residuals 20984 691   30.37
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Ttest
##               Estimate Std. Error  t value Pr(>|t|)
## (Intercept)    3.78679    0.46635   8.1200  < 2e-16 ***
## biome - COCO -9.29837    0.70151 -13.2547  < 2e-16 ***
## biome - COMX -2.31376    0.52629  -4.3964    1e-05 ***
## biome - COST -5.49444    0.81957  -6.7041  < 2e-16 ***
## biome - HODE -0.11811    0.98963  -0.1193  0.90503
## biome - TEDE  5.69047    1.04511   5.4449  < 2e-16 ***
## biome - TUND -7.09429    2.03812  -3.4808  0.00053 ***
## biome - WAMX  6.60221    2.47430   2.6683  0.00780 **
## biome - WAST  3.74400    0.72681   5.1513  < 2e-16 ***
## biome - XERO  8.28231    1.15853   7.1490  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Une carte à tester pour finir

```
library(leaflet)
pal <- colorNumeric(palette=c(low="blue",high="red"),domain=ss700["tann"])
m <- leaflet() %>% addTiles() %>%
  addCircles(ss700[,"long"],ss700[,"lati"], color=pal(ss700[,"tann"]),
             fillOpacity=1, opacity=1)
m
```

Plan

Les anti sèches de RStudio

R base

RStudio

RMarkdown

Importation

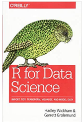
Manipulation

Visualisation

Des livres

- A Language and Environment for Statistical Computing (R Core Team, 2017), https://www.R-project.org/

- R for Data science (Wickham & Grolemund, 2016), https://r4ds.had.co.nz/



- R pour la statistique et la science des données (Cornillon et al., 2018), https://r-stat-sc-donnees.github.io/

–>