Executive Master S & BG: Modèle linéaire généralisé et Choix de modèles

Météo à Bâle

Souade FAJRI

14 Juillet 2023

Contents

1. Telechargement des données

1.1 Importation des données

```
d = read.table("meteo.train.csv",header=T,sep=",")
#summary(d)
library(ggplot2)
library(corrplot)
library(caret)
library(pROC)
library(FactoMineR)
library(factoextra)
```

1.2 Intitulés des variables

Nous allons dans cette section renommer les noms des variables pour faciliter la rédaction de ce projet et mieux s'approprier le jeu de données.

```
##  [1] "X"                    "Year"                 "Month"
##  [4] "Day"                  "Hour"                 "Minute"
##  [7] "Temp.mean"            "Hum.mean"             "Press.mean"
## [10] "Precipitation"        "Snowfall"             "Total.cloud.mean"
## [13] "High.cloud.mean"      "Med.cloud.mean"       "Low.cloud.mean"
## [16] "Sunshine"             "Radiation"            "Wind.speed.10m.mean"
## [19] "Wind.direc.10.m"      "Wind.speed.80m.mean"  "Wind.direc.80.m"
## [22] "Wind.speed.900m.mean" "Wind.direc.900.m"     "Gust.mean"
## [25] "Temp.max"             "Temp.min"             "Hum.max"
## [28] "Hum.min"              "Press.max"            "Press.min"
## [31] "Total.cloud.max"      "Total.cloud.min"      "High.cloud.max"
## [34] "High.cloud.min"       "Med.cloud.max"        "Med.cloud.min"
## [37] "Low.cloud.max"        "Low.cloud.min"        "Wind.speed.10m.max"
## [40] "Wind.speed.10m.min"   "Wind.speed.80.max"    "Wind.speed.80m.min"
## [43] "Wind.speed.900.max"   "Wind.speed.900m.min"  "Gust.max"
## [46] "Gust.min"             "pluie.demain"
```

2. Optimisation de la dimension

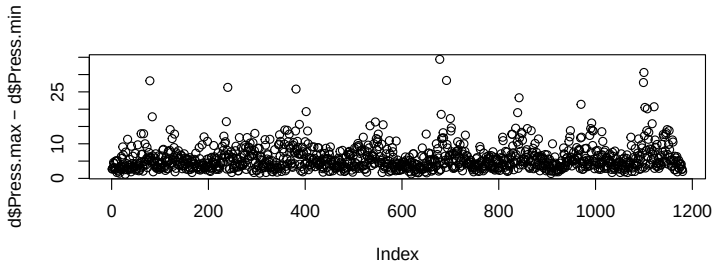
Dans cette partie, nous analysons les variables qui semblent apporter peu d'information et décidons si nous les conservons pour la suite de l'exercice.

L'objectif est de nettoyer la base de données et de réduire le nombre de variables.

2.1 Variables ajoutées

Press

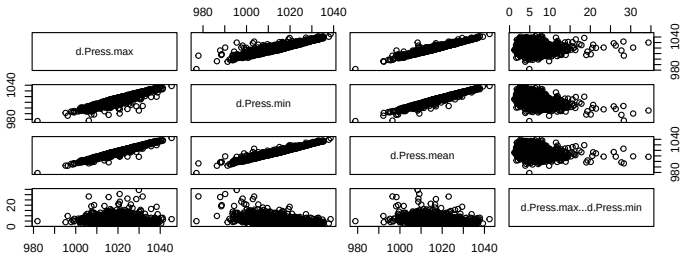
```
plot(d$Press.max - d$Press.min)
```




```
temp = data.frame(d$Press.max,d$Press.min,d$Press.mean,d$Press.max - d$Press.min )
cor(temp)
```

```
##                            d.Press.max d.Press.min d.Press.mean
## d.Press.max                 1.000000000   0.9047461    0.9722008
## d.Press.min                 0.904746052   1.0000000    0.9735478
## d.Press.mean                0.972200761   0.9735478    1.0000000
## d.Press.max...d.Press.min -0.009935123  -0.4349191   -0.2302201
##                           d.Press.max...d.Press.min
## d.Press.max                           -0.009935123
## d.Press.min                           -0.434919125
## d.Press.mean                          -0.230220060
## d.Press.max...d.Press.min              1.000000000
```

```
pairs(temp)
```



```
d["Press.var"] = d$Press.max - d$Press.min
```

Pour évaluer la corrélation entre les variables Pressure.max, Pressure.min et Pressure.mean, nous constatons qu'elles sont fortement corrélées. Par conséquent, nous prenons la décision de conserver uniquement la variable de pression moyenne, Pressure.mean, et d'ajouter une nouvelle variable, (Pressure.max-Pressure.min), afin de tenter de saisir les jours présentant des variations importantes de pression.

Notre hypothèse est qu'un changement significatif de pression peut être associé à des situations orageuses et donc à des périodes de pluie.

En ajoutant cette nouvelle variable, nous espérons capturer cette relation potentielle entre les variations de pression et les événements pluvieux.

Gust.var

```
d["Gust.var"] = d$Gust.max -d$Gust.min
```

Nous prenons également la décision d'ajouter la variation des rafales de vent (Gust.var) afin de capturer les événements orageux, événements qui sont souvent associés à des périodes pluvieuses.

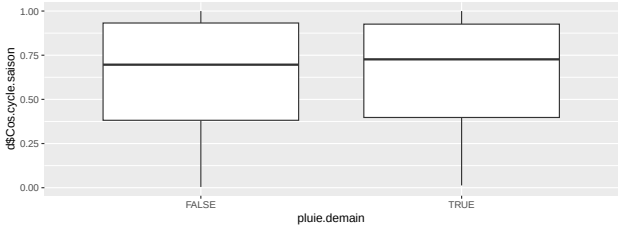
En considérant la variation des rafales de vent, nous cherchons à saisir les changements brusques et intenses du vent, souvent associés aux phénomènes orageux.

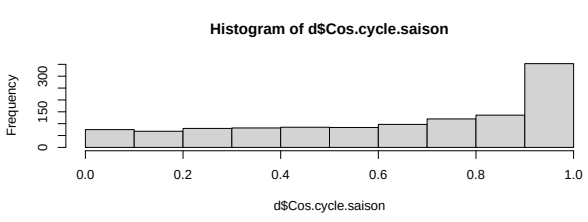
En incluant cette variable dans notre modèle, nous espérons améliorer notre capacité à prédire la présence de pluie lors de ces événements.

Cos.cycle.saison

```
  tmp = do.call(paste, list(d$Month, d$Day, d$Year))
  tmp = as.Date(tmp, format=c("%m %d %Y"))
  d$Cos.cycle.saison = abs(cos(as.numeric(format(tmp, "%j"))/365*4*pi))
```

```
ggplot(d, aes(x=pluie.demain, y=d$Cos.cycle.saison )) + geom_boxplot()
hist(d$Cos.cycle.saison)
```





Nous prenons également la décision d'ajouter une nouvelle variable relative au cycle des saisons.

En effet, l'intuition suggère que le cycle des saisons peut avoir une influence sur la pluie. Les saisons sont des périodes de l'année caractérisées par des changements climatiques spécifiques qui se répètent de manière cyclique. Ces changements climatiques peuvent inclure des variations de température, d'humidité, de pression atmosphérique, de vents, et bien sûr, de la quantité de précipitations, y compris la pluie.

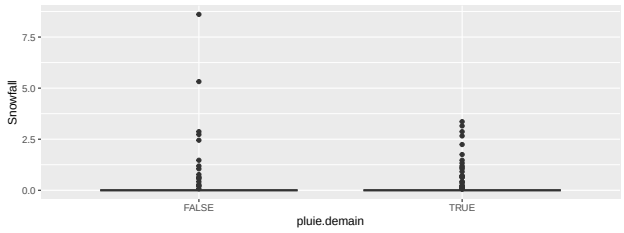
Pour cela, nous allons créer une covariable reprenant le cosinus du jour avec une périodicité de 4 mois. Cette nouvelle variable nous permettra de tenir compte des variations saisonnières potentielles dans notre modèle, ce qui pourrait améliorer notre compréhension et notre capacité à prédire les événements de pluie.

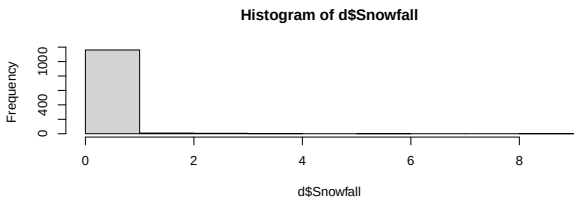
2.2 Variables supprimées

Grâce aux modalités des variables et à leur distribution par rapport aux valeurs de la variable à prédire, nous allons déterminer si ces variables sont utiles ou non dans le cadre de la réalisation de ce projet.

Snowfall

```
ggplot(d, aes(x=pluie.demain, y=Snowfall)) + geom_boxplot()
hist(d$Snowfall)
```

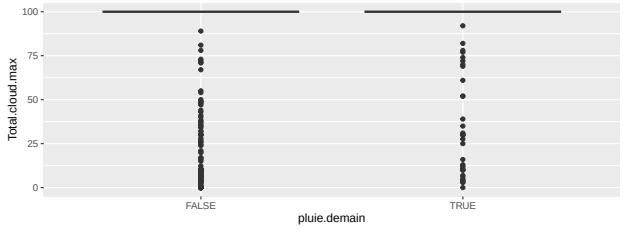


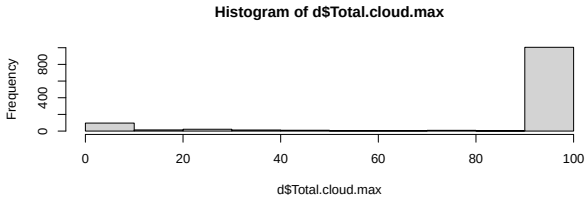


Distribution identique sur les modalités (FALSE & TRUE) de la variable pluie.demain. Au constate via l'histogramme un important d'observations égales à 0.

Total.cld.max

```
ggplot(d, aes(x=pluie.demain, y=Total.cloud.max)) + geom_boxplot()
hist(d$Total.cloud.max)
```

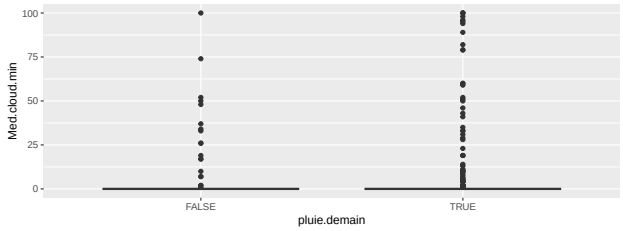


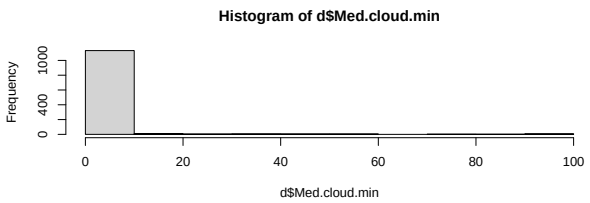


Distribution identique sur les modalités (FALSE & TRUE) de la variable pluie.demain. Au constate via l'histogramme un important d'observations superieurs à 90.

Med.cld.min

```
ggplot(d, aes(x=pluie.demain, y=Med.cloud.min )) + geom_boxplot()
hist(d$Med.cloud.min)
```

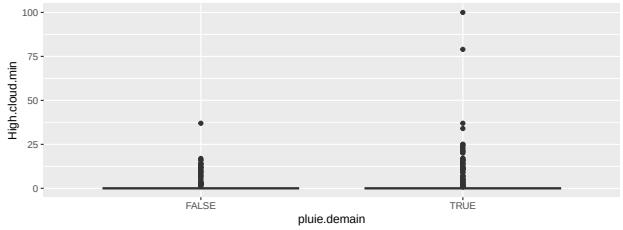


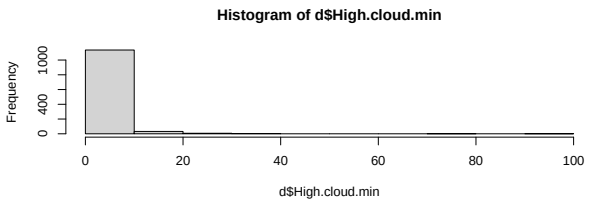


Distribution identique sur les modalités (FALSE & TRUE) de la variable pluie.demain. Au constate via l'histogramme un important d'observations égale à 0.

High.cloud.min

```
ggplot(d, aes(x=pluie.demain, y=High.cloud.min)) + geom_boxplot()
hist(d$High.cloud.min)
```

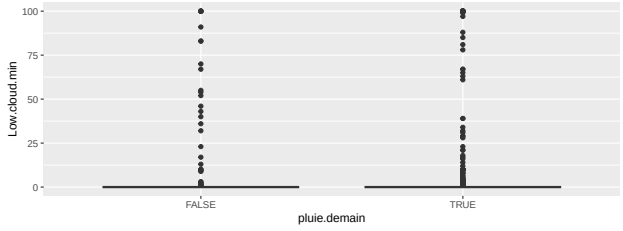


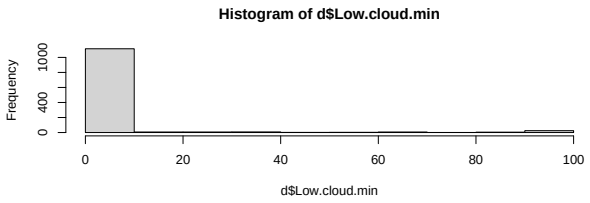


Distribution identique sur les modalités (FALSE & TRUE) de la variable pluie.demain. Au constate via l'histogramme un important d'observations égales à 0.

Low.cloud.min

```
ggplot(d, aes(x=pluie.demain, y=Low.cloud.min )) + geom_boxplot()
hist(d$Low.cloud.min)
```





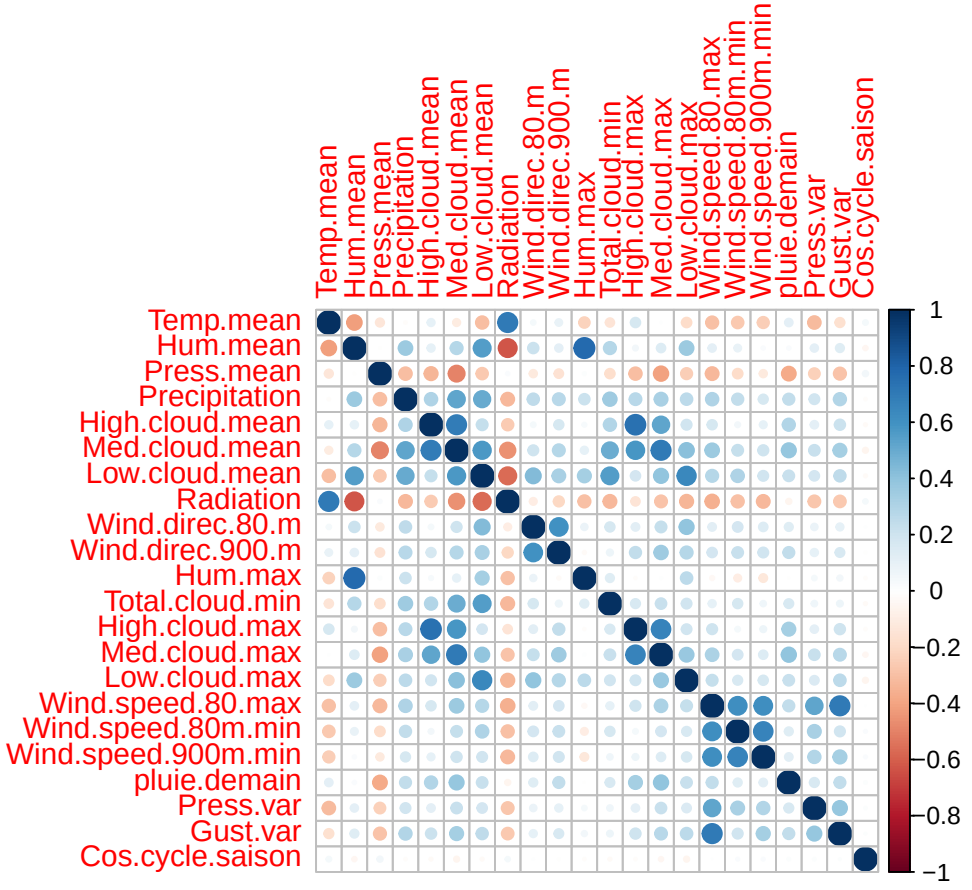
Distribution identique sur les modalités (FALSE & TRUE) de la variable pluie.demain. Au constate via l'histogramme un important d'observations égales à 0.

Corrélation > à 0.80

Après avoir éliminé les variables présentant une forte corrélation (dans le but d'avoir uniquement des variables distinctes et d'éviter la redondance dans le modèle), nous allons supprimer les variables qui ne semblent pas apporter d'information significative au modèle.

```
d = d[,-which(names(d) %in% c("X","Year","Month","Day","Hour","Minute",
  "High.cloud.min","Snowfall","Total.cloud.max","Med.cloud.min","Low.cloud.min","Press.max","Press.min"
          ))]
```

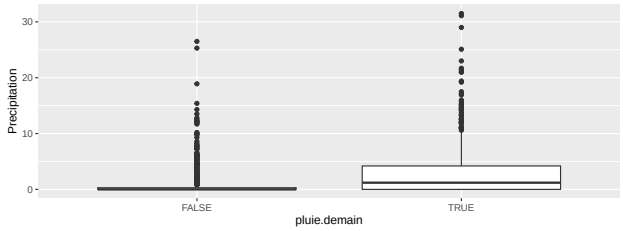
```
dfcor = abs(cor(d))
hc = findCorrelation(dfcor, cutoff=0.80) # putt any value as a "cutoff"
hc = sort(hc)
reduced_Data = d[,-c(hc)]
#print (reduced_Data)
d = reduced_Data
corrplot(cor(d, use="complete"))
```

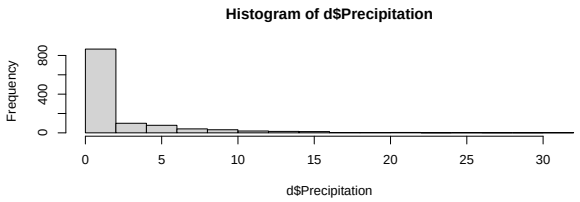


2.3 Variables retenues

Precipitation

```
ggplot(d, aes(x=pluie.demain, y=Precipitation)) + geom_boxplot()
hist(d$Precipitation)
```



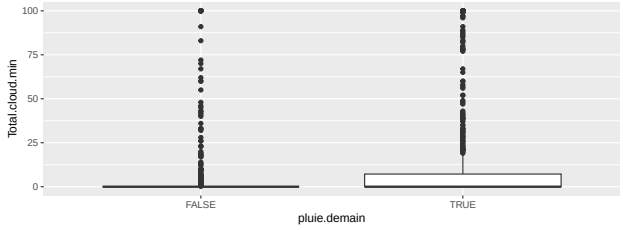


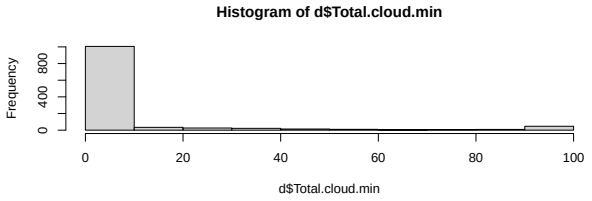
L'histogramme montre une concentration des observations vers la gauche, ce qui indique que la majorité des valeurs de la variable considérée sont regroupées à des niveaux inférieurs. Cependant, l'observation du boxplot révèle que la variable "precip" semble fournir des indications sur la variable "pluie.demain".

Cette interprétation suggère que les observations où "pluie.demain" est égal à TRUE ont tendance à présenter des niveaux de précipitation supérieurs à zéro. Cela signifie qu'il y a une relation positive entre la variable "precip" et la probabilité qu'il pleuve le lendemain.

Total.cld.min

```
ggplot(d, aes(x=pluie.demain, y=Total.cloud.min)) + geom_boxplot()
hist(d$Total.cloud.min)
```

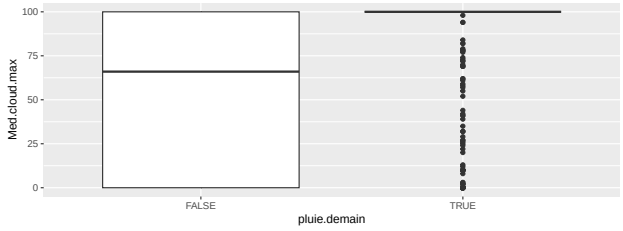


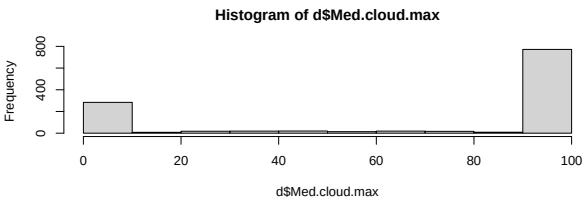


Cette interprétation suggère que les observations où "pluie.demain" est égal à TRUE ont tendance à présenter des niveaux de précipitation supérieurs à zéro. Cela signifie qu'il y a une relation positive entre la variable "Total.cloud.min" et la probabilité qu'il pleuve le lendemain.

Med.cloud.max

```
ggplot(d, aes(x=pluie.demain, y=Med.cloud.max )) + geom_boxplot()
hist(d$Med.cloud.max)
```

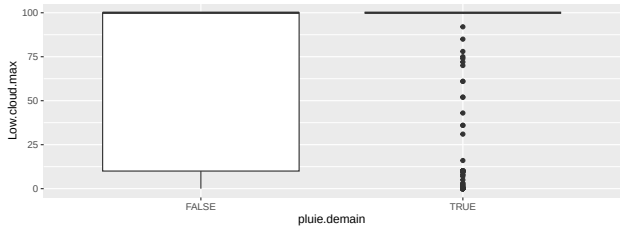


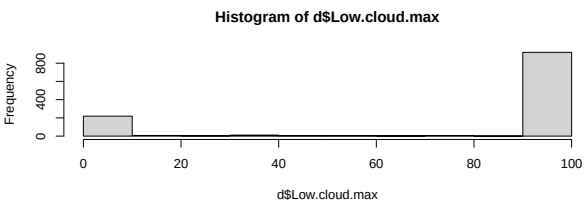


Cette interprétation suggère que les observations où “pluie.demain” est égal à FALSE ont tendance à présenter des niveaux de nébulosité à zéro. Cela signifie qu’il y a une relation positive entre la variable “Med.cloud.max” et la probabilité qu’il pleuve le lendemain.

Low.cloud.max

```
ggplot(d, aes(x=pluie.demain, y=Low.cloud.max )) + geom_boxplot()
hist(d$Low.cloud.max)
```





Cette interprétation suggère que les observations où “pluie.demain” est égal à FALSE ont tendance à présenter des niveaux de nébulosité à zéro. Cela signifie qu’il y a une relation positive entre la variable “Low.cloud.max” et la probabilité qu’il pleuve le lendemain.

3. Construction des modèles

3.1 Modèle 1: modélisation avec toutes les covariables du jeu de données

Dans le cadre de la construction de ce premier modèle, nous allons prendre en compte toutes les covariables de notre jeu de données ajusté.

On obtient un modèle composé de 13 variables explicatives.

```
m1 = glm(formula = pluie.demain ~ .,family = binomial,data = d)
print(mean(abs(round(predict(m1, d, type = "response"))-d$pluie.demain)))
```

```
## [1] 0.2669492
```

```
f1 = formula(m1) # formule du modèle
n1 = colnames(d) # nom des covariables retenues
summary(m1)
```

```
##
## Call:
## glm(formula = pluie.demain ~ ., family = binomial, data = d)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2641  -0.8777   0.4056   0.8427   2.8251
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          5.167e+01  1.154e+01   4.476 7.60e-06 ***
## Temp.mean            3.509e-02  1.744e-02   2.012 0.044169 *
## Hum.mean            -1.005e-02  1.640e-02  -0.612 0.540315
## Press.mean          -5.498e-02  1.124e-02  -4.892 9.99e-07 ***
## Precipitation        4.615e-03  2.397e-02   0.193 0.847315
## High.cloud.mean     -7.082e-04  5.893e-03  -0.120 0.904349
## Med.cloud.mean       5.997e-03  4.942e-03   1.213 0.224962
## Low.cloud.mean       4.278e-03  4.363e-03   0.980 0.326862
## Radiation            1.206e-04  7.059e-05   1.709 0.087514 .
## Wind.direc.80.m     -3.679e-03  1.584e-03  -2.324 0.020152 *
## Wind.direc.900.m     4.760e-03  1.318e-03   3.611 0.000305 ***
## Hum.max              1.417e-02  1.618e-02   0.876 0.381149
## Total.cloud.min      6.820e-03  4.149e-03   1.644 0.100194
## High.cloud.max       3.886e-03  2.741e-03   1.418 0.156172
## Med.cloud.max        8.214e-03  2.656e-03   3.093 0.001982 **
## Low.cloud.max        4.237e-03  2.559e-03   1.655 0.097839 .
## Wind.speed.80.max    1.515e-02  1.342e-02   1.129 0.258764
## Wind.speed.80m.min  -6.337e-03  2.029e-02  -0.312 0.754773
## Wind.speed.900m.min  5.351e-03  9.079e-03   0.589 0.555589
## Press.var            3.760e-02  2.393e-02   1.572 0.116059
## Gust.var             9.248e-03  9.302e-03   0.994 0.320128
```

```
## Cos.cycle.saison     2.893e-01  2.229e-01   1.298 0.194367
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1635.4  on 1179  degrees of freedom
## Residual deviance: 1286.8  on 1158  degrees of freedom
## AIC: 1330.8
##
## Number of Fisher Scoring iterations: 4
```

```
BIC(m1)
```

```
## [1] 1442.412
```

3.2 Modèle 2: step AIC

Pour la construction de ce deuxieme modèle, nous allons faire une recherche pas à pas en utilisant le critère AIC .

```
fit1 = glm(pluie.demain ~ ., family = binomial, data = d)
fit2 = glm(pluie.demain ~ 1, family = binomial, data = d)
mAIC = step(fit2,direction="both"     ,scope=list(upper=fit1,lower=fit2))
```

```
fAIC = formula(mAIC) # formule du modèle
nAIC = names(mAIC$coefficients) # nom des covariables retenues
summary(mAIC)
```

```
##
## Call:
## glm(formula = pluie.demain ~ Med.cloud.max + Press.mean + Wind.direc.900.m +
##     Temp.mean + Med.cloud.mean + Press.var + Low.cloud.max +
##     Wind.direc.80.m + Total.cloud.min + Radiation + Wind.speed.80.max +
##     High.cloud.max + Cos.cycle.saison, family = binomial, data = d)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3141  -0.8826   0.4209   0.8481   2.7823
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        5.262e+01  1.143e+01   4.603 4.16e-06 ***
## Med.cloud.max      7.909e-03  2.563e-03   3.086 0.002031 **
## Press.mean        -5.540e-02  1.113e-02  -4.979 6.40e-07 ***
## Wind.direc.900.m   4.745e-03  1.288e-03   3.683 0.000231 ***
## Temp.mean          3.707e-02  1.650e-02   2.247 0.024655 *
## Med.cloud.mean     6.887e-03  4.214e-03   1.634 0.102213
## Press.var          4.010e-02  2.378e-02   1.686 0.091791 .
## Low.cloud.max      5.999e-03  2.220e-03   2.702 0.006895 **
## Wind.direc.80.m   -3.375e-03  1.500e-03  -2.250 0.024473 *
## Total.cloud.min    8.068e-03  3.750e-03   2.151 0.031451 *
## Radiation          1.087e-04  5.606e-05   1.939 0.052527 .
## Wind.speed.80.max  2.317e-02  8.259e-03   2.805 0.005031 **
## High.cloud.max     3.464e-03  2.166e-03   1.599 0.109746
## Cos.cycle.saison   3.166e-01  2.217e-01   1.428 0.153162
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1635.4  on 1179  degrees of freedom
## Residual deviance: 1290.7  on 1166  degrees of freedom
## AIC: 1318.7
##
## Number of Fisher Scoring iterations: 4
```

```r
BIC(mAIC)
```

```
## [1] 1389.694
```

AIC modèle 2: 1318.7 vs AIC modèle 1 = 1330.8

BIC modèle 2: 1389.7 vs modèle 1 = 1442.4

3.3 Modèle 3: step BIC

Pour la construction de ce troisième modèle, nous allons effectuer une recherche pas à pas en utilisant le critère BIC.

Le critère BIC privilégie les modèles avec peu de covariables, et on obtient donc dans ce troisième modèle 6 variables explicatives.

```
fit1 = glm(pluie.demain ~ ., family = binomial, data = d)
fit2 = glm(pluie.demain ~ 1, family = binomial, data = d)
mBIC = step(fit2,direction="both"    ,scope=list(upper=fit1,lower=fit2) ,k=log(nrow(d)))
```

```
fBIC = formula(mBIC) # formule du modèle
nBIC = names(mBIC$coefficients) # nom des covariables retenues
summary(mBIC)
```

```
##
## Call:
## glm(formula = pluie.demain ~ Med.cloud.max + Press.mean + Wind.direc.900.m +
##     Gust.var + Temp.mean + Med.cloud.mean, family = binomial,
##     data = d)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1728  -0.8991   0.4177   0.8692   2.5054
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)      59.7925504 10.9219628   5.475 4.39e-08 ***
## Med.cloud.max     0.0098358  0.0022814   4.311 1.62e-05 ***
## Press.mean       -0.0614197  0.0106743  -5.754 8.72e-09 ***
## Wind.direc.900.m  0.0030635  0.0009642   3.177 0.001487 **
## Gust.var          0.0222911  0.0064034   3.481 0.000499 ***
## Temp.mean         0.0436541  0.0104100   4.193 2.75e-05 ***
## Med.cloud.mean    0.0099953  0.0033919   2.947 0.003210 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1635.4  on 1179  degrees of freedom
## Residual deviance: 1312.1  on 1173  degrees of freedom
## AIC: 1326.1
##
## Number of Fisher Scoring iterations: 4
```

3.4 Modèle 4: interaction des variables

Pour trouver ce modèle, nous avons utilisé les variables communes des modèles 2 et 3, puis nous avons cherché manuellement les combinaisons de variables qui pourraient apporter de l'information supplémentaire.

Les combinaisons suivantes ont été ajoutées :

-Temp.mean:Press.mean -Temp.mean:Wind.direc.900.m -Temp.mean:Med.cloud.max -Med.cloud.max:Med.cloud.mean

```
m.inter = glm(pluie.demain ~
Med.cloud.max
+Press.mean
+Wind.direc.900.m
+Temp.mean
+Med.cloud.mean
+Temp.mean:Press.mean
+Temp.mean:Wind.direc.900.m
+Temp.mean:Med.cloud.max
+Med.cloud.max:Med.cloud.mean
, family = binomial, data = d)
```

```
  finter = formula(m.inter) # formule du modèle
  ninter = names(m.inter$coefficients) # nom des covariables retenues
  summary(m.inter)
```

```
##
## Call:
## glm(formula = pluie.demain ~ Med.cloud.max + Press.mean + Wind.direc.900.m +
##     Temp.mean + Med.cloud.mean + Temp.mean:Press.mean + Temp.mean:Wind.direc.900.m +
##     Temp.mean:Med.cloud.max + Med.cloud.max:Med.cloud.mean, family = binomial,
##     data = d)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4847  -0.8562   0.3191   0.8438   2.3720
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -1.231e+01  1.747e+01  -0.705   0.4810
## Med.cloud.max                  8.625e-04  4.249e-03   0.203   0.8391
## Press.mean                     9.126e-03  1.707e-02   0.535   0.5928
## Wind.direc.900.m               1.055e-02  1.940e-03   5.437 5.43e-08 ***
## Temp.mean                      1.043e+01  1.849e+00   5.641 1.69e-08 ***
## Med.cloud.mean                 1.160e-01  7.603e-02   1.526   0.1271
## Press.mean:Temp.mean          -1.015e-02  1.812e-03  -5.601 2.13e-08 ***
## Wind.direc.900.m:Temp.mean    -5.950e-04  1.474e-04  -4.037 5.42e-05 ***
## Med.cloud.max:Temp.mean        6.596e-04  2.663e-04   2.476   0.0133 *
## Med.cloud.max:Med.cloud.mean -1.039e-03  7.529e-04  -1.380   0.1676
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1635.4  on 1179  degrees of freedom
## Residual deviance: 1265.2  on 1170  degrees of freedom
## AIC: 1285.2
##
## Number of Fisher Scoring iterations: 4
```

En moyenne, la probabilité d'avoir de la pluie le lendemain est principalement influencée par deux facteurs : la température moyenne (Temp.mean) et la couverture nuageuse à moyennes altitudes (Med.cloud.mean). Ces deux variables ont tendance à augmenter la probabilité que la variable pluie.demain soit positive.

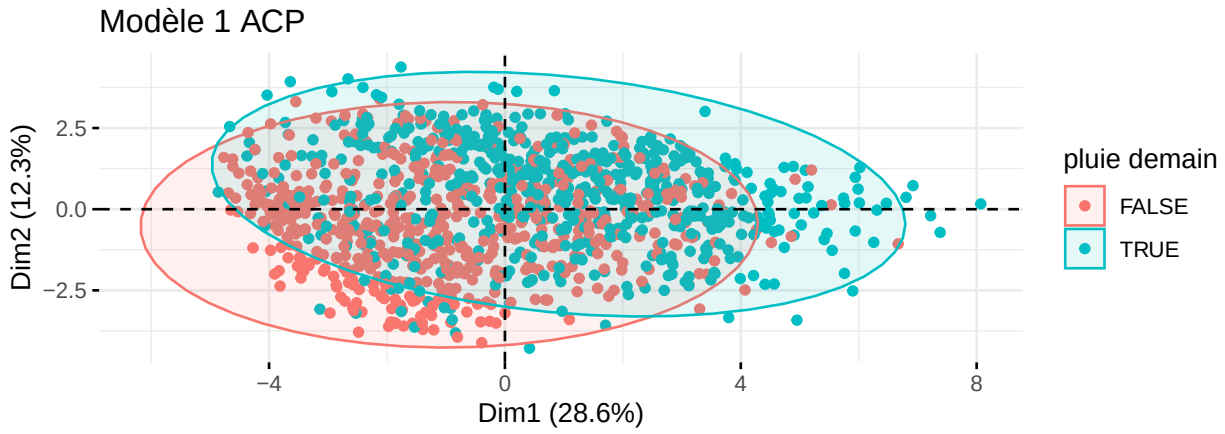
De plus, lorsqu'il y a une interaction entre la pression atmosphérique et la température, cet effet combiné est celui qui a le plus grand impact sur l'augmentation de la probabilité d'avoir de la pluie le lendemain.

4. Visualisation des tendances

Afin d'identifier le modèle le plus performant, c'est à dire celui qui discrimine le mieux la variable "pluie.demain", nous allons réaliser l'ACP de chacun des modeles constuits ci-dessus.

4.1 Modèle 1: modélisation avec toutes les covariables du jeu de données

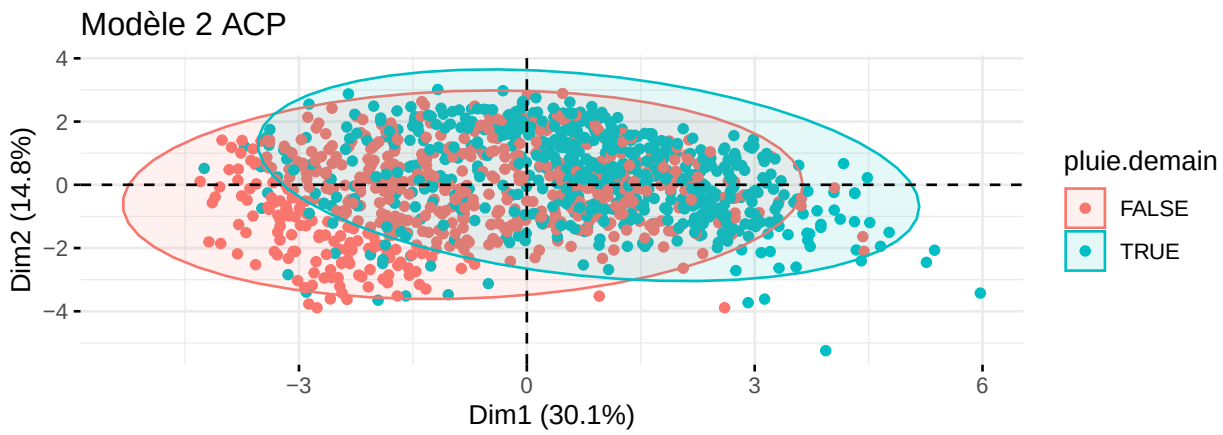
```
D1 = d
res.pca = PCA(D1,quali.sup = which(colnames(D1)=="pluie.demain"),graph=FALSE)
fviz_pca_ind (res.pca, geom.ind="point", col.ind=d$pluie.demain ,
              legend.title="pluie demain", addEllipses = T, title="Modèle 1 ACP")
```




```
x1 = data.frame(res.pca$ind$coord,res.pca$call$quali.sup$quali.sup)
x1$pluie.demain = x1$pluie.demain=="TRUE"
```

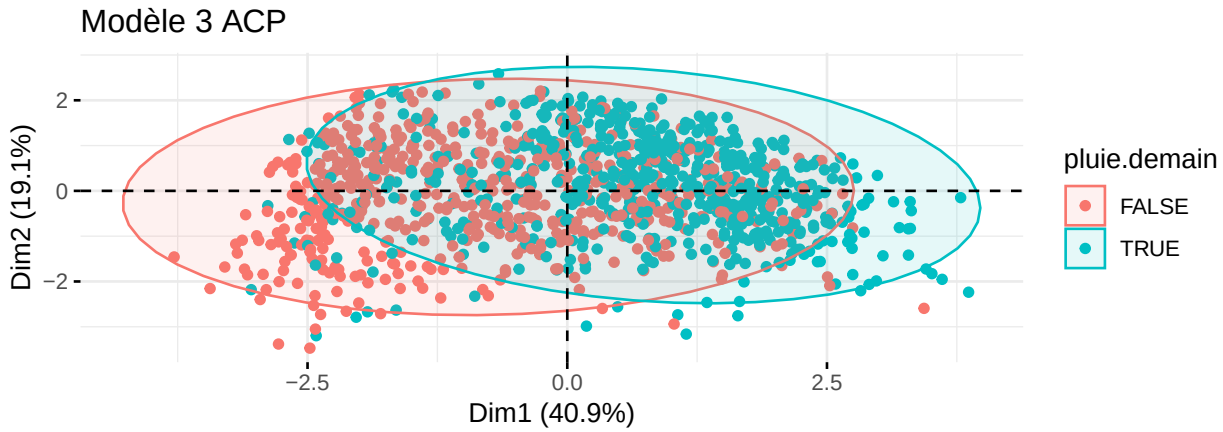
4.2 Modèle 2: step AIC

```
D2 = d[,(colnames(d) %in% nAIC) | colnames(d)=="pluie.demain"]
res.pca = PCA(D2,quali.sup = which(colnames(D2)=="pluie.demain"),graph=FALSE)
fviz_pca_ind (res.pca, geom.ind="point", col.ind=d$pluie.demain ,
              legend.title="pluie.demain", addEllipses = T, title="Modèle 2 ACP")
```



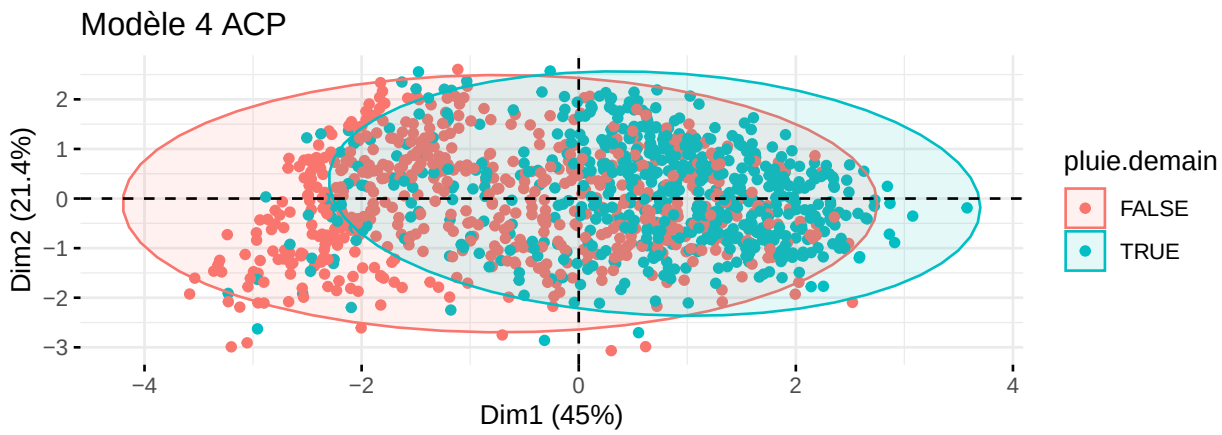
4.3 Modèle 3: step BIC

```
D3 = d[,(colnames(d) %in% nBIC) | colnames(d)=="pluie.demain"]
res.pca = PCA(D3,quali.sup = which(colnames(D3)=="pluie.demain"),graph=FALSE)
fviz_pca_ind (res.pca, geom.ind="point", col.ind=d$pluie.demain ,
              legend.title="pluie.demain", addEllipses = T, title="Modèle 3 ACP")
```



4.4 Modèle 4: interaction des variables

```
D4 = d[,(colnames(d) %in% ninter) | colnames(d)=="pluie.demain"]
  res.pca = PCA(D4,quali.sup = which(colnames(D4)=="pluie.demain"),graph=FALSE)
  fviz_pca_ind (res.pca, geom.ind="point", col.ind=d$pluie.demain ,
                legend.title="pluie.demain", addEllipses = T, title="Modèle 4 ACP")
```



4.5 Conclusions

Le modèle 3 (BIC) et 4 (interaction) semblent être les meilleurs modèle au vu des:

- ellipses un peu séparés
- 61% expliqués par 2 PC pour le modèle 3 et 66,4% pour le modèle 4
- des AIC, BIC et p-value analysés ci-dessus

5. Comparaison de la performance de prédiction

5.1 Validation croisée

Fonction

```
myCrossValidation = function(formule,dataFrame,nParts)
{
  # Initialization
    errV = numeric(0)
  # Boucle for sur le nombre de parties
    for (k in 1:nParts)
    {
      # Calcul des indices du jeu de test
        indTest = seq(k,nrow(dataFrame),nParts)
        df_test = dataFrame[indTest,]
        df_train = dataFrame[-indTest,]
      # Calcul du modèle
        modele = glm(formule, family = binomial, data = df_train)
      # Calcul des coefficient du modèle final
        if (k==1)
          { MODEL = modele
            N_TRAIN = nrow(df_train)
            MODEL$coefficients = modele$coefficients*N_TRAIN
            N_SUM = N_TRAIN }
        else
          { N_TRAIN = nrow(df_train)
            MODEL$coefficients = MODEL$coefficients + modele$coefficients*N_TRAIN
            N_SUM = N_SUM + N_TRAIN }
      # Prédiction pour la validation crois
        pred = predict(modele, df_test, type = "response")
      # Calcul de l'erreur de prediction
        err = mean(abs(round(pred)-df_test$pluie.demain))
        #print(mean(abs(round(predict(modele, dataFrame, type = "response"))-dataFrame$pluie.demain))
      # Ajout au vecteur
        errV = rbind(errV,err)
    }
  # Output
    MODEL$coefficients = MODEL$coefficients/N_SUM
    return(list(erreur = errV, modele = MODEL))
}
```

Application de la fonction au projet

```
cv_m1 = myCrossValidation(f1, d, 10)
cv_mAIC = myCrossValidation(fAIC  , d, 10)
cv_mBIC = myCrossValidation(fBIC  , d, 10)
cv_m.inter= myCrossValidation(pluie.demain ~ . , d, 10)
```

5.2 Observation des résultats

Voici l'ensemble des taux d'erreur pour les différentes méthodes

```
Rst = data.frame(cv_m1$erreur,cv_mAIC$erreur,cv_mBIC$erreur,cv_m.inter$erreur)
Rst
```

```
##       cv_m1.erreur cv_mAIC.erreur cv_mBIC.erreur cv_m.inter.erreur
## err      0.2881356      0.2796610      0.2796610         0.2881356
## err.1    0.2796610      0.2796610      0.2711864         0.2796610
## err.2    0.2542373      0.2627119      0.1949153         0.2542373
## err.3    0.2711864      0.2966102      0.3050847         0.2711864
## err.4    0.2966102      0.2966102      0.2796610         0.2966102
## err.5    0.2711864      0.2542373      0.2457627         0.2711864
## err.6    0.2796610      0.2711864      0.2881356         0.2796610
## err.7    0.2542373      0.2542373      0.2711864         0.2542373
## err.8    0.2881356      0.2796610      0.2711864         0.2881356
## err.9    0.3220339      0.3305085      0.3305085         0.3220339
```

```
summary(Rst)
```

```
##   cv_m1.erreur    cv_mAIC.erreur   cv_mBIC.erreur   cv_m.inter.erreur
##  Min.   :0.2542   Min.   :0.2542   Min.   :0.1949   Min.   :0.2542
##  1st Qu.:0.2712   1st Qu.:0.2648   1st Qu.:0.2712   1st Qu.:0.2712
##  Median :0.2797   Median :0.2797   Median :0.2754   Median :0.2797
##  Mean   :0.2805   Mean   :0.2805   Mean   :0.2737   Mean   :0.2805
##  3rd Qu.:0.2881   3rd Qu.:0.2924   3rd Qu.:0.2860   3rd Qu.:0.2881
##  Max.   :0.3220   Max.   :0.3305   Max.   :0.3305   Max.   :0.3220
```

Au niveau des critère AIC BIC, le modèle 4 est le plus performant. Sur le plan de la prédiction, le modèle est 4 est également le plus performant, il obtient la moyenne la plus faible avec **0.3220**.

Il nous semble donc pertinent de selectionner le modèle 4 pour la prédiction de pluie.demain dans le fichier test.

6. Prédiction avec le modèle 4 - interaction

6.1 Importation du fichier test et prédiction

```
d.test = read.table("meteo.test.csv",header=T,sep=",")
```

6.3 Modèle final retenu (modèle 4 - interaction)

```
m.final.pred = glm(pluie.demain ~
Med.cloud.max
+Press.mean
+Wind.direc.900.m
+Temp.mean
+Med.cloud.mean
+Temp.mean:Press.mean
+Temp.mean:Wind.direc.900.m
+Temp.mean:Med.cloud.max
+Med.cloud.max:Med.cloud.mean
, family = binomial, data = d)
```

Nous rappelons qu'en moyenne, la probabilité d'avoir de la pluie le lendemain est davantage influencée par deux facteurs : la température moyenne (Temp.mean) et la couverture nuageuse à moyennes altitudes (Med.cld.mean). Ces deux variables ont tendance à augmenter la probabilité que la variable pluie.demain soit positive.

De plus, lorsque la pression atmosphérique agit conjointement sur la température, cet effet combiné est celui qui a le plus grand impact sur l'augmentation de la probabilité d'avoir de la pluie le lendemain.

6.4 Prédiction & Export

```
subtest = subset(d.test, select = c(Med.cloud.max, Press.mean, Wind.direc.900.m, Temp.mean,Med.cloud.mea
pred.test = predict(m.final.pred, newdata=subtest,type="response")
prediction = data.frame(d.test$X,pred.test >= 0.5)
write.csv(x = prediction, file = "ProjetGLM_prediction_pluie.demain.csv")
```