[Generalized.Linear.Model-Model.Selection] Meteo Bâle-Suisse

Miora ANDRIANTSEHENO

09/01/2022

Contents

Introduction . . . 2
I. Analyse préliminaire des données . . . 3
I.1 Préparation des données . . . 3
I.2 Corrélogramme . . . 7
I.3 Analyse graphique des données . . . 8
II. Approche par échantillon de validation (apprentissage/validation) . . . 18
II.1 Partitionnement de Meteotrain (80/20) . . . 18
II.2 Sélection de variables . . . 20
II.2.1 Modèle saturé . . . 20
II.2.2 Modèles avec Best Subset Selection . . . 20
II.2.3 Modèles pas à pas : Forward, Backward, Stepwise (indice AIC) . . . 27
II.2.4 Modèles pas à pas : Forward, Backward, Stepwise (indice Bic) . . . 30
II.2.5 Approche par Analyse en Composantes Principales . . . 34
II.2.6 Tableau récapitulatif des modèles . . . 73
II.3 Première évaluation des modèles (Wald, pseudos R2) . . . 75
II.3.1 Test de Wald (évaluation de la normalité asymptotique des estimateurs) 75
II.3.2 Pseudos R2 . . . 76
II.4 Evaluation et validation des modèles . . . 78
II.4.1 Scores d'affectation et erreurs de prédiction . . . 78
II.4.2 Qualité d'ajustement du modèle aux données . . . 80
II.4.3 Tableau récapitulatif sur l'évaluation des modèles . . . 97
II.5 Analyse des résidus . . . 100
III. Approche par validation croisée en K blocs . . . 110
IV. Approche par validation croisée Leave-One-Out (LOOCV) . . . 112
V. Decision du modèle retenu et application sur l'échantillon meteotest . . . 113
Conclusion . . . 114

Introduction

Nous disposons dans notre dataset de données météorologiques de toute nature (température, pression, nébulosité, vent...) se rapportant aux journées entre 2010 et 2018 dans la ville de Bâle en Suisse. Nous allons les exploiter pour faire de la prédiction sur la prévision de pluie aux journées supplémentaires contenues dans meteo.test.

Pour ce faire, nous allons dans un premier temps nous approprier les données, en effectuant divers traitements sur les données.

Dans cette première étape, nous allons observer les données par le summary fourni afin d'avoir un premier aperçu de celles-ci, tracer le corrélogramme, et analyser graphiquement ce que la variable cible et les variables explicatives apportent comme premières informations.

En utilisant plusieurs approches, nous allons mettre en oeuvre une régression logistique.

Pour ce faire, en premier lieu, nous réaliserons l'approche par l'échantillon de validation : nous divisons la dataset Meteotrain en 2, une dataset pour l'apprentissage, et l'autre pour la validation.
Avec différentes méthodes, nous procéderons à la sélection de variables :
- Sélection par recherche exhaustive en best subset (Cp Mallows, R2 ajusté, RSS, BIC)
- Sélection en méthodes pas à pas (forward, backward, stepwise) en considérant 3 critères (AIC, BIC, Cp)
Ceci nous conduira à un certain nombre de modèles.

Comme notre dataset est de grande dimension, il a semblé pertinent de réaliser une analyse en composantes principales pour réduire sa taille, et utiliser ses axes factoriels retenus comme régresseurs. Avec les 2 méthodes déjà utilisées auparavant, nous obtiendrons d'autres modèles.
Afin d'évaluer leur qualité d'ajustement aux données et leur performance prédictive, chaque modèle sera testé au moyen de différents critères.

En deuxième lieu, nous verrons la mise en oeuvre de l'approche par validation croisée en k-folds, qui viendra challenger les modèles obtenus.

Enfin, en troisième lieu, la validation croisée LOOCV viendra également pour conforter les résultats précédents.

Dans une dernière étape, nous déciderons du modèle retenu qui satisfait le mieux et procéderons aux prédictions sur Meteotest.

I. Analyse préliminaire des données

Chargeons les données

```
# Chargement des 2 jeux de données #
rm(list=ls())
meteotrain <- read.csv("meteo.train.csv")
meteotest <- read.csv("meteo.test.csv")
```

Voyons ce qu'elles contiennent

```
summary(meteotrain)
summary(meteotest)
```

Les 2 jeux contiennent les mêmes variables à l'exception près que dans le jeu meteotest, la variable dépendante pluie.demain n'y est pas.

C'est la variable que nous souhaitons prédire grâce au premier jeu de données qui servira d'apprentissage et de validation. Nous verrons plus loin qu'il faudra diviser meteotrain en 2 ou plusieurs partitions selon l'approche utilisée.

Toutes nos variables sont quantitatives hormis la variable que nous cherchons à expliquer pluie.demain, qui se présente comme une variable logique (True,False). Notre summary nous indique une information cruciale : il semblerait que nous ayons quasiment autant de valeurs TRUE que de FALSE pour cette variable.
==> Ceci nous amène à penser que la probabilité est d'environ 0.5 qu'il y ait de la pluie le lendemain selon les données de l'échantillon. Cette variable suit bien une loi de bernoulli de probabilité 0.5, et cela nous conforte dans notre mise en application de la régression logistique.

Il semblerait que les variables Hour et Minute soient nulles partout. Dans la préparation des données, nous les supprimerons des bases.

I.1 Préparation des données

Pour la préparation de nos 2 jeux de données, nous allons faire les traitements suivants : suppression des données manquantes, suppression de données inutiles, recodage des colonnes pour avoir des noms de variables plus courts et lisibles, suppression de variables inutiles, vérification du type de la variable pluie.demain et conversion en factor.

```
# Préparation données : renommer les variables de Meteotrain et Meteotest #
names(meteotrain)
meteotrain <- rename.variable(meteotrain,
                              "Day","Jour")
meteotrain <- rename.variable(meteotrain,                                "High.Cloud.Cover.daily.max..hig
meteotrain <- rename.variable(meteotrain,
    "High.Cloud.Cover.daily.mean..high.cld.lay.","Nebulosite.forte.moy")
meteotrain <- rename.variable(meteotrain,    "High.Cloud.Cover.daily.min..high.cld.lay.","Nebulosite.fo
meteotrain <- rename.variable(meteotrain,
                              "Hour","Heure")
```

```
meteotrain <- rename.variable(meteotrain,    "Low.Cloud.Cover.daily.max..low.cld.lay.","Nebulosite.faibl
meteotrain <- rename.variable(meteotrain,    "Low.Cloud.Cover.daily.mean..low.cld.lay.","Nebulosite.fail
meteotrain <- rename.variable(meteotrain,
"Low.Cloud.Cover.daily.min..low.cld.lay.","Nebulosite.faible.min")
meteotrain <- rename.variable(meteotrain,    "Mean.Sea.Level.Pressure.daily.max..MSL.","Pression.max")
meteotrain <- rename.variable(meteotrain,
"Mean.Sea.Level.Pressure.daily.mean..MSL.","Pression.moy")
meteotrain <- rename.variable(meteotrain,
"Mean.Sea.Level.Pressure.daily.min..MSL.","Pression.min")
meteotrain <- rename.variable(meteotrain,    "Medium.Cloud.Cover.daily.max..mid.cld.lay.","Nebulosite.m
meteotrain <- rename.variable(meteotrain,    "Medium.Cloud.Cover.daily.mean..mid.cld.lay.","Nebulosite.
meteotrain <- rename.variable(meteotrain,    "Medium.Cloud.Cover.daily.min..mid.cld.lay.","Nebulosite.m
meteotrain <- rename.variable(meteotrain,
                              "Minute","Minute")
meteotrain <- rename.variable(meteotrain,
                              "Month","Mois")
meteotrain <- rename.variable(meteotrain, "Relative.Humidity.daily.max..2.m.above.gnd.","Humidite.max")
meteotrain <- rename.variable(meteotrain,   "Relative.Humidity.daily.mean..2.m.above.gnd.","Humidite.moy
meteotrain <- rename.variable(meteotrain,    "Relative.Humidity.daily.min..2.m.above.gnd.","Humidite.mi
meteotrain <- rename.variable(meteotrain,    "Shortwave.Radiation.daily.sum..sfc.","Rayonnement_solR")
meteotrain <- rename.variable(meteotrain,    "Snowfall.amount.raw.daily.sum..sfc.","Enneigement")
meteotrain <- rename.variable(meteotrain,    "Sunshine.Duration.daily.sum..sfc.","Ensoleillement")
meteotrain <- rename.variable(meteotrain,    "Temperature.daily.max..2.m.above.gnd.","Temperature.max")
meteotrain <- rename.variable(meteotrain,    "Temperature.daily.mean..2.m.above.gnd.","Temperature.moy"
meteotrain <- rename.variable(meteotrain,    "Temperature.daily.min..2.m.above.gnd.","Temperature.min")
meteotrain <- rename.variable(meteotrain,    "Total.Cloud.Cover.daily.max..sfc.","TotalNebulosite.max")
meteotrain <- rename.variable(meteotrain,    "Total.Cloud.Cover.daily.mean..sfc.","TotalNebulosite.moy"
meteotrain <- rename.variable(meteotrain,    "Total.Cloud.Cover.daily.min..sfc.","TotalNebulosite.min")
meteotrain <- rename.variable(meteotrain,    "Total.Precipitation.daily.sum..sfc.","Precipitations")
meteotrain <- rename.variable(meteotrain,    "Wind.Direction.daily.mean..10.m.above.gnd.","DirectionVen
meteotrain <- rename.variable(meteotrain,    "Wind.Direction.daily.mean..80.m.above.gnd.","DirectionVen
meteotrain <- rename.variable(meteotrain,    "Wind.Direction.daily.mean..900.mb.","DirectionVent_900hpa
meteotrain <- rename.variable(meteotrain,
"Wind.Gust.daily.max..sfc.","RafalesVent.max")
meteotrain <- rename.variable(meteotrain,
"Wind.Gust.daily.mean..sfc.","RafalesVent.moy")
meteotrain <- rename.variable(meteotrain,
"Wind.Gust.daily.min..sfc.","RafalesVent.min")
meteotrain <- rename.variable(meteotrain,    "Wind.Speed.daily.max..10.m.above.gnd.","VitesseVent_10m.ma
meteotrain <- rename.variable(meteotrain,    "Wind.Speed.daily.max..80.m.above.gnd.","VitesseVent_80m.ma
meteotrain <- rename.variable(meteotrain,    "Wind.Speed.daily.max..900.mb.","VitesseVent_900hpa.max")
meteotrain <- rename.variable(meteotrain,    "Wind.Speed.daily.mean..10.m.above.gnd.","VitesseVent_10m.
meteotrain <- rename.variable(meteotrain,    "Wind.Speed.daily.mean..80.m.above.gnd.","VitesseVent_80m.
meteotrain <- rename.variable(meteotrain,    "Wind.Speed.daily.mean..900.mb.","VitesseVent_900hpa.moy")
meteotrain <- rename.variable(meteotrain,    "Wind.Speed.daily.min..10.m.above.gnd.","VitesseVent_10m.mi
 meteotrain <- rename.variable(meteotrain,   "Wind.Speed.daily.min..80.m.above.gnd.","VitesseVent_80m.mi
meteotrain <- rename.variable(meteotrain,    "Wind.Speed.daily.min..900.mb.","VitesseVent_900hpa.min")
meteotrain <- rename.variable(meteotrain,
                              "X","X")
meteotrain <- rename.variable(meteotrain,
                              "Year","Annee")

## Renommage de meteotest ##
```

```
names(meteotest)
meteotest <- rename.variable(meteotest,
                             "Day","Jour")
meteotest <- rename.variable(meteotest,                                "High.Cloud.Cover.daily.max..high
meteotest <- rename.variable(meteotest,
    "High.Cloud.Cover.daily.mean..high.cld.lay.","Nebulosite.forte.moy")
meteotest <- rename.variable(meteotest,    "High.Cloud.Cover.daily.min..high.cld.lay.","Nebulosite.forte
meteotest <- rename.variable(meteotest,
                             "Hour","Heure")
meteotest <- rename.variable(meteotest,    "Low.Cloud.Cover.daily.max..low.cld.lay.","Nebulosite.faible
meteotest <- rename.variable(meteotest,    "Low.Cloud.Cover.daily.mean..low.cld.lay.","Nebulosite.faible
meteotest <- rename.variable(meteotest,
"Low.Cloud.Cover.daily.min..low.cld.lay.","Nebulosite.faible.min")
meteotest <- rename.variable(meteotest,    "Mean.Sea.Level.Pressure.daily.max..MSL.","Pression.max")
meteotest <- rename.variable(meteotest,
"Mean.Sea.Level.Pressure.daily.mean..MSL.","Pression.moy")
meteotest <- rename.variable(meteotest,
"Mean.Sea.Level.Pressure.daily.min..MSL.","Pression.min")
meteotest <- rename.variable(meteotest,    "Medium.Cloud.Cover.daily.max..mid.cld.lay.","Nebulosite.moy
meteotest <- rename.variable(meteotest,    "Medium.Cloud.Cover.daily.mean..mid.cld.lay.","Nebulosite.moy
meteotest <- rename.variable(meteotest,    "Medium.Cloud.Cover.daily.min..mid.cld.lay.","Nebulosite.moy
meteotest <- rename.variable(meteotest,
                             "Minute","Minute")
meteotest <- rename.variable(meteotest,
                             "Month","Mois")
meteotest <- rename.variable(meteotest, "Relative.Humidity.daily.max..2.m.above.gnd.","Humidite.max")
meteotest <- rename.variable(meteotest,   "Relative.Humidity.daily.mean..2.m.above.gnd.","Humidite.moy"
meteotest <- rename.variable(meteotest,    "Relative.Humidity.daily.min..2.m.above.gnd.","Humidite.min"
meteotest <- rename.variable(meteotest,    "Shortwave.Radiation.daily.sum..sfc.","Rayonnement_solR")
meteotest <- rename.variable(meteotest,    "Snowfall.amount.raw.daily.sum..sfc.","Enneigement")
meteotest <- rename.variable(meteotest,    "Sunshine.Duration.daily.sum..sfc.","Ensoleillement")
meteotest <- rename.variable(meteotest,    "Temperature.daily.max..2.m.above.gnd.","Temperature.max")
meteotest <- rename.variable(meteotest,    "Temperature.daily.mean..2.m.above.gnd.","Temperature.moy")
meteotest <- rename.variable(meteotest,    "Temperature.daily.min..2.m.above.gnd.","Temperature.min")
meteotest <- rename.variable(meteotest,    "Total.Cloud.Cover.daily.max..sfc.","TotalNebulosite.max")
meteotest <- rename.variable(meteotest,    "Total.Cloud.Cover.daily.mean..sfc.","TotalNebulosite.moy")
meteotest <- rename.variable(meteotest,    "Total.Cloud.Cover.daily.min..sfc.","TotalNebulosite.min")
meteotest <- rename.variable(meteotest,    "Total.Precipitation.daily.sum..sfc.","Precipitations")
meteotest <- rename.variable(meteotest,    "Wind.Direction.daily.mean..10.m.above.gnd.","DirectionVent_
meteotest <- rename.variable(meteotest,    "Wind.Direction.daily.mean..80.m.above.gnd.","DirectionVent_8
meteotest <- rename.variable(meteotest,    "Wind.Direction.daily.mean..900.mb.","DirectionVent_900hpa.m
meteotest <- rename.variable(meteotest,
"Wind.Gust.daily.max..sfc.","RafalesVent.max")
meteotest <- rename.variable(meteotest,
"Wind.Gust.daily.mean..sfc.","RafalesVent.moy")
meteotest <- rename.variable(meteotest,
"Wind.Gust.daily.min..sfc.","RafalesVent.min")
meteotest <- rename.variable(meteotest,    "Wind.Speed.daily.max..10.m.above.gnd.","VitesseVent_10m.max
meteotest <- rename.variable(meteotest,    "Wind.Speed.daily.max..80.m.above.gnd.","VitesseVent_80m.max
meteotest <- rename.variable(meteotest,    "Wind.Speed.daily.max..900.mb.","VitesseVent_900hpa.max")
meteotest <- rename.variable(meteotest,    "Wind.Speed.daily.mean..10.m.above.gnd.","VitesseVent_10m.moy
meteotest <- rename.variable(meteotest,    "Wind.Speed.daily.mean..80.m.above.gnd.","VitesseVent_80m.moy
meteotest <- rename.variable(meteotest,    "Wind.Speed.daily.mean..900.mb.","VitesseVent_900hpa.moy")
```

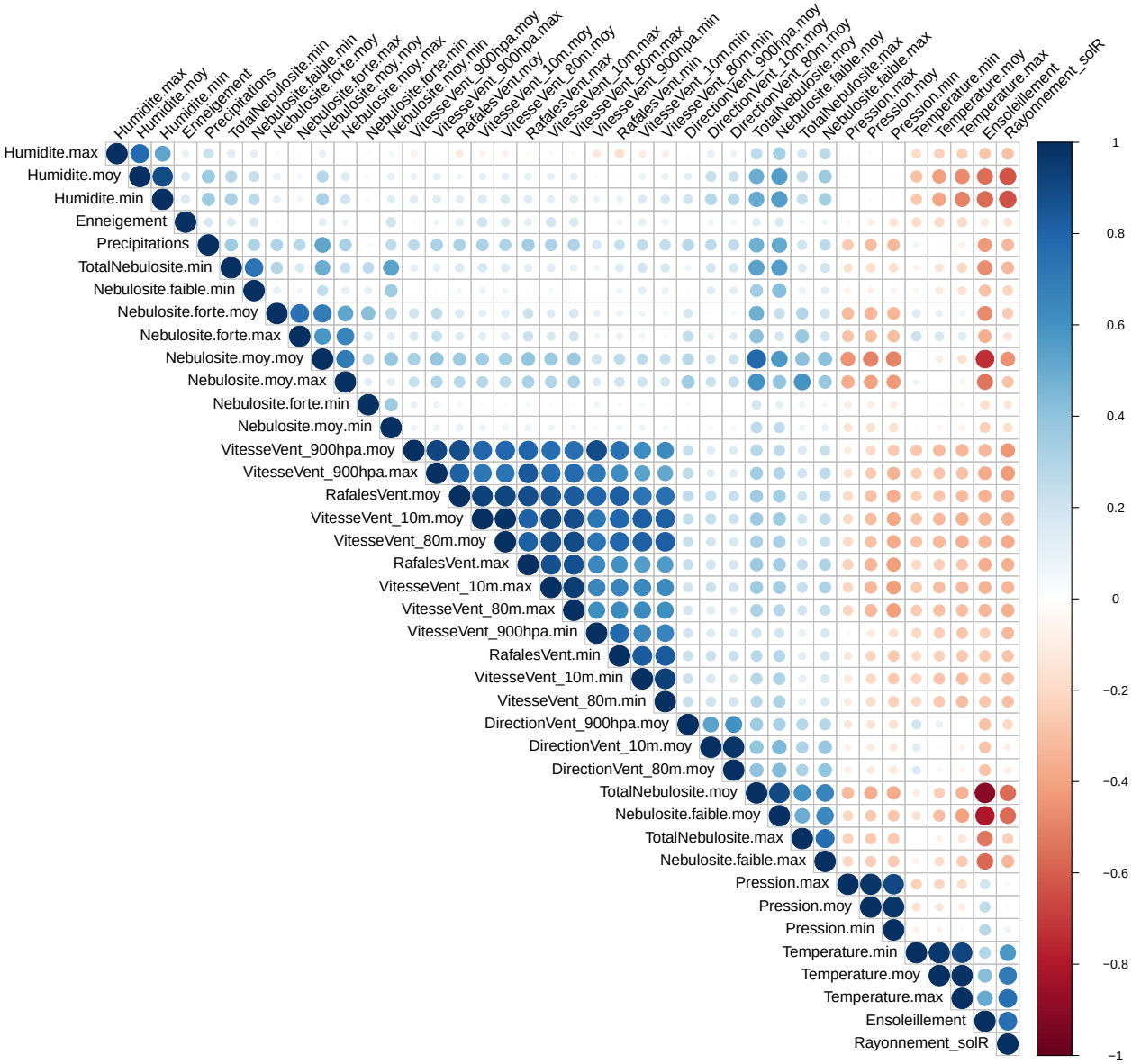
```
meteotest <- rename.variable(meteotest,    "Wind.Speed.daily.min..10.m.above.gnd.","VitesseVent_10m.min
 meteotest <- rename.variable(meteotest,   "Wind.Speed.daily.min..80.m.above.gnd.","VitesseVent_80m.min
meteotest <- rename.variable(meteotest,    "Wind.Speed.daily.min..900.mb.","VitesseVent_900hpa.min")
meteotest <- rename.variable(meteotest,
                             "X","X")
meteotest <- rename.variable(meteotest,
                             "Year","Annee")
# Suppression des variables Heure et Minute qui ont des valeurs nulles partout #
meteotrain <- meteotrain[,-c(5:6)]
meteotest <- meteotest[,-c(5:6)]

# Traitement des données manquantes #
sum(is.na.data.frame(meteotrain))
sum(is.na.data.frame(meteotest))
# Il n'y a pas de données manquantes, cela est parfait.

# Vérification de la classe de pluie.demain et conversion #
class(meteotrain$pluie.demain) # elle est de classe logique
meteotrain$pluie.demain= as.factor(meteotrain$pluie.demain)
class(meteotrain$pluie.demain)
```

Dans un premier temps, nous souhaitons voir s'il existe certaines corrélations entre les variables qui nous intéressent, étant donné que nous avons un très grand nombre de variables (> 40 variables). Pour ce faire, nous allons tracer le corrélogramme.

Les 4 premières variables sont des variables liées au temps (X, annee, jour, mois), n'ayant aucun lien avec les autres variables explicatives météorologiques. Il conviendrait de les exclure, mais nous vérifierons plus loin au préalable leur significativité.

Pour l'étude du corrélogramme, nous allons néanmoins les exclure.

I.2 Corrélogramme



Au vu du corrélogramme, nous nous apercevons entre autres que :

- Toutes les variables liées à la Vitesse du Vent (10m, 80m, et 900hPa) sont fortement corrélées aux Rafales vents.
- Les variables Nebulosite.moy.moy, Nebulosite.moy.max, Nebulosite.forte.moy et Nebulosite.forte.max sont aussi très corrélées.
- Les variables liées à l'Humidite sont aussi très corrélées entre elles.
- DirectionVent_10m_moy et DirectionVent_80m_moy sont très corrélées.
- Les variables TotalNebulosite_moy, Nebulosite_faible_moy, TotalNebulosite_max, Nebulosite_faible_max sont fortement corrélées.
- Ensoleillement et RayonnementsolR sont fortement corrélées.

- Les variables Temperature sont fortement corrélées, de même pour les variables Pression entre elles.

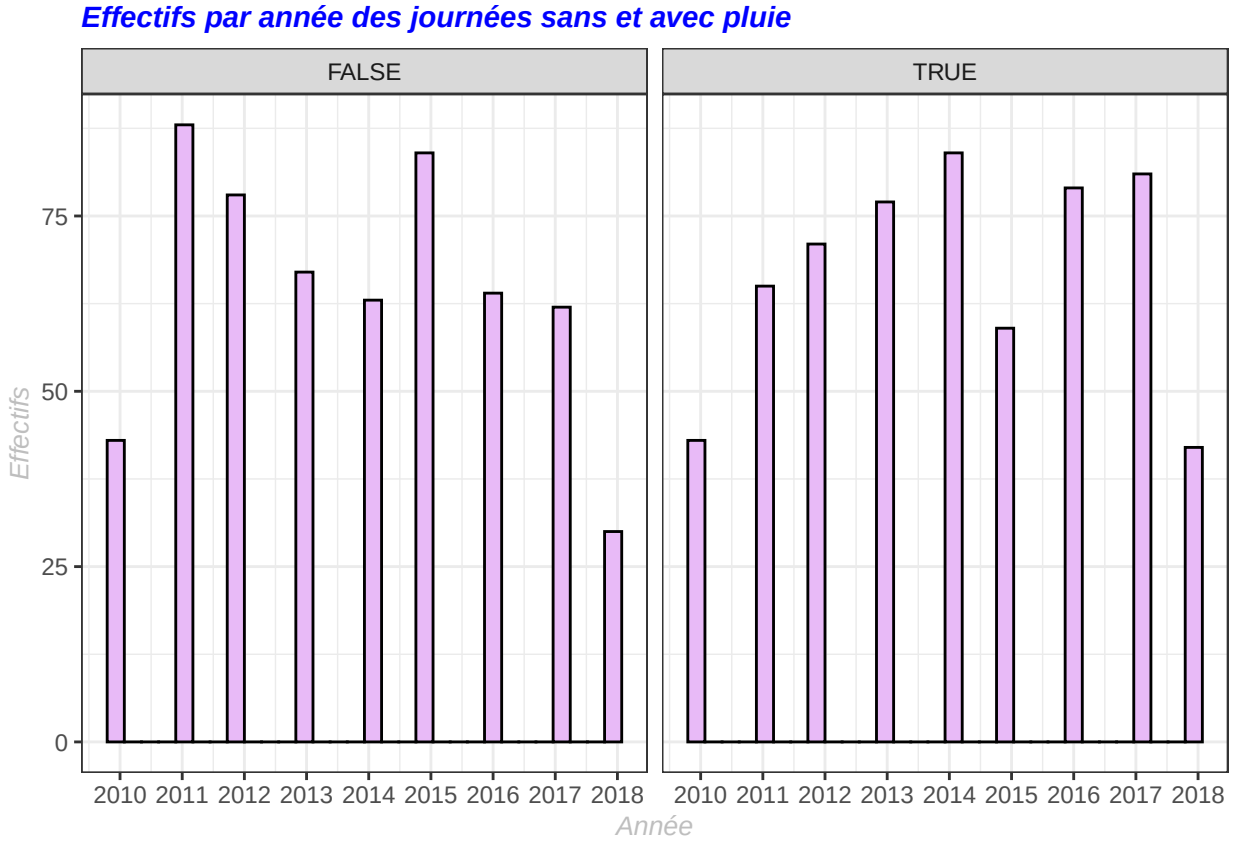
Aussi, nous pouvons voir que la Température max est corrélée avec l'ensoleillement et le rayonnement solaire, ce qui est assez évident.

I.3 Analyse graphique des données

Faisons succinctement une analyse graphique. Nous allons analyser séparément les journées avec et sans pluie en les mettant en comparaison.

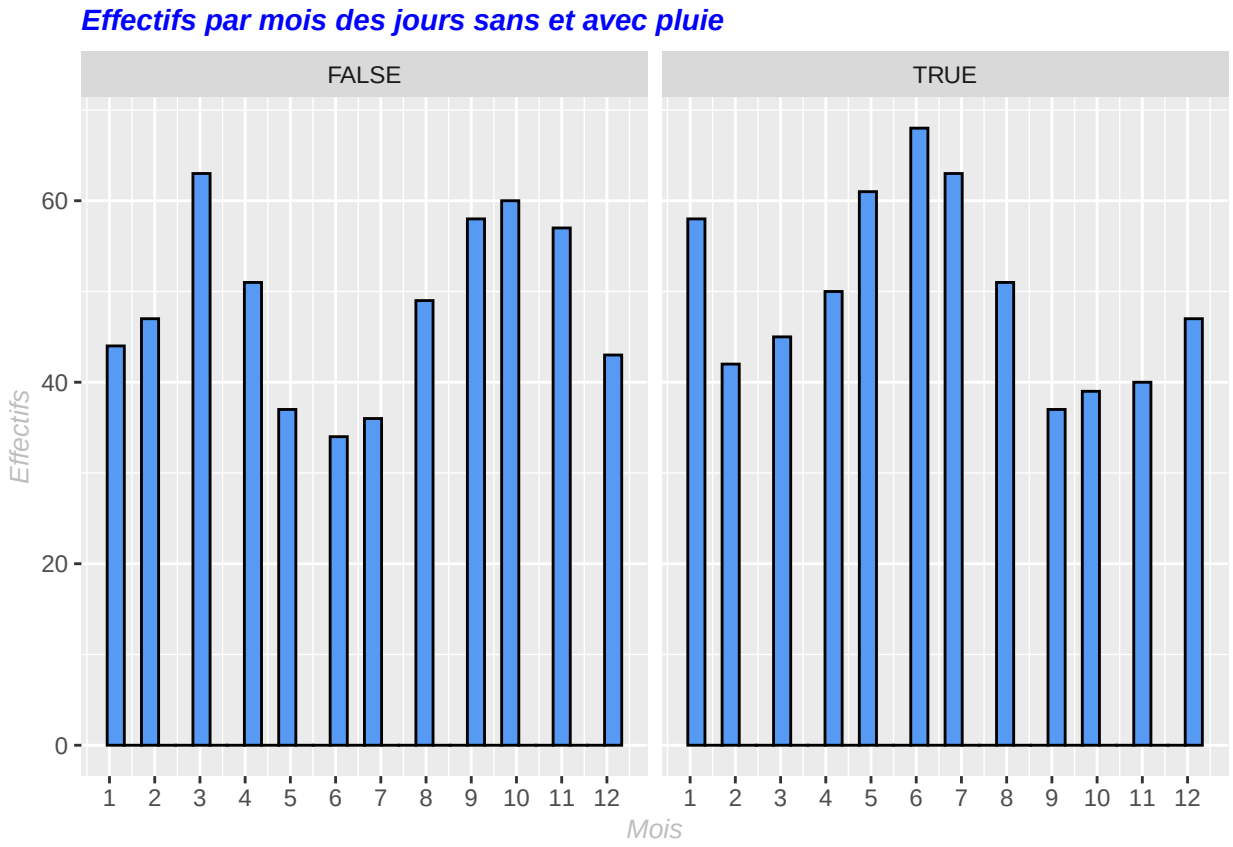
La répartition des journées sèches et pluvieuses selon les années et mois

```
ggplot(meteotrain) +
  aes(x = Annee) +
  geom_histogram(colour="black", fill="#E8BAF7") +
  scale_x_continuous(breaks=
        c(2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018),
        labels=c(2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018)) +
  ggtitle("Effectifs par année des journées sans et avec pluie") +
  xlab("Année") +
  ylab("Effectifs")+
  facet_wrap(~pluie.demain) +
  theme_bw() +
  theme(
    plot.title = element_text(face = "bold.italic",colour="blue",size=11),
    axis.title.x = element_text(face = "italic",colour = "grey",size = 10),
    axis.title.y  = element_text(face = "italic",colour = "grey",size = 10)
  )
```




```
ggplot(meteotrain) +
  aes(x = Mois) +
  geom_histogram(colour="black", fill="#569AF3") +
  scale_x_continuous(breaks=c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12),
                     labels=c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)) +
  ggtitle("Effectifs par mois des jours sans et avec pluie") +
  xlab("Mois") +
```

```
  ylab("Effectifs")+
  facet_wrap(~pluie.demain) +
  theme_gray() +
  theme(
    plot.title = element_text(face = "bold.italic",colour="blue",size=11),
    axis.title.x = element_text(face = "italic",colour = "grey",size = 10),
    axis.title.y  = element_text(face = "italic",colour = "grey",size = 10)
  )
```



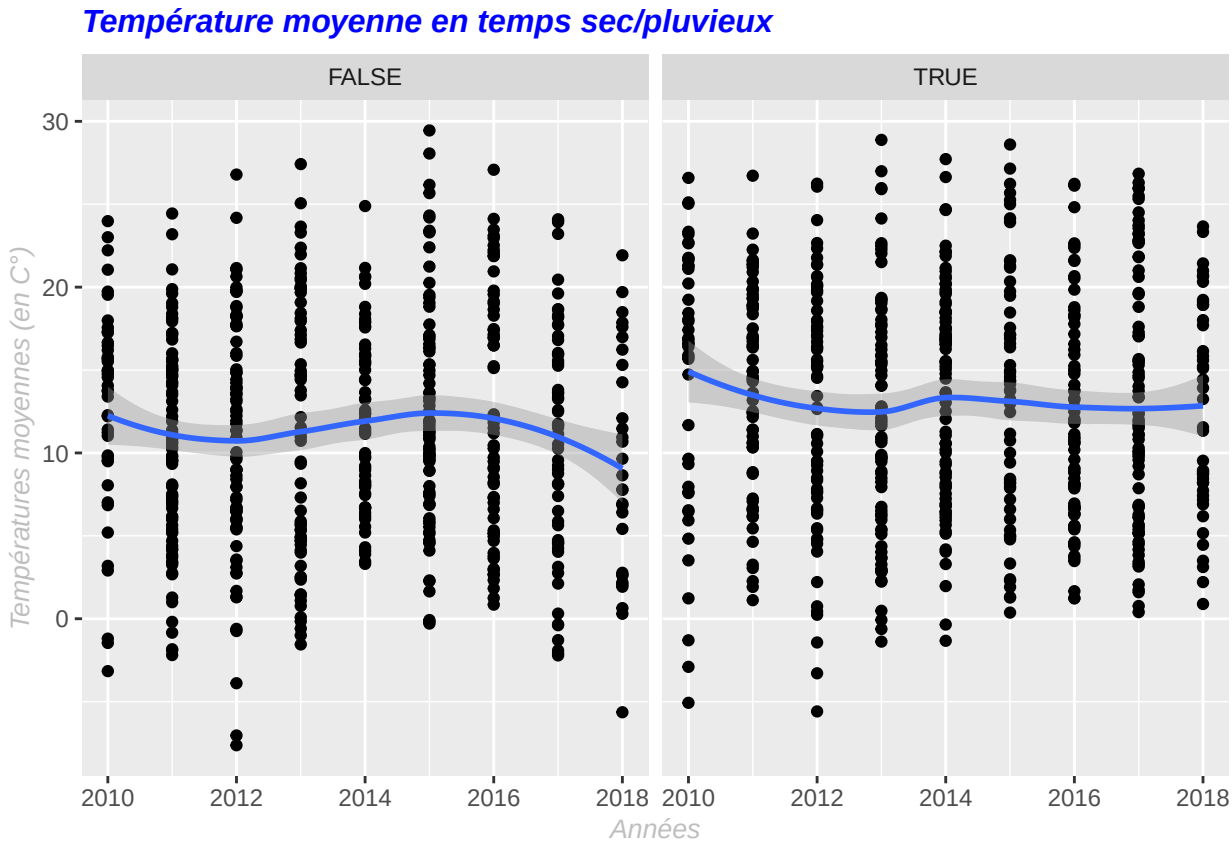
Les années 2014 et 2017 sont les années marquées par des journées de pluie plus nombreuses. Entre 2011 et 2014, les effectifs de journées de pluie n'ont cessé d'augmenter jusqu'à atteindre un pic en 2014. 2015 a été moins pluvieuse puis 2016 et 2017 sont à nouveau des années retrouvant des niveaux de pluie comme 2014. 2018 est quant à elle l'année avec la moins pluvieuse.

Les années 2011 et 2015 sont marquées par des journées sèches plus nombreuses. 2010 et 2018 sont celles connaissant le moins de journées sèches.

Les mois les plus pluvieux sont les mois de mai-juin-juillet pour toutes les années confondues. Et les mois de mars et octobre sont les mois les plus secs, pour toutes les années.

La Température moyenne en temps sans/avec pluie

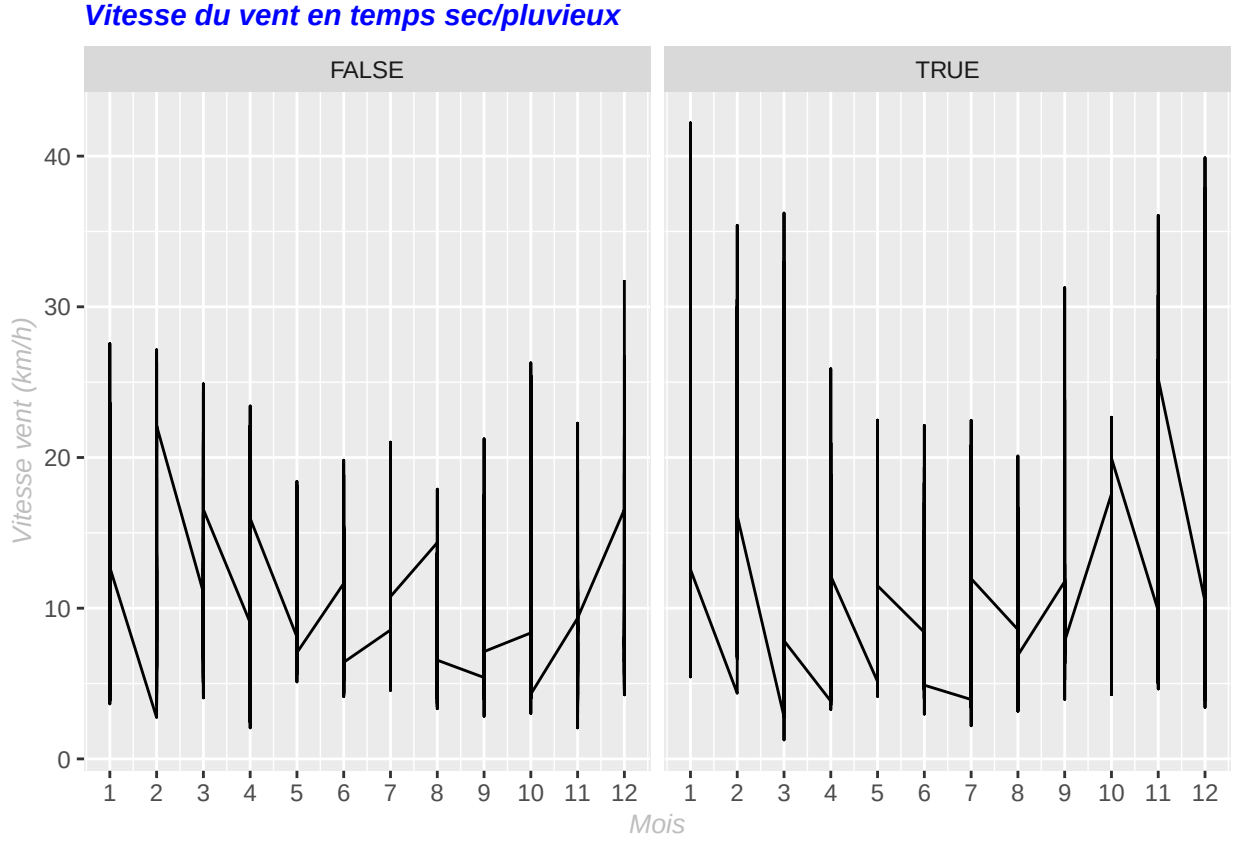
```
ggplot(meteotrain) +
  aes(x= Annee, y = Temperature.moy) +
  geom_point() +
  geom_smooth() +
  ggtitle("Température moyenne en temps sec/pluvieux") +
  xlab("Années") +
  ylab("Températures moyennes (en C°)")+
  facet_wrap(~pluie.demain) +
  theme_gray() +
  theme(
    plot.title = element_text(face = "bold.italic",colour="blue",size=12),
    axis.title.x = element_text(face = "italic",colour = "grey",size = 10),
    axis.title.y  = element_text(face = "italic",colour = "grey",size = 10)
  )
```



Les journées pluvieuses connaissent une température moyenne plus élevée en moyenne sur toutes les années de 2010 à 2018. (excepté en 2015-2016 où la température moyenne entre temps de pluie et temps sec semble identique).

Vitesse du vent en temps sans/avec pluie (VitesseVent_10m.moy)

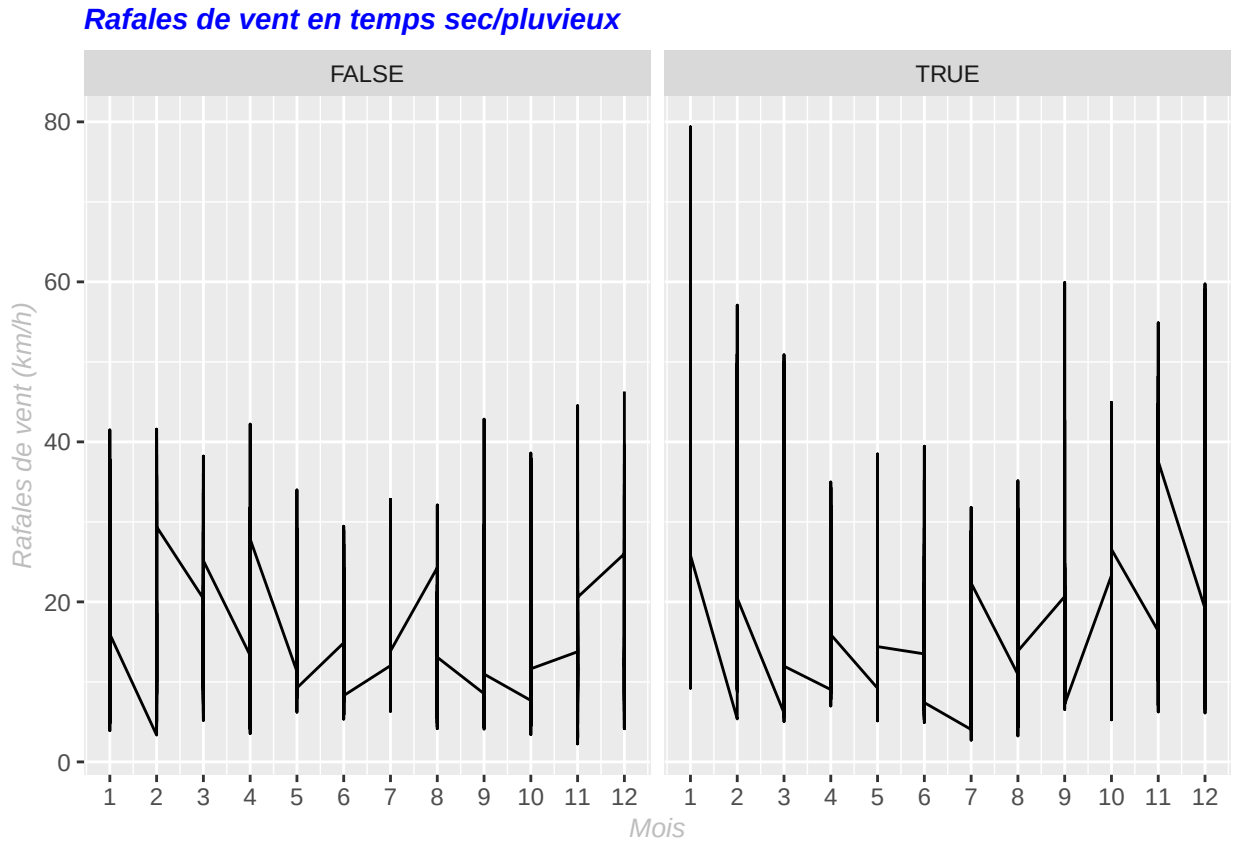
```
ggplot(meteotrain) +
  aes(x= Mois, y = VitesseVent_10m.moy) +
  geom_line() +
  scale_x_continuous(breaks=c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12),
                     labels=c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)) +
  ggtitle("Vitesse du vent en temps sec/pluvieux") +
  xlab("Mois") +
  ylab("Vitesse vent (km/h)")+
facet_wrap(~pluie.demain) +
  theme_gray() +
  theme(
    plot.title = element_text(face = "bold.italic",colour="blue",size=11),
    axis.title.x = element_text(face = "italic",colour = "grey",size = 10),
    axis.title.y  = element_text(face = "italic",colour = "grey",size = 10)
  )
```



En temps de pluie, la vitesse du vent est d'une plus grande intensité sur tous les mois de l'année. Nous notons néanmoins que les mois d'été et celui d'octobre sont légèrement moins marqués par cette hausse.

Rafales de vent en temps sans/avec pluie (RafalesVent.moy)

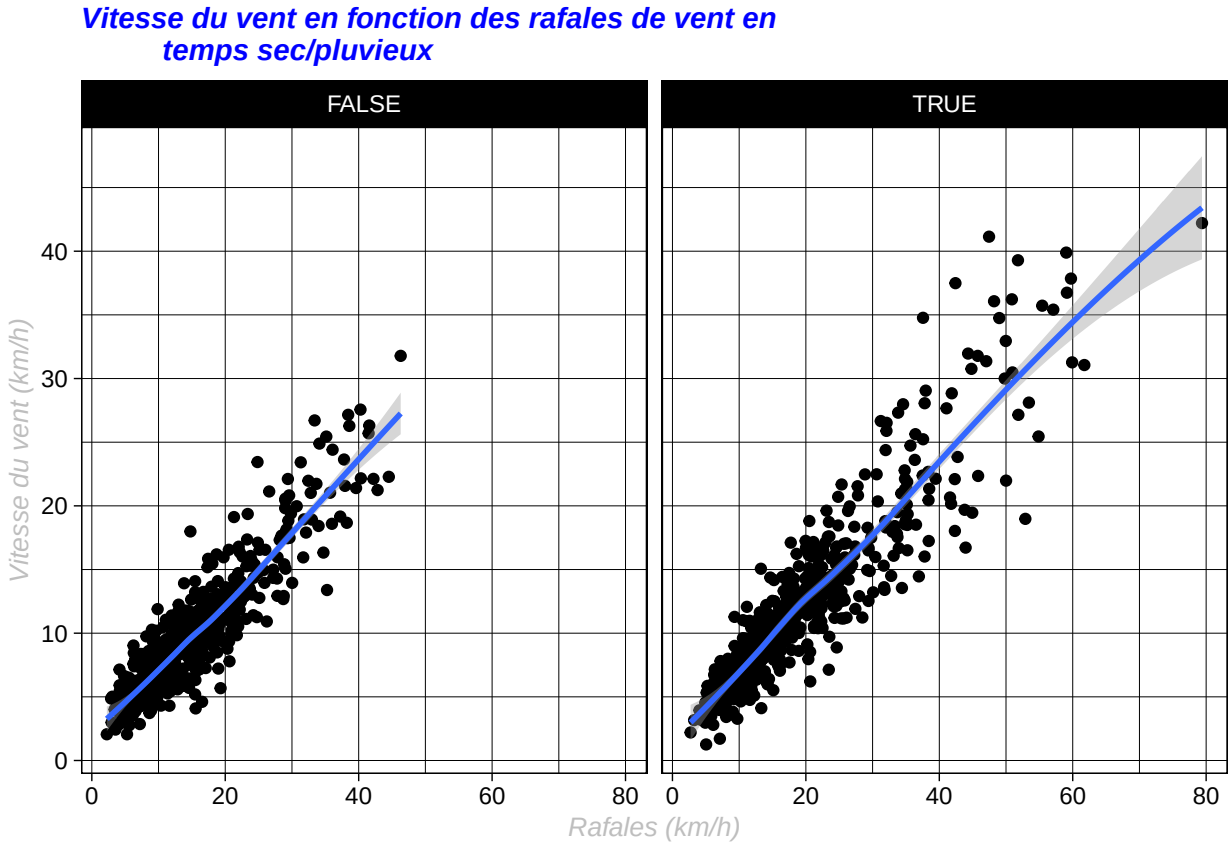
```
ggplot(meteotrain) +
  aes(x= Mois, y = RafalesVent.moy) +
  geom_line() +
  scale_x_continuous(breaks=c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12),
                     labels=c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)) +
  ggtitle("Rafales de vent en temps sec/pluvieux") +
  xlab("Mois") +
  ylab("Rafales de vent (km/h)") +
  facet_wrap(~pluie.demain) +
  theme_gray() +
  theme(
    plot.title = element_text(face = "bold.italic",colour="blue",size=11),
    axis.title.x = element_text(face = "italic",colour = "grey",size = 10),
    axis.title.y  = element_text(face = "italic",colour = "grey",size = 10)
  )
```



De la même manière que la vitesse du vent, les rafales sont largement intensifiées en temps pluvieux sur toute l'année hormis aux mois d'avril et juillet.

La Vitesse du vent et les Rafales de Vent en temps sans/avec pluie

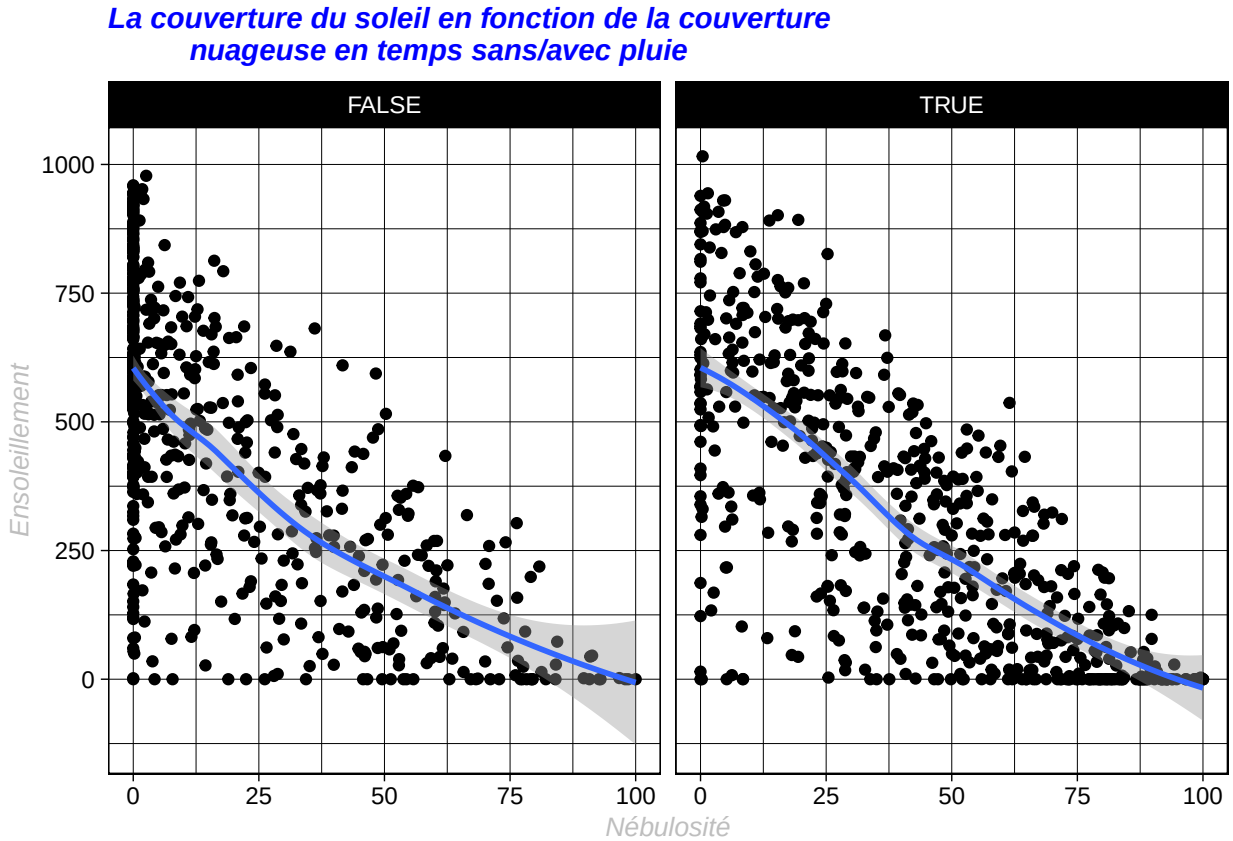
```
qplot(RafalesVent.moy,VitesseVent_10m.moy , data = meteotrain) +
  geom_smooth() +
  ggtitle("Vitesse du vent en fonction des rafales de vent en
          temps sec/pluvieux") +
  xlab("Rafales (km/h)") +
  ylab("Vitesse du vent (km/h)") +
  facet_wrap(~ pluie.demain)+
  theme_linedraw() +
  theme(
    plot.title = element_text(face = "bold.italic",colour="blue",size=11),
    axis.title.x = element_text(face = "italic",colour = "grey",size = 10),
    axis.title.y  = element_text(face = "italic",colour = "grey",size = 10)
  )
```



Nous voyons que lorsqu'il pleut les rafales de vents sont plus importantes, s'accompagnant d'une vitesse beaucoup plus élevée. Par ailleurs, nos graphiques démontrent que la vitesse du vent croît de façon proportionnelle aux rafales de vent, et ce, qu'il fasse beau ou mauvais temps (sec ou pluvieux). Ceci nous confirme bien la corrélation importante observée sur le corrélogramme entre les variables VitesseVent et RafalesVent.

La Nébulosité moyenne et l'Ensoleillement par temps sans/avec pluie

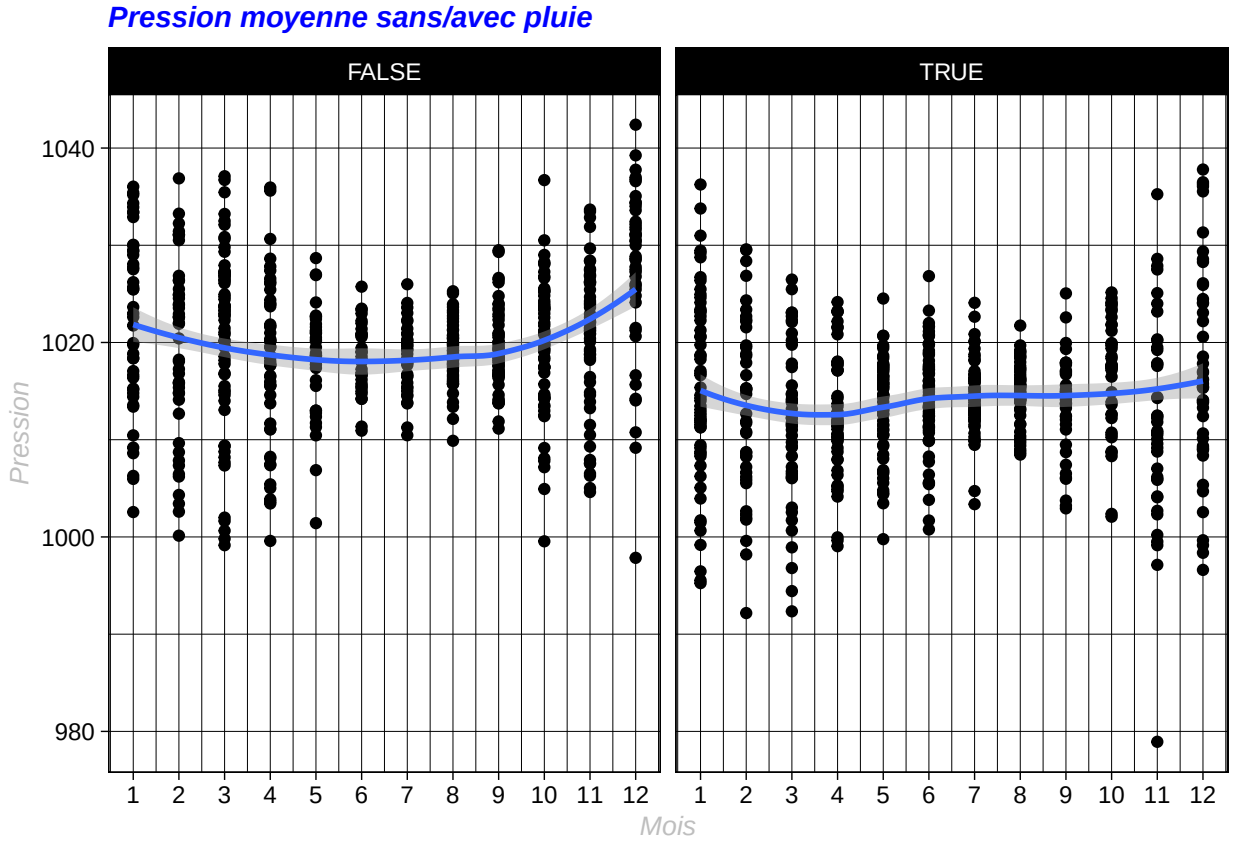
```
qplot(Nebulosite.moy.moy,Ensoleillement , data = meteotrain) +
  geom_smooth() +
  ggtitle("La couverture du soleil en fonction de la couverture
          nuageuse en temps sans/avec pluie") +
  xlab("Nébulosité") +
  ylab("Ensoleillement") +
  facet_wrap(~ pluie.demain)+
  theme_linedraw() +
  theme(
    plot.title = element_text(face = "bold.italic",colour="blue",size=11),
    axis.title.x = element_text(face = "italic",colour = "grey",size = 10),
    axis.title.y  = element_text(face = "italic",colour = "grey",size = 10)
  )
```



Nous voyons que la plage d'ensoleillement diminue lorsque la couverture nuageuse augmente.

La pression en temps sans/avec pluie

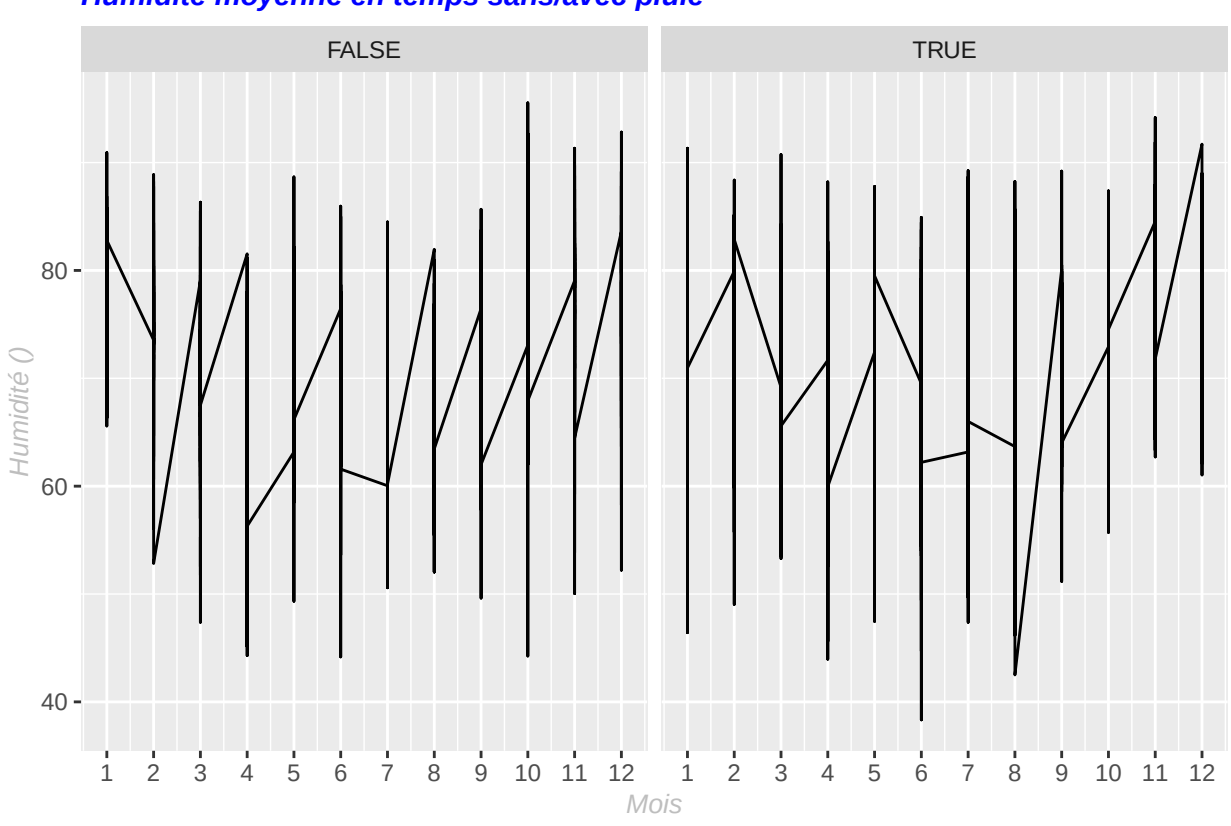
```
ggplot(meteotrain) +
  aes(x= Mois, y = Pression.moy) +
  geom_point() +
  geom_smooth()+
  scale_x_continuous(breaks=c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12),
                     labels=c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)) +
  ggtitle("Pression moyenne sans/avec pluie") +
  xlab("Mois") +
  ylab("Pression")+
  facet_wrap(~pluie.demain) +
  theme_linedraw() +
  theme(
    plot.title = element_text(face = "bold.italic",colour="blue",size=11),
    axis.title.x = element_text(face = "italic",colour = "grey",size = 10),
    axis.title.y  = element_text(face = "italic",colour = "grey",size = 10)
  )
```



La pression moyenne est plus importante en temps sec, qu'en temps pluvieux.

Humidité par temps sans/avec pluie

```
ggplot(meteotrain) +
  aes(x= Mois, y = Humidite.moy) +
  geom_line() +
  scale_x_continuous(breaks=c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12),
                     labels=c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)) +
  ggtitle("Humidité moyenne en temps sans/avec pluie") +
  xlab("Mois") +
  ylab("Humidité ()") +
  facet_wrap(~pluie.demain) +
  theme_gray() +
  theme(
    plot.title = element_text(face = "bold.italic",colour="blue",size=11),
    axis.title.x = element_text(face = "italic",colour = "grey",size = 10),
    axis.title.y  = element_text(face = "italic",colour = "grey",size = 10)
  )
```



II. Approche par échantillon de validation (apprentissage/validation)

II.1 Partitionnement de Meteotrain (80/20)

Avant tout, nous allons partitionner notre échantillon Meteotrain en 2 avec 80% d'échantillon d'apprentissage et 20% d'échantillon de validation. La première servira à l'ajustement, la seconde à valider le modèle.

```
set.seed(12345) # nous fixons le seed pour que ce soit reproductible
training = sample(c(T, F), nrow(meteotrain), replace = T, prob = c(.8, .2)) # construction des index al

meteotrain$pluie.demain=as.factor(ifelse(meteotrain$pluie.demain=="TRUE",1,0))

tr.meteotrain = meteotrain[training,]
# notre échantillon d'apprentissage a 948 lignes
test.meteotrain = meteotrain[!training,]
# notre échantillon de validation a 232 lignes
```

Avant la mise en oeuvre de la sélection de modèles proprement dite, arrêtons-nous un moment sur les variables temporelles énoncées plus haut, et observons leur significativité par rapport au modèle saturé initial.

```
lg.modelsat = glm(
 pluie.demain ~ ., family = binomial,data = tr.meteotrain)

summary(lg.modelsat)
```

Ce modèle saturé est, par définition, le modèle qui ajuste le plus "parfaitement" possible les données.

Test de déviance de modèles emboités sur les 4 variables temporelles

```
lg.modeljour = glm(pluie.demain~. -Jour, family = binomial,
data = tr.meteotrain)
anova(lg.modelsat,lg.modeljour,test = "LRT")

lg.modelmois = glm(pluie.demain~. -Jour-Mois, family = binomial,
                   data = tr.meteotrain)
anova(lg.modelsat,lg.modelmois, test = "LRT")
```

```
lg.modelannee = glm(pluie.demain~. -Jour-Mois-Annee, family = binomial,
                    data = tr.meteotrain)
anova(lg.modelsat,lg.modelannee, test = "LRT")

lg.modelX = glm(pluie.demain~. -Jour-Mois-Annee-X, family = binomial,
                data = tr.meteotrain)
anova(lg.modelsat,lg.modelX, test = "LRT")
```

Tous les tests montrent que la p-value est supérieure à 0,05.
Cela nous amène à toujours considérer que le modèle réduit est préféré (l'hypothèse H0 de nullité des coefficients n'est pas rejetée). Nous allons donc les exclure pour la suite dans la tr.meteotrain (base d'apprentissage).

```
# Redéfinition de notre data set #
tr.meteotrain1 <- tr.meteotrain[,-c(1:4)]
```

Notre base de données ne contient plus que 41 variables.

II.2 Sélection de variables

```
# Spécifions de nouveau le modèle saturé lg.modelsat et
# le modèle de départ lg.modelnull  (qui seront les références)
lg.modelsat = glm(
 pluie.demain ~ ., family = binomial,data = tr.meteotrain1)

lg.modelnull = glm(
 pluie.demain ~ 1, family = binomial,data = tr.meteotrain1)
```

II.2.1 Modèle saturé

Notre premier modèle lg.modelsat est le modèle complet qui nous servira de référence.

```
summary(lg.modelsat)
lg.modelsat$deviance
lg.modelsat$null.deviance
lg.modelsat$aic
```

Celui-ci a un critère AIC à 1066.8 et une déviance à 984.8

II.2.2 Modèles avec Best Subset Selection

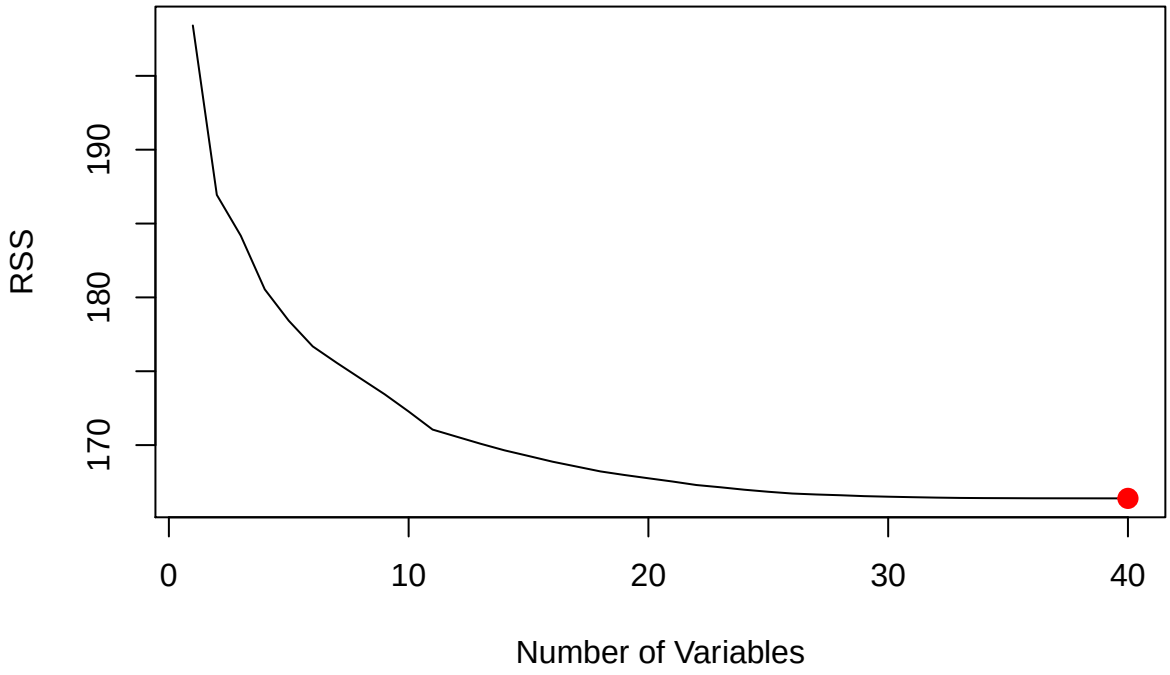
Commençons par la méthode de recherche exhaustive avec les critères (RSS, R2 ajusté, Cp et Bic). Le critère R2 ne serait pas d'une utilité ici, car, nous sommes en présence d'un jeu de grande dimension, le R2 sélectionnera a fortiori le modèle à 40 variables, puisque, par définition, le meilleur R2 sera celui possédant le maximum de variables.
Nous allons ainsi appliquer la méthode en sélectionnant que le meilleur modèle (nbest=1) par taille de modèles. (ex : le meilleur modèle pour les modèles de dimension 1 variable, puis le meilleur modèle parmi les modèles de dimension 2 etc..)

```
# Méthode Best subset selection
library(leaps)
lg.modelbsubset <- regsubsets(pluie.demain ~., tr.meteotrain1,
                              nbest=1, int=T,nvmax = 40)
lg.summary=summary(lg.modelbsubset)

with(lg.summary,data.frame(rsq,adjr2,cp,rss,bic,outmat))
quartz()
```

Nous pouvons observer pour chaque critère, le nombre et les variables sélectionnées, pour analyser au mieux la pertinence de ces modèles.

RSS

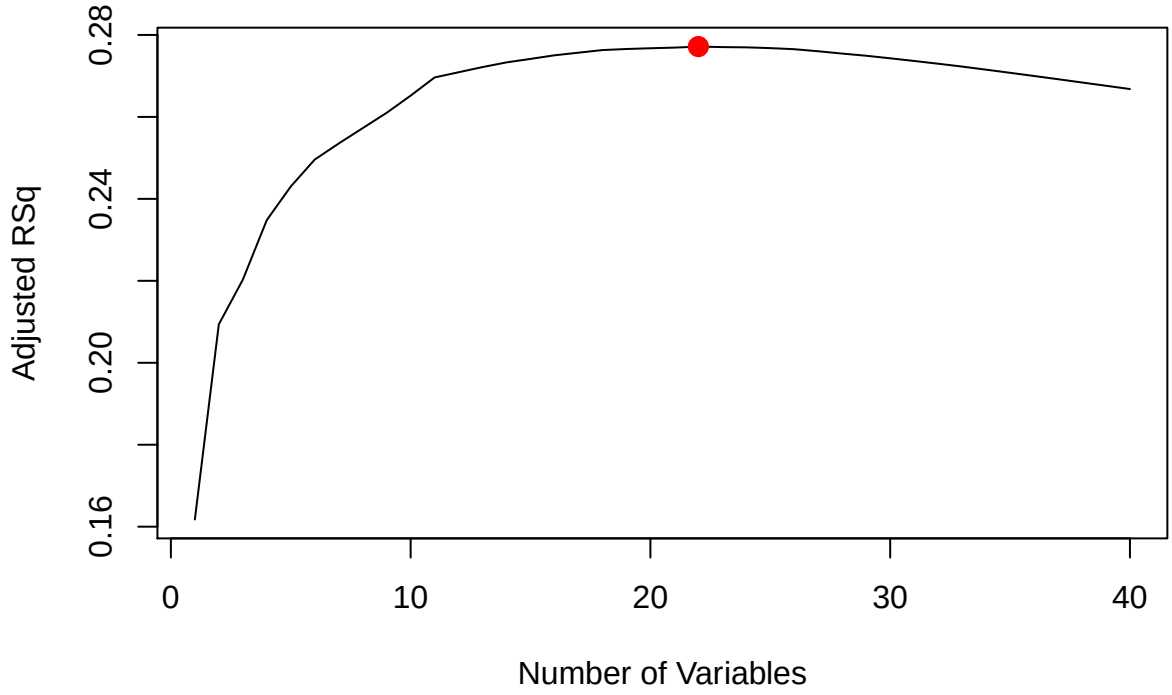
```
which.min(lg.summary$rss)
plot(lg.summary$rss , xlab =" Number of Variables " , ylab =" RSS " ,type ="l")
points(which.min(lg.summary$rss), lg.summary$rss[which.min(lg.summary$rss)],
       col =" red " , cex =2 , pch =20)
```



Le critère RSS avec le minimum souhaité semble être le modèle avec le maximum de variables (le point rouge indiquant là où le RSS atteint son minimum). Cependant, nous pouvons noter que la décroissance du RSS est assez abrupte de 0 à 10 variables environ ; à partir de 11 variables ajoutées au modèle, le RSS est quasiment au plus bas de son niveau, jusqu'à 40 variables. Nous pourrions donc, suivant ce critère, nous focaliser sur un modèle contenant entre 10 et 20 variables, prenons 18 variables par exemple.

R2 Ajusté

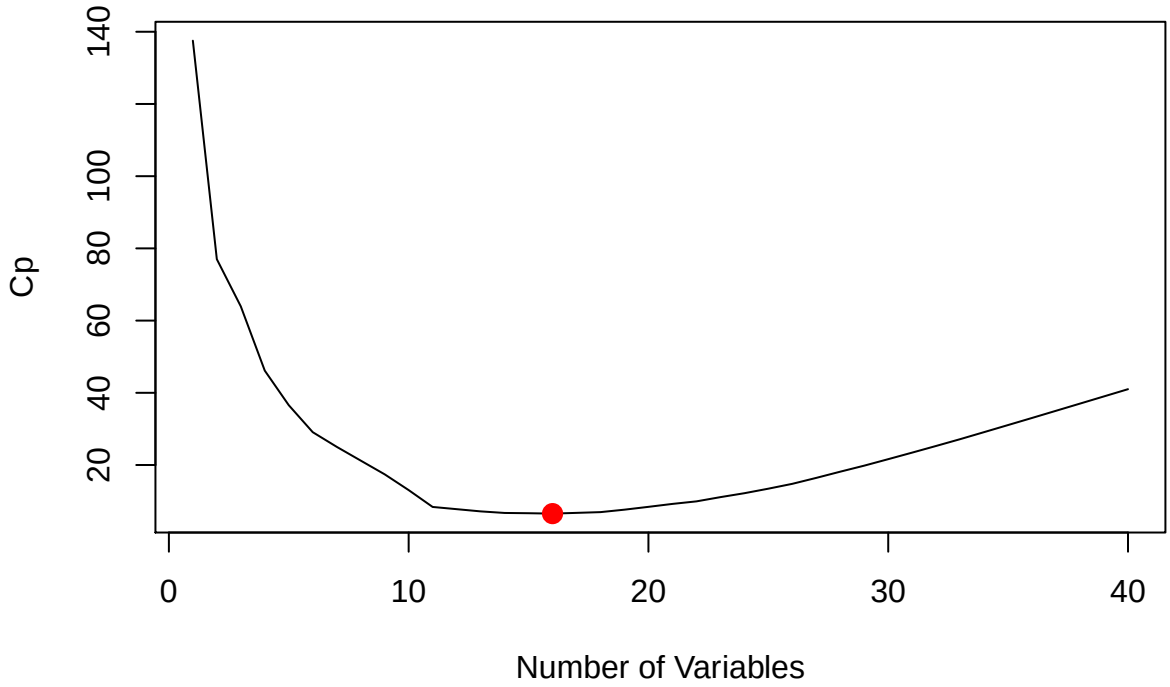
```
which.max(lg.summary$adjr2)
plot(lg.summary$adjr2 , xlab =" Number of Variables " ,
     ylab =" Adjusted RSq " , type ="l")
points(which.max(lg.summary$adjr2),
       lg.summary$adjr2[which.max(lg.summary$adjr2)],
       col =" red " , cex =2 , pch =20)
```



Avec ce critère, le modèle ayant le R2 ajusté maximum est un modèle avec 22 variables.

Cp de Mallows

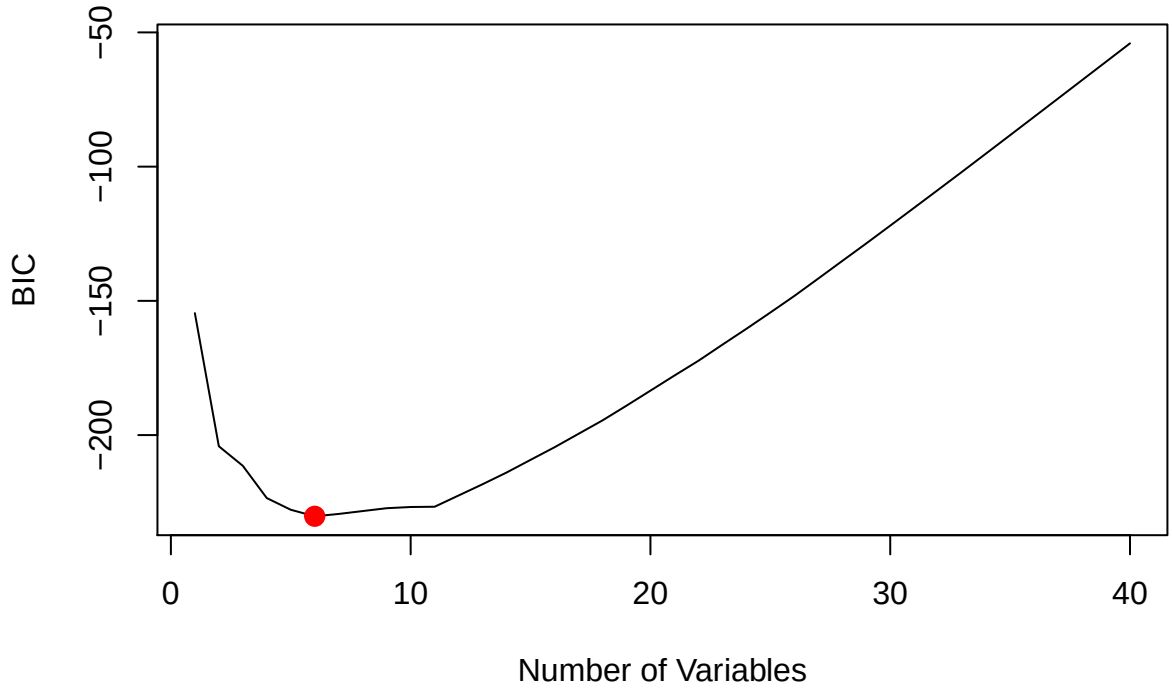
```
which.min(lg.summary$cp)
plot(lg.summary$cp , xlab =" Number of Variables " ,
     ylab =" Cp " ,type ="l")
points(which.min(lg.summary$cp), lg.summary$cp[which.min(lg.summary$cp)],
       col =" red " , cex =2 , pch =20)
```



Avec le critère Cp, le modèle ayant le Cp minimum est un modèle à 16 variables.

Bic

```
which.min(lg.summary$bic)
plot(lg.summary$bic , xlab =" Number of Variables " ,
     ylab =" BIC " ,type ="l")
points(which.min(lg.summary$bic), lg.summary$bic[which.min(lg.summary$bic)],
       col =" red " , cex =2 , pch =20)
```



Le critère Bic issu de cette méthode exhaustive est négatif. Nous ne savons pas si cela est normal ou pas. Cela dit, le modèle ayant le Bic le plus petit est un modèle parcimonieux à 6 variables.

Selon chaque critère, nous pouvons voir quelles sont les variables retenues.

```
plot(lg.modelbsubset,scale= "r2")
```

```
plot(lg.modelbsubset,scale="adjr2")
```

```
plot(lg.modelbsubset,scale="Cp")
```

```
plot(lg.modelbsubset,scale= "bic")
```

En fonction du critère, les modèles et variables retenus diffèrent :

-RSS : le meilleur modèle est à 40 variables. D'après nos remarques précédentes, si nous devions nous baser sur ce critère, nous retiendrons plutôt aux alentours de 18 variables (les 18 premières ajoutées et l'intercept). D'après le graphe, en effet, nous pouvons confirmer ce choix, en constatant que le R2 semble stagner autour de 0.3 à partir du modèle à 18 variables. Ce modèle serait le suivant :

Modèle lg.BSSRSSModel

```
coef(lg.modelbsubset,18,scale="r2")
lg.BSSRSSModel <- glm(pluie.demain~Pression.moy+
                        Precipitations+
                        Enneigement+
                        Nebulosite.moy.moy+
                        Ensoleillement+
                        VitesseVent_80m.moy+
                        DirectionVent_80m.moy+
                        DirectionVent_900hpa.moy+
                        Temperature.max+
                        Pression.max+
                        Pression.min+
                        TotalNebulosite.min+
                        Nebulosite.forte.max+
                        Nebulosite.moy.max+
                        Nebulosite.faible.max+
                        VitesseVent_10m.max+
                        VitesseVent_10m.min+
                        RafalesVent.max,
                      family = binomial,
                      data = tr.meteotrain1)
summary(lg.BSSRSSModel)
AIC(lg.BSSRSSModel)
BIC(lg.BSSRSSModel)
```

Il affiche un :

- AIC = 1034.5
- BIC = 1126.7

Les variables sélectionnées n'apparaissent pas toutes significatives au sens de Wald. Seules 11 le sont.

-R2 ajusté : le meilleur modèle est à 22 variables et serait le suivant :
Modèle lg.BSSR2ADJModel

```
coef(lg.modelbsubset,22,scale="r2adj")
lg.BSSR2ADJModel <- glm(pluie.demain~Pression.moy+
                        Precipitations+
                        Enneigement+
                        Nebulosite.moy.moy+
                        Ensoleillement+
                        VitesseVent_10m.moy+
                        VitesseVent_80m.moy+
                        DirectionVent_80m.moy+
                        DirectionVent_900hpa.moy+
                        RafalesVent.moy+
                        Temperature.max+
                        Temperature.min+
```

```
                          Humidite.max+
                          Pression.max+
                          Pression.min+
                          TotalNebulosite.min+
                          Nebulosite.forte.max+
                          Nebulosite.moy.max+
                          Nebulosite.faible.max+
                          VitesseVent_10m.max+
                          VitesseVent_10m.min+
                          VitesseVent_900hpa.max,
                        family = binomial,
                        data = tr.meteotrain1)

summary(lg.BSSR2ADJModel)
AIC(lg.BSSR2ADJModel)
BIC(lg.BSSR2ADJModel)
```

Il affiche un :
- AIC = 1036.9
- BIC = 1148.6

Les variables sélectionnées n'apparaissent pas toutes significatives au sens de Wald. Seules 11 le sont (comme au modèle précédent).

- Cp de Mallows : le meilleur modèle est à 16 variables. Celui-ci correspond à :
Modèle lg.BSSCpModel

```
coef(lg.modelbsubset,16,scale="cp")

  lg.BSSCpModel <- glm(pluie.demain~Pression.moy+
                          Precipitations+
                          Enneigement+
                          Nebulosite.moy.moy+
                          VitesseVent_80m.moy+
                          DirectionVent_80m.moy+
                          DirectionVent_900hpa.moy+
                          Temperature.max+
                          Pression.max+
                          Pression.min+
                          Nebulosite.forte.max+
                          Nebulosite.moy.max+
                          Nebulosite.faible.max+
                          VitesseVent_10m.max+
                          VitesseVent_10m.min+
                          RafalesVent.max,
                        family = binomial,
                        data = tr.meteotrain1)
summary(lg.BSSCpModel)
AIC(lg.BSSCpModel)
BIC(lg.BSSCpModel)
```

Il affiche un :
- AIC = 1034
- BIC = 1116.5

Les 11 même variables significatives précédentes sont encore sélectionnées et pparaissent significatives au sens de Wald.

-Bic : le meilleur modèle est à 6 variables, et est celui-ci :
Modèle lg.BSSBicModel

```
coef(lg.modelbsubset,6,scale="bic")

lg.BSSBicModel <- glm(pluie.demain~
                        Nebulosite.moy.moy+
                        Temperature.max+
                        Pression.min+
                        Nebulosite.moy.max+
                        Nebulosite.faible.max+
                        RafalesVent.max,
                      family = binomial,
                      data = tr.meteotrain1)
summary(lg.BSSBicModel)
AIC(lg.BSSBicModel)
BIC(lg.BSSBicModel)
```

Il affiche un :
- AIC = 1057.1
- BIC = 1091
==> ici, nous nous apercevons que ce n'est pas le même Bic qui a été observé lors de l'utilisation de la regsubset... Aussi, nous nous interrogeons sur la pertinence de l'utilisation de cette méthode de recherche exhaustive sur nos données. Est-ce nos variables qui sont trop nombreuses et inappropriées à cette fonction ? Est-ce la fonction regsubset qui est inadaptée à la régression logistique ? Sans réponse, à ce stade, nous préférons ne pas nous avancer avec les modèles choisis par cette approche et regarderons plutôt une autre méthode.

A présent, nous allons voir la méthode pas à pas avec les critères AIC et BIC.

II.2.3 Modèles pas à pas : Forward, Backward, Stepwise (indice AIC)

Modèle lg.FWDAicModel

```
library(MASS)
lg.fwdaic.model <- stepAIC(lg.modelnull,scope=list(lower=lg.modelnull,
                                                   upper=lg.modelsat),
data = tr.meteotrain1, direction = "forward")

coef(lg.fwdaic.model,which.min(summary(lg.fwdaic.model)$aic))
```

Le modèle lg.FwdAicModel retenu est celui-ci :

```
lg.FWDAicModel <- glm(formula = pluie.demain ~
                        Nebulosite.moy.max +
                        Pression.min +
                        Temperature.min +
                        RafalesVent.max +
                        Nebulosite.faible.max +
                        Nebulosite.forte.max +
                        Temperature.max +
                        DirectionVent_900hpa.moy +
                        TotalNebulosite.min +
                        DirectionVent_80m.moy +
                        Pression.moy +
                        Pression.max +
                        Nebulosite.moy.moy, family = binomial, data = tr.meteotrain1)

summary(lg.FWDAicModel)
AIC(lg.FWDAicModel)
```

Il a 13 variables et un AIC = 1045.2

Modèle lg.BWDAicModel

```
lg.bwdaic.model <- stepAIC(lg.modelsat,data = tr.meteotrain1,
                           direction = "backward")

coef(lg.bwdaic.model,which.min(summary(lg.bwdaic.model)$aic))
```

Le modèle retenu est le suivant :

```
 lg.BWDAicModel <- glm(formula = pluie.demain ~
                         Pression.moy +
                         Precipitations +
                         Enneigement +
                         Nebulosite.moy.moy +
                         VitesseVent_80m.moy +
                         DirectionVent_80m.moy +
                         DirectionVent_900hpa.moy +
                         Temperature.max +
                         Temperature.min +
                         Pression.max +
                         Pression.min +
                         TotalNebulosite.min +
                         Nebulosite.forte.max +
                         Nebulosite.moy.max +
                         Nebulosite.faible.max +
                         VitesseVent_10m.max +
                         VitesseVent_10m.min, family = binomial,
```

```
data = tr.meteotrain1)

summary(lg.BWDAicModel)
AIC(lg.BWDAicModel)
```

Le modèle lg.BWDAicModel retenu abouti à un AIC = 1033.7 et 17 variables.

==> a priori, les 2 méthodes ne retiennent pas exactement les mêmes variables. La Backward semble rajouter des variables par rapport à la Forward. En terme de critère AIC, c'est la Backward qui est préférée pour le moment, même si toutes les variables sélectionnées n'apparaissent pas toutes significatives au sens de Wald.

Modèle lg.STEPWAicModel

```
lg.stepwaic.model <- stepAIC(lg.modelnull,
                             scope=list(upper=lg.modelsat),
data = tr.meteotrain1, direction = "both")

coef(lg.stepwaic.model,which.min(summary(lg.stepwaic.model)$aic))
```

Le modèle retenu est celui-ci :

```
lg.STEPWAicModel <- glm(formula = pluie.demain ~
                          Nebulosite.moy.max +
                          Pression.min +
                          RafalesVent.max +
                          Nebulosite.faible.max +
                          Temperature.max +
                          DirectionVent_900hpa.moy +
                          TotalNebulosite.min +
                          DirectionVent_80m.moy +
                          Pression.moy +
                          Pression.max +
                          Nebulosite.moy.moy, family = binomial,
data = tr.meteotrain1)

summary(lg.STEPWAicModel)
AIC(lg.STEPWAicModel)
```

Le modèle lg.STEPWAicModel retenu abouti à un modèle avec un AIC = 1043.9 et 11 variables.

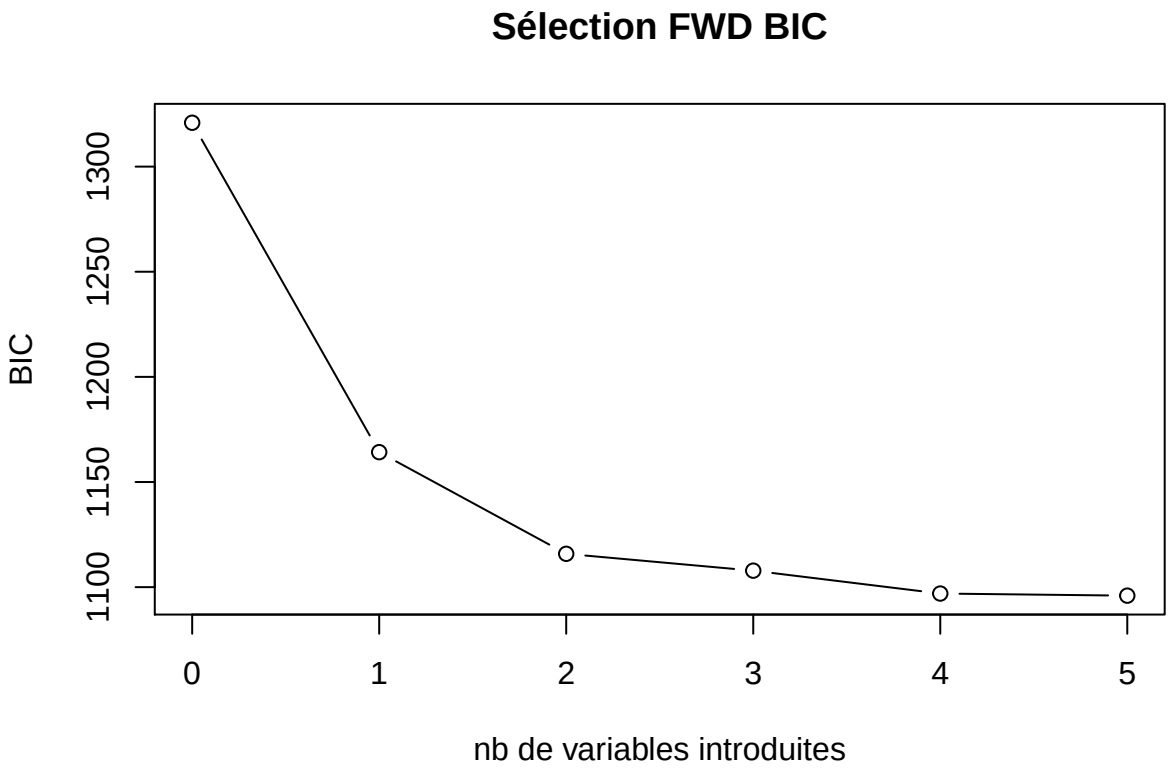
==> l'approche qui obtient l'AIC minimum est la Backward selection, en sélectionnant 17 variables dans son modèle.

Choisissons un autre critère (Bic), pour construire d'autres modèles pas à pas.

II.2.4 Modèles pas à pas : Forward, Backward, Stepwise (indice Bic)

Modèle lg.FWDBicModel

```
lg.fwdbic.model <- stepAIC(lg.modelnull,scope=list(lower=lg.modelnull,
                                                   upper=lg.modelsat),
data = tr.meteotrain1, direction = "forward", k=log(nrow(tr.meteotrain1)))

summary(lg.fwdbic.model)


coef(lg.fwdbic.model,which.min(summary(lg.fwdbic.model)$aic))
print(lg.fwdbic.model$anova)

plot(0:(nrow(lg.fwdbic.model$anova)-1),lg.fwdbic.model$anova[,"AIC"],type = "b",xlab = "nb de variables
     main = "Sélection FWD BIC")
```



Dès la 3ème variable ajoutée, on voit qu'on améliore assez peu le BIC. Le modèle lg.FWDBicModel retenu a 5 variables, c'est le suivant :

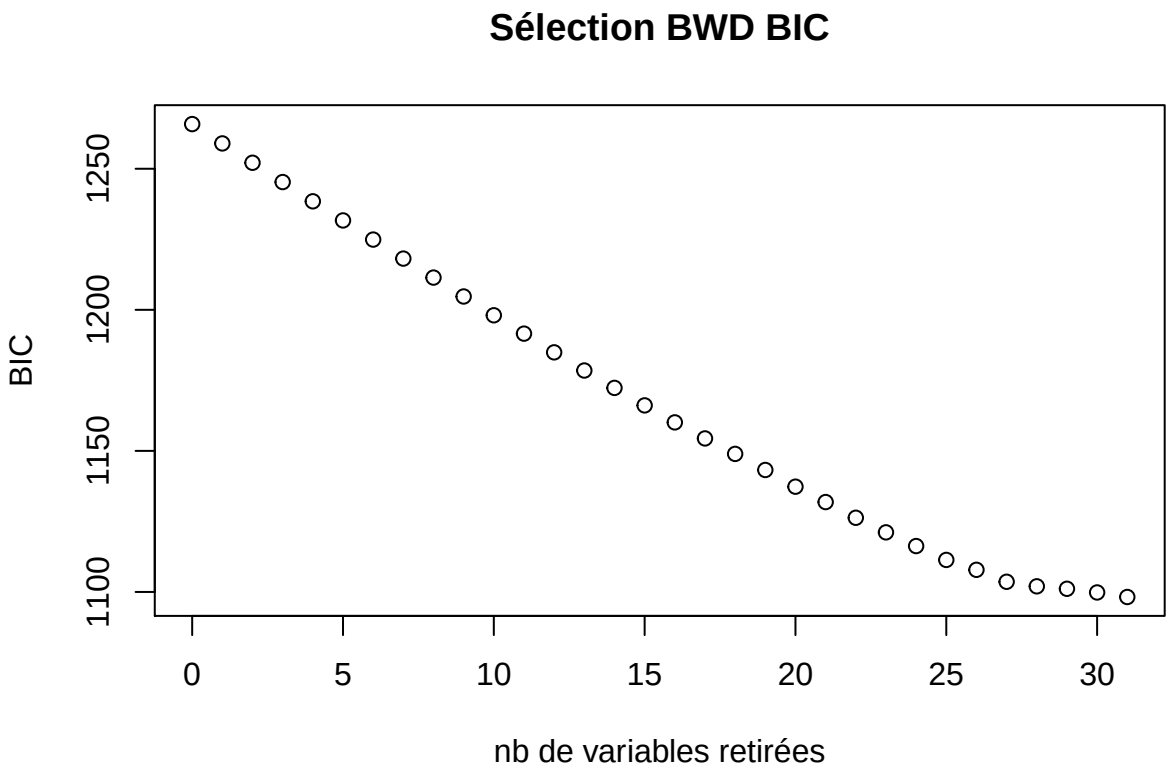
```
# Bic #

lg.FWDBicModel <- glm(formula = pluie.demain ~
                        Nebulosite.moy.max +
                        Pression.min +
                        Temperature.min +
                        RafalesVent.max +
```

```
                        Nebulosite.faible.max
                        , family = binomial,
            data = tr.meteotrain1)

summary(lg.FWDBicModel)
BIC(lg.FWDBicModel)
```

Le modèle retenu a un Bic = 1096.

Modèle lg.BWDBicModel

```
lg.bwdbic.model <- stepAIC(lg.modelsat,data = tr.meteotrain1,
                           direction = "backward",k=log(nrow(tr.meteotrain1)))
summary(lg.bwdbic.model)


coef(lg.bwdbic.model,which.min(summary(lg.bwdbic.model)$aic))
print(lg.bwdbic.model$anova)

plot(0:(nrow(lg.bwdbic.model$anova)-1),lg.bwdbic.model$anova[,"AIC"],type = "b",xlab = "nb de variables
     main = "Sélection BWD BIC")
```



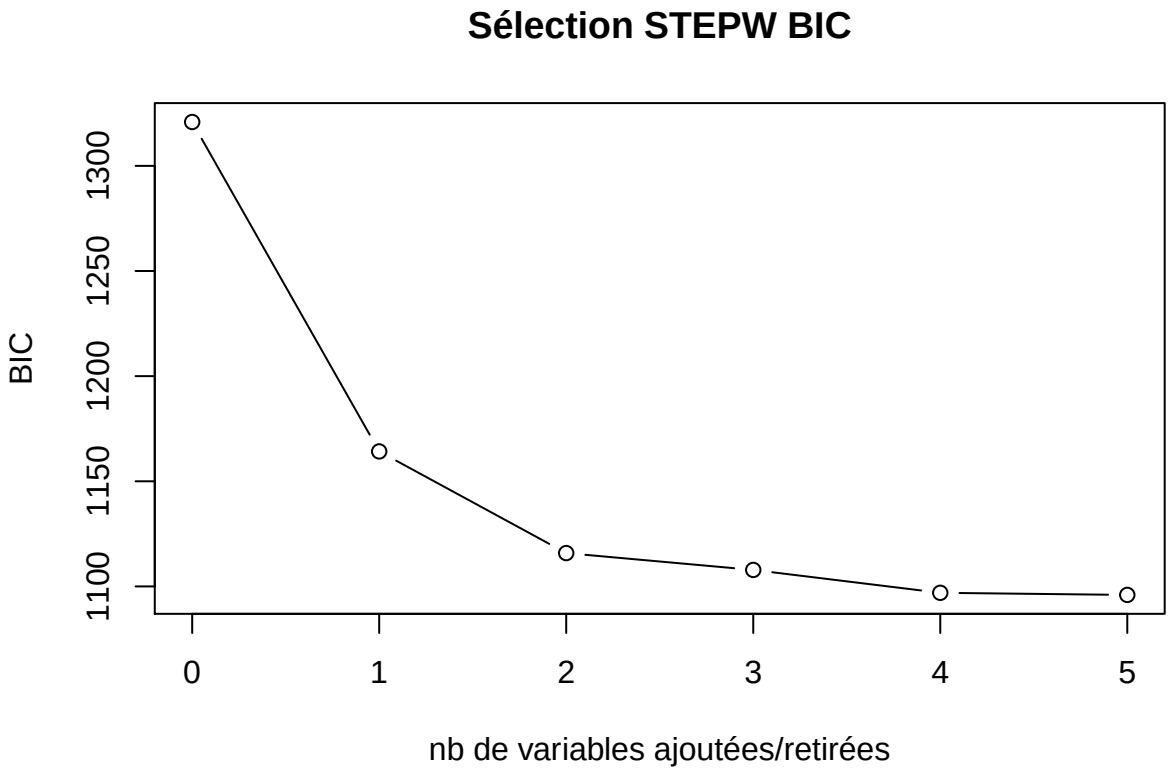
Nous avons donc retiré plus de 30 variables avant d'obtenir un modèle au BIC minimum. Nous voyons que la baisse du BIC au fur et à mesure de la suppression de variables se fait progressivement. Le modèle lg.BWDBicModel retenu a 9 variables et est le suivant :

```
# Bic #

lg.BWDBicModel <- glm(formula = pluie.demain ~
                        Pression.moy+
                        VitesseVent_80m.moy +
                        Temperature.max +
                        Pression.max +
                        Pression.min+
                        Nebulosite.moy.max +
                        Nebulosite.faible.max +
                        VitesseVent_10m.max +
                        VitesseVent_10m.min
                         , family = binomial,
            data = tr.meteotrain1)

summary(lg.BWDBicModel)
BIC(lg.BWDBicModel)
```

Le modèle lg.BWDBicModel retenu a 9 variables, un Bic = 1098.2.

Modèle lg.STEPWBicModel

```
lg.stepwbic.model <- stepAIC(lg.modelnull,
                             scope=list(upper=lg.modelsat),
data = tr.meteotrain1, direction = "both", k=log(nrow(tr.meteotrain1)))
summary(lg.stepwbic.model)

coef(lg.stepwbic.model,which.min(summary(lg.stepwbic.model)$aic))
print(lg.stepwbic.model$anova)

plot(0:(nrow(lg.stepwbic.model$anova)-1),lg.stepwbic.model$anova[,"AIC"],type = "b",xlab = "nb de variab
     main = "Sélection STEPW BIC")
```



Le modèle lg.STEPWBicModel retenu est le suivant :

```
# Bic #

lg.STEPWBicModel <- glm(formula = pluie.demain ~
                          Nebulosite.moy.max +
                          Pression.min +
                          Temperature.min +
                          RafalesVent.max +
                          Nebulosite.faible.max
                          , family = binomial,
                        data = tr.meteotrain1)
summary(lg.STEPWBicModel)
BIC(lg.STEPWBicModel)
```

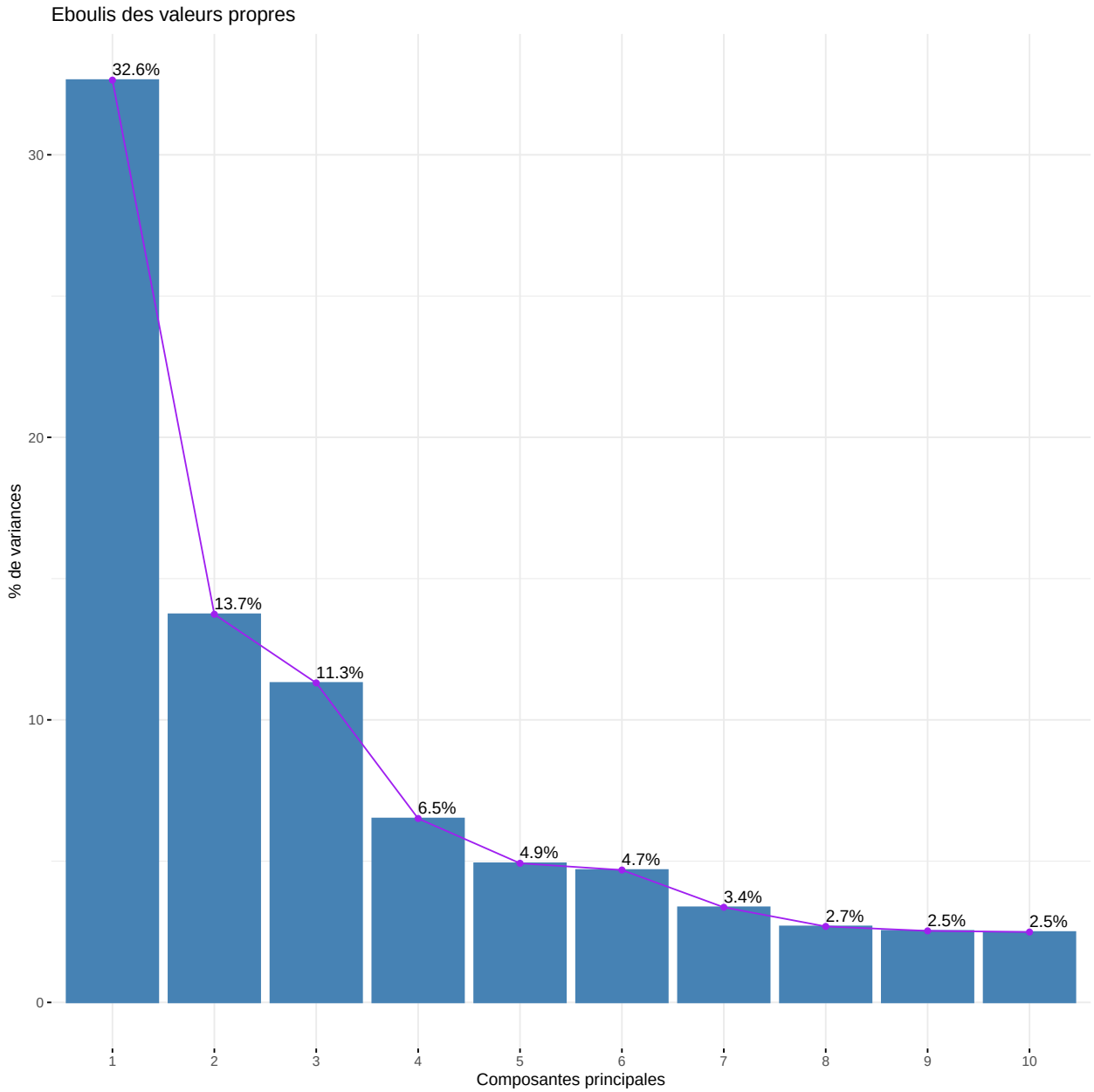
Le modèle lg.STEPWBicModel retenu a 6 variables (le même que le Forward), et un BIC = 1096.

Une autre approche plus judicieuse, étant donné la taille de la matrice de design est de procéder à une ACP. Cette procédure présente un avantage certain qui est celui de diminuer la dimension de notre matrice initiale tout en résumant au maximum l'information. (elle conserve le maximum de variance expliquée)

II.2.5 Approche par Analyse en Composantes Principales

Plus haut nous avions déjà effectué le corrélogramme pour observer les éventuelles corrélations entre les variables explicatives.

En premier lieu, nous allons préparer notre base initiale où nous allons effectuer l'analyse.
La variable pluie.demain sera mise en variable qualitative supplémentaire. Aussi, nous procéderons à l'ACP en considérant les variables X, Annee, Jour et mois comme des variables supplémentaires (celles que nous avons citées comme étant des variables temporelles, elles ne serviront qu'à identifier les points sur les nuages).



Si nous observons le pourcentage cumulé de variance expliquée suivant les 4 premières composantes, nous avons déjà 64% de variance expliquée.

La méthode de Kaiser nous indique qu'il faut retenir les valeurs propres 1 à 9 qui sont supérieures à 1.

Le critère du coude ou screeplot, quant à lui, démontre une première rupture de la courbe à partir de la 2ème

valeur propre, puis une autre cassure observée à la 4ème valeur propre. Cela insinuerait que les 3 premières composantes principales devraient être conservées.

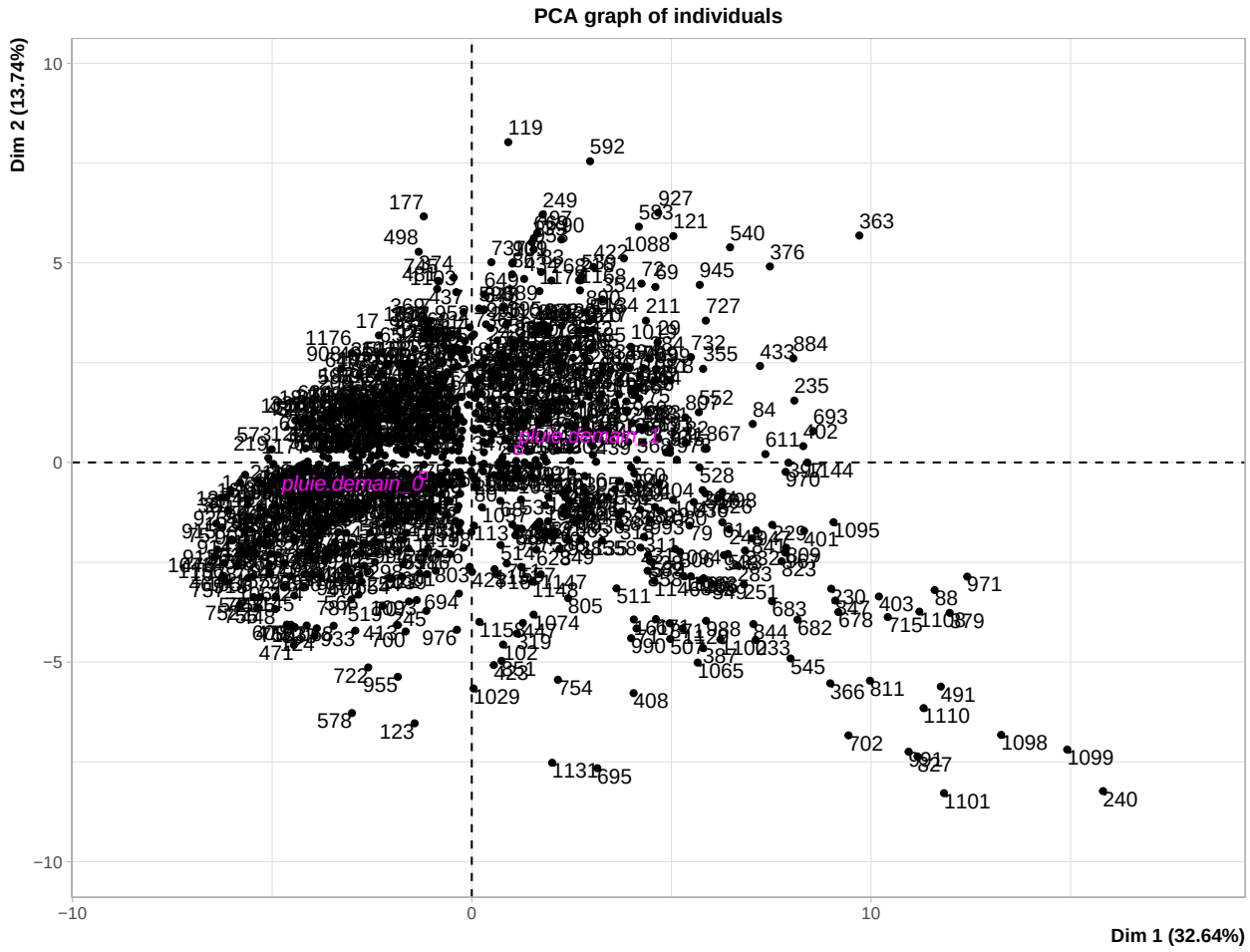
Nous voyons que la 1ère valeur propre comporte presque 33% de l'inertie totale : ceci est déjà un bon résumé car supérieur à 1/8=12,5%. Le second axe explique presque 14% de la dispersion soit une information résumée également. La 3ème valeur propre explique à elle seule 11% de l'inertie et pourrait aussi apporter des informations intéressantes. Si nous faisons le choix des 8 premières composantes principales, cela nous conduirait à presque 80% de l'inertie totale, le critère du 80% de l'information conservée serait quasi-satisfaite.

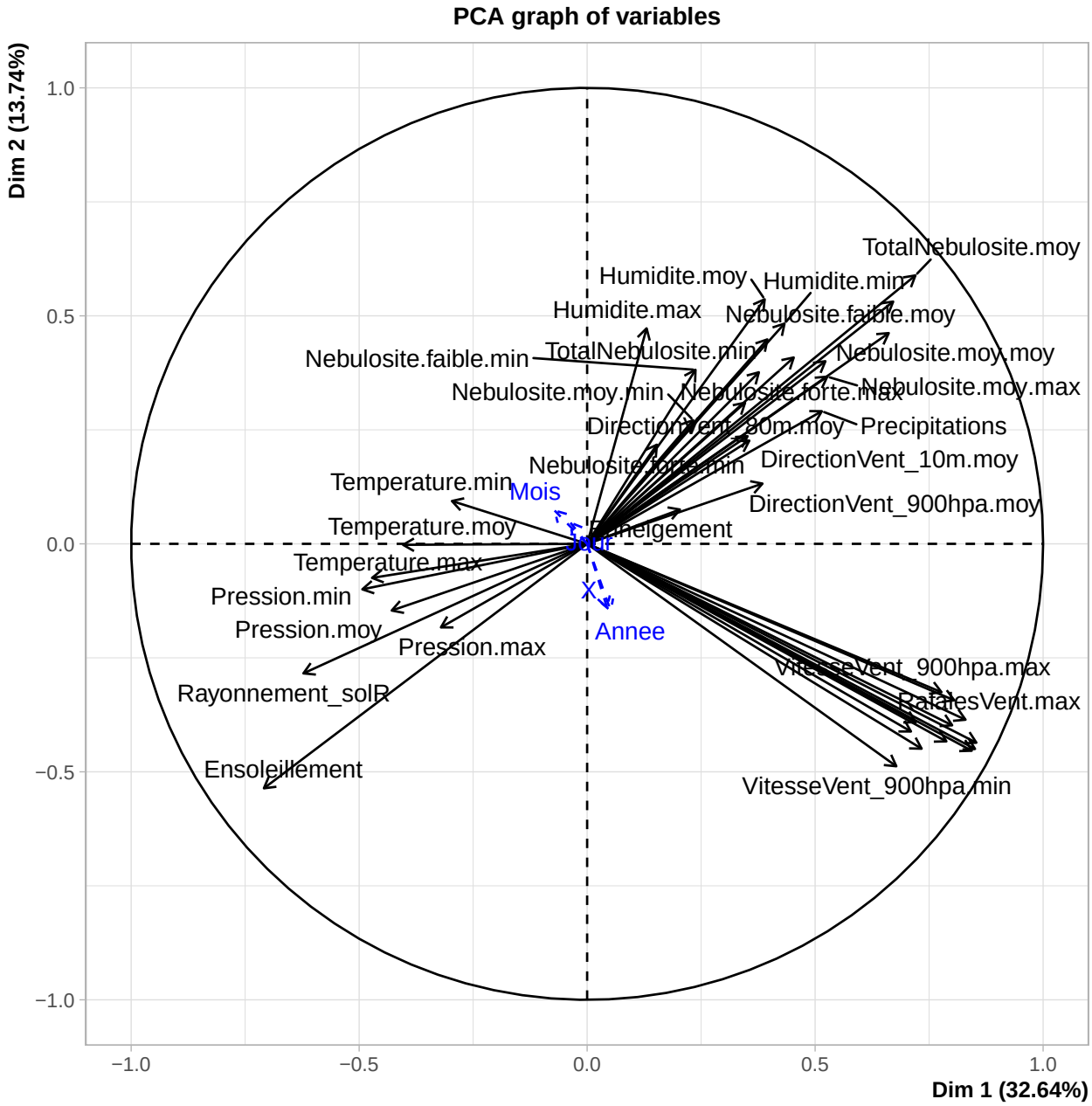
Etudions ces 8 premières composantes, ce qui représente déjà presque 79% de variance expliquée.

Nous allons partir sur ce choix de 8 axes.

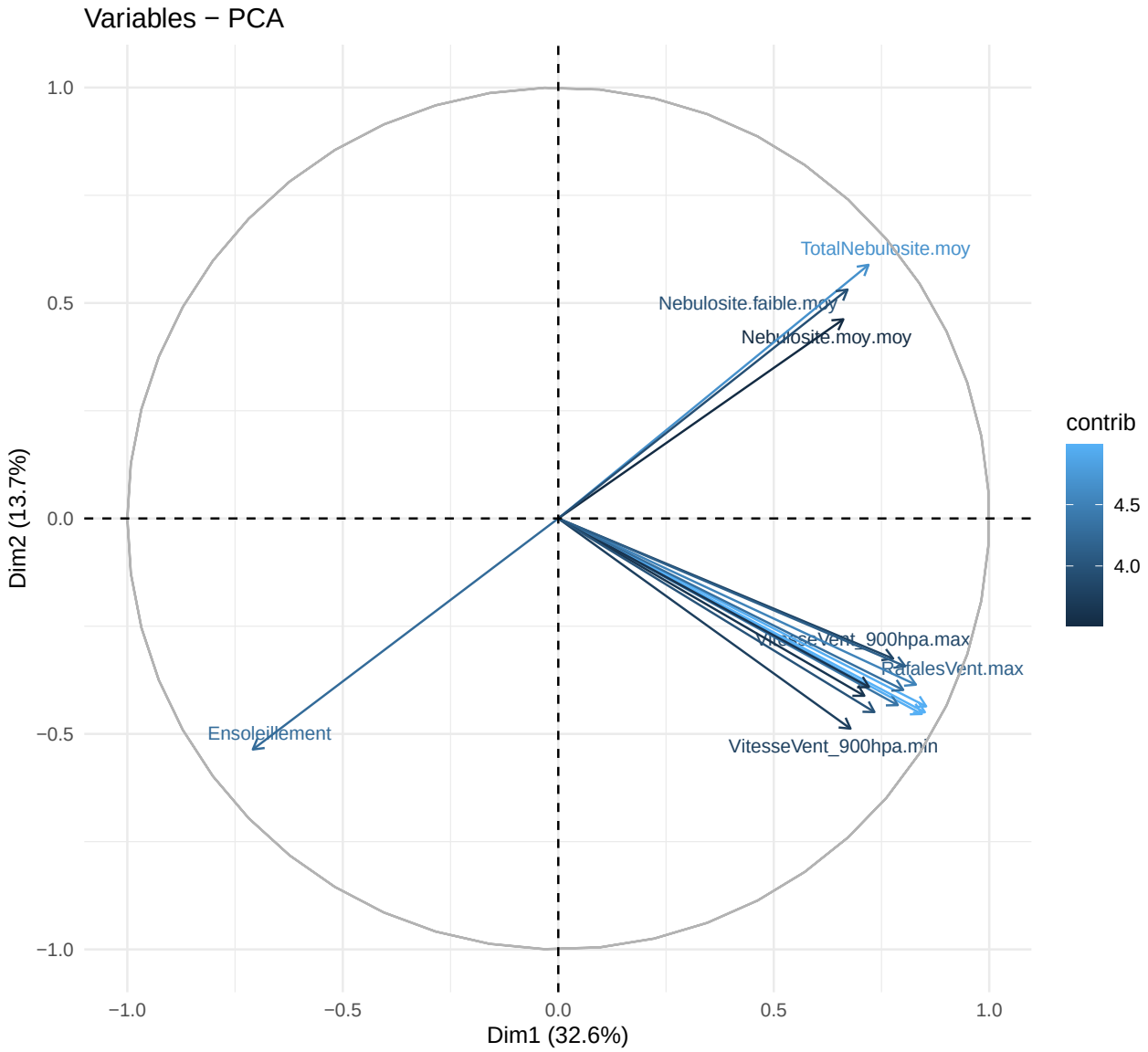
Analyse du cercle de corrélation des variables

```
acp.res <- PCA(base.acp, ncp=8,scale.unit=T,quali.sup = 45,
                  quanti.sup = 1:4,graph = T)
```






```
###################  cercle de corrélation  ########################

fviz_pca_var(acp.res ,axes = c(1,2),geom = c("arrow", "text"),
             labelsize = 3,
             repel = T,
             fill.var = "white",
             col.var = "contrib",
             col.circle = "grey70",
             select.var = list(names=NULL,cos2=0.5
                               )) + theme_minimal()
```




```
acp.res$var$cos2 # la qualité de représentation sur les axes #
acp.res$var$contrib  #,# la contribution sur les axes #
acp.res$var$cor # la corrélation de chaque variable sur les axes #
```

En n'affichant que les variables de qualité de représentation cos2>=0.5, nous constatons pour l'effet taille sur l'axe 1, que les variables qui sont toutes corrélées positivement (du même côté de l'axe 1) sont :
- les variables qui se rapportent à VitesseVent/RafalesVent - les 3 variables relatives à la Nebulosite qui sont TotalNebulosite.moy, Nebulosite.moy.moy et Nebulosite.faible.moy

Pour l'effet forme sur cet axe 1, il semble s'opposer le groupe des variables des caractéristiques VitesseVent/Rafales citées ci-dessus à la variable Ensoleillement, si nous ne retenons que la qualité de représentation supérieure à 0.5.

Sur l'axe 2, pour l'effet taille, nous notons que le groupe des 3 variables de Nebulosite sont du même côté, donc corrélées positivement.

Pour l'effet forme, il y a opposition du groupe des variables liées à la Nebulosite au groupe des variables constitué par les variables VitesseVent/RafalesVent et Ensoleillement.

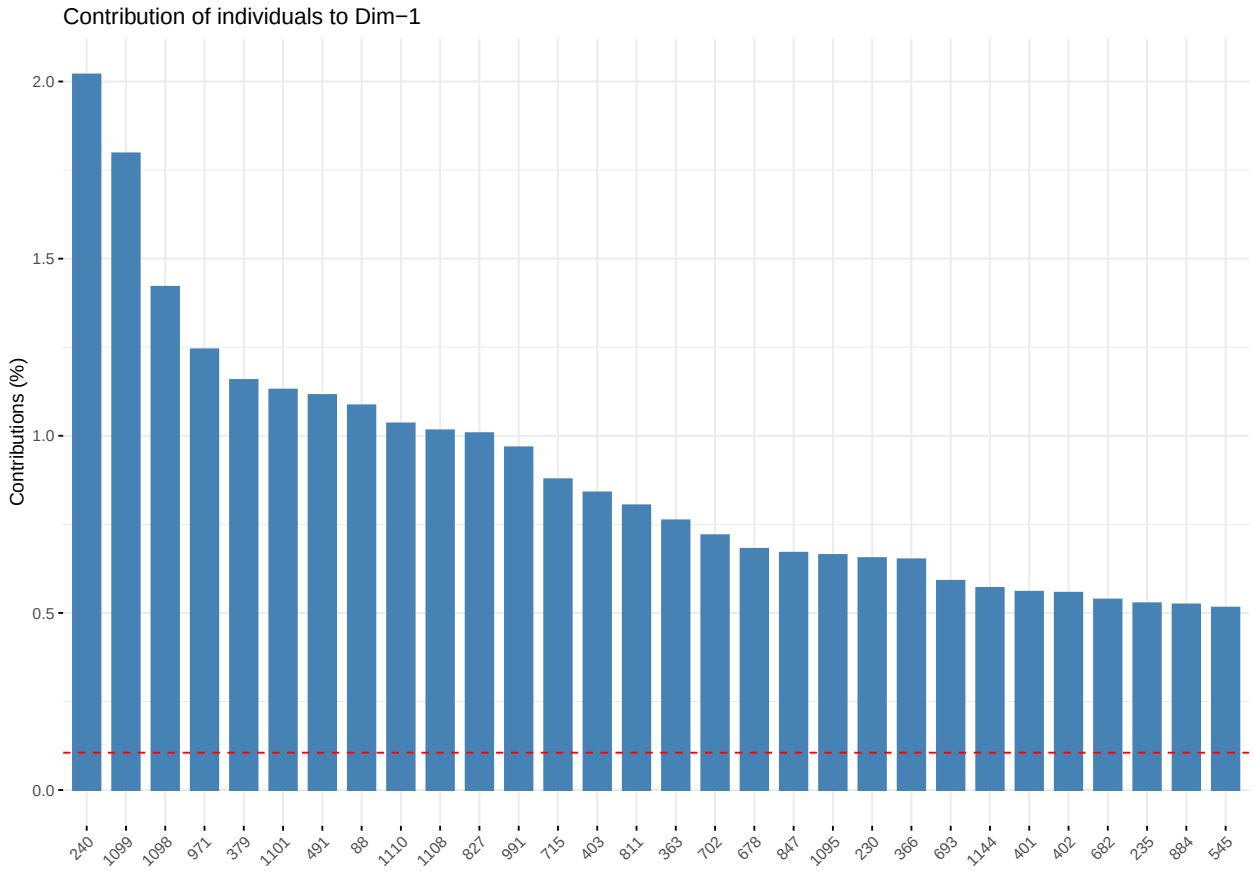
Ceci étant dit, la variable Ensoleillement est corrélée négativement au groupe des variables VitesseV-

ent/RafalesVent. Aussi, les 3 variables de Nebulosite sont corrélés négativement avec Ensoleillement. Ceci semble tout à fait logique car dès que le ciel se couvre de nuages, la durée d'ensoleillement s'écourte au cours de la journée.

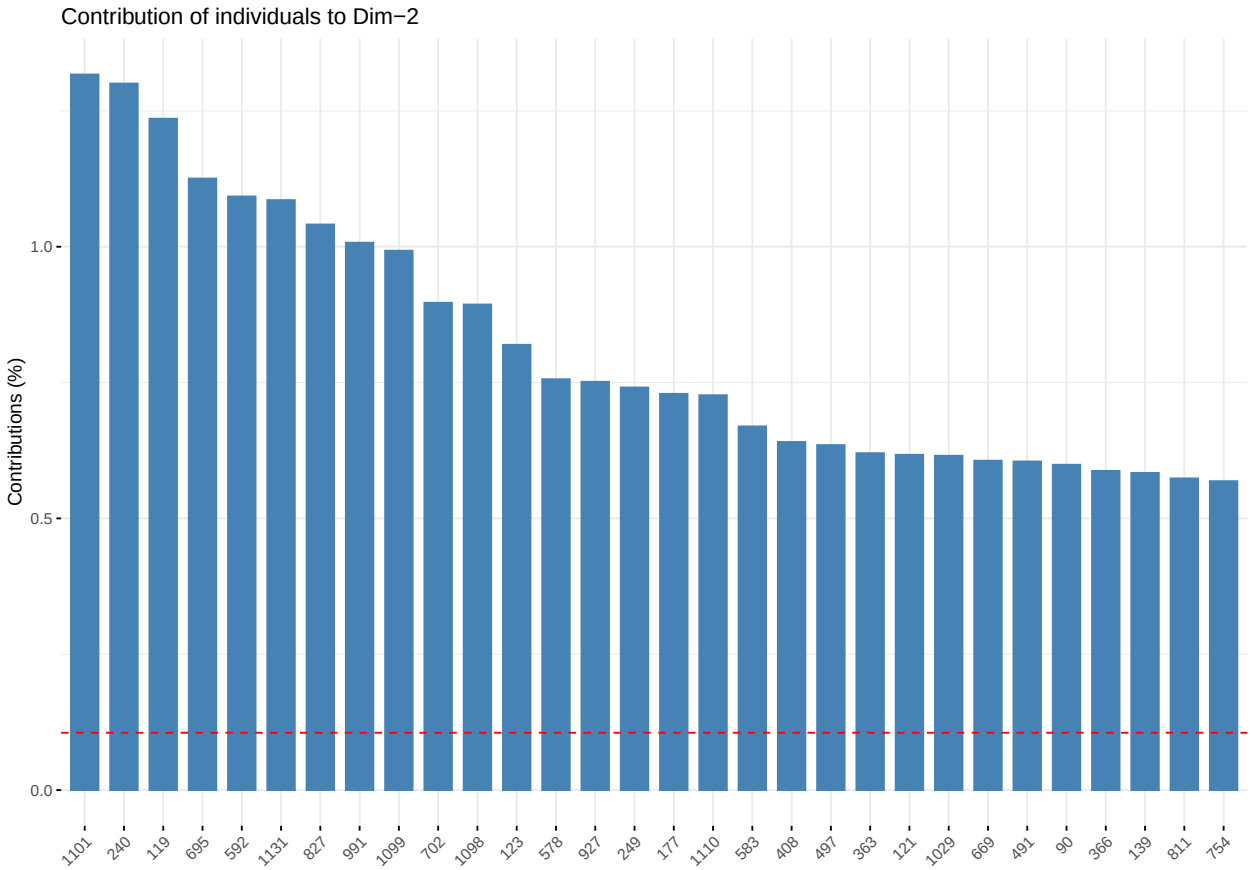
Analyse du nuage des individus

```
##################### Nuage des individus #########################
fviz_pca_ind(acp.res ,geom= c("point","text"),
             repel =T,
             axes = c(1,2),
             palette = NULL,
             pointsize = 0.3,
             labelsize=2,
             addEllipses = T,
             ellipse.level=0.95,
             cex = 0.7,
             col.ind = base.acp$pluie.demain,
             legend.title = "Pluie/Sec",
             select.ind=list(contrib =600))
```


```
## Contribution des individus ##
fviz_contrib(acp.res, choice = "ind", axes = 1,
             top = 30 )
```




```
fviz_contrib(acp.res, choice = "ind", axes = 2,
             top = 30 )
```



Nous constatons d'emblée la prédominance des journées pluvieuses du côté droit du graphe que ce soit en haut ou en bas.
Avec l'analyse des variables faite plus haut, nous en déduisons que ces journées sont donc caractérisées par l'importance des nuages (pour le quart haut droite) et celles du vent de toutes natures (pour le quart bas droite) Dans le quart bas gauche, nous voyons se concentrer la plupart des journées à temps sec (même si elles sont quand même un peu dispersées ailleurs aussi).

L'axe 1 :
Des points remarquables sont à noter comme les observations : 240, 1099 et 1098. Ce sont en effet les 3 observations qui contribuent le plus dans la construction de l'axe 1.
Nous pouvons ainsi faire le lien avec la représentation des variables énoncées en haut : ces journées sont celles qui sont le plus marquées par le vent et les rafales.
Dans le quart haut droite, nous avons également des points remarquables comme les observations : 693, 884, 402 ; ceux-ci figurent également parmi les plus grands contributeurs de l'axe 1.

```
topindivcontrib1 = meteotrain[c(240,1099,1098,693,884,402),]
```

Ce sont des jours tombant principalement en décembre, janvier ou février, soit en plein hiver.
Ils sont marqués par des nébulosités très importantes, des vents et rafales largement plus que la moyenne des autres journées, voire au maximum.

L'axe 2 :
Les points 1101, 119 et 592 sont parmi ceux qui contribuent le plus à l'axe. Il y a également les points 695, 1131 et 240 (point qui se retrouve également contributeur de l'axe 1).
Nous ne pouvons pas vraiment distinguer la particularité de ces jours en relation avec les variables (du graphe des variables) car ces journées traduisent des effets différents. Ce sont des points qui ne se ressemblent pas vraiment. Nous verrons plus bas au détail par axe que cet axe n'est pas vraiment représenté par des variables précises.
Par ailleurs, nous voyons bien que ces points remarquables sont situés tantôt dans le quart haut droite, tantôt

dans le quart bas droite voire extrême droite.

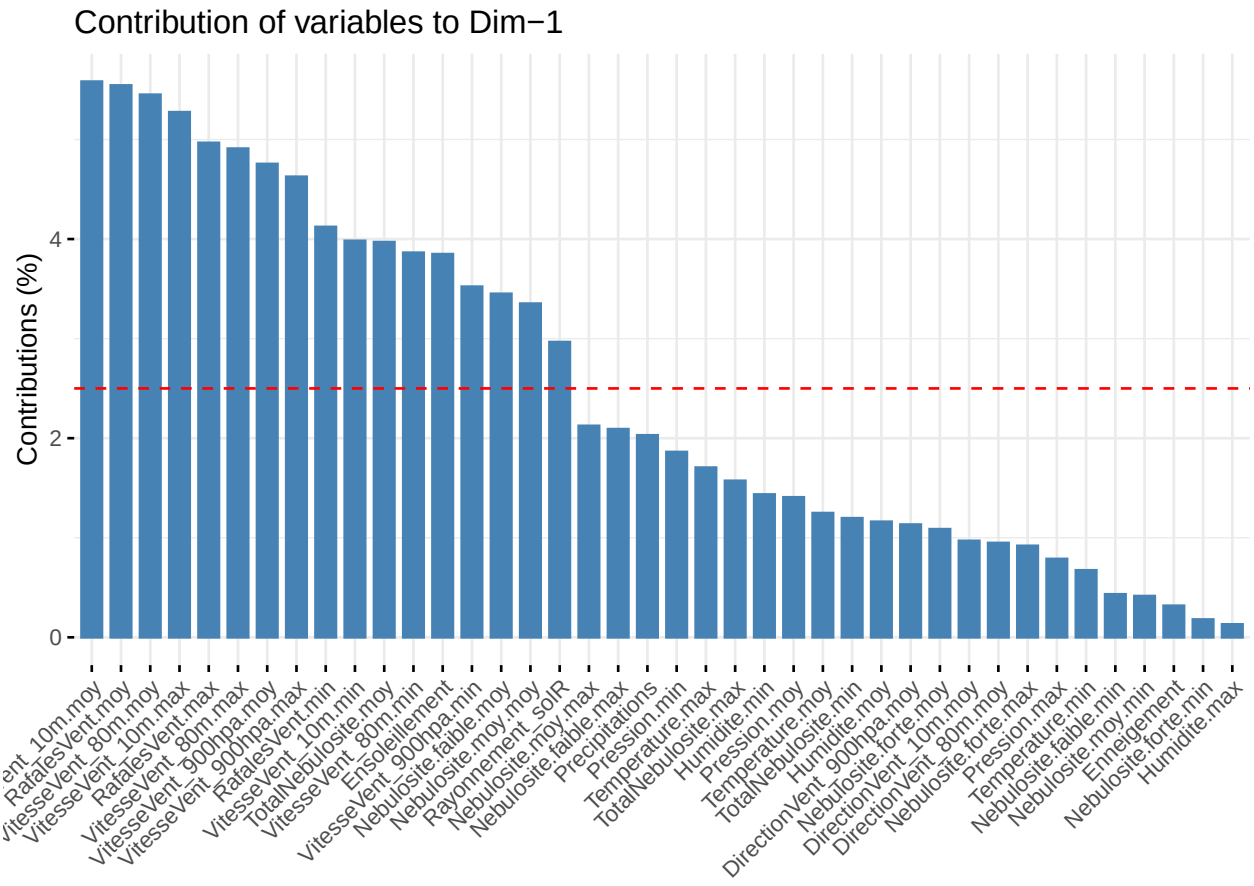
```
topindivcontrib2 = meteotrain[c(1101,119,592,695,1131,240),]
```

Ces journées traduisent ainsi à la fois du temps sec/venteux, mais également du temps pluvieux/nuageux, et du temps pluvieux/venteux qui ont eu lieu principalement aux mois de décembre, janvier, février et mars.

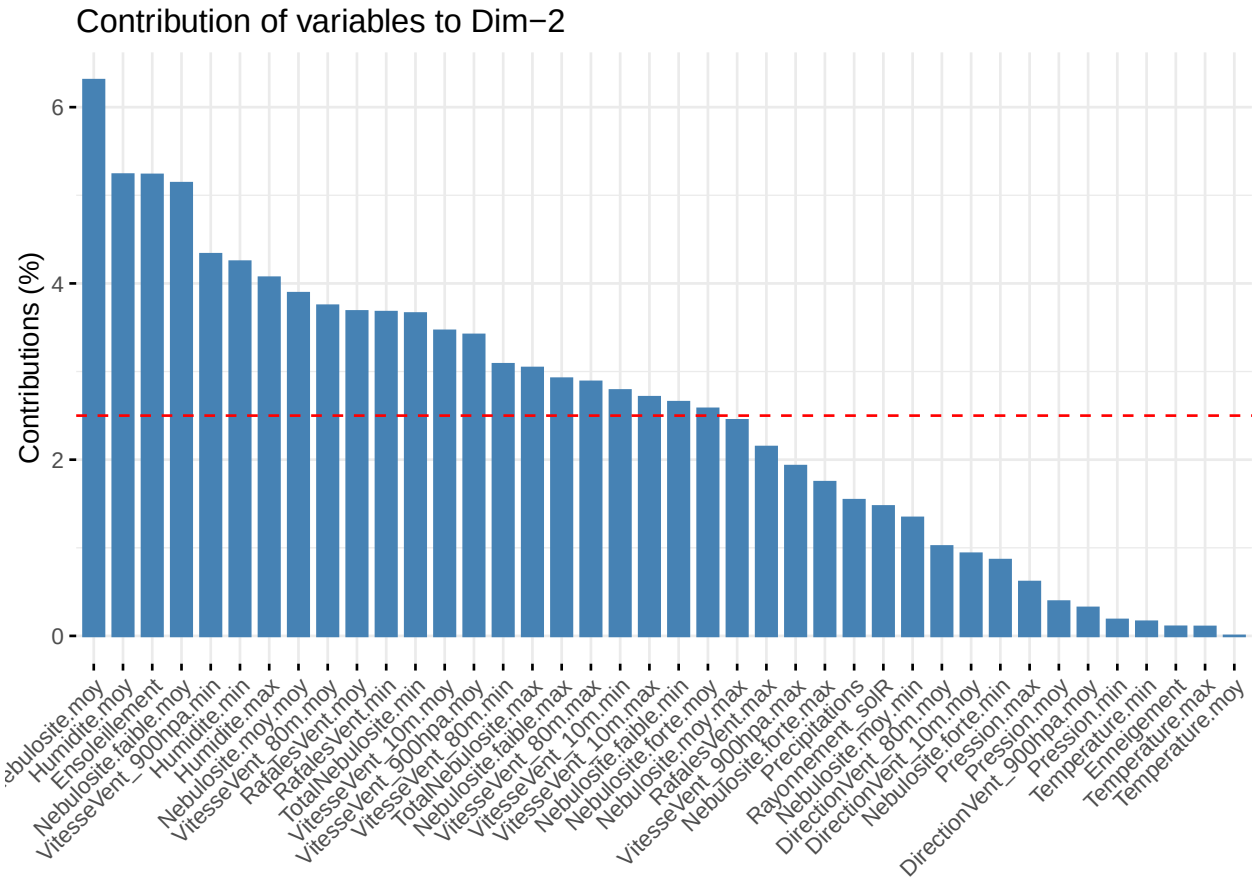
Nous allons voir comment interpréter toutes ces relations en approfondissant l'analyse axe par axe avec les variables-individus.

Analyse du 1er plan factoriel (1,2)

```
# Contribution des variables #

fviz_contrib(acp.res, choice = "var", axes = 1)
```




```
fviz_contrib(acp.res, choice = "var", axes = 2)
```




```
# Description automatique des axes #
dimdesc = dimdesc(acp.res, axes = 1:8, proba=0.05)
dimdesc$Dim.1
dimdesc$Dim.2
```

L'axe 1 : Les variables qui contribuent le pus fortement sont toutes les variables VitesseVent/RafalesVent déjà vues plus haut.

Avec des contributions entre 4 et 5 chacun et des cos2 proches de 0.5 à 0.7, nous pouvons dire qu'ils ont une bonne qualité de représentation sur cet axe et contribuent en grande partie à celui-ci. ===> cet axe 1 représenterait donc les journées venteuses (avec vitesse et rafales importantes).

Dans une moindre mesure, nous avons aussi les variables liées à la Nebulosité moyenne qui contribuent assez à l'axe 1 (avec des contributions à un peu moins de 4 et une qualité de représentation avoisinant 0.5).

Toutes ces variables citées ont une corrélation forte avec l'axe 1 (entre 0.6 et 0.85).

A l'opposé, les variables liées à l'ensoleillement et le rayonnement solaire ont d'importants coefficients de corrélation négatifs avec l'axe 1, et une bonne qualité de représentation (notamment pour l'ensoleillement cos2 à 0.5).

==> cet axe 1 oppose donc le groupe des journées très venteuses et pluvieuses (avec de la nébulosité) aux journées ensoleillées. On peut résumer cet axe 1 par la prédominance du vent et des nuages durant les journées.

L'axe 2 :
Les variables relatives à la Nebulosité sont parmi celles qui contribuent à cet axe avec des contributions oscillant entre 4 et 6 mais une qualité de représentation peu convaincante aux alentours de 0.2-0.3. Aussi, nous voyons que les variables liées à l'humidité sont aussi parmi celles qui contribuent le plus à cet axe

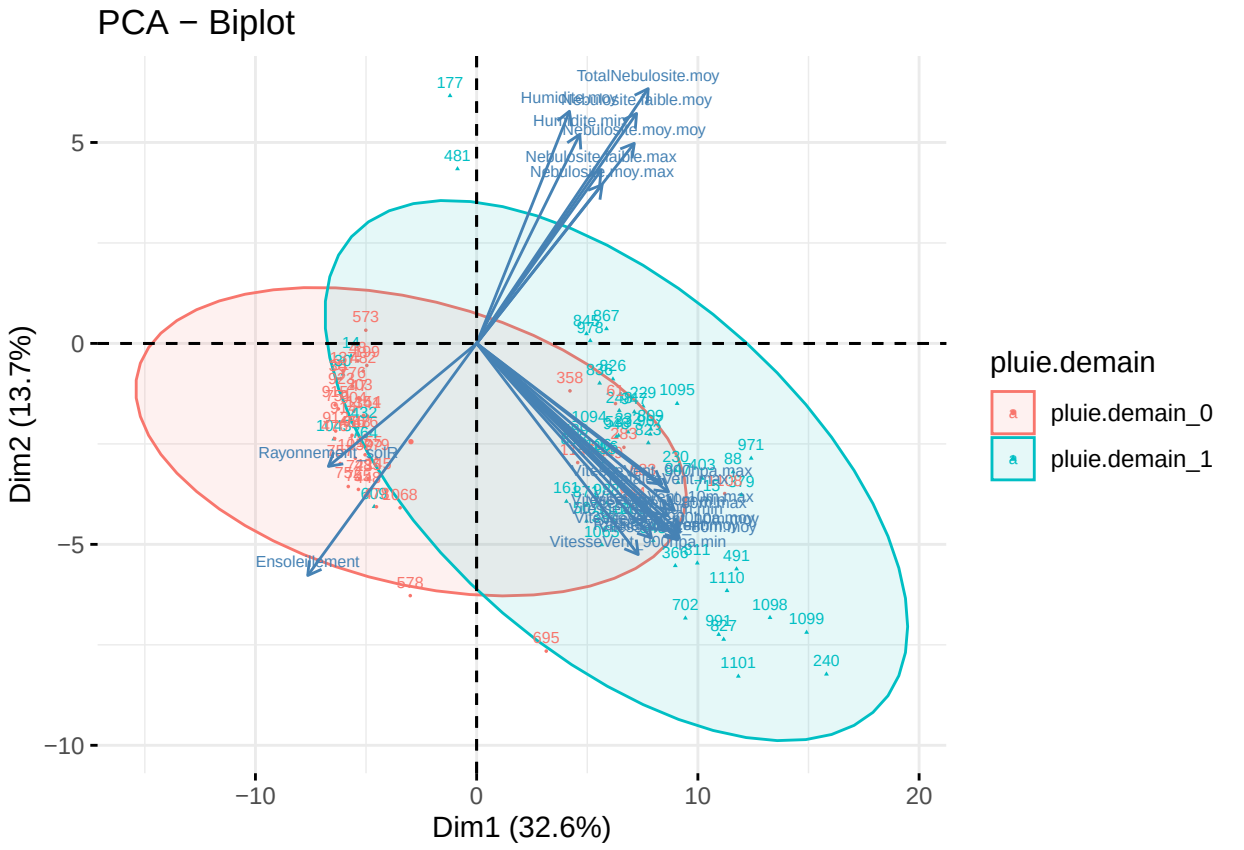
(contributions proches de 4-5), ainsi que celles des vent (côté bas droit du graphique). Cependant, toutes ont une faible qualité de représentation.
A l'opposé sur cet axe, nous avons l'ensoleillement qui contribue également pour plus de 5 (mais une qualité à 0.3).

==> cet axe 2 ne semble pas être bien représenté par une catégorie particulière de journées.

Les variables qui ont les corrélations les plus importantes (environ 0.5) avec l'axe sont celles se rapportant à l'humidité et la nébulosité moyenne. A l'opposé de cet axe, nous avons l'ensoleillement qui a la corrélation négative la plus significative (à -0.54).

Etant donné ces corrélations assez faibles, nous ne pouvons déterminer précisément la caractéristique principale de cet axe.

```
fviz_pca_biplot(acp.res,habillage = 45,
                pointsize = 0.3,
                labelsize = 2,
                addEllipses = T,
                ellipse.level=0.95,
                cex = 0.7,
                select.ind=list(cos2=0.7, contrib =400),
                select.var=list(cos2=0.4))
```



Sur ce graphe, nous voyons se concentrer les journées pluvieuses dans le quart bas droite, en lien avec les fortes rafales de vent et une vitesse importante des vents de toutes altitudes.

Les journées à temps sec et ensoleillées se concentrent en revanche dans le quart bas gauche relatives à l'ensoleillement naturellement.

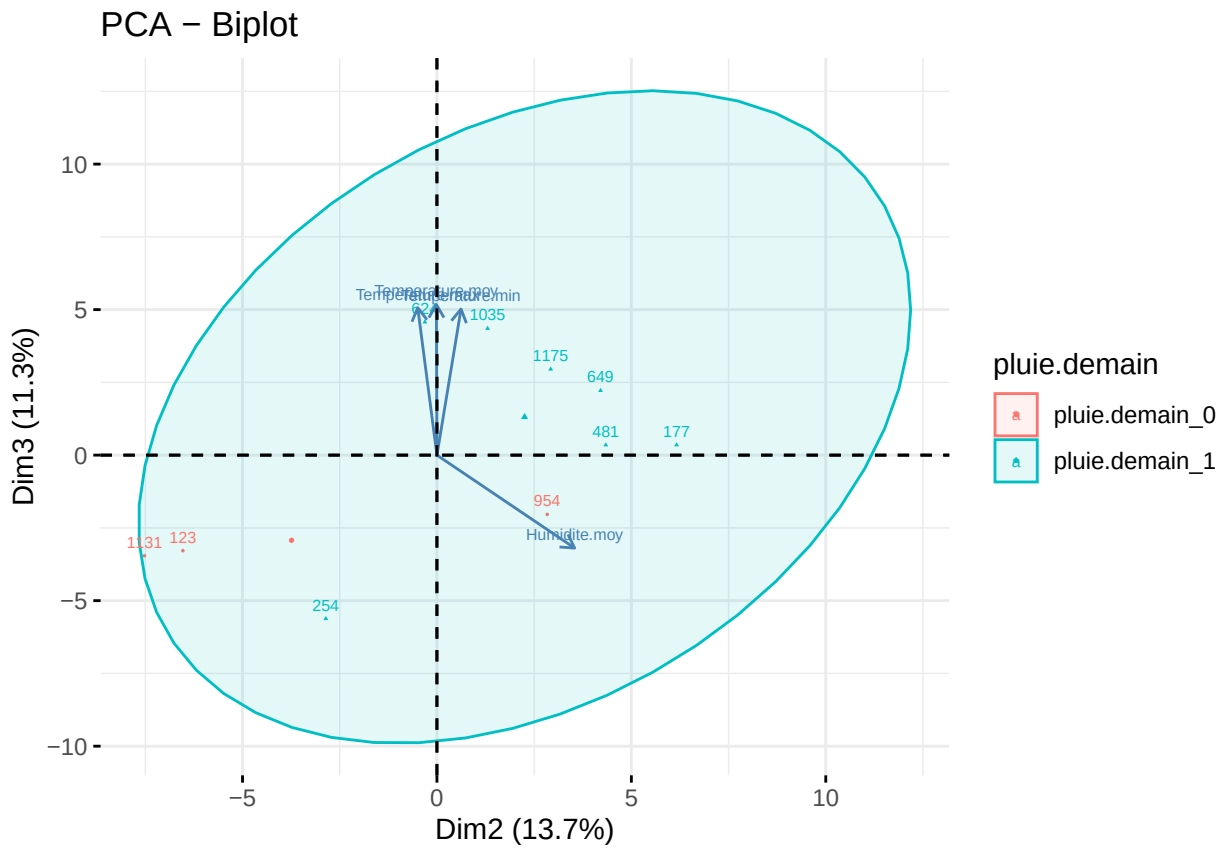
Si nous abaissons le seuil d'affichage des cos2 des variables à 0.4, nous apercevons également les variables

liées à l'humidité, qui sont corrélées positivement à celles de la nébulosité. Leur qualité de représentation sur les axes 1 ou 2 ne dépassant pas 0.4, ils sont moins significatifs que les autres sur ce plan, nous verrons plus loin sur d'autres plans factoriels leur présence plus pertinente.

En bas à gauche, nous voyons également mais de qualité cos2 inférieure à 0,4 le rayonnement solaire qui est corrélé positivement à l'ensoleillement, et corrélé négativement aux variables de nébulosité et du groupe vent/rafales.

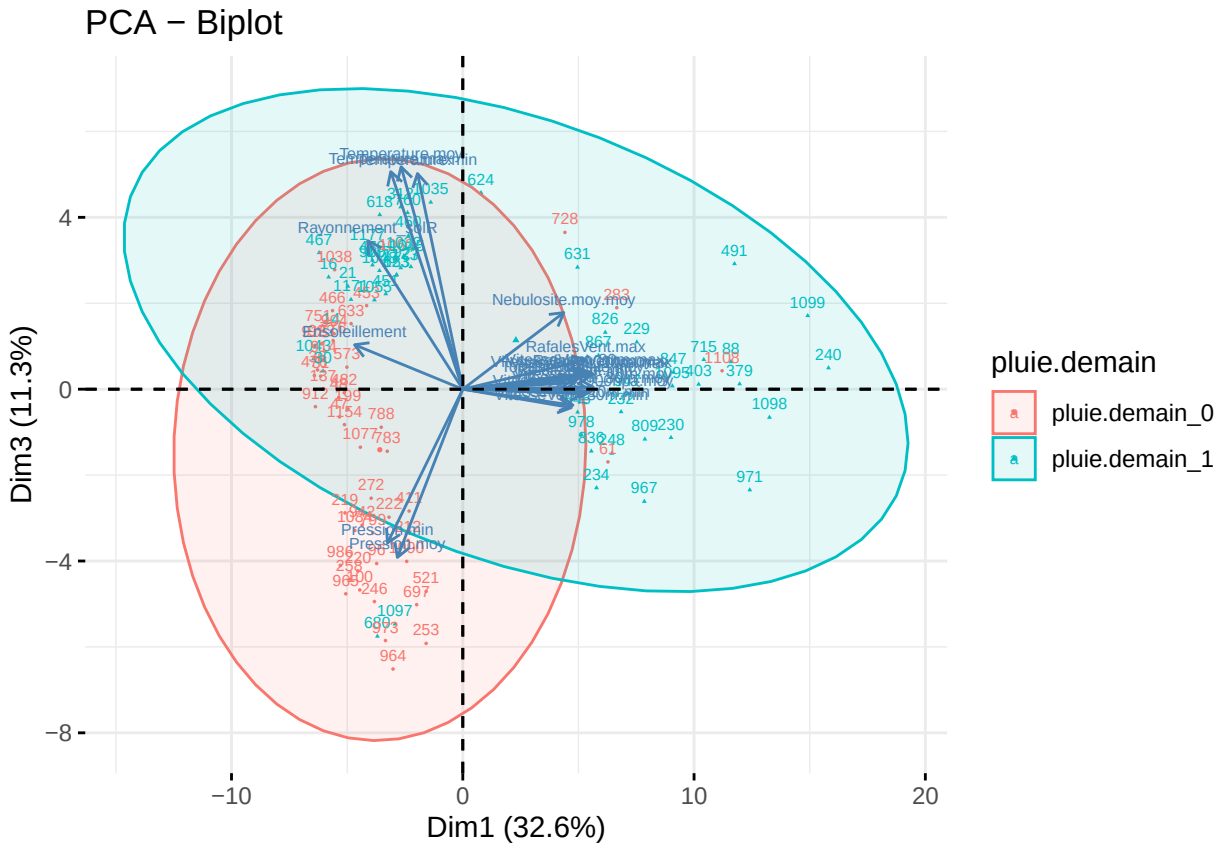
Analyse du 2ème et 3ème plan factoriel (1,3) et (2,3)

```
# Plan axes 2 et 3 et axes 1 et 3 #
fviz_pca_biplot(acp.res,habillage = 45,
                axes = c(2,3),
                pointsize = 0.3,
                labelsize = 2,
                addEllipses = T,
                ellipse.level=0.95,
                cex = 0.7,
                select.ind=list(cos2=0.7, contrib =900),
                select.var=list(cos2=0.5))
```

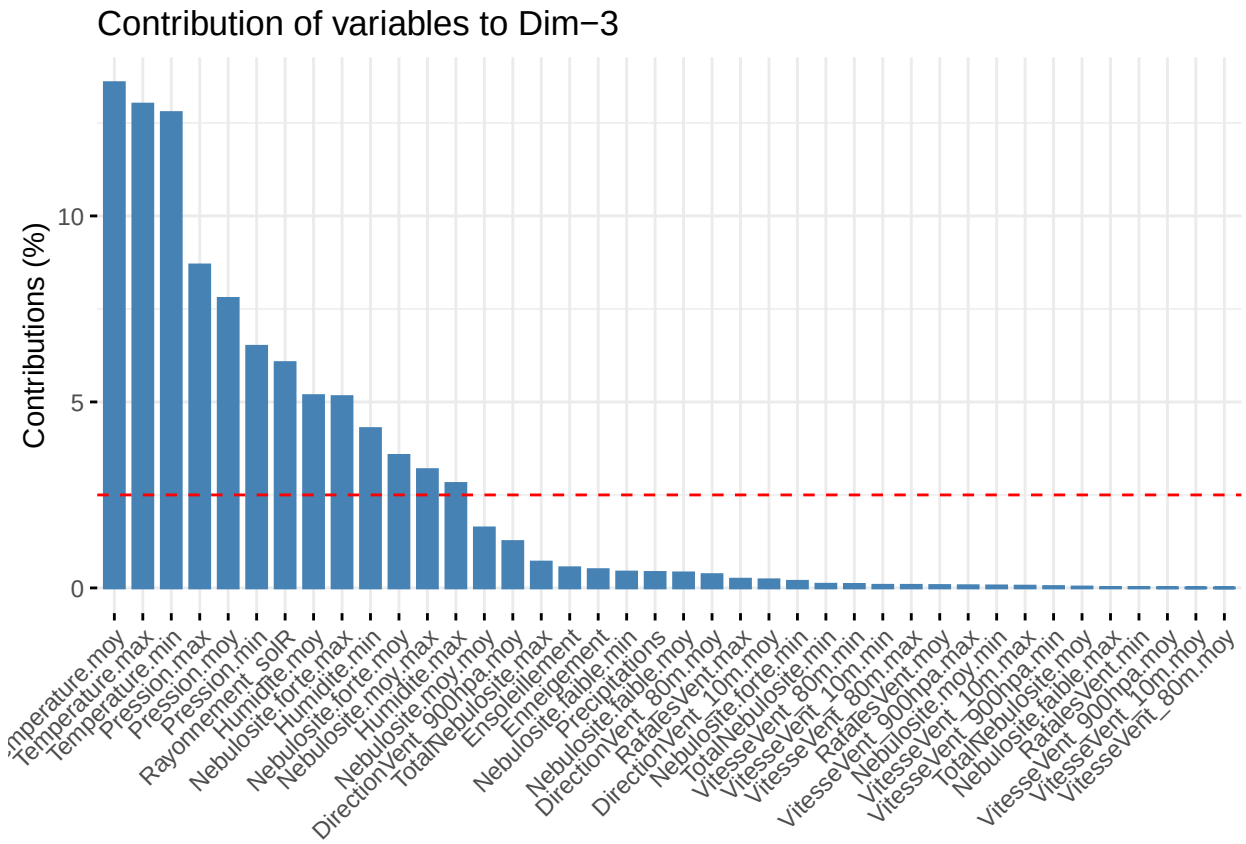



```
# Plan axes 1 et 3 #
fviz_pca_biplot(acp.res,habillage = 45,
                axes = c(1,3),
```

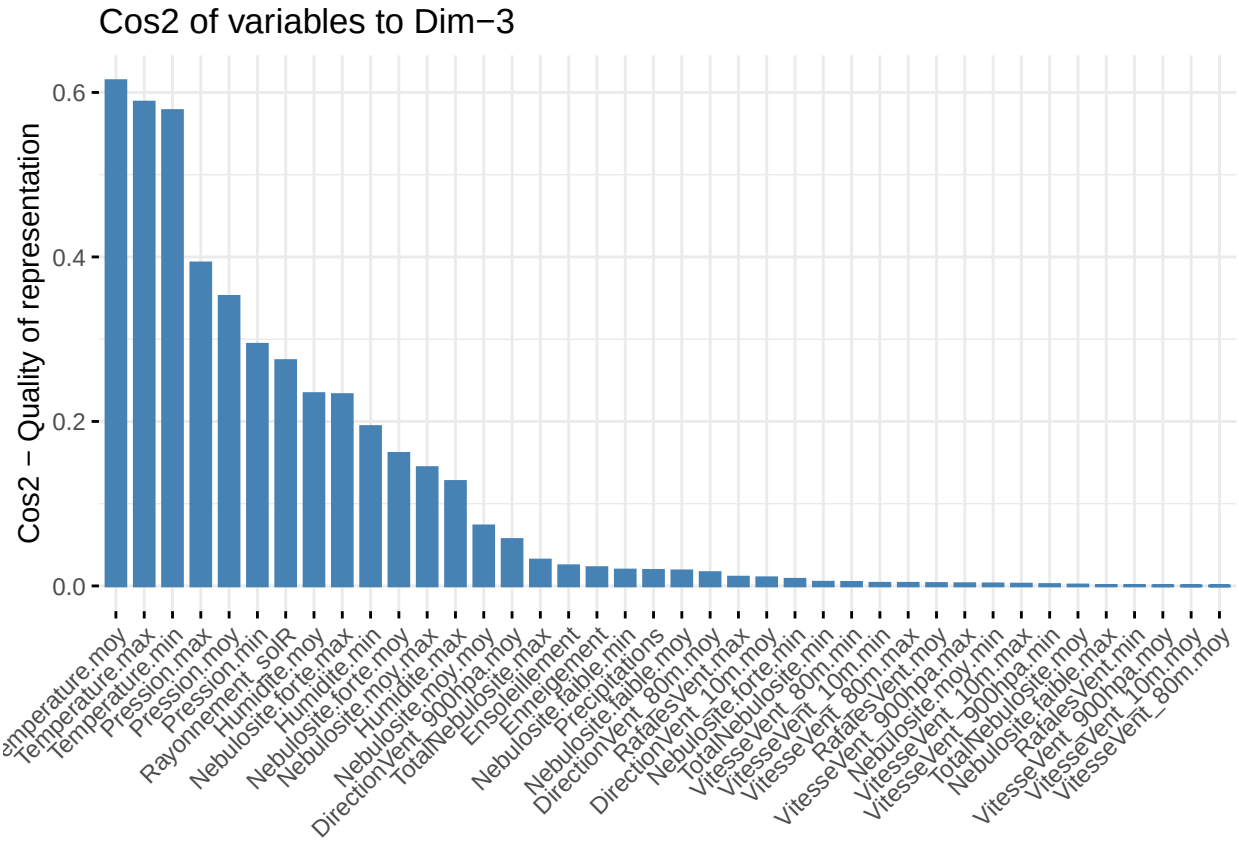
```
                pointsize = 0.3,
                labelsize = 2,
                addEllipses = T,
                ellipse.level=0.95,
                cex = 0.7,
                select.ind=list(cos2=0.7, contrib =900),
                select.var=list(cos2=0.5))
```




```
# Contribution des variables à l'axe 3 #
fviz_contrib(acp.res,axes = 3,choice ="var")
```




```
# Qualité de représentation cos2 à l'axe 3 #
fviz_cos2(acp.res,axes = 3, choice = "var")
```




```
# Description automatique des axes #
dimdesc$Dim.3
```

Sur le plan composé des axes 2 et 3, nous n'avons que très peu d'informations.

Si nous observons le plan factoriel avec les axes 1 et 3, nous avons nettement plus d'informations.

L'axe 1 :
Nous confirmons notre analyse ci-dessus, car nous avons bien les variables liées aux VitesseVent et RafalesVent qui sont presque confondues à l'axe 1.

L'axe 3 :
Les variables de Température sont celles qui contribuent le plus fortement à cet axe, avec des contributions à plus de 13 et de bonne qualité de représentation à 0.6. Les coefficients de corrélation à presque 0.8 le confirment.
De plus, les variables Pression sont aussi de grandes contributrices à cet axe, avoisinant une contribution à 8, mais la qualité de représentation à 0.3 est assez peu importante. On pourrait cependant les conserver mais nous les reverrons sur un autre plan. De même, le rayonnement solaire apporte sa contribution aussi (plus de 6) à cet axe avec un cos2 moyen à 0.3, mais présente une bonne corrélation avec cette dimension 3.

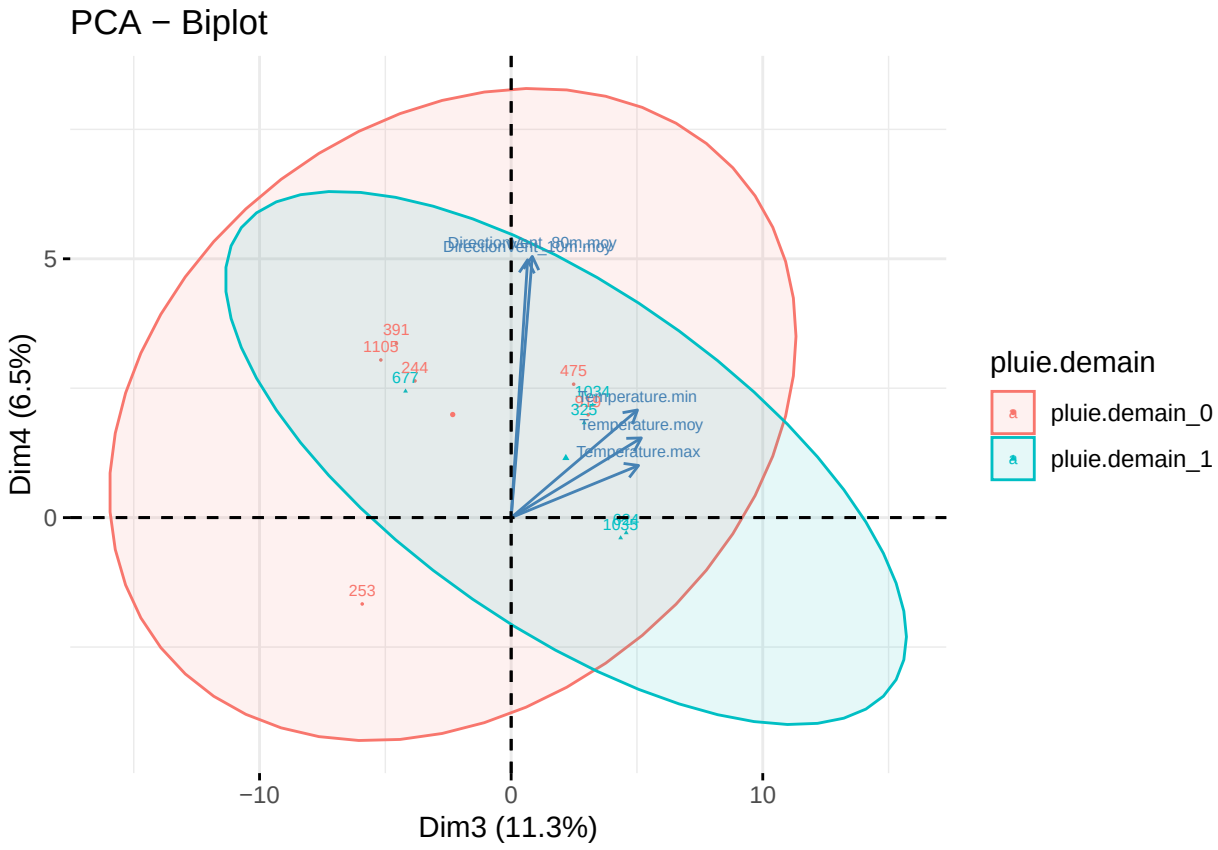
Quant aux individus (journées), nous voyons que du côté positif de cet axe, nous retrouvons les journées pluvieuses.
Du côté négatif, ce sont les journées non pluvieuses.

==> l'axe 3 semble donc représenter la température en grande partie, et la pression (également le rayonnement solaire) dans une moindre mesure. Ces deux groupes de variables sont corrélés de façon opposée sur cet axe (les coefficients de corrélation des pression étant négatifs).
Les journées pluvieuses sont situées sur le haut du graphe en relation avec la température (quart haut gauche) et les vents et rafales de toutes natures (quart haut droite).
Les journées à temps sec sont plus situées dans le quart bas gauche en relation avec la pression, et aussi dans

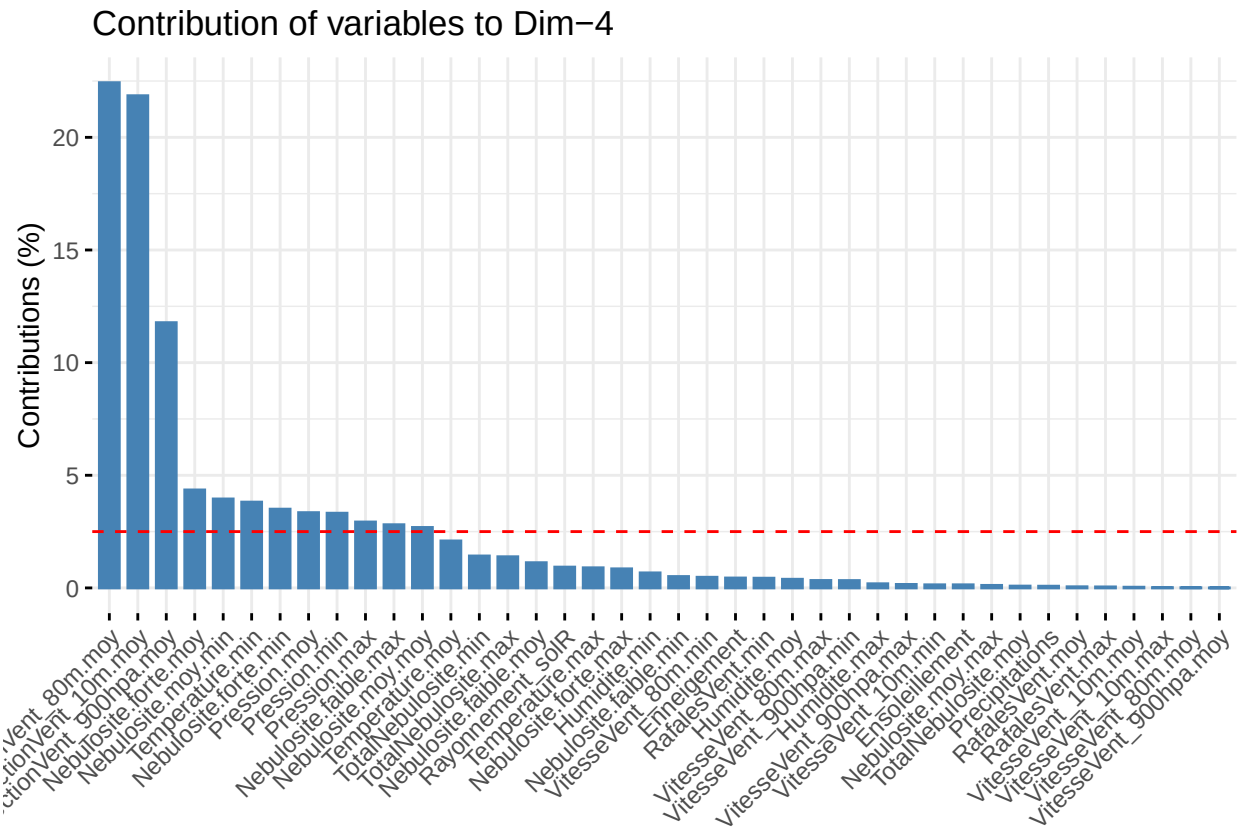
le quart haut gauche proche de l'axe 1.

Analyse du 4ème plan factoriel (3,4)

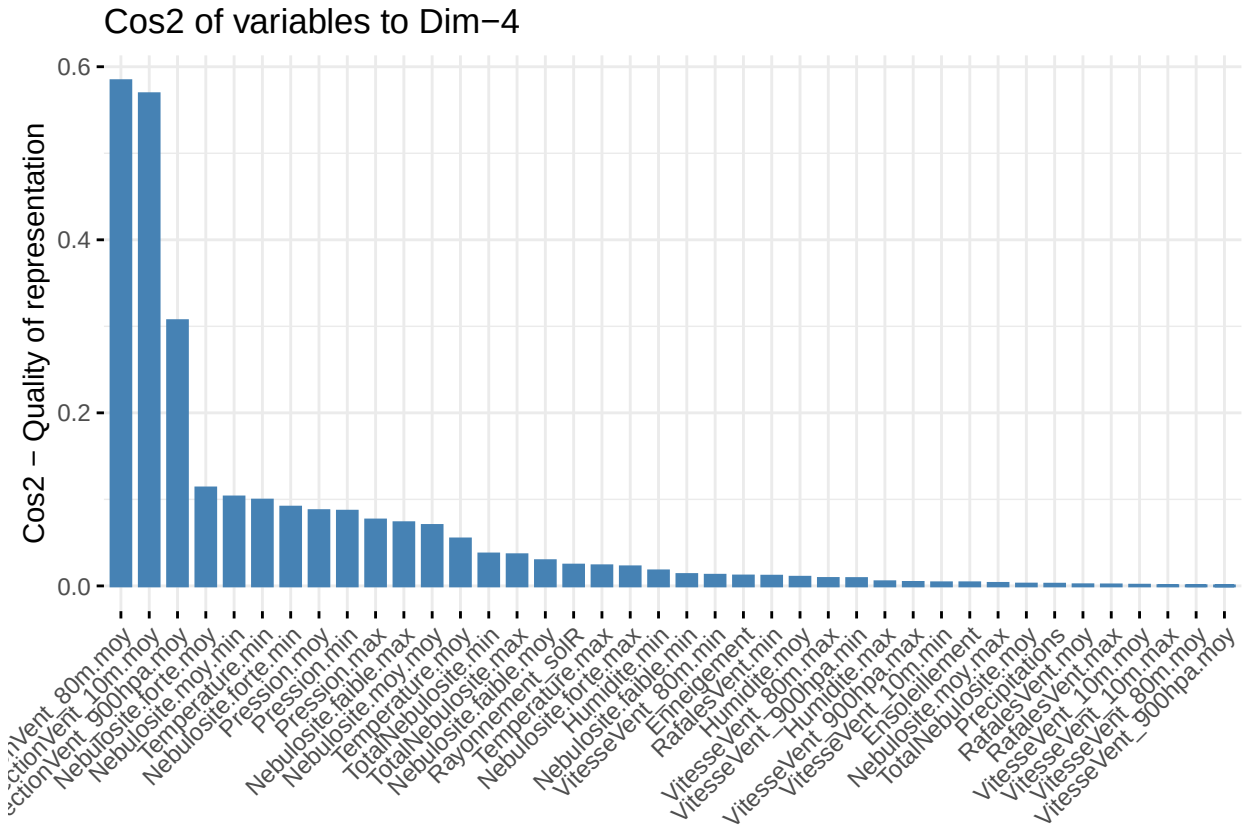
```
# Plan axes 3 et 4 #
fviz_pca_biplot(acp.res,habillage = 45,
                axes = c(3,4),
                pointsize = 0.3,
                labelsize = 2,
                addEllipses = T,
                ellipse.level=0.95,
                cex = 0.7,
                select.ind=list(cos2=0.7, contrib =900),
                select.var=list(cos2=0.5))
```




```
# Contribution des variables à l'axe 4 #
fviz_contrib(acp.res,axes = 4,choice ="var")
```




```
# Qualité de représentation cos2 à l'axe 4 #
fviz_cos2(acp.res,axes = 4, choice = "var")
```




```
# Description automatique des axes #
dimdesc$Dim.4
```

L'axe 3 :
Nous avons vu plus haut que cet axe est représenté par les Températures et Pression.

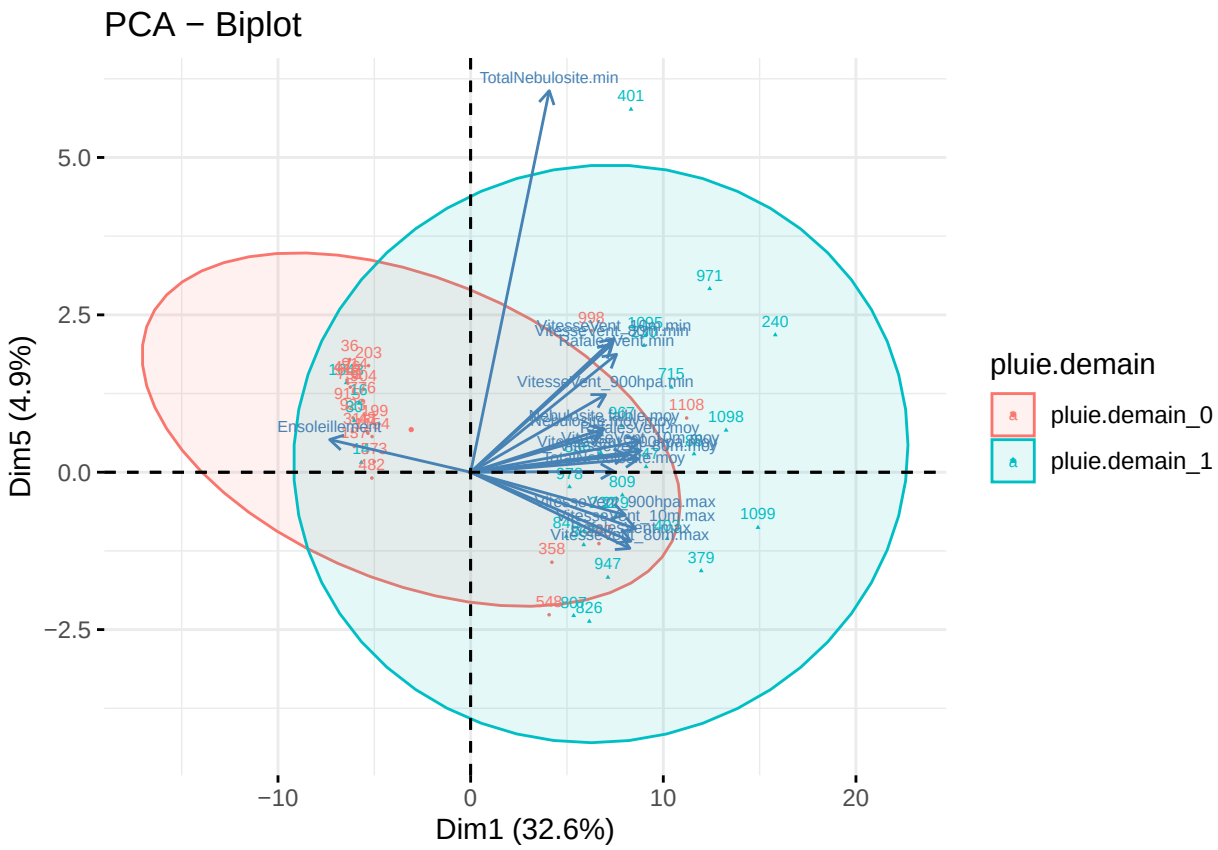
L'axe 4 :
Au vu du plan factoriel composé des axes 3 et 4, celui-ci nous montre que ce sont les variables DirectionVent de toutes altitudes qui y contribuent le plus (entre 11 et 22) et une bonne qualité de représentation aux alentours de 0.6. Ces variables sont fortement corrélées à cette dimension à plus de 0.7.

==> l'axe 4 semble donc représenter la Direction du vent avec les variables DirectionVent_10m.moy, DirectionVent_80m.moy (avec fortes corrélations autour de 0,6 à 0.7) et DirectionVent_900hpa.moy dans une moindre mesure.

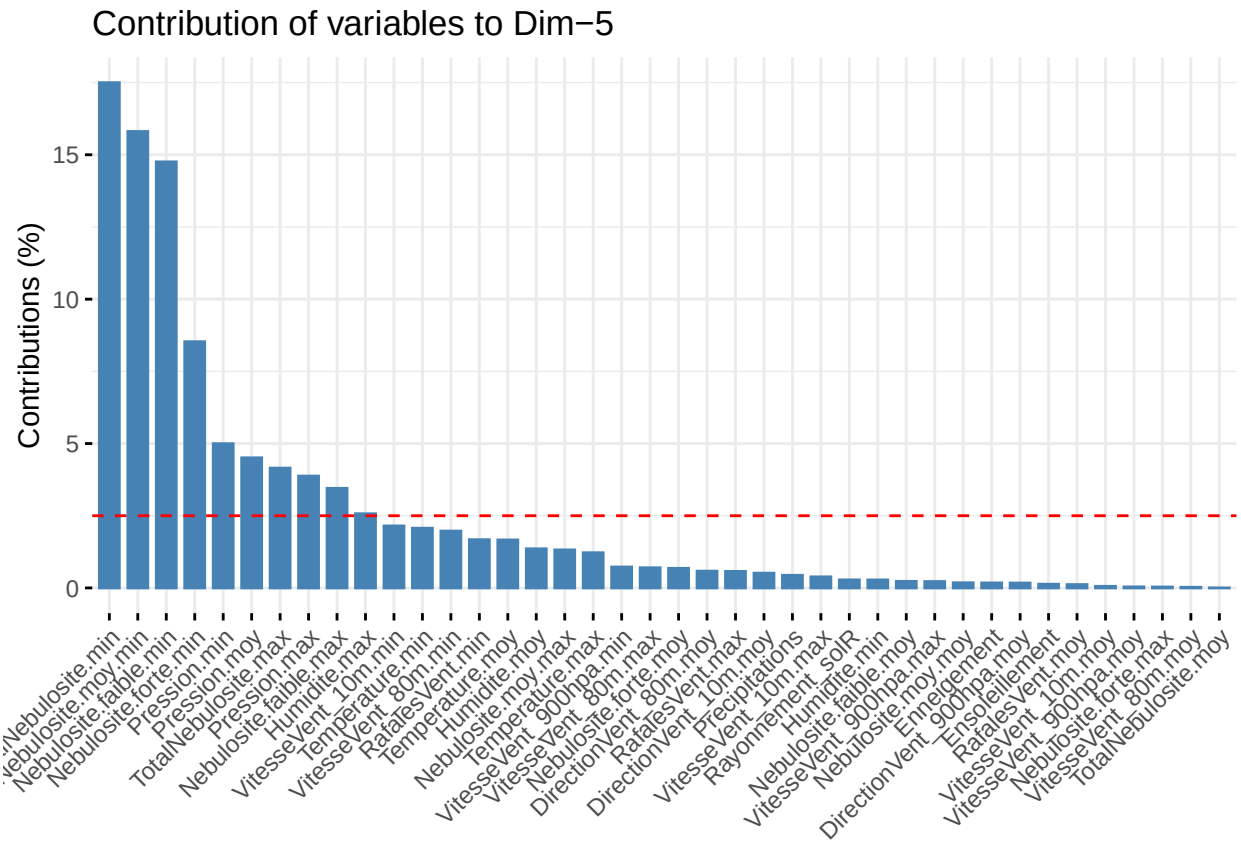
Analyse du 5ème plan factoriel (1,5)

```
# Plan axes 1 et 5 #
fviz_pca_biplot(acp.res,habillage = 45,
                axes = c(1,5),
                pointsize = 0.3,
                labelsize = 2,
                addEllipses = T,
                ellipse.level=0.95,
                cex = 0.7,
```

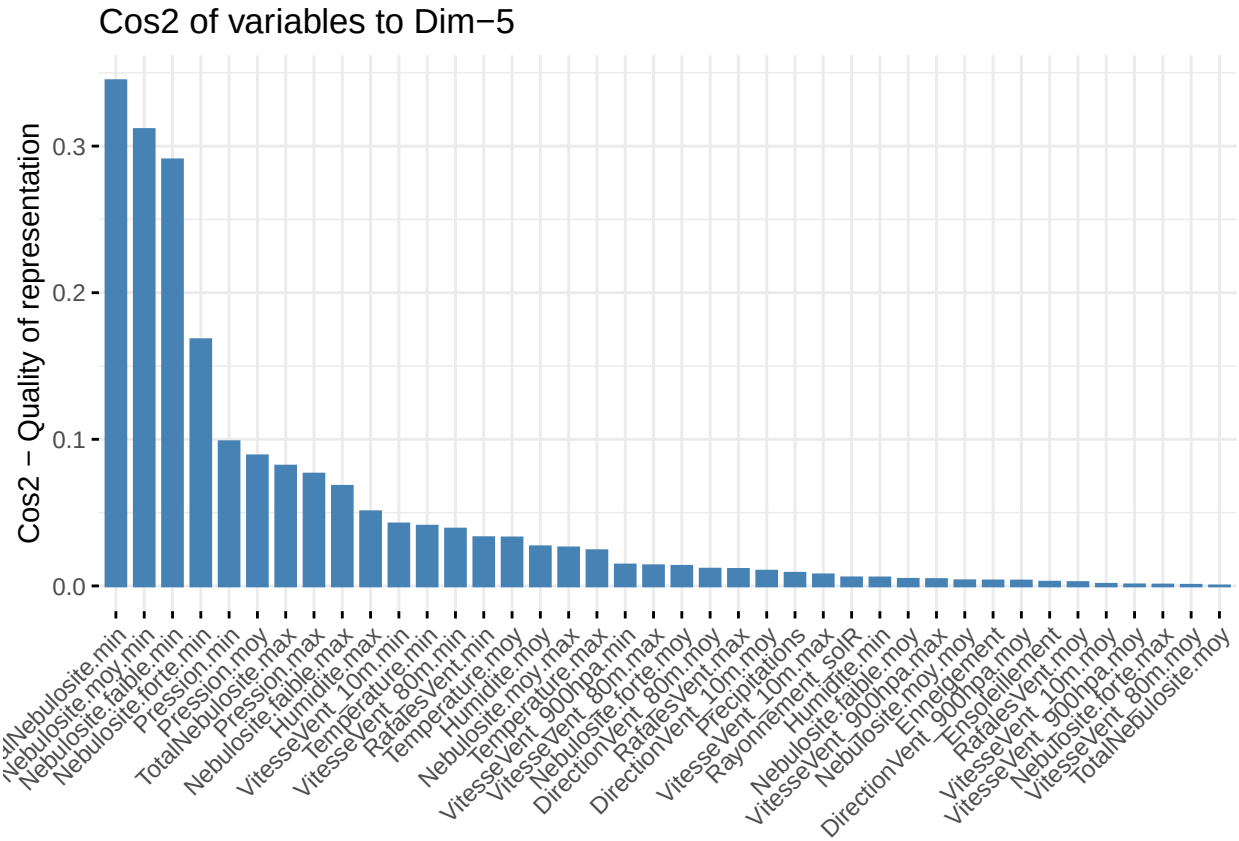
```
                select.ind=list(cos2=0.7, contrib =900),
                select.var=list(cos2=0.4))
```




```
# Contribution des variables à l'axe 5 #
fviz_contrib(acp.res,axes = 5,choice ="var")
```




```
# Qualité de représentation cos2 à l'axe 5 #
fviz_cos2(acp.res,axes = 5, choice = "var")
```




```
# Description automatique des axes #
dimdesc$Dim.5
```

Après observations, nous avons vu que seul ce plan factoriel (1,5) donne une assez bonne représentation de l'axe 5. L'analyse automatique dimdesc nous aidera à mieux l'interpréter, ainsi que les valeurs de contributions et de cos2.

L'axe 1 :
Nous avons vu plus haut son interprétation.

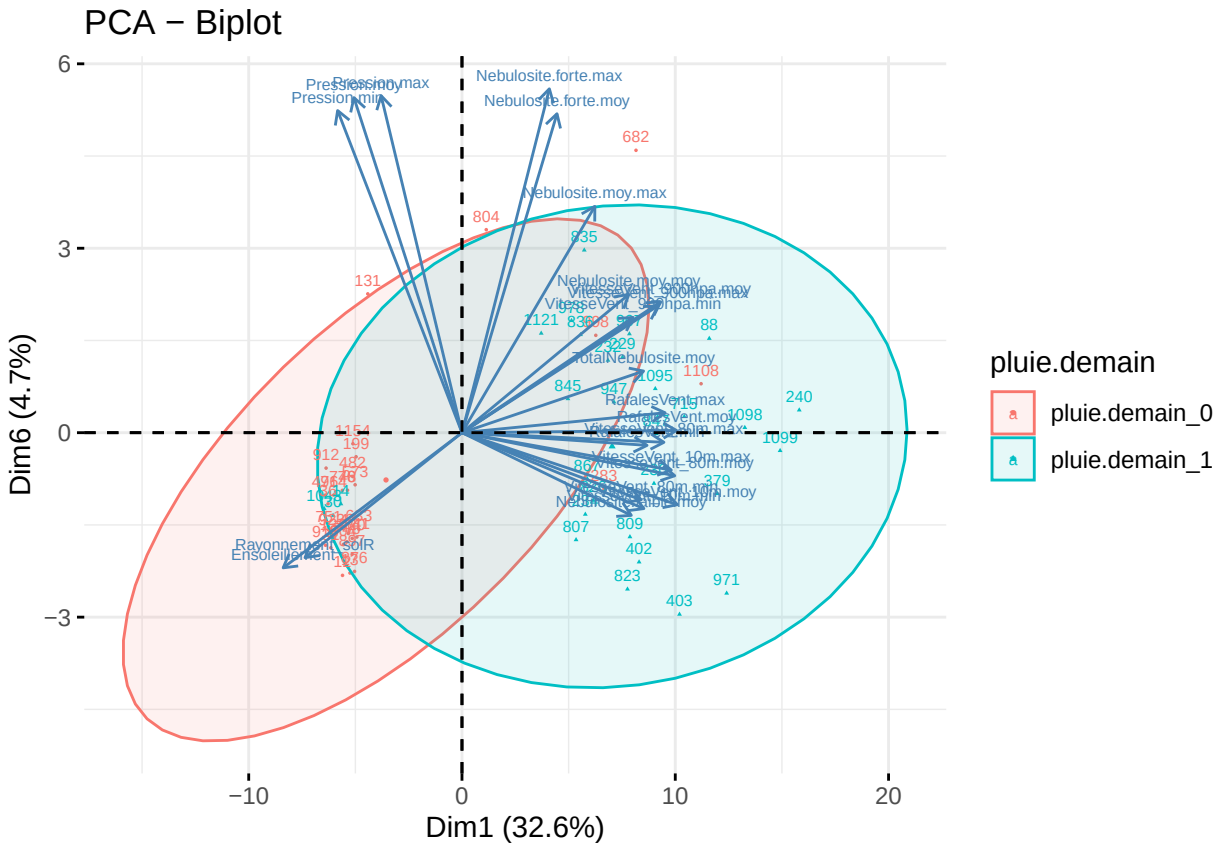
L'axe 5 :
Au vu du plan factoriel composé des axes 1 et 5 et des analyses de contribution et cos2 et dimdesc sur la dimension 5, nous notons que ce sont les variables de nébulosité min (TotalNebulosite.min, Nebulosite.moy.min, Nebulosite.faible.min) qui y contribuent le plus (presque tous au dessus de 15) et une moyenne qualité de représentation aux alentours de 0.3.

==> l'axe 5 semble donc représenter la Nébulosité min avec les variables TotalNebulosite.min, Nebulosite.moy.min et Nebulosite.faible.min (avec de fortes corrélations autour de 0.5 à 0.6).

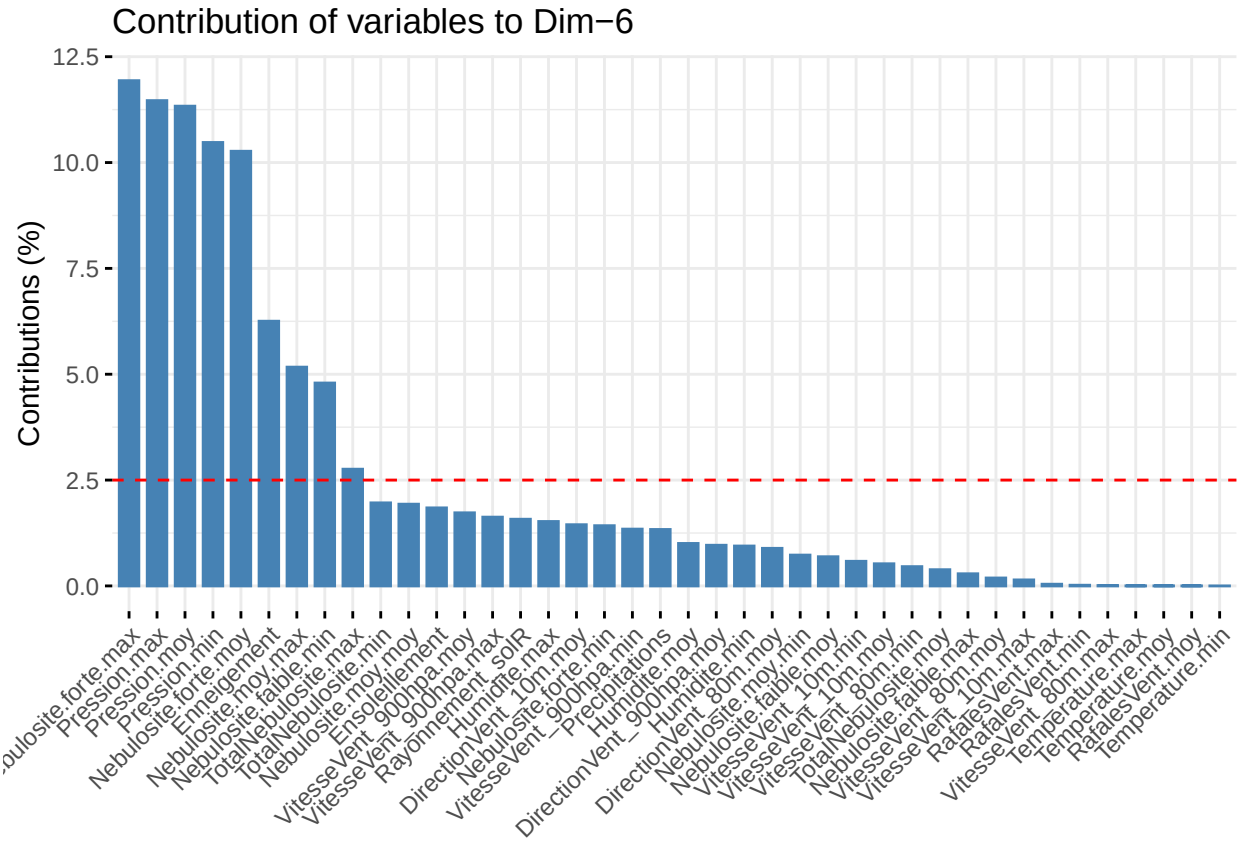
Analyse du 6ème plan factoriel (1,6)

```
# Plan axes 1 et 6 #
fviz_pca_biplot(acp.res,habillage = 45,
                axes = c(1,6),
                pointsize = 0.3,
                labelsize = 2,
```

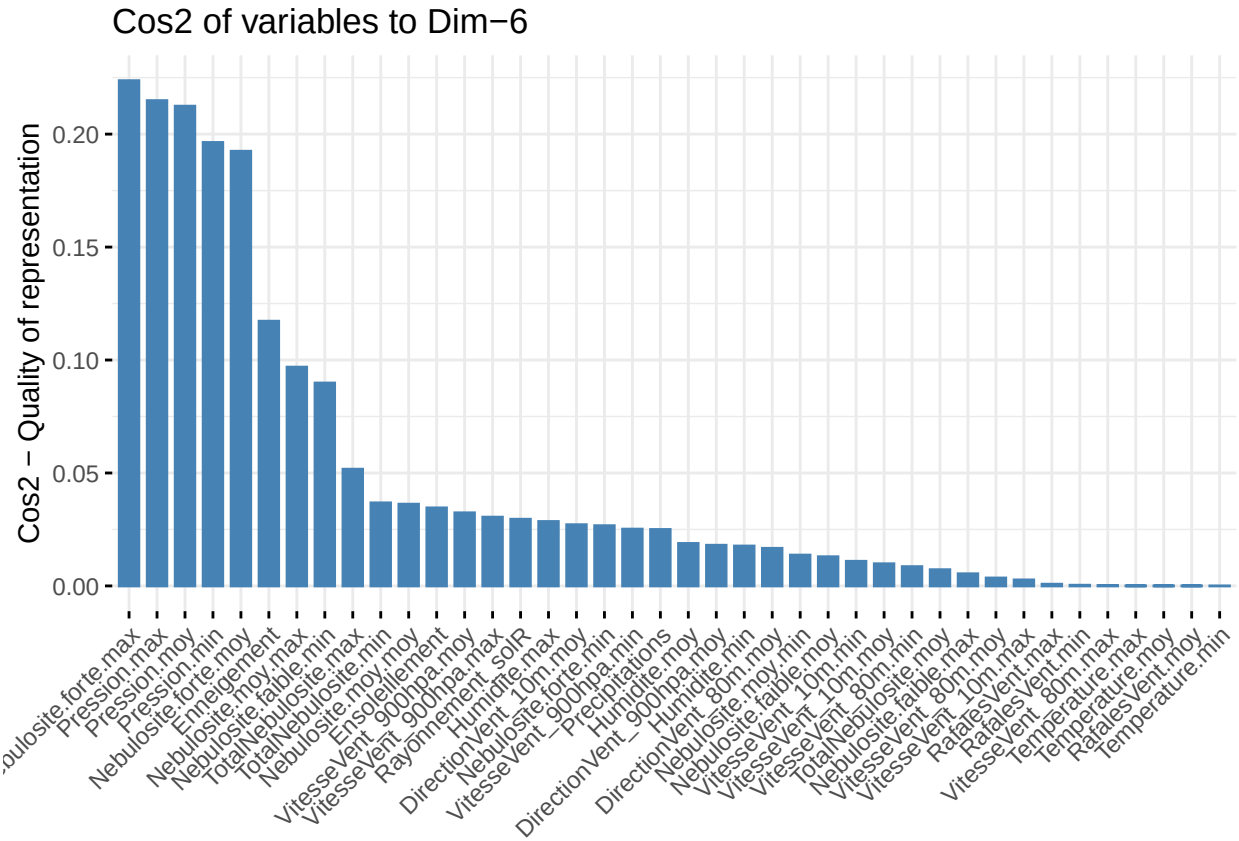
```
              addEllipses = T,
              ellipse.level=0.95,
              cex = 0.7,
              select.ind=list(cos2=0.7, contrib =900),
              select.var=list(cos2=0.3))
```




```
# Contribution des variables à l'axe 6 #
fviz_contrib(acp.res,axes = 6,choice ="var")
```




```
# Qualité de représentation cos2 à l'axe 6 #
fviz_cos2(acp.res,axes = 6, choice = "var")
```




```
# Description automatique des axes #
dimdesc$Dim.6
```

Après quelques essais, nous avons vu que ce plan factoriel (1,6) donne une assez bonne représentation de l'axe 6 pour le graphique.

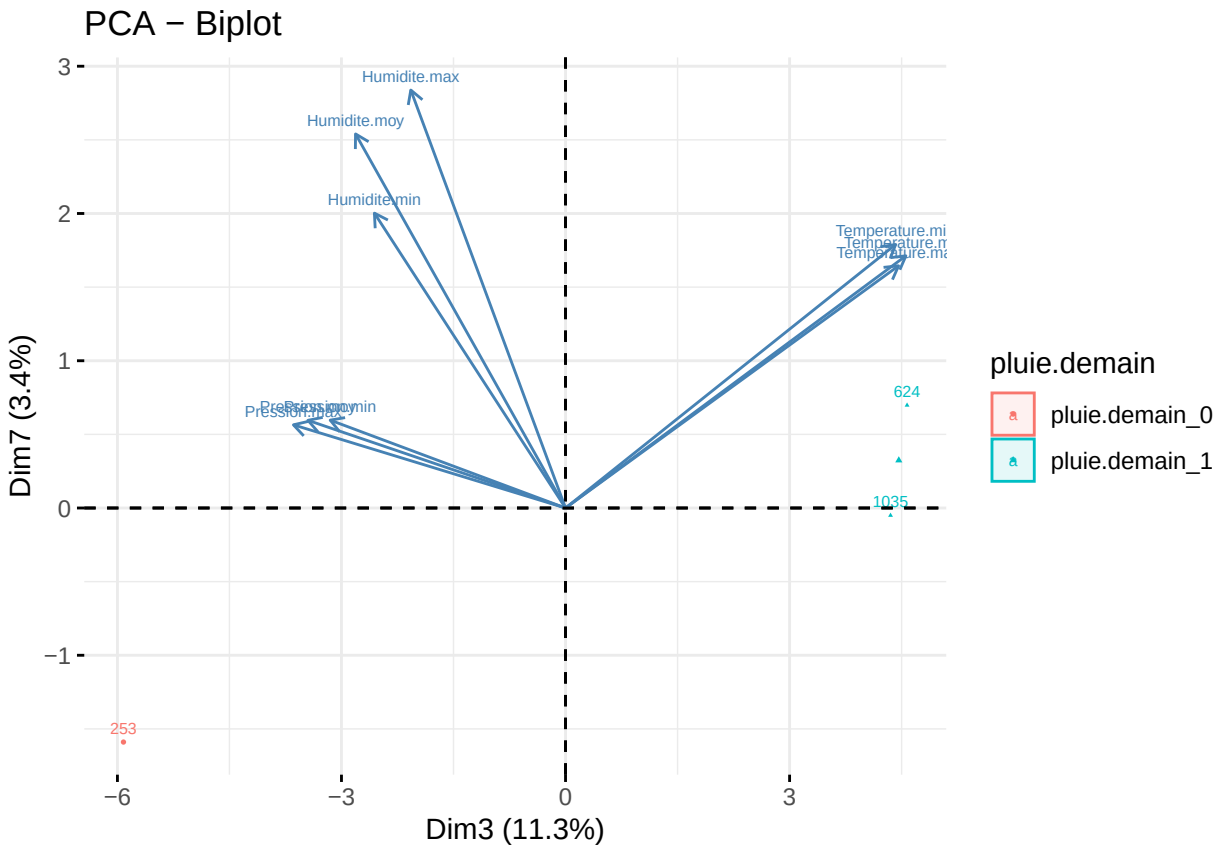
L'axe 6 :
Sur cette dimension, nous notons que ce sont les variables de nébulosité forte (Nebulosite.forte.max, Nebulosite.forte.moy) qui y contribuent le plus (presque tous au dessus de 10) mais une très moyenne qualité de représentation aux alentours de 0.2.
Aussi, les variables de pression Pression.max, Pression.moy et Pression.min (forte contribution à plus de 10) contribuent à cet axe, mais avec une qualité cos2 quasi faible.

==> l'axe 6 semble donc représenter la Nébulosité forte avec les variables Nebulosite.forte.max et Nebulosite.forte.moy et la pression (avec de moyennes corrélations autour de 0.4 à 0.5).

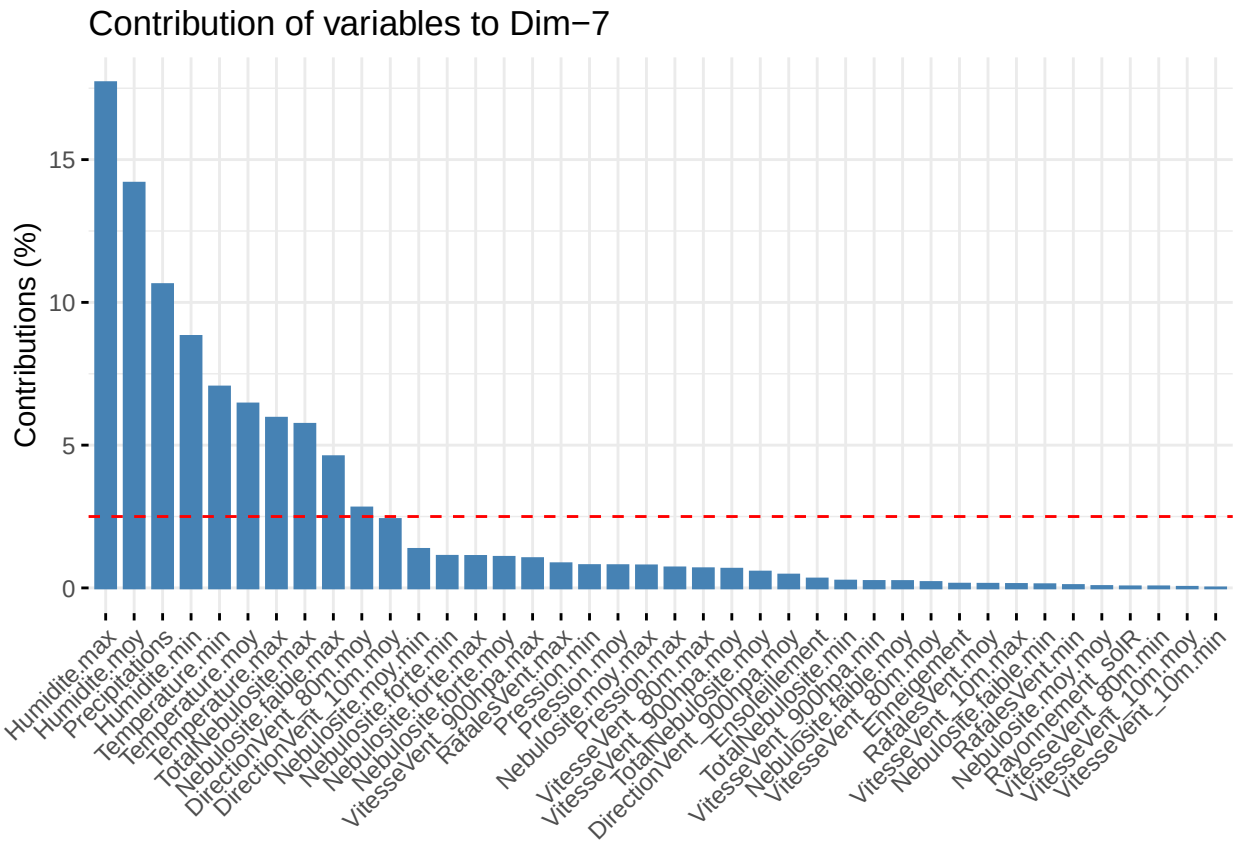
Analyse du 7ème plan factoriel (3,7)

```
# Plan axes 3 et 7 #
fviz_pca_biplot(acp.res,habillage = 45,
                axes = c(3,7),
                pointsize = 0.3,
                labelsize = 2,
                addEllipses = T,
                ellipse.level=0.95,
```

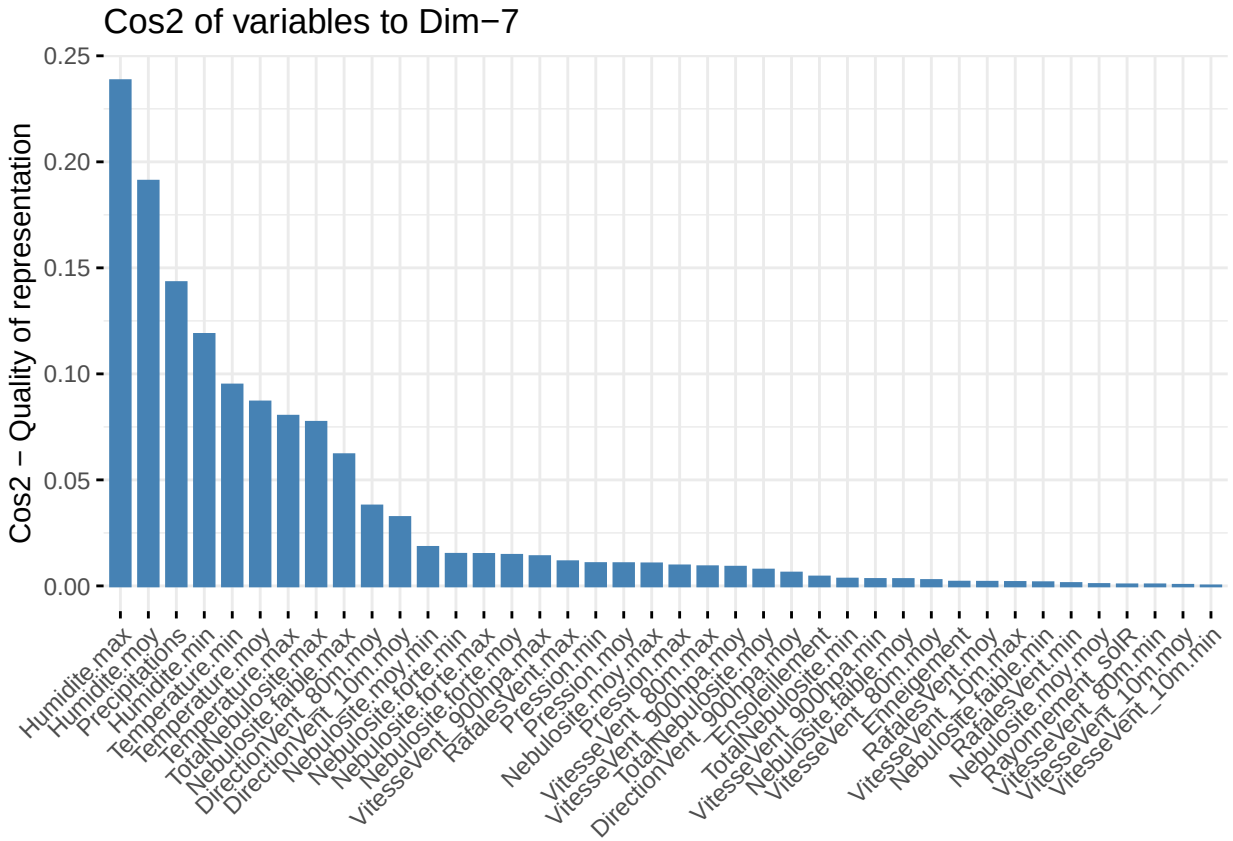
```
              cex = 0.7,
              select.ind=list(cos2=0.7, contrib =900),
              select.var=list(cos2=0.3))
```




```
# Contribution des variables à l'axe 7 #
fviz_contrib(acp.res,axes = 7,choice ="var")
```




```
# Qualité de représentation cos2 à l'axe 7 #
fviz_cos2(acp.res,axes = 7, choice = "var")
```



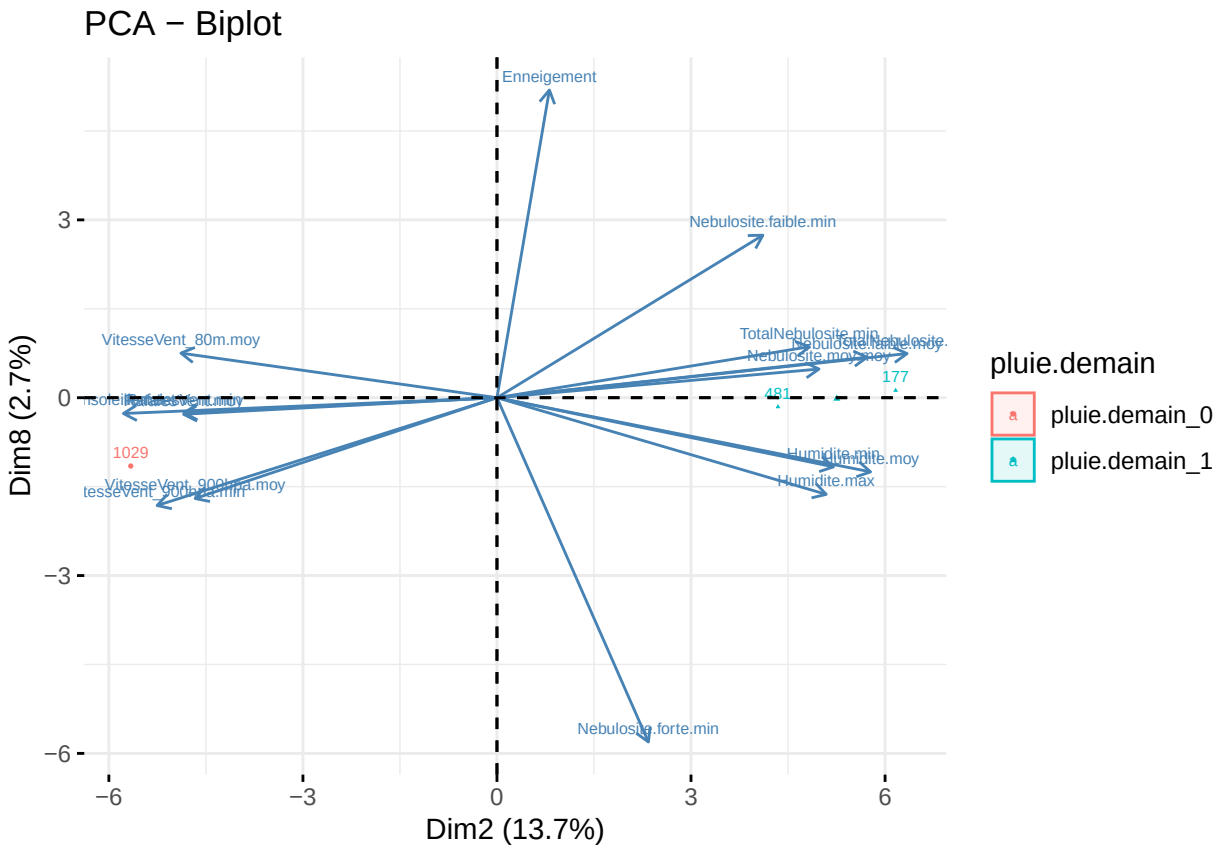

```
# Description automatique des axes #
dimdesc$Dim.7
```

Après quelques essais, nous avons vu que ce plan factoriel (3,7) donne une assez bonne représentation de l'axe 7 pour le graphique.

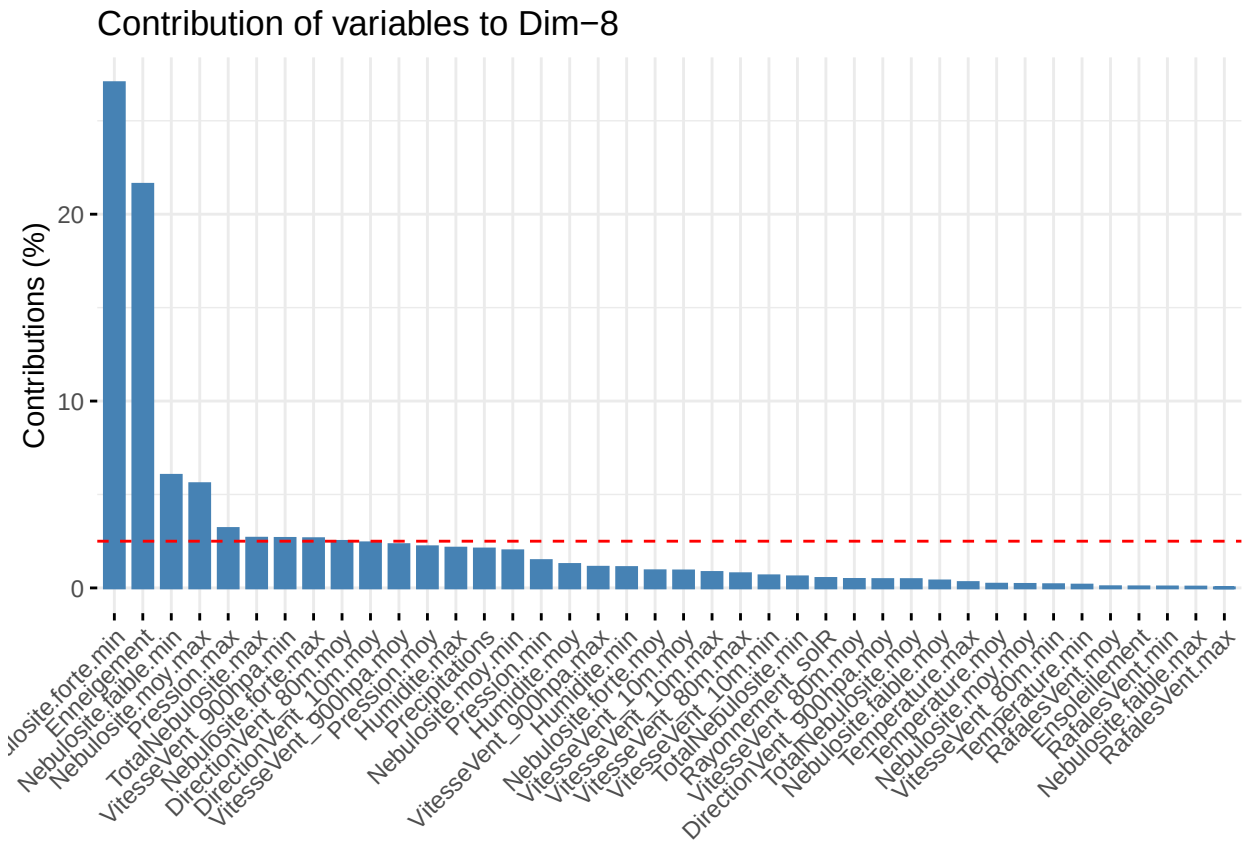
L'axe 7 :
Il semble représenter l'humidité avec les variables Humidite.max, Humidite.moy, Humidite.min et les précipitations (avec de moyennes corrélations autour de 0.4 à 0.5). Ce sont les variables ayant les meilleures contributions (de 10 jusqu'à 17) avec un cos2 relativement faible.

Analyse du 8ème plan factoriel (2,8)

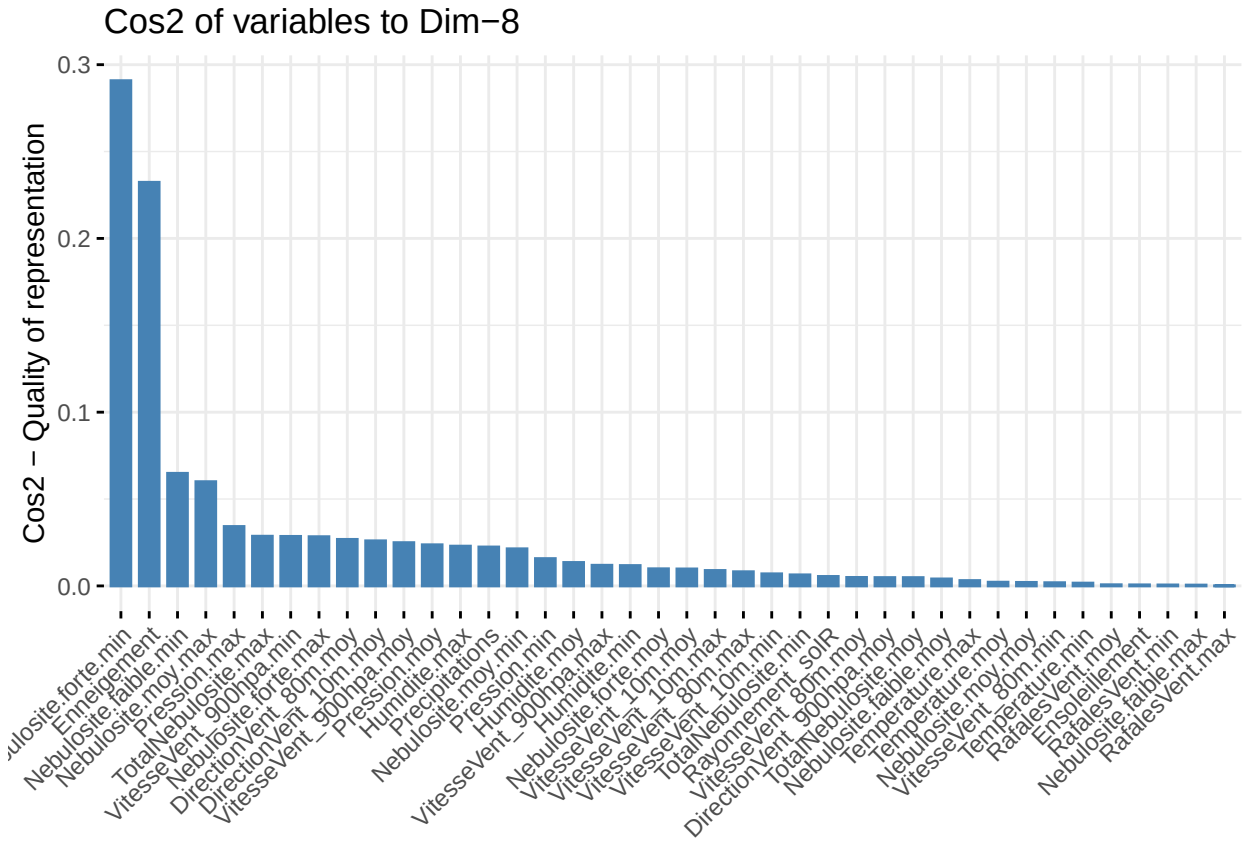
```
# Plan axes 2 et 8 #
fviz_pca_biplot(acp.res,habillage = 45,
                axes = c(2,8),
                pointsize = 0.3,
                labelsize = 2,
                addEllipses = T,
                ellipse.level=0.95,
                cex = 0.7,
                select.ind=list(cos2=0.7, contrib =900),
                select.var=list(cos2=0.2))
```




```
# Contribution des variables à l'axe 8 #
fviz_contrib(acp.res,axes = 8,choice ="var")
```




```
# Qualité de représentation cos2 à l'axe 8 #
fviz_cos2(acp.res,axes = 8, choice = "var")
```



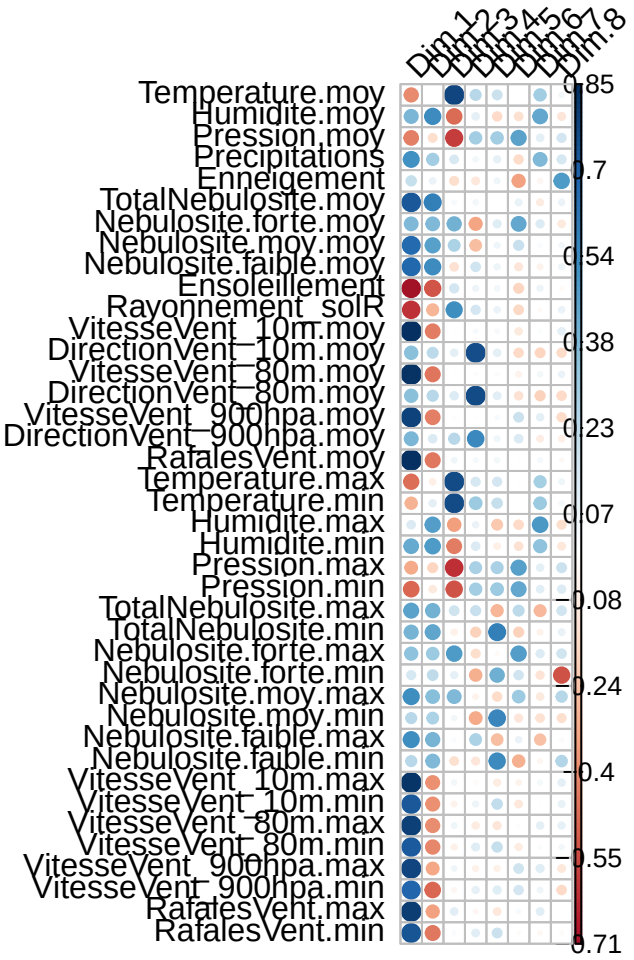

```
# Description automatique des axes #
dimdesc$Dim.8
```

L'axe 8 :
Il semble représenter la nébulosité forte min et l'enneigement (avec de bonnes corrélations opposées autour de 0.5). Ce sont les 2 variables ayant les meilleures contributions (à 21 et 27 avec un cos2 autour de 0.3.

==> l'axe 8 nous informe donc sur les journées enneigées (du côté haut du graphe) qui s'oppose aux journées avec de la couverture nuageuse forte min. (côté bas)

Pour résumer, nous voyons à travers le graphe ci-dessous axe par axe, les variables qui obtiennent les coefficients de corrélation les plus significatifs (positif comme négatif).

```
library (corrplot)
corrplot(acp.res$var$coord[,1:8], is.corr = FALSE, tl.col="black", tl.srt=45)
```



Mes déductions sont les suivantes :

Axe 1 : VitesseVent_10m.moy+VitesseVent_80m.moy+VitesseVent_900hpa.moy+RafalesVent.moy+VitesseVent_10m.max+ VitesseVent_10m.min+VitesseVent_80m.max+VitesseVent_80m.min+ VitesseVent_900hpa.max+VitesseVent_900hpa.min+ RafalesVent.max+RafalesVent.min+Nebulosite.moy.moy+ Nebulosite.faible.moy+ TotalNebulosite.moy+Ensoleillement.

Axe 2 : il n'y a pas vraiment de variables dominantes. Toutes celles qui apparaissaient vont contribuer plus dans les autres dimensions.

Axe 3 : Temperature.moy+Temperature.max+Temperature.min+Rayonnement_solR.

Axe 4 : DirectionVent_10m.moy+DirectionVent_80m.moy+DirectionVent_900hpa.moy.

Axe 5 : TotalNebulosite.min+Nebulosite.moy.min+Nebulosite.faible.min.

Axe 6 : Nebulosite.forte.max+Nebulosite.forte.moy+Pression.moy+Pression.max+Pression.min.

Axe 7 : Humidite.max+Humidite.moy+Humidite.min+Precipitations.

Axe 8 : Enneigement+Nebulosite.forte.min.

Nous allons à présent préparer les nouvelles matrices tr.meteotrain_proj et test.meteotrain_proj qui vont servir à la régression sur les composantes principales.
A l'aide des résultats de l'ACP, nous récupérons la projection des individus sur les axes factoriels, les valeurs propres et également la projection des variables sur les axes factoriels.
Avant de projeter les données test.meteotrain sur les axes, il faut au préalable procéder à la standardisation de celle-ci (par scale).

Il est aisé ensuite de projeter cette nouvelle matrice centrée-réduite sur les axes factoriels choisis par l'ACP (8 composantes principales).

Ainsi, nous avons une nouvelle matrice tr.meteotrain_proj contenant les individus projetés sur les composantes principales de l'ACP, et la nouvelle matrice de test test.meteotrain_proj projetés sur les axes factoriels.

Avant la mise en oeuvre de la régression logistique, il conviendra de remettre la variable cible pluie.demain dans chacune des matrices, qui est la variable que nous cherchons à prédire.

```
# Récupération des individus et variables projetés sur
#les axes factoriels #
tr.meteotrain_proj <- as.data.frame(acp.res$ind$coord)
acp.res.var.coord <- as.data.frame(acp.res$var$coord)
acp.eig.val = as.data.frame(acp.res$eig) # les valeurs propres #

# Standardisation de la base test.meteotrain sans les variables
# supplémentaires de l'acp #
test.meteotrain.std <- as.data.frame(scale(test.meteotrain[,-c(1:4,45)],
                        center = TRUE,scale = TRUE))

# Projection de la matrice test.meteotrain sur les axes factoriels de ACP #
pdtmat = as.matrix(test.meteotrain.std) %*% as.matrix(acp.res.var.coord)

acp.test.proj <- matrix (rep(0, nrow(test.meteotrain.std)*
                               ncol(acp.res$ind$coord)),
                         nrow(test.meteotrain.std),
                         ncol(acp.res$ind$coord))
for (i in 1:8){
  acp.test.proj[,i] <- pdtmat[,i]/sqrt(acp.eig.val[i,1])
}

test.meteotrain_proj = as.data.frame(acp.test.proj)

# tr.meteotrain_proj  nouvelle matrice de tr.meteotrain
# test.meteotrain_proj nouvelle matrice de test.meteotrain

# Ajout de la variable cible pluie.demain
tr.meteotrain_proj$pluie.demain = tr.meteotrain$pluie.demain
test.meteotrain_proj$pluie.demain = test.meteotrain$pluie.demain

# Renommage des colonnes
colnames(test.meteotrain_proj) = colnames(tr.meteotrain_proj)
```

Procédons désormais à la régression logistique sur la matrice tr.meteotrain_proj suivant les 8 composantes principales. Avec la réduction des variables de 41 à 8, nous avons quand même bien épuré notre base.
Nous allons opter pour les méthodes pas à pas Forward, Backward et Stepwise (avec les critères AIC, BIC) et de recherche exhaustive de leaps (critère Cp de Mallows) pour construire les modèles.

En premier modèle, nous conservons déjà le modèle saturé lg.modelsat_acp, qui contient les 8 composantes principales.
Modèle lg.FWDAicModel_acp

```
# Spécifions de nouveau le modèle saturé lg.modelsat.acp et le modèle de départ lg.modelnull.acp (qui s

lg.modelsat.acp = glm(
 pluie.demain ~ ., family = binomial,data = tr.meteotrain_proj)
AIC(lg.modelsat.acp)
BIC(lg.modelsat.acp)

lg.modelnull.acp = glm(
 pluie.demain ~ 1, family = binomial,data = tr.meteotrain_proj)


lg.fwdaic.model.acp <- stepAIC(lg.modelnull.acp,
                               scope=list(upper=lg.modelsat.acp),
data = tr.meteotrain_proj, direction = "forward")

coef(lg.fwdaic.model.acp,which.min(summary(lg.fwdaic.model.acp)$aic))
```

Le modèle retenu à 4 variables est celui-ci :

```
lg.FWDAicModel_acp <- glm(formula = pluie.demain ~Dim.3 + Dim.1 + Dim.2 + Dim.5, family = binomial,
data = tr.meteotrain_proj)

summary(lg.FWDAicModel_acp)
AIC(lg.FWDAicModel_acp)
```

Son AIC = 1065.4

Modèle lg.BWDAicModel_acp

```
lg.bwdaic.model.acp <- stepAIC(lg.modelsat.acp,data = tr.meteotrain_proj,
                               direction = "backward")

coef(lg.bwdaic.model.acp,which.min(summary(lg.bwdaic.model.acp)$aic))
```

Le modèle retenu est celui-ci :

```
lg.BWDAicModel_acp <- glm(formula = pluie.demain ~Dim.3 + Dim.1 + Dim.2 + Dim.5, family = binomial,
data = tr.meteotrain_proj)

summary(lg.BWDAicModel_acp)
AIC(lg.BWDAicModel_acp)
```

C'est le même modèle que celui du Forward, avec le critère AIC.

Modèle lg.STEPWAicModel_acp

```
lg.stepwaic.model.acp <- stepAIC(lg.modelnull.acp,
                                 scope=list(upper=lg.modelsat.acp),
data = tr.meteotrain_proj, direction = "both")

coef(lg.stepwaic.model.acp,which.min(summary(lg.stepwaic.model.acp)$aic))
```

Le modèle retenu est le même que les 2 précédents.

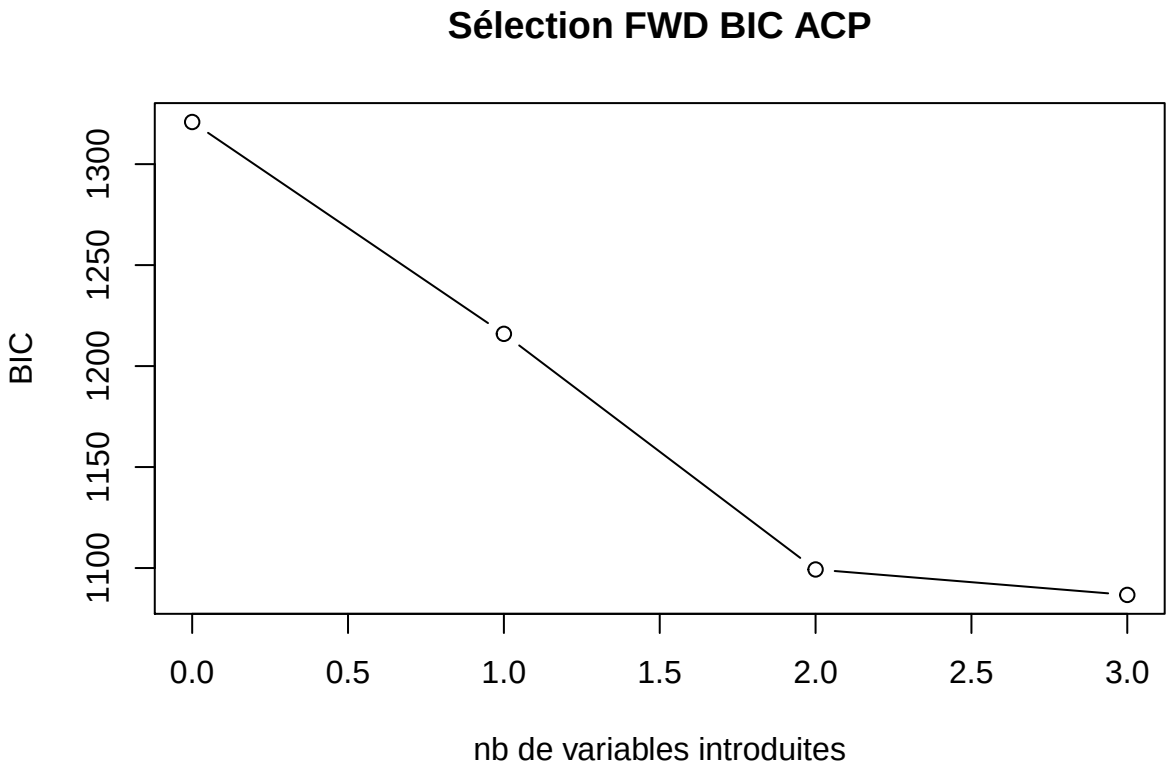
```
lg.STEPWAicModel_acp <- glm(formula = pluie.demain ~ Dim.3 + Dim.1 + Dim.2 +
                            Dim.5, family = binomial,
data = tr.meteotrain_proj)

summary(lg.STEPWAicModel_acp)
AIC(lg.STEPWAicModel_acp)
```

Peu importe la méthode Forward, Backward ou Stepwise, ce sont les mêmes variables qui sont sélectionnées pour construire le modèle au meilleur AIC (il vaut 1065.4).

Regardons le critère BIC.

Modèle lg.FWDBicModel_acp

```
lg.fwdbic.acp.model <- stepAIC(lg.modelnull.acp,scope=list(lower=lg.modelnull.acp,
                                                           upper=lg.modelsat.acp),
data = tr.meteotrain_proj, direction = "forward", k=log(nrow(tr.meteotrain_proj)))

summary(lg.fwdbic.acp.model)


coef(lg.fwdbic.acp.model,which.min(summary(lg.fwdbic.acp.model)$aic))
print(lg.fwdbic.acp.model$anova)

plot(0:(nrow(lg.fwdbic.acp.model$anova)-1),lg.fwdbic.acp.model$anova[,"AIC"],type = "b",xlab = "nb de va
     main = "Sélection FWD BIC ACP")
```



Dès la 2ème composante ajoutée, on voit qu'on améliore assez peu le BIC. Finalement, le modèle lg.FWDBicModel_acp retenu a 3 variables, c'est le suivant :

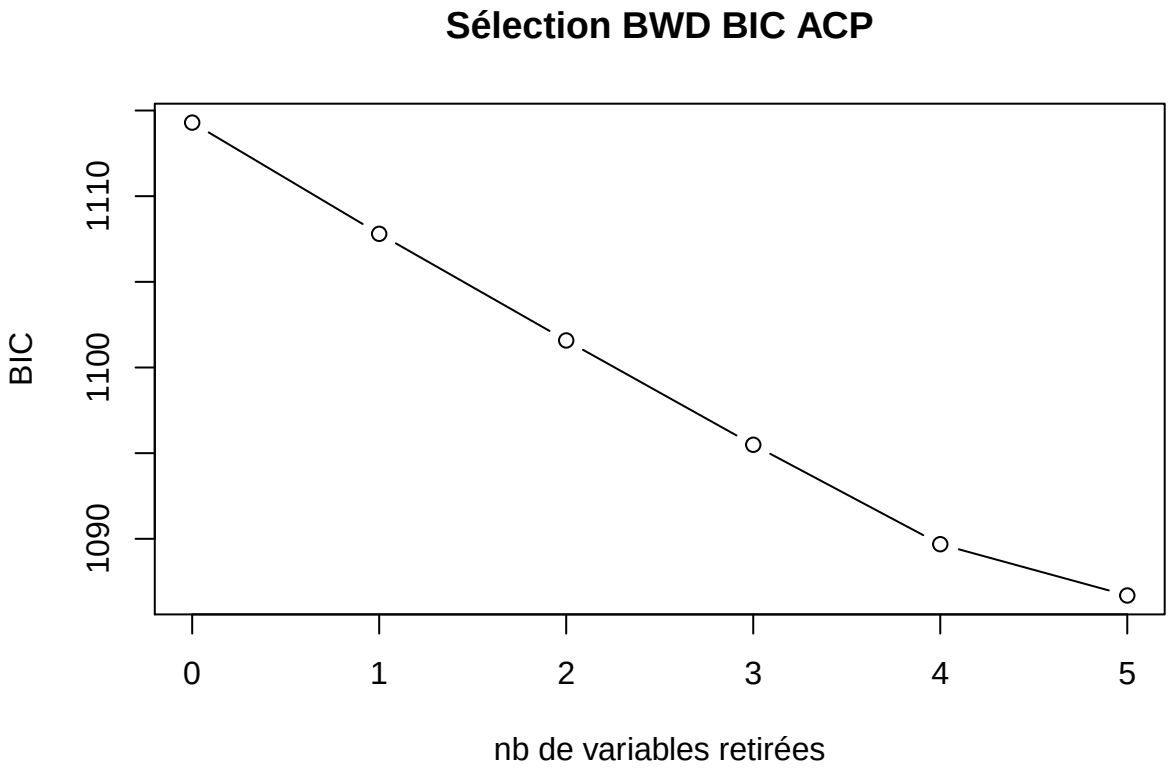
```
# Bic #

lg.FWDBicModel_acp <- glm(formula = pluie.demain ~
                          Dim.3 +
                          Dim.1 +
                          Dim.2
                          , family = binomial,
            data = tr.meteotrain_proj)

summary(lg.FWDBicModel_acp)
BIC(lg.FWDBicModel_acp)
```

Le modèle retenu a un Bic = 1086.7.

Modèle lg.BWDBicModel_acp

```
lg.bwdbic.acp.model <- stepAIC(lg.modelsat.acp,data = tr.meteotrain_proj,
                           direction = "backward",k=log(nrow(tr.meteotrain_proj)))
summary(lg.bwdbic.acp.model)


coef(lg.bwdbic.acp.model,which.min(summary(lg.bwdbic.acp.model)$aic))
print(lg.bwdbic.acp.model$anova)
```

```
plot(0:(nrow(lg.bwdbic.acp.model$anova)-1),lg.bwdbic.acp.model$anova[,"AIC"],type = "b",
     xlab = "nb de variables retirées",ylab = "BIC",
     main = "Sélection BWD BIC ACP")
```



Nous avons donc retiré 5 composantes pour obtenir un modèle au BIC minimum. Le modèle lg.BWDBicModel_acp retenu a 3 composantes(les mêmes que précédemment) et est le suivant :

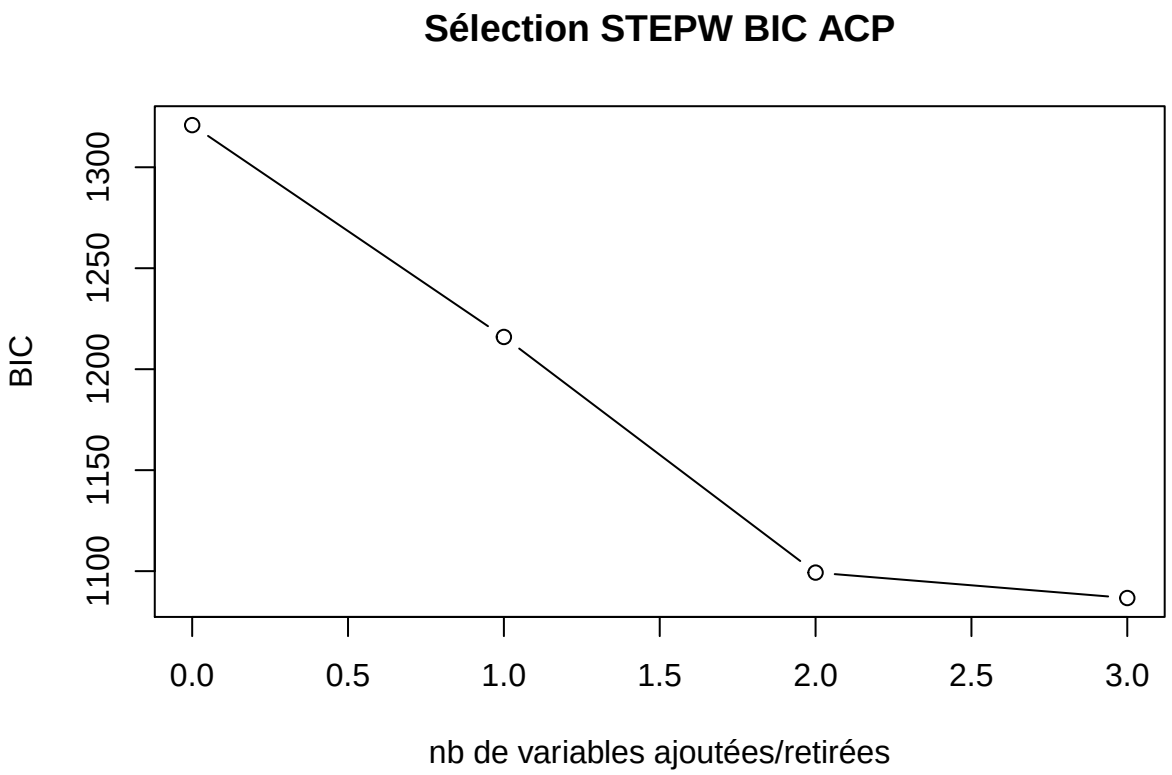
```
# Bic #

lg.BWDBicModel_acp <- glm(formula = pluie.demain ~
                          Dim.3 +
                          Dim.1 +
                          Dim.2
                           , family = binomial,
            data = tr.meteotrain_proj)

summary(lg.BWDBicModel_acp)
BIC(lg.BWDBicModel_acp)
```

Le modèle lg.BWDBicModel_acp retenu a les 3 mêmes composantes, un Bic = 1086.7.

Modèle lg.STEPWBicModel_acp

```
lg.stepwbic.acp.model <- stepAIC(lg.modelnull.acp,
                                 scope=list(upper=lg.modelsat.acp),
data = tr.meteotrain_proj, direction = "both", k=log(nrow(tr.meteotrain_proj)))
```

```
summary(lg.stepwbic.acp.model)

coef(lg.stepwbic.acp.model,which.min(summary(lg.stepwbic.acp.model)$aic))
print(lg.stepwbic.acp.model$anova)

plot(0:(nrow(lg.stepwbic.acp.model$anova)-1),lg.stepwbic.acp.model$anova[,"AIC"],type = "b",xlab = "nb d
     main = "Sélection STEPW BIC ACP")
```



Le modèle lg.STEPWBicModel retenu est exactement le même :

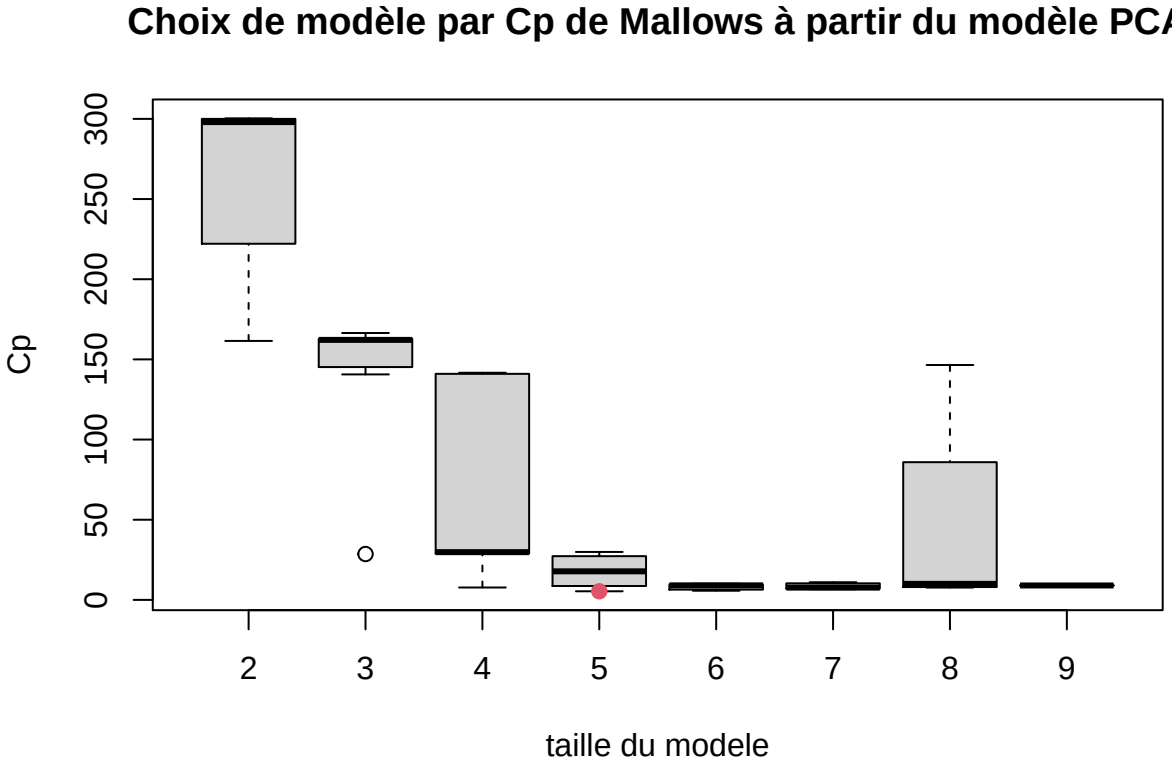
```
# Bic #

lg.STEPWBicModel_acp <- glm(formula = pluie.demain ~
                            Dim.3 +
                            Dim.1 +
                            Dim.2
                            , family = binomial,
                          data = tr.meteotrain_proj)
summary(lg.STEPWBicModel_acp)
BIC(lg.STEPWBicModel_acp)
```

Peu importe la méthode ascendante, descendante ou dans les 2 sens, le modèle retenu est le même avec 3 composantes. Rappelons que par définition, le critère BIC retiendra le modèle le plus parcimonieux, il n'est donc pas étonnant de voir nos résultats ainsi.

Pour tenter le critère du Cp de Mallows, nous pouvons essayer avec la méthode leaps du package leaps. Nous noterons que cette méthode n'est recommandée qu'en présence d'une quinzaine de variables au plus. Cela paraît donc raisonnable si nous l'utilisons avec la matrice réduite issue de notre ACP.

Modele lg.FWDCpModel_acp

```
colnum <- which(names(tr.meteotrain_proj) %in%
  c("Dim.1","Dim.2","Dim.3","Dim.4", "Dim.5","Dim.6","Dim.7","Dim.8"))
library(leaps)

lg.fwdcp.acp.model<-leaps(tr.meteotrain_proj[,colnum],
                          tr.meteotrain_proj[,9],method="Cp")

best.fwdcp = which.min(lg.fwdcp.acp.model$Cp)

best.fwd.cp.acp=lg.fwdcp.acp.model$which[best.fwdcp,]
boxplot(lg.fwdcp.acp.model$Cp ~lg.fwdcp.acp.model$size,xlab = "taille du modele",ylab = 'Cp')
points(lg.fwdcp.acp.model$size[best.fwdcp]-1,lg.fwdcp.acp.model$Cp[best.fwdcp],pch=20,col=2,cex=1.5)
title("Choix de modèle par Cp de Mallows à partir du modèle PCA")

print(names(tr.meteotrain_proj)[colnum][best.fwd.cp.acp])
```



On voit qu'on en revient au modèle sélectionné par le Backward/Forward/Stepwise avec Aic.

Pour aller plus loin (mais ceci sort du cadre du cours), nous pourrions faire les méthode de filtres de "ranking" en amont des variables, associée à la méthode de "Wrapper". Mais cela ne sera peut-être pas étudié dans le cadre de la formation donc nous ne pourrons l'expérimenter.

Retenons ainsi les 3 modèles suivants dont nous testerons la validité avec les autres modèles précédents :
- lg.FWDAicModel_acp (qui est le même pour BWD et STEPW, et également celle avec le critère de Cp de

Mallows)
- lg.FWDBicModel_acp (qui est le même pour BWD et STEPW)
- lg.modelsat_acp (qui correspond au modèle saturé avec toutes les composantes principales de l'Acp)

Table 1: Tableau récapitulatif des modèles

Methode	Modele	nbvar	valAIC	valBIC
Forward	lg.FWDAicModel	13	1045.16032197021	1113.12128500178
Forward	lg.FWDBicModel	5	1066.83520518449	1095.96133219802
Backward	lg.BWDAicModel	17	1033.74243142297	1121.12081246356
Backward	lg.BWDBicModel	9	1049.67299261325	1098.2165376358
Stepwise	lg.STEPWAicModel	11	1043.89770489857	1102.14995892563
Stepwise	lg.STEPWBicModel	5	1043.89770489857	1095.96133219802
Forward ACP	lg.FWDAicModel_acp	4	1065.41357976079	1089.68535227207
Forward ACP	lg.FWDBicModel_acp	3	1067.27404651361	1086.69146452263
Total ACP	lg.modelsat_acp	8	1070.60538500963	1114.29457552993

Faisons un tableau récapitulatif de tous nos modèles.

II.2.6 Tableau récapitulatif des modèles

```
Modele = c("lg.FWDAicModel","lg.FWDBicModel","lg.BWDAicModel","lg.BWDBicModel","lg.STEPWAicModel","lg.S
nbvar = c(      13,5,
                17,9,
                11,5,4,3,8)
valAIC = c(AIC(lg.FWDAicModel),AIC(lg.FWDBicModel),
           AIC(lg.BWDAicModel),AIC(lg.BWDBicModel),
           AIC(lg.STEPWAicModel),AIC(lg.STEPWAicModel),
           AIC(lg.FWDAicModel_acp),AIC(lg.FWDBicModel_acp),
           AIC(lg.modelsat.acp))
valBIC = c(BIC(lg.FWDAicModel),BIC(lg.FWDBicModel),
           BIC(lg.BWDAicModel),BIC(lg.BWDBicModel),
           BIC(lg.STEPWAicModel),BIC(lg.STEPWBicModel),
           BIC(lg.FWDAicModel_acp),BIC(lg.FWDBicModel_acp),
           BIC(lg.modelsat.acp))

tablo=cbind(Modele,nbvar,valAIC,valBIC)

tabl.res = as.data.frame(tablo)
```

```
library(kableExtra)
library(tidyverse)


Methode=c(rep("Forward",2), rep("Backward",2), rep("Stepwise",2),rep("Forward ACP",2),"Total ACP")

tabl.res = cbind(Methode,tablo)

knitr::kable(tabl.res,
             caption = "Tableau récapitulatif des modèles")
```

Avant de décider du modèle à retenir, il faut au préalable évaluer la performance de prédiction de chaque modèle sur la base de validation et sa capacité à ajuster correctement les données. Ici, nous voyons que

le modèle ayant l'Aic le plus petit est le lg.BWDAicModel. Quant au plus petit Bic, c'est le modèle lg.FWDBicModel_acp qui le possède. Rien à ce stade ne nous permet de désigner quel est le meilleur modèle. En effet, ce n'est pas parce que la méthode Backward (critère Aic) a déterminé un modèle avec un Aic petit que c'est d'emblée le meilleur modèle à choisir. Il nous faut évaluer les prédictions pour juger. Nous allons dans le paragraphe suivant évaluer les modèles.

II.3 Première évaluation des modèles (Wald, pseudos R2)

```
# Les 9 modèles construits #

lg.FWDAicModel <- glm(formula = pluie.demain ~ Nebulosite.moy.max +
Pression.min + Temperature.min + RafalesVent.max + Nebulosite.faible.max + Nebulosite.forte.max + Tempe

lg.BWDAicModel <- glm(formula = pluie.demain ~ Pression.moy +
Precipitations + Enneigement + Nebulosite.moy.moy + VitesseVent_80m.moy + DirectionVent_80m.moy + Direc
Temperature.min + Pression.max + Pression.min + TotalNebulosite.min +
Nebulosite.forte.max + Nebulosite.moy.max + Nebulosite.faible.max +
VitesseVent_10m.max + VitesseVent_10m.min, family = binomial,
data = tr.meteotrain1)

lg.STEPWAicModel <- glm(formula = pluie.demain ~ Nebulosite.moy.max +
Pression.min + RafalesVent.max + Nebulosite.faible.max + Temperature.max + DirectionVent_900hpa.moy + T
family = binomial, data = tr.meteotrain1)

lg.FWDBicModel <- glm(formula = pluie.demain ~ Nebulosite.moy.moy +
Temperature.min + Pression.min + Nebulosite.faible.max +RafalesVent.max,
family = binomial, data = tr.meteotrain1)


lg.BWDBicModel <- glm(formula = pluie.demain ~ Pression.moy+
VitesseVent_80m.moy +Temperature.max + Pression.max + Pression.min+            Nebulosite.moy.max +Nebul
  VitesseVent_10m.min, family = binomial,
data = tr.meteotrain1)

lg.STEPWBicModel <- glm(formula = pluie.demain ~ Nebulosite.moy.moy +
Temperature.min + Pression.min+Nebulosite.moy.max +Nebulosite.faible.max+
RafalesVent.max, family = binomial, data = tr.meteotrain1)

lg.FWDAicModel_acp <- glm(formula = pluie.demain ~Dim.3 +
                            Dim.1 + Dim.2 + Dim.5,
family = binomial, data = tr.meteotrain_proj)

lg.FWDBicModel_acp <- glm(formula = pluie.demain ~ Dim.1+Dim.2+Dim.3,
family = binomial, data = tr.meteotrain_proj)

lg.modelsat_acp <- glm(formula = pluie.demain ~Dim.1 + Dim.2 + Dim.3 +Dim.4+ Dim.5+Dim.6+Dim.7+Dim.8,
family = binomial, data = tr.meteotrain_proj)
```

II.3.1 Test de Wald (évaluation de la normalité asymptotique des estimateurs)

Pour chaque modèle, nous allons rapidement étudier les significativités de Wald des variables, au seuil de 5%.

```
summary(lg.FWDAicModel)
summary(lg.FWDBicModel)
summary(lg.BWDAicModel)
summary(lg.BWDBicModel)
summary(lg.STEPWAicModel)
summary(lg.STEPWBicModel)
summary(lg.FWDAicModel_acp)
summary(lg.FWDBicModel_acp)
summary(lg.modelsat_acp)
```

Nous pouvons passer en revue chacun des modèles (sans compter l'intercept). Au seuil de 5%, nous avons :

- lg.FWDAicModel : 8 significatives sur 13
- lg.FWDBicModel : toutes les 5 significatives
- lg.BWDAicModel : 10 significatives sur 17
- lg.BWDBicModel : toutes les 9 significatives
- lg.STEPWAicModel : 9 significatives sur 11
- lg.STEPWBicModel : toutes les 6 significatives
- lg.FWDAicModel_acp : toutes significatives
- lg.FWDBicModel_acp : toutes significatives
- lg.modelsat_acp : 4 significatives sur 8

Avec cette analyse globale, nous voyons que les variables sélectionnées sont presque toutes significatives au sens de Wald dans les modèles utilisant comme critères le Bic et les 2 modèles de l'Acp (hormis le modèle saturé de l'Acp). Ceux utilisant le critère de l'Aic ont des variables significatives pour plus de la moitié d'entre elles.

Si nous regardons les Pseudos R2 :

II.3.2 Pseudos R2

```
library(DescTools)
PseudoR2(lg.FWDAicModel,which=c("McFadden","CoxSnell"))
PseudoR2(lg.FWDBicModel,which=c("McFadden","CoxSnell"))
PseudoR2(lg.BWDAicModel,which=c("McFadden","CoxSnell"))
PseudoR2(lg.BWDBicModel,which=c("McFadden","CoxSnell"))
PseudoR2(lg.STEPWAicModel,which=c("McFadden","CoxSnell"))
PseudoR2(lg.STEPWBicModel,which=c("McFadden","CoxSnell"))
PseudoR2(lg.FWDAicModel_acp,which=c("McFadden","CoxSnell"))
PseudoR2(lg.FWDBicModel_acp,which=c("McFadden","CoxSnell"))
PseudoR2(lg.modelsat_acp,which=c("McFadden","CoxSnell"))
```

```
##  McFadden  CoxSnell
## 0.2259056 0.2688395
##  McFadden  CoxSnell
## 0.1889365 0.2303969
##  McFadden  CoxSnell
## 0.2406833 0.2836636
## McFadden CoxSnell
## 0.216383 0.259125
##  McFadden  CoxSnell
## 0.2238223 0.2667252
##  McFadden  CoxSnell
## 0.2011723 0.2433392
##  McFadden  CoxSnell
## 0.1967935 0.2387327
##  McFadden  CoxSnell
## 0.1938555 0.2356264
##  McFadden  CoxSnell
## 0.1989306 0.2409845
```

Les résultats sur les pseudos R2 ne nous permettent pas de prendre de décision claire. Tous les pseudos R2 McFadden sont dans la fourchette [0.19,0.24] et ceux de Cox and Snell sont dans la fourchette [0.23,0.28]. Nous savons que théoriquement le pseudo R2 doit être compris dans [0,1], plus il est proche de 1, meilleur il est. Nous allons approfondir l'analyse avec la performance prédictive de chaque modèle.

II.4 Evaluation et validation des modèles

Nous allons évaluer la performance prédictive des modèles ainsi que leur qualité d'ajustement aux données Meteotest.

II.4.1 Scores d'affectation et erreurs de prédiction

Calculons les erreurs de prédictions sur l'échantillon de validation avec les 9 modèles construits.

Nous allons passer au crible les 9 modèles en passant à la loupe leur capacité prédictive. Il existe plusieurs façons d'obtenir les erreurs de prédictions. Calculons ici : le MAE (mean absolute error), le RMSEP (root mean squared error of prediction) et le MSEP (mean squared error of prediction) pour voir quel est le modèle qui propose le minimum sur chaque critère. Ces 3 critères mesurent respectivement la moyenne, l'écart-type et la variance des résidus. Plus leur valeur est faible, meilleure est le modèle en terme de calculs d'erreurs.

```
# Calculs des scores (sur test.meteotrain et test.meteotrain_proj)

pred.FWDAicmodel = predict(lg.FWDAicModel, test.meteotrain, type = "response")
pred.BWDAicmodel = predict(lg.BWDAicModel, test.meteotrain, type = "response")
pred.STEPWAicmodel = predict(lg.STEPWAicModel, test.meteotrain, type = "response")
pred.FWDBicmodel = predict(lg.FWDBicModel, test.meteotrain, type = "response")
pred.BWDBicmodel = predict(lg.BWDBicModel, test.meteotrain, type = "response")
pred.STEPWBicmodel = predict(lg.STEPWBicModel, test.meteotrain, type = "response")
pred.modelsat_acp = predict(lg.modelsat.acp, test.meteotrain_proj, type = "response")
pred.FWDAicmodel_acp = predict(lg.FWDAicModel_acp, test.meteotrain_proj, type = "response")
pred.FWDBicmodel_acp = predict(lg.FWDBicModel_acp, test.meteotrain_proj, type = "response")
```

Nous vérifions ce que contiennent nos scores pour chaque modèle.

```
summary(pred.FWDAicmodel)
summary(pred.FWDBicmodel)
summary(pred.BWDAicmodel)
summary(pred.BWDBicmodel)
summary(pred.STEPWAicmodel)
summary(pred.STEPWBicmodel)
summary(pred.FWDAicmodel_acp)
summary(pred.FWDBicmodel_acp)
summary(pred.modelsat_acp)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.01145 0.25470 0.48209 0.48865 0.72488 0.99629
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.03577 0.28518 0.46251 0.48503 0.69583 0.96900
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.01249 0.25752 0.49016 0.48937 0.73142 0.99613
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.01595 0.26883 0.50745 0.49038 0.72057 0.99132
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.01295 0.26949 0.48187 0.48890 0.72235 0.99643
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.02775 0.26114 0.47948 0.48384 0.70747 0.95308
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

```
## 0.02704 0.31679 0.48919 0.50066 0.70942 0.95657
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.03262 0.33170 0.48157 0.50105 0.71297 0.96561
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.03095 0.30994 0.48637 0.50104 0.71528 0.95637
```

Nous nous rassurons que nos scores sont bien compris entre 0 et 1 (c'est bien ce que nous attendions car ce sont des probabilités).

```
test.meteotrain$pluie.demain=as.numeric(test.meteotrain$pluie.demain)
test.meteotrain_proj$pluie.demain=as.numeric(test.meteotrain_proj$pluie.demain)

# Calculs des MAE #

mae.lg.FWDAicModel=mean(abs(pred.FWDAicmodel - test.meteotrain$pluie.demain),
                        na.rm = T)
mae.lg.BWDAicModel=mean(abs(pred.BWDAicmodel - test.meteotrain$pluie.demain),
                        na.rm = T)
mae.lg.STEPWAicModel=mean(abs(pred.STEPWAicmodel - test.meteotrain$pluie.demain),
                          na.rm = T)
mae.lg.FWDBicModel=mean(abs(pred.FWDBicmodel - test.meteotrain$pluie.demain),
                        na.rm = T)
mae.lg.BWDBicModel=mean(abs(pred.BWDBicmodel - test.meteotrain$pluie.demain),
                        na.rm = T)
mae.lg.STEPWBicModel=mean(abs(pred.STEPWBicmodel - test.meteotrain$pluie.demain),
                          na.rm = T)
mae.lg.FWDAicModel_acp=mean(abs(pred.FWDAicmodel_acp - test.meteotrain$pluie.demain),
                            na.rm = T)
mae.lg.FWDBicModel_acp=mean(abs(pred.FWDBicmodel_acp - test.meteotrain$pluie.demain),
                            na.rm = T)
mae.lg.modelsat_acp=mean(abs(pred.modelsat_acp - test.meteotrain$pluie.demain),
                         na.rm = T)


# Calculs des RMSEP #


msep.lg.FWDAicModel=sqrt(mean((pred.FWDAicmodel - test.meteotrain$pluie.demain)^2))
msep.lg.BWDAicModel=sqrt(mean((pred.BWDAicmodel - test.meteotrain$pluie.demain)^2))
msep.lg.STEPWAicModel=sqrt(mean((pred.STEPWAicmodel - test.meteotrain$pluie.demain)^2))
msep.lg.FWDBicModel=sqrt(mean((pred.FWDBicmodel - test.meteotrain$pluie.demain)^2))
msep.lg.BWDBicModel=sqrt(mean((pred.BWDBicmodel - test.meteotrain$pluie.demain)^2))
msep.lg.STEPWBicModel=sqrt(mean((pred.STEPWBicmodel - test.meteotrain$pluie.demain)^2))
msep.lg.FWDAicModel_acp=sqrt(mean((pred.FWDAicmodel_acp - test.meteotrain_proj$pluie.demain)^2))
msep.lg.FWDBicModel_acp=sqrt(mean((pred.FWDBicmodel_acp - test.meteotrain_proj$pluie.demain)^2))
msep.lg.modelsat_acp=sqrt(mean((pred.modelsat_acp - test.meteotrain_proj$pluie.demain)^2))

# Calculs des MSEP #

rmsep.lg.FWDAicModel=mean((pred.FWDAicmodel - test.meteotrain$pluie.demain)^2)
rmsep.lg.BWDAicModel=mean((pred.BWDAicmodel - test.meteotrain$pluie.demain)^2)
rmsep.lg.STEPWAicModel=mean((pred.STEPWAicmodel - test.meteotrain$pluie.demain)^2)
rmsep.lg.FWDBicModel=mean((pred.FWDBicmodel - test.meteotrain$pluie.demain)^2)
```

```
rmsep.lg.BWDBicModel=mean((pred.BWDBicmodel - test.meteotrain$pluie.demain)^2)
rmsep.lg.STEPWBicModel=mean((pred.STEPWBicmodel - test.meteotrain$pluie.demain)^2)
rmsep.lg.FWDAicModel_acp=mean((pred.FWDAicmodel_acp - test.meteotrain_proj$pluie.demain)^2)
rmsep.lg.FWDBicModel_acp=mean((pred.FWDBicmodel_acp - test.meteotrain_proj$pluie.demain)^2)
rmsep.lg.modelsat_acp=mean((pred.modelsat_acp - test.meteotrain_proj$pluie.demain)^2)
```

Les 3 indicateurs d'erreurs calculés nous conduisent à des taux d'erreurs les plus faibles en faveur des modèles issus de l'Acp.

Nous allons poursuivre l'analyse de la performance prédictive avec l'ensemble des modèles pour tous les diagnostics.

Voyons la matrice de confusion (et le taux d'erreur) et la courbe ROC, qui retracent le pouvoir discriminant du modèle. Nous calculerons de ce fait l'Auc (aire sous la courbe) sur chaque modèle. En parallèle, nous pouvons vérifier la calibration du modèle en effectuant les diagrammes de fiabilité et le test de Hosmer-Lemeshow de chaque modèle.

C'est à l'issue de cette étape que nous déciderons du modèle à retenir : car c'est bien pour sa qualité prédictive et sa capacité à s'ajuster au mieux aux données qu'un modèle est déterminément choisi parmi plusieurs modèles.

II.4.2 Qualité d'ajustement du modèle aux données

Matrice de confusion pour un point de coupure=0.5 et courbe ROC

```
# Les probabilités obtenues

prd.FWDAicmodel = as.factor(ifelse(pred.FWDAicmodel>0.5,1,0))
prd.BWDAicmodel = as.factor(ifelse(pred.BWDAicmodel>0.5,1,0))
prd.STEPWAicmodel = as.factor(ifelse(pred.STEPWAicmodel>0.5,1,0))
prd.FWDBicmodel = as.factor(ifelse(pred.FWDBicmodel>0.5,1,0))
prd.BWDBicmodel = as.factor(ifelse(pred.BWDBicmodel>0.5,1,0))
prd.STEPWBicmodel = as.factor(ifelse(pred.STEPWBicmodel>0.5,1,0))
prd.FWDAicmodel_acp = as.factor(ifelse(pred.FWDAicmodel_acp>0.5,1,0))
prd.FWDBicmodel_acp = as.factor(ifelse(pred.FWDBicmodel_acp>0.5,1,0))
prd.modelsat_acp = as.factor(ifelse(pred.modelsat_acp>0.5,1,0))

# Matrices de confusion

library (caret)

test.meteotrain_proj$pluie.demain=
  as.factor(ifelse(test.meteotrain_proj$pluie.demain==1,0,1))
test.meteotrain$pluie.demain=
  as.factor(ifelse(test.meteotrain$pluie.demain==1,0,1))


conf.mat.lg.FWDAicModel=confusionMatrix(data=prd.FWDAicmodel, reference = test.meteotrain$pluie.demain)
print(conf.mat.lg.FWDAicModel)
tauxerreurFWDAicModel =1.0-sum(diag(table(test.meteotrain$pluie.demain,prd.FWDAicmodel)))/sum(table(tes
print(tauxerreurFWDAicModel)
```

```
conf.mat.lg.BWDAicModel=confusionMatrix(data=prd.BWDAicmodel, reference = test.meteotrain$pluie.demain)
print(conf.mat.lg.BWDAicModel)
tauxerreurBWDAicModel =1.0-sum(diag(table(test.meteotrain$pluie.demain,prd.BWDAicmodel)))/sum(table(tes
print(tauxerreurBWDAicModel)

conf.mat.lg.STEPWAicModel=confusionMatrix(data=prd.STEPWAicmodel, reference = test.meteotrain$pluie.dema
print(conf.mat.lg.STEPWAicModel)
tauxerreurSTEPWAicModel =1.0-sum(diag(table(test.meteotrain$pluie.demain,prd.STEPWAicmodel)))/sum(table
print(tauxerreurSTEPWAicModel)

conf.mat.lg.FWDBicModel=confusionMatrix(data=prd.FWDBicmodel, reference = test.meteotrain$pluie.demain)
print(conf.mat.lg.FWDBicModel)
tauxerreurFWDBicModel =1.0-sum(diag(table(test.meteotrain$pluie.demain,prd.FWDBicmodel)))/sum(table(tes
print(tauxerreurFWDBicModel)

conf.mat.lg.BWDBicModel=confusionMatrix(data=prd.BWDBicmodel, reference = test.meteotrain$pluie.demain)
print(conf.mat.lg.BWDBicModel)
tauxerreurBWDBicModel =1.0-sum(diag(table(test.meteotrain$pluie.demain,prd.BWDBicmodel)))/sum(table(tes
print(tauxerreurBWDBicModel)

conf.mat.lg.STEPWBicModel=confusionMatrix(data=prd.STEPWBicmodel, reference = test.meteotrain$pluie.dema
print(conf.mat.lg.STEPWBicModel)
tauxerreurSTEPWBicModel =1.0-sum(diag(table(test.meteotrain$pluie.demain,prd.STEPWBicmodel)))/sum(table
print(tauxerreurSTEPWBicModel)

conf.mat.lg.FWDAicModel_acp=confusionMatrix(data=prd.FWDAicmodel_acp, reference = test.meteotrain_proj$p
print(conf.mat.lg.FWDAicModel_acp)
tauxerreurFWDAicModel_acp =1.0-sum(diag(table(test.meteotrain$pluie.demain,prd.FWDAicmodel_acp)))/sum(ta
print(tauxerreurFWDAicModel_acp)

conf.mat.lg.FWDBicModel_acp=confusionMatrix(data=prd.FWDBicmodel_acp, reference = test.meteotrain_proj$p
print(conf.mat.lg.FWDBicModel_acp)
tauxerreurFWDBicModel_acp =1.0-sum(diag(table(test.meteotrain$pluie.demain,prd.FWDBicmodel_acp)))/sum(ta
print(tauxerreurFWDBicModel_acp)

conf.mat.lg.modelsat_acp=confusionMatrix(data=prd.modelsat_acp, reference = test.meteotrain_proj$pluie.d
print(conf.mat.lg.modelsat_acp)
tauxerreurmodelsat_acp =1.0-sum(diag(table(test.meteotrain$pluie.demain,prd.modelsat_acp)))/sum(table(te
print(tauxerreurmodelsat_acp)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 84 39
##          1 28 81
##
##                Accuracy : 0.7112
##                  95% CI : (0.6483, 0.7686)
##     No Information Rate : 0.5172
##     P-Value [Acc > NIR] : 1.358e-09
##
##                   Kappa : 0.4236
```

```
##
##  Mcnemar's Test P-Value : 0.2218
##
##             Sensitivity : 0.7500
##             Specificity : 0.6750
##          Pos Pred Value : 0.6829
##          Neg Pred Value : 0.7431
##              Prevalence : 0.4828
##          Detection Rate : 0.3621
##    Detection Prevalence : 0.5302
##       Balanced Accuracy : 0.7125
##
##        'Positive' Class : 0
##
## [1] 0.2887931
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 82 39
##          1 30 81
##
##                Accuracy : 0.7026
##                  95% CI : (0.6393, 0.7606)
##     No Information Rate : 0.5172
##     P-Value [Acc > NIR] : 7.041e-09
##
##                   Kappa : 0.4061
##
##  Mcnemar's Test P-Value : 0.3355
##
##             Sensitivity : 0.7321
##             Specificity : 0.6750
##          Pos Pred Value : 0.6777
##          Neg Pred Value : 0.7297
##              Prevalence : 0.4828
##          Detection Rate : 0.3534
##    Detection Prevalence : 0.5216
##       Balanced Accuracy : 0.7036
##
##        'Positive' Class : 0
##
## [1] 0.2974138
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 82 37
##          1 30 83
##
##                Accuracy : 0.7112
##                  95% CI : (0.6483, 0.7686)
##     No Information Rate : 0.5172
##     P-Value [Acc > NIR] : 1.358e-09
```

```
##
##                   Kappa : 0.4229
##
##  Mcnemar's Test P-Value : 0.4635
##
##             Sensitivity : 0.7321
##             Specificity : 0.6917
##          Pos Pred Value : 0.6891
##          Neg Pred Value : 0.7345
##              Prevalence : 0.4828
##          Detection Rate : 0.3534
##    Detection Prevalence : 0.5129
##       Balanced Accuracy : 0.7119
##
##        'Positive' Class : 0
##
## [1] 0.2887931
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 89 40
##          1 23 80
##
##                Accuracy : 0.7284
##                  95% CI : (0.6664, 0.7846)
##     No Information Rate : 0.5172
##     P-Value [Acc > NIR] : 3.948e-11
##
##                   Kappa : 0.459
##
##  Mcnemar's Test P-Value : 0.04382
##
##             Sensitivity : 0.7946
##             Specificity : 0.6667
##          Pos Pred Value : 0.6899
##          Neg Pred Value : 0.7767
##              Prevalence : 0.4828
##          Detection Rate : 0.3836
##    Detection Prevalence : 0.5560
##       Balanced Accuracy : 0.7307
##
##        'Positive' Class : 0
##
## [1] 0.2715517
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 76 39
##          1 36 81
##
##                Accuracy : 0.6767
##                  95% CI : (0.6124, 0.7365)
```

```
##     No Information Rate : 0.5172
##     P-Value [Acc > NIR] : 6.093e-07
##
##                   Kappa : 0.3533
##
##  Mcnemar's Test P-Value : 0.8174
##
##             Sensitivity : 0.6786
##             Specificity : 0.6750
##          Pos Pred Value : 0.6609
##          Neg Pred Value : 0.6923
##              Prevalence : 0.4828
##          Detection Rate : 0.3276
##    Detection Prevalence : 0.4957
##       Balanced Accuracy : 0.6768
##
##        'Positive' Class : 0
##
## [1] 0.3232759
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 80 40
##          1 32 80
##
##                Accuracy : 0.6897
##                  95% CI : (0.6258, 0.7486)
##     No Information Rate : 0.5172
##     P-Value [Acc > NIR] : 7.152e-08
##
##                   Kappa : 0.38
##
##  Mcnemar's Test P-Value : 0.4094
##
##             Sensitivity : 0.7143
##             Specificity : 0.6667
##          Pos Pred Value : 0.6667
##          Neg Pred Value : 0.7143
##              Prevalence : 0.4828
##          Detection Rate : 0.3448
##    Detection Prevalence : 0.5172
##       Balanced Accuracy : 0.6905
##
##        'Positive' Class : 0
##
## [1] 0.3103448
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 83 38
##          1 29 82
##
```

```
##                Accuracy : 0.7112
##                  95% CI : (0.6483, 0.7686)
##     No Information Rate : 0.5172
##     P-Value [Acc > NIR] : 1.358e-09
##
##                   Kappa : 0.4233
##
##  Mcnemar's Test P-Value : 0.3284
##
##             Sensitivity : 0.7411
##             Specificity : 0.6833
##          Pos Pred Value : 0.6860
##          Neg Pred Value : 0.7387
##              Prevalence : 0.4828
##          Detection Rate : 0.3578
##    Detection Prevalence : 0.5216
##       Balanced Accuracy : 0.7122
##
##        'Positive' Class : 0
##
## [1] 0.2887931
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 85 38
##          1 27 82
##
##                Accuracy : 0.7198
##                  95% CI : (0.6573, 0.7766)
##     No Information Rate : 0.5172
##     P-Value [Acc > NIR] : 2.414e-10
##
##                   Kappa : 0.4408
##
##  Mcnemar's Test P-Value : 0.2148
##
##             Sensitivity : 0.7589
##             Specificity : 0.6833
##          Pos Pred Value : 0.6911
##          Neg Pred Value : 0.7523
##              Prevalence : 0.4828
##          Detection Rate : 0.3664
##    Detection Prevalence : 0.5302
##       Balanced Accuracy : 0.7211
##
##        'Positive' Class : 0
##
## [1] 0.2801724
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 84 35
```

```
##          1 28 85
##
##                Accuracy : 0.7284
##                  95% CI : (0.6664, 0.7846)
##     No Information Rate : 0.5172
##     P-Value [Acc > NIR] : 3.948e-11
##
##                   Kappa : 0.4574
##
##  Mcnemar's Test P-Value : 0.4497
##
##             Sensitivity : 0.7500
##             Specificity : 0.7083
##          Pos Pred Value : 0.7059
##          Neg Pred Value : 0.7522
##              Prevalence : 0.4828
##          Detection Rate : 0.3621
##    Detection Prevalence : 0.5129
##       Balanced Accuracy : 0.7292
##
##        'Positive' Class : 0
##
## [1] 0.2715517
```

Un "bon" modèle serait celui qui aurait le taux d'erreur et le taux de faux positifs faibles (proche de 0), et des indicateurs élevées (proches de 1) pour la sensitivity, specificity, pos.pred.Value et neg.pred.value.

Par conséquent, le modèle qui semble avoir les meilleurs scores basés sur ces critères d'évaluation est :**lg.modelsat_acp**.
Nous pouvons toutefois considérer celui en 2ème bonne position qui obtient de très bons scores qui est le **lg.FWDBicModel_acp**

Les taux d'erreur de ces 2 modèles sont respectivement de 27,2% et 28%. Le taux d'erreur signifie que si nous choisissons ce modèle nous avons 27,2% (respectivement 28%) de chances de nous tromper sur la prédiction.

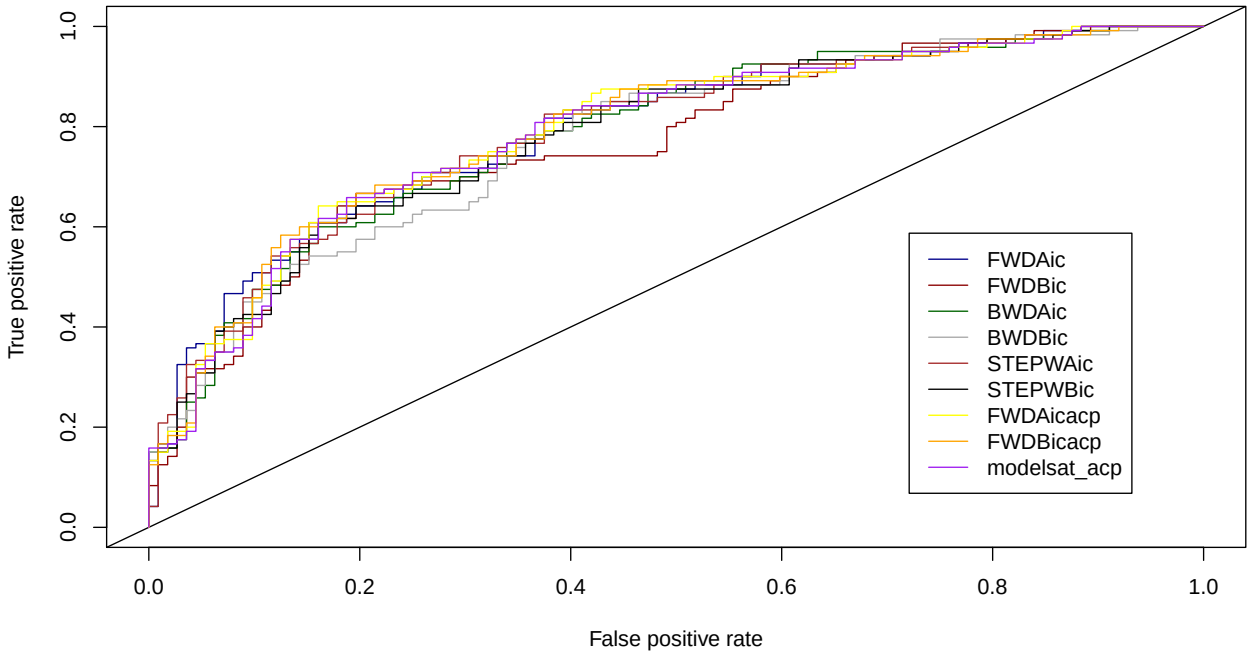
Traçons les courbes ROC et calculons les Auc respectifs.

Courbe ROC et Auc

```
#install.packages("ROCR")
library(ROCR)

 pFWDAic = prediction(pred.FWDAicmodel, test.meteotrain$pluie.demain)
 pFWDBic = prediction(pred.FWDBicmodel, test.meteotrain$pluie.demain)
 pBWDAic = prediction(pred.BWDAicmodel, test.meteotrain$pluie.demain)
 pBWDBic = prediction(pred.BWDBicmodel, test.meteotrain$pluie.demain)
 pSTEPWAic = prediction(pred.STEPWAicmodel, test.meteotrain$pluie.demain)
 pSTEPWBic = prediction(pred.STEPWBicmodel, test.meteotrain$pluie.demain)
```

```
 pFWDAicacp = prediction(pred.FWDAicmodel_acp, test.meteotrain_proj$pluie.demain)
 pFWDBicacp = prediction(pred.FWDBicmodel_acp, test.meteotrain_proj$pluie.demain)
 pmodelsatacp = prediction(pred.modelsat_acp, test.meteotrain_proj$pluie.demain)


 plot(performance( pFWDAic, "tpr", "fpr"), col="darkblue")
abline(0,1)
 plot(performance( pFWDBic, "tpr", "fpr"), col="darkred",add=T)
 plot(performance( pBWDAic, "tpr", "fpr"), col="darkgreen",add=T)
 plot(performance( pBWDBic, "tpr", "fpr"), col="darkgrey",add=T)
 plot(performance( pSTEPWAic, "tpr", "fpr"), col="brown",add=T)
 plot(performance( pSTEPWBic, "tpr", "fpr"), col="black",add=T)
 plot(performance( pFWDAicacp, "tpr", "fpr"), col="yellow",add=T)
 plot(performance( pFWDBicacp, "tpr", "fpr"), col="orange",add=T)
 plot(performance( pmodelsatacp, "tpr", "fpr"), col="purple",add=T)

perfFWDAic = performance(pFWDAic,"auc")
auc.lg.FWDAicModel=perfFWDAic@y.values[[1]]

perfFWDBic = performance(pFWDBic,"auc")
auc.lg.FWDBicModel=perfFWDBic@y.values[[1]]

perfBWDAic = performance(pBWDAic,"auc")
auc.lg.BWDAicModel=perfBWDAic@y.values[[1]]

perfBWDBic = performance(pBWDBic,"auc")
auc.lg.BWDBicModel=perfBWDBic@y.values[[1]]

perfSTEPWAic = performance(pSTEPWAic,"auc")
auc.lg.STEPWAicModel=perfSTEPWAic@y.values[[1]]

perfSTEPWBic = performance(pSTEPWBic,"auc")
auc.lg.STEPWBicModel=perfSTEPWBic@y.values[[1]]

perfFWDAicacp = performance(pFWDAicacp,"auc")
auc.lg.FWDAicModel_acp=perfFWDAicacp@y.values[[1]]

perfFWDBicacp = performance(pFWDBicacp,"auc")
auc.lg.FWDBicModel_acp=perfFWDBicacp@y.values[[1]]

perfmodelsatacp = performance(pmodelsatacp,"auc")
auc.lg.modelsat_acp=perfmodelsatacp@y.values[[1]]

legend("bottomright", inset = 0.1, legend = c("FWDAic",
                                              "FWDBic",
                                              "BWDAic",
                                              "BWDBic",
                                              "STEPWAic",
                                              "STEPWBic",
                                              "FWDAicacp",
                                              "FWDBicacp",
                                              "modelsat_acp"), lty = 1,
  col = c("darkblue", "darkred", "darkgreen","darkgrey",
```

```
          "brown","black","yellow","orange","purple"))
```



Le modèle qui détient l'aire sous la courbe ROC la plus élevée est la lg.FWDBicmodel_acp (courbe orange) avec un Auc = 0.7932. Notre classifieur fait largement mieux que l'attribution au hasard puisque l'Auc est bien supérieur à 0.5.
En théorie, un Auc entre 0.7 et 0.8 indique une discrimination acceptable. Entre 0.8 et 0.9, notre discrimination est excellente.
L'Auc étant à 0.7, nous avons une discrimination très satisfaisante qui frôle presque 0.8.

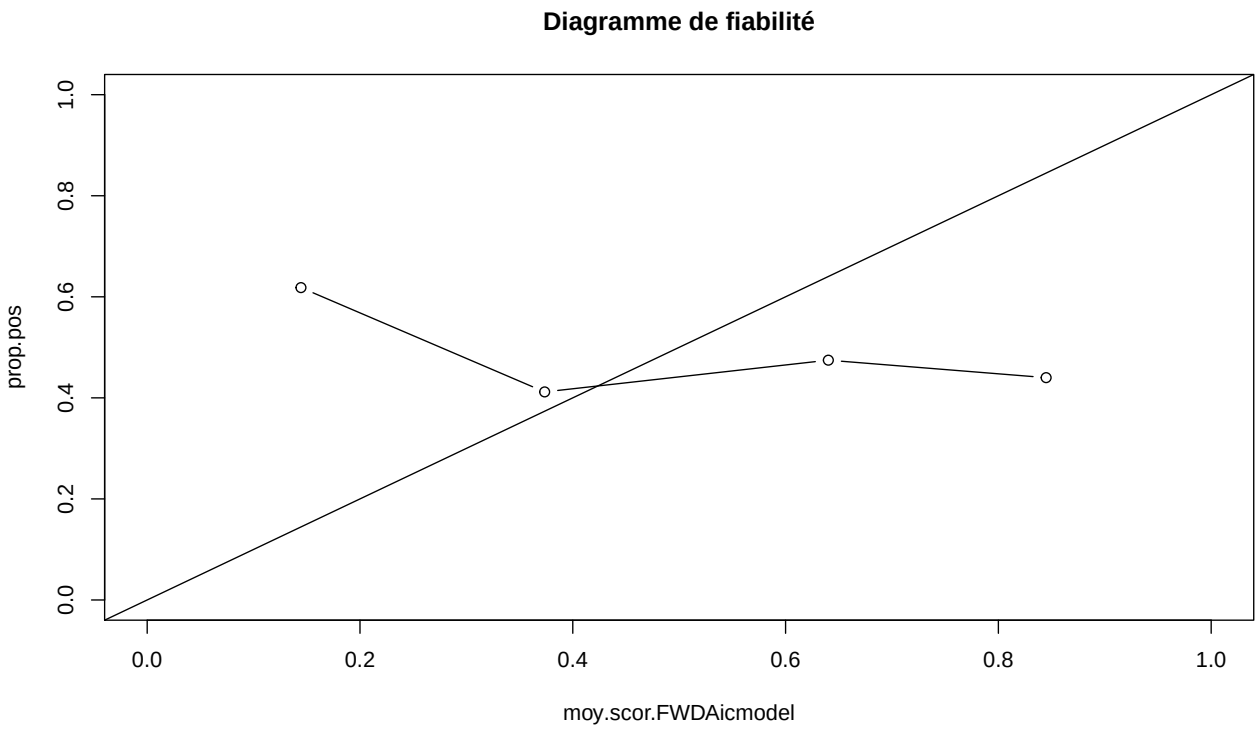
Pour graver notre décision, nous pouvons faire une autre méthode de vérification de calibration des modèles aux données pour nous permettre de décider.
Testons le diagramme de fiabilité et le test de Hosmer-Lemeshow sur les modèles (pas tous les modèles pour le diagramme).

Diagramme de fiabilité

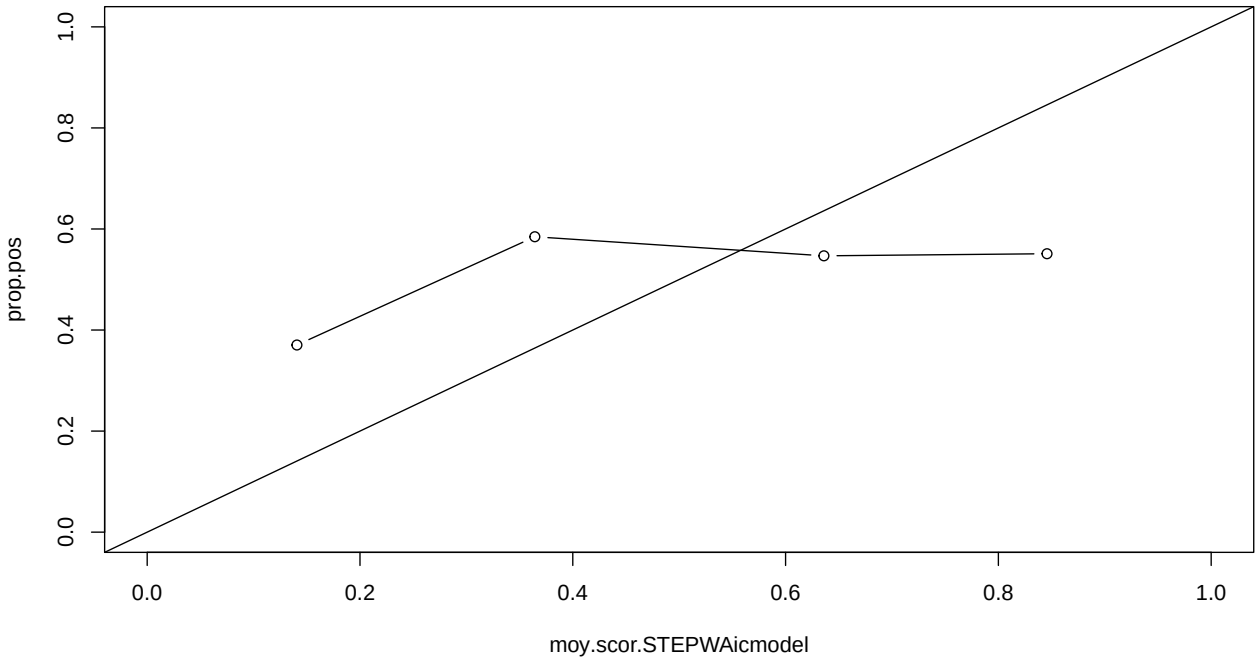
Vérifions la calibration des modèles, si les scores prédits sont compatibles avec les scores réels.

```
# Modèle lg.FWDAicmodel #
# Tri décroissant et découpage des scores en groupes #
scor.FWDAicmodel <- cut(sort(pred.FWDAicmodel),breaks=seq(from=0.0, to=1.0, by=0.25),include.lowest = TR
table(scor.FWDAicmodel)

# Calcul de la moyenne sur chaque groupe #
moy.scor.FWDAicmodel <- tapply(sort(pred.FWDAicmodel), scor.FWDAicmodel, mean)
print(moy.scor.FWDAicmodel)

# Proportion des positifs par groupe #
test.meteotrain$pluie.demain=
  as.factor(ifelse(test.meteotrain$pluie.demain==1,0,1))
table(scor.FWDAicmodel,test.meteotrain$pluie.demain)
prop.pos <- apply(table(scor.FWDAicmodel,
```

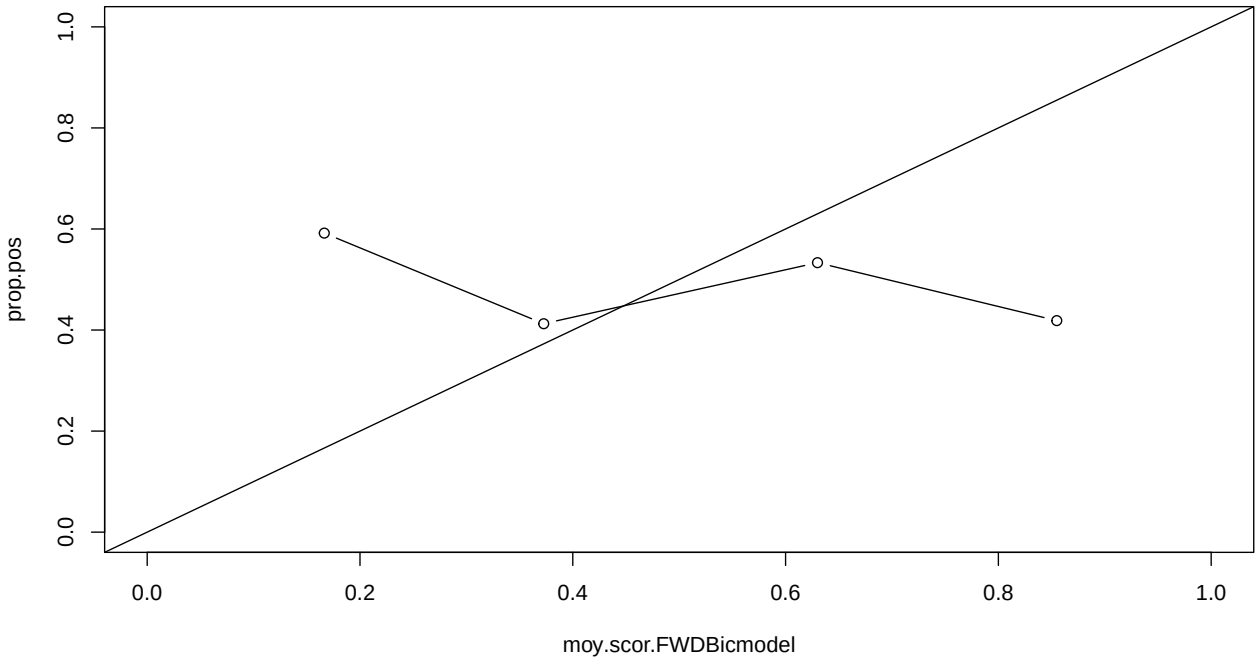
```
              test.meteotrain$pluie.demain),1,function(x){x[2]/sum(x)})

# Graphe #
plot(moy.scor.FWDAicmodel, prop.pos, main="Diagramme de fiabilité", type="b", xlim=c(0,1), ylim=c(0,1))
abline(a=0,b=1)
```



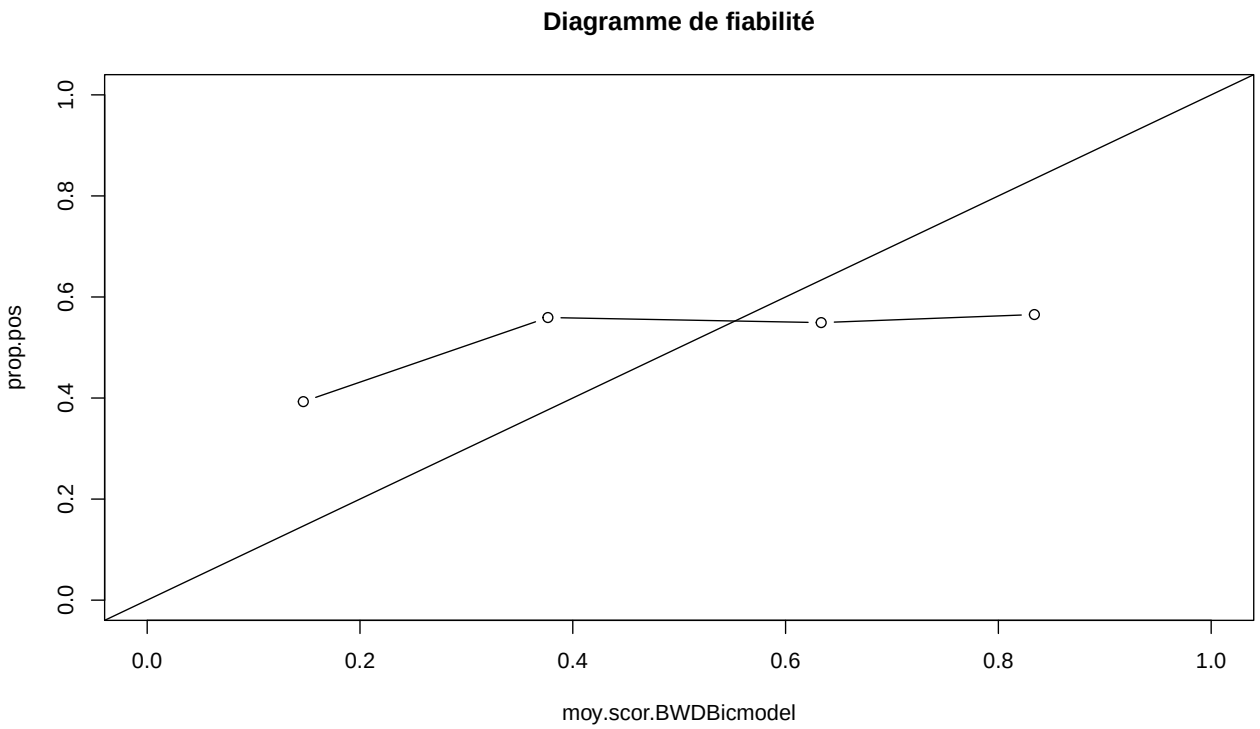

```
# Modèle lg.STEPWAicmodel #
# Tri décroissant et découpage des scores en groupes #
scor.STEPWAicmodel <- cut(sort(pred.STEPWAicmodel),breaks=seq(from=0.0, to=1.0, by=0.25),include.lowest
table(scor.STEPWAicmodel)

# Calcul de la moyenne sur chaque groupe #
moy.scor.STEPWAicmodel <- tapply(sort(pred.STEPWAicmodel),
                                 scor.STEPWAicmodel, mean)
print(moy.scor.STEPWAicmodel)

# Proportion des positifs par groupe #
test.meteotrain$pluie.demain=
  as.factor(ifelse(test.meteotrain$pluie.demain==1,0,1))
table(scor.STEPWAicmodel,test.meteotrain$pluie.demain)
prop.pos <- apply(table(scor.STEPWAicmodel,test.meteotrain$pluie.demain),
                  1,function(x){x[2]/sum(x)})

# Graphe #
plot(moy.scor.STEPWAicmodel, prop.pos, main="Diagramme de fiabilité", type="b", xlim=c(0,1), ylim=c(0,1
abline(a=0,b=1)
```



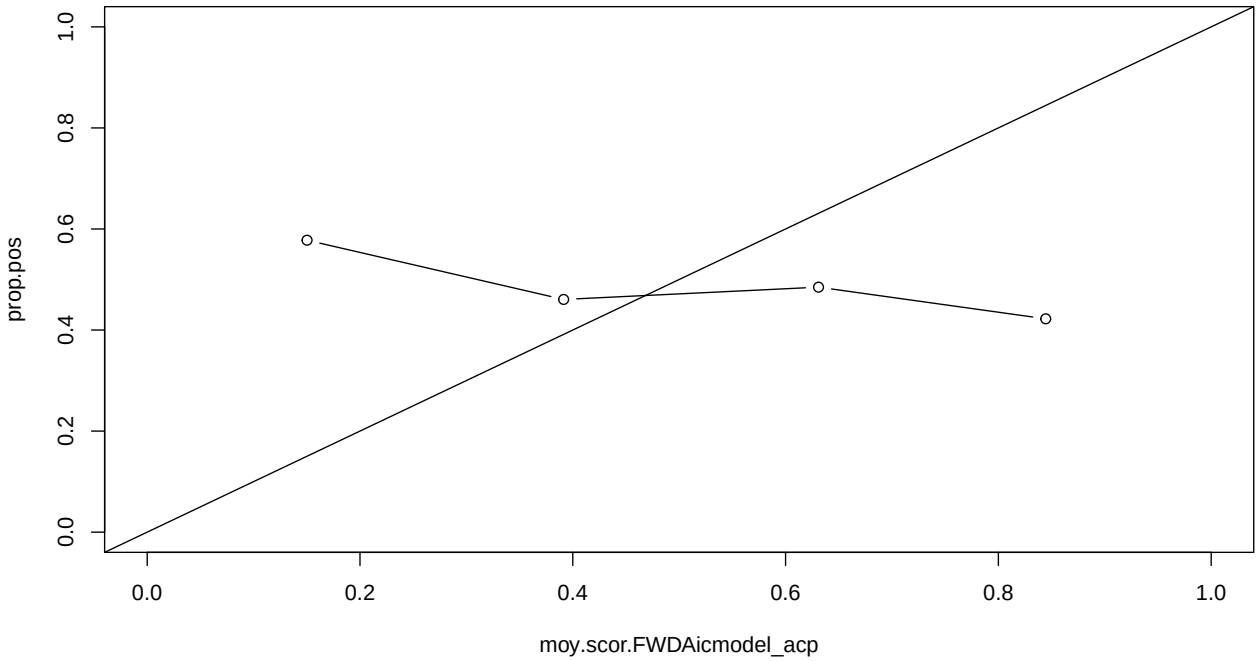

```
# Modèle lg.FWDBicmodel #
# Tri décroissant et découpage des scores en groupes #
scor.FWDBicmodel <- cut(sort(pred.FWDBicmodel),breaks=seq(from=0.0, to=1.0, by=0.25),include.lowest = TR
table(scor.FWDBicmodel)

# Calcul de la moyenne sur chaque groupe #
moy.scor.FWDBicmodel <- tapply(sort(pred.FWDBicmodel), scor.FWDBicmodel, mean)
print(moy.scor.FWDBicmodel)

# Proportion des positifs par groupe #
test.meteotrain$pluie.demain=
  as.factor(ifelse(test.meteotrain$pluie.demain==1,0,1))
table(scor.FWDAicmodel,test.meteotrain$pluie.demain)
prop.pos <- apply(table(scor.FWDBicmodel,test.meteotrain$pluie.demain),
                  1,function(x){x[2]/sum(x)})

# Graphe #
plot(moy.scor.FWDBicmodel, prop.pos, main="Diagramme de fiabilité", type="b", xlim=c(0,1), ylim=c(0,1))
abline(a=0,b=1)
```



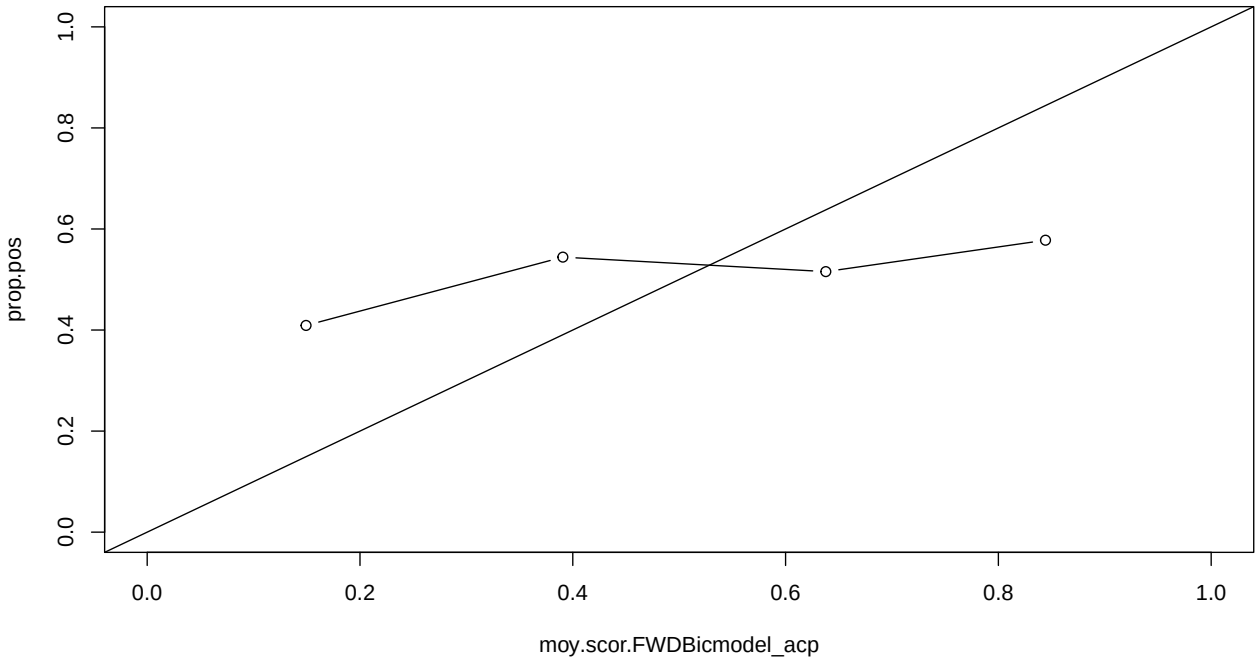

```
# Modèle lg.BWDBicmodel #
# Tri décroissant et découpage des scores en groupes #
scor.BWDBicmodel <- cut(sort(pred.BWDBicmodel),breaks=seq(from=0.0, to=1.0, by=0.25),include.lowest = TR
table(scor.BWDBicmodel)

# Calcul de la moyenne sur chaque groupe #
moy.scor.BWDBicmodel <- tapply(sort(pred.BWDBicmodel), scor.BWDBicmodel, mean)
print(moy.scor.BWDBicmodel)

# Proportion des positifs par groupe #
test.meteotrain$pluie.demain=
  as.factor(ifelse(test.meteotrain$pluie.demain==1,0,1))
table(scor.BWDBicmodel,test.meteotrain$pluie.demain)
prop.pos <- apply(table(scor.BWDBicmodel,test.meteotrain$pluie.demain),
                  1,function(x){x[2]/sum(x)})

# Graphe #
plot(moy.scor.BWDBicmodel, prop.pos, main="Diagramme de fiabilité", type="b", xlim=c(0,1), ylim=c(0,1))
abline(a=0,b=1)
```



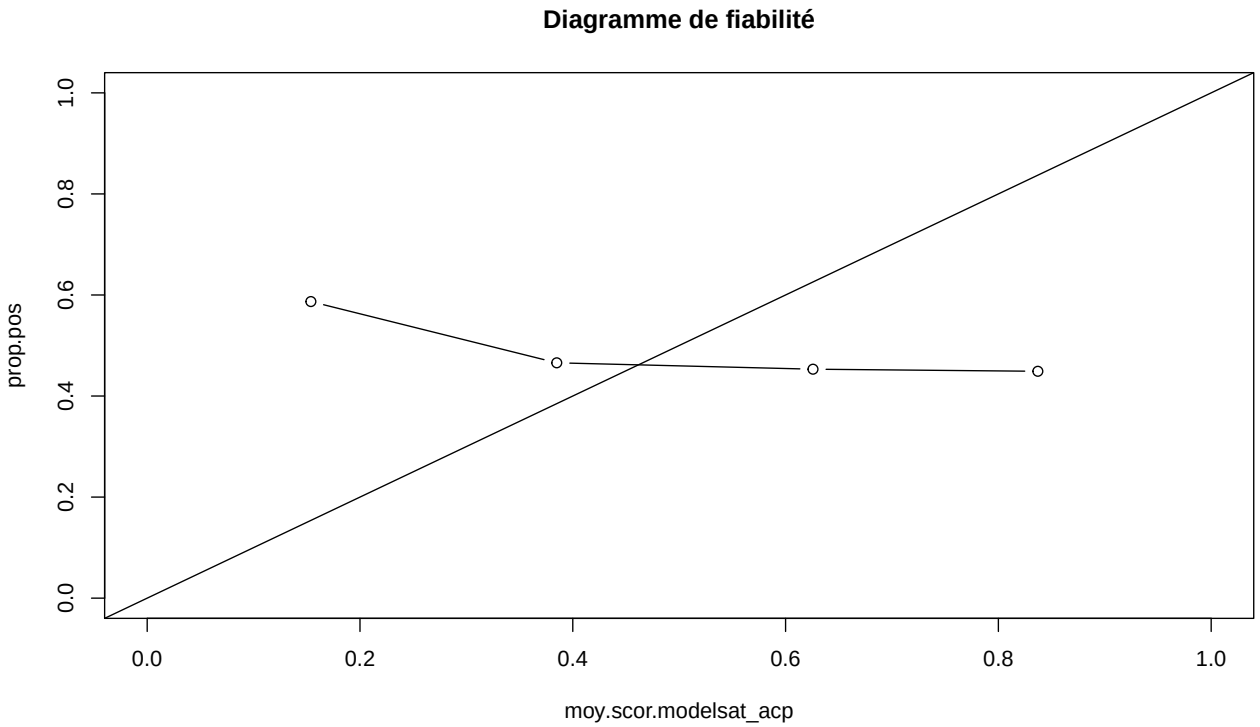

```
# Modèle lg.FWDAicmodel_acp #
# Tri décroissant et découpage des scores en groupes #
scor.FWDAicmodel_acp <- cut(sort(pred.FWDAicmodel_acp),
                            breaks=seq(from=0.0, to=1.0, by=0.25),
                            include.lowest = TRUE)
table(scor.FWDAicmodel_acp)

# Calcul de la moyenne sur chaque groupe #
moy.scor.FWDAicmodel_acp <- tapply(sort(pred.FWDAicmodel_acp),
                                   scor.FWDAicmodel_acp, mean)
print(moy.scor.FWDAicmodel_acp)

# Proportion des positifs par groupe #
test.meteotrain_proj$pluie.demain=
  as.factor(ifelse(test.meteotrain_proj$pluie.demain==1,0,1))
table(scor.FWDAicmodel_acp,test.meteotrain_proj$pluie.demain)
prop.pos <- apply(table(scor.FWDAicmodel_acp,test.meteotrain_proj$pluie.demain),
                  1,function(x){x[2]/sum(x)})

# Graphe #
plot(moy.scor.FWDAicmodel_acp, prop.pos, main="Diagramme de fiabilité",
     type="b", xlim=c(0,1), ylim=c(0,1))
abline(a=0,b=1)
```




```
# Modèle lg.FWDBicmodel_acp #
# Tri décroissant et découpage des scores en groupes #
scor.FWDBicmodel_acp <- cut(sort(pred.FWDBicmodel_acp),
                            breaks=seq(from=0.0, to=1.0, by=0.25),
                            include.lowest = TRUE)
table(scor.FWDBicmodel_acp)

# Calcul de la moyenne sur chaque groupe #
moy.scor.FWDBicmodel_acp <- tapply(sort(pred.FWDBicmodel_acp),
                                   scor.FWDBicmodel_acp, mean)
print(moy.scor.FWDBicmodel_acp)

# Proportion des positifs par groupe #
test.meteotrain_proj$pluie.demain=
  as.factor(ifelse(test.meteotrain_proj$pluie.demain==1,0,1))
table(scor.FWDBicmodel_acp,test.meteotrain_proj$pluie.demain)
prop.pos <- apply(table(scor.FWDBicmodel_acp,test.meteotrain_proj$pluie.demain),
                  1,function(x){x[2]/sum(x)})

# Graphe #
plot(moy.scor.FWDBicmodel_acp, prop.pos, main="Diagramme de fiabilité", type="b", xlim=c(0,1), ylim=c(0
abline(a=0,b=1)
```




```
# Modèle lg.modelsat_acp #
# Tri décroissant et découpage des scores en groupes #
scor.modelsat_acp <- cut(sort(pred.modelsat_acp),breaks=seq(from=0.0, to=1.0, by=0.25),include.lowest =
table(scor.modelsat_acp)

# Calcul de la moyenne sur chaque groupe #
moy.scor.modelsat_acp <- tapply(sort(pred.modelsat_acp), scor.modelsat_acp, mean)
print(moy.scor.modelsat_acp)

# Proportion des positifs par groupe #
test.meteotrain_proj$pluie.demain=
  as.factor(ifelse(test.meteotrain_proj$pluie.demain==1,0,1))
table(scor.modelsat_acp,test.meteotrain_proj$pluie.demain)
prop.pos <- apply(table(scor.modelsat_acp,test.meteotrain_proj$pluie.demain),
                  1,function(x){x[2]/sum(x)})

# Graphe #
plot(moy.scor.modelsat_acp, prop.pos, main="Diagramme de fiabilité", type="b", xlim=c(0,1), ylim=c(0,1))
abline(a=0,b=1)
```



A travers les diagrammes ci-dessus, nous n'en avons pas un dont la courbe suive nettement la bissectrice principale, tout au moins, nous pourrions accepter le diagramme du lg.STEPWAicModel , celui du lg.FWDBicmodel_acp et du lg.BWDBicmodel.

Essayons de voir le test de Hosmer-Lemeshow.

Test de Hosmer et Lemeshow

Faisons tourner les 9 modèles sur les données de test :

```
#install.packages("ResourceSelection")
library(ResourceSelection)

hl.FWDAic = hoslem.test(lg.FWDAicModel$y,fitted(lg.FWDAicModel), g=10)
hl.FWDAic
cbind(hl.FWDAic$observed,hl.FWDAic$expected)

hl.FWDBic = hoslem.test(lg.FWDBicModel$y,fitted(lg.FWDBicModel), g=10)
hl.FWDBic
cbind(hl.FWDBic$observed,hl.FWDBic$expected)

hl.BWDAic = hoslem.test(lg.BWDAicModel$y,fitted(lg.BWDAicModel), g=10)
hl.BWDAic
cbind(hl.BWDAic$observed,hl.BWDAic$expected)

hl.BWDBic = hoslem.test(lg.BWDBicModel$y,fitted(lg.BWDBicModel), g=10)
hl.BWDBic
```

```
cbind(hl.BWDBic$observed,hl.BWDBic$expected)

hl.STEPWAic = hoslem.test(lg.STEPWAicModel$y,fitted(lg.STEPWAicModel), g=10)
hl.STEPWAic
cbind(hl.STEPWAic$observed,hl.STEPWAic$expected)

hl.STEPWBic = hoslem.test(lg.STEPWBicModel$y,fitted(lg.STEPWBicModel), g=10)
hl.STEPWBic
cbind(hl.STEPWBic$observed,hl.STEPWBic$expected)

hl.FWDAic_acp = hoslem.test(lg.FWDAicModel_acp$y,fitted(lg.FWDAicModel_acp), g=10)
hl.FWDAic_acp
cbind(hl.FWDAic_acp$expected,hl.FWDAic_acp$observed)

hl.FWDBic_acp = hoslem.test(lg.FWDBicModel_acp$y,fitted(lg.FWDBicModel_acp), g=10)
hl.FWDBic_acp
cbind(hl.FWDBic_acp$expected,hl.FWDBic_acp$observed)

hl.modelsat_acp = hoslem.test(lg.modelsat_acp$y,fitted(lg.modelsat_acp), g=10)
hl.modelsat_acp
cbind(hl.modelsat_acp$expected,hl.modelsat_acp$observed)
```

D'après ce test, tous les modèles présentent une bonne calibration aux données car la p-value est supérieure à 5%.

II.4.3 Tableau récapitulatif sur l'évaluation des modèles

Etablissons un tableau récapitulatif sur l'évaluation prédictive de chaque modèle et la qualité d'ajustement.

Tableau récapitulatif d'évaluation des modèles

```
nom_modele = c("lg.FWDAicModel","lg.FWDBicModel",
               "lg.BWDAicModel","lg.BWDBicModel",
               "lg.STEPWAicModel","lg.STEPWBicModel",
               "lg.FWDAicModel_acp",
               "lg.FWDBicModel_acp","lg.modelsat_acp")

MAE = c(mae.lg.FWDAicModel,
        mae.lg.FWDBicModel,
        mae.lg.BWDAicModel,
        mae.lg.BWDBicModel,
        mae.lg.STEPWAicModel,
        mae.lg.STEPWBicModel,
        mae.lg.FWDAicModel_acp,
        mae.lg.FWDBicModel_acp,
        mae.lg.modelsat_acp)

MSEP = c(msep.lg.FWDAicModel,msep.lg.FWDBicModel,
         msep.lg.BWDAicModel,msep.lg.BWDBicModel,
         msep.lg.STEPWAicModel,msep.lg.STEPWBicModel,
         msep.lg.FWDAicModel_acp,
         msep.lg.FWDBicModel_acp,
         msep.lg.modelsat_acp)

RMSEP = c(rmsep.lg.FWDAicModel,rmsep.lg.FWDBicModel,
          rmsep.lg.BWDAicModel,rmsep.lg.BWDBicModel,
          rmsep.lg.STEPWAicModel,rmsep.lg.STEPWBicModel,
          rmsep.lg.FWDAicModel_acp,
          rmsep.lg.FWDBicModel_acp,
          rmsep.lg.modelsat_acp)

Accuracy = c(conf.mat.lg.FWDAicModel$overall['Accuracy'],
             conf.mat.lg.FWDBicModel$overall['Accuracy'],
             conf.mat.lg.BWDAicModel$overall['Accuracy'],
             conf.mat.lg.BWDBicModel$overall['Accuracy'],
             conf.mat.lg.STEPWAicModel$overall['Accuracy'],
             conf.mat.lg.STEPWBicModel$overall['Accuracy'],
             conf.mat.lg.FWDAicModel_acp$overall['Accuracy'],
             conf.mat.lg.FWDBicModel_acp$overall['Accuracy'],
             conf.mat.lg.modelsat_acp$overall['Accuracy'])

Sensitivity = c(conf.mat.lg.FWDAicModel$byClass['Sensitivity'],
                conf.mat.lg.FWDBicModel$byClass['Sensitivity'],
                conf.mat.lg.BWDAicModel$byClass['Sensitivity'],
                conf.mat.lg.BWDBicModel$byClass['Sensitivity'],
                conf.mat.lg.STEPWAicModel$byClass['Sensitivity'],
                conf.mat.lg.STEPWBicModel$byClass['Sensitivity'],
```

```
                conf.mat.lg.FWDAicModel_acp$byClass['Sensitivity'],
                conf.mat.lg.FWDBicModel_acp$byClass['Sensitivity'],
                conf.mat.lg.modelsat_acp$byClass['Sensitivity'])

Specificity = c(conf.mat.lg.FWDAicModel$byClass['Specificity'],
                conf.mat.lg.FWDBicModel$byClass['Specificity'],
                conf.mat.lg.BWDAicModel$byClass['Specificity'],
                conf.mat.lg.BWDBicModel$byClass['Specificity'],
                conf.mat.lg.STEPWAicModel$byClass['Specificity'],
                conf.mat.lg.STEPWBicModel$byClass['Specificity'],
                conf.mat.lg.FWDAicModel_acp$byClass['Specificity'],
                conf.mat.lg.FWDBicModel_acp$byClass['Specificity'],
                conf.mat.lg.modelsat_acp$byClass['Specificity'])

TxErreur = c(tauxerreurFWDAicModel,tauxerreurFWDBicModel,
               tauxerreurBWDAicModel,tauxerreurBWDBicModel,
               tauxerreurSTEPWAicModel,tauxerreurSTEPWBicModel,
               tauxerreurFWDAicModel_acp,tauxerreurFWDBicModel_acp,
               tauxerreurmodelsat_acp)

Auc = c(auc.lg.FWDAicModel,auc.lg.FWDBicModel,
        auc.lg.BWDAicModel,auc.lg.BWDBicModel,
        auc.lg.STEPWAicModel,auc.lg.STEPWBicModel,
        auc.lg.FWDAicModel_acp,auc.lg.FWDBicModel_acp,
        auc.lg.modelsat_acp)

PvalHL = c(hl.FWDAic$p.value,
                 hl.FWDBic$p.value,
                 hl.BWDAic$p.value,
                 hl.BWDBic$p.value,
                 hl.STEPWAic$p.value,
                 hl.STEPWBic$p.value,
                 hl.FWDAic_acp$p.value,
                 hl.FWDBic_acp$p.value,
                 hl.modelsat_acp$p.value)

tablo.val=cbind(MAE,MSEP,RMSEP,Accuracy,
                Sensitivity,Specificity,TxErreur,Auc,PvalHL)

tablo1 = as.data.frame(apply(tablo.val, 2, as.numeric))

tabl.res.eval = cbind(nom_modele,tablo1)
```

```
#library(kableExtra)
#library(tidyverse)

knitr::kable(tabl.res.eval,caption = "Tableau récapitulatif
             d'évaluation de modèles", digits = 4)
```

Les modèles issus de l'Acp proposent des scores très satisfaisants (surtout le modèle saturé lg.modelsat_acp

Table 2: Tableau récapitulatif d'évaluation de modèles

nom_modele	MAE	MSEP	RMSEP	Accuracy	Sensitivity	Specificity	TxErreur	Auc	PvalHL
lg.FWDAicModel	1.0286	1.1153	1.2438	0.7112	0.7500	0.6750	0.2888	0.7932	0.4138
lg.FWDBicModel	1.0322	1.1226	1.2603	0.7284	0.7946	0.6667	0.2716	0.7683	0.3989
lg.BWDAicModel	1.0279	1.1160	1.2455	0.7026	0.7321	0.6750	0.2974	0.7855	0.1092
lg.BWDBicModel	1.0269	1.1170	1.2477	0.6767	0.6786	0.6750	0.3233	0.7748	0.2871
lg.STEPWAicModel	1.0283	1.1149	1.2430	0.7112	0.7321	0.6917	0.2888	0.7929	0.0619
lg.STEPWBicModel	1.0334	1.1217	1.2581	0.6897	0.7143	0.6667	0.3103	0.7830	0.3306
lg.FWDAicModel_acp	1.0166	1.1042	1.2192	0.7112	0.7411	0.6833	0.2888	0.7926	0.9414
lg.FWDBicModel_acp	1.0162	1.1038	1.2184	0.7198	0.7589	0.6833	0.2802	0.7932	0.8265
lg.modelsat_acp	1.0162	1.1043	1.2194	0.7284	0.7500	0.7083	0.2716	0.7894	0.6377

et lg.FWDBicModel_acp). Au vu des résultats, nous devons donc choisir entre ces 2 modèles car ce sont ceux qui possédent la meilleure qualité d'ajustement aux données. Revenons sur la composition de chacun des modèles, nous rappelons que :
- le modèle saturé Acp lg.modelsat_acp est un modèle avec les 8 composantes principales (soit presque 80% de l'information résumée), mais les variables ne sont pas toutes significatives.
- le modèle forward Bic lg.FWDBicModel_acp est un modèle avec les 3 premières dimensions, il est plus parcimonieux, et les variables sont toutes significatives au sens de Wald.

Réalisons un test de rapport de vraisemblance pour nous aiguiller dans notre choix. Nous allons voir si le modèle réduit à 3 composantes peut être adopté au détriment du modèle complet.

```
lg.modelsat_acp <- glm(pluie.demain~.,data = tr.meteotrain_proj,
                       family = binomial)
lg.FWDBicModel_acp <- glm(pluie.demain~Dim.1+Dim.2+Dim.3,
                          data = tr.meteotrain_proj,
                          family = binomial)
anova(lg.modelsat_acp,lg.FWDBicModel_acp,test = "LRT")
```

Visiblement, les 5 dernières composantes principales ne sont donc pas significativement différentes de 0, nous pouvons donc accepter l'hypothèse H0. Le modèle lg.FWDBicModel_acp sera préféré, c'est le modèle qui présente les meilleures caractéristiques.

Nous décidons au vu de ces différents résultats de retenir le modèle issu de l'acp **lg.FWDBicModel_acp** .

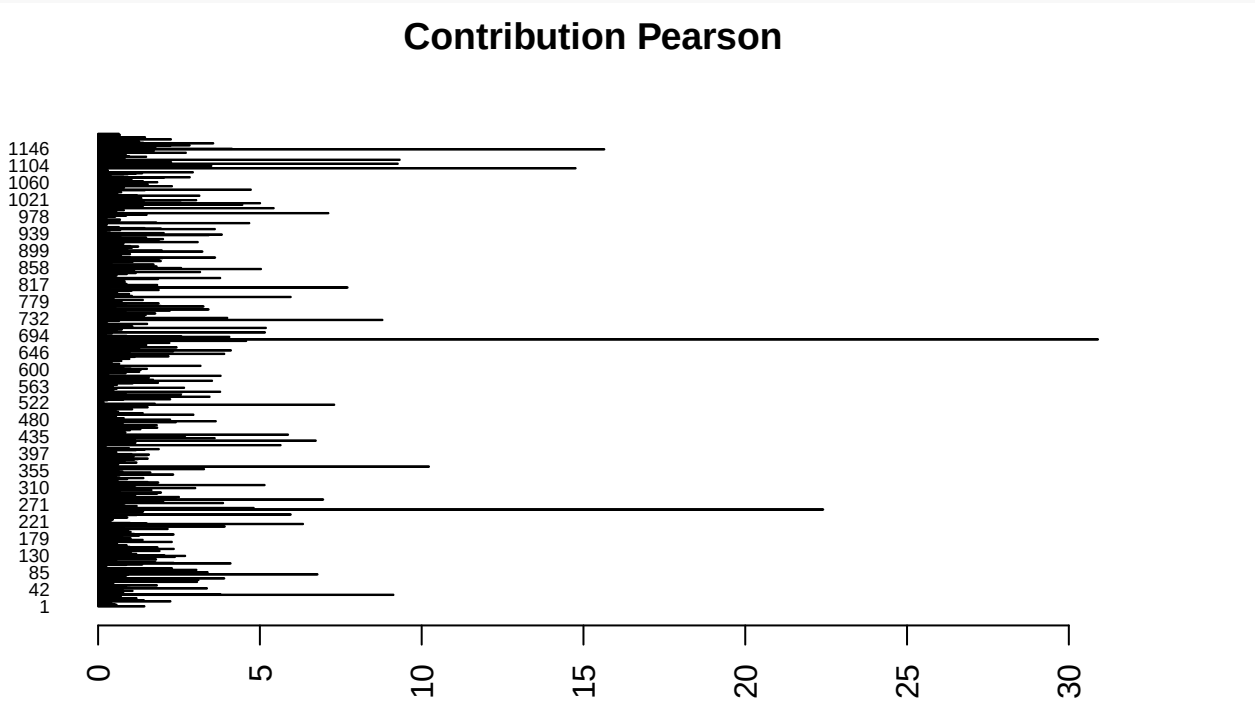
A présent, nous allons faire un diagnostic des résidus pour vérifier la qualité de restitution par le modèle de chaque observation, cela mettra en exergue d'éventuels points atypiques.

A travers cela, nous allons mettre en évidence :
- les points aberrants et/ou mal modélisés (qui pourraient être exclus)
- les points influents (qui, s'ils étaient amenés à être exclus, modifieraient totalement le modèle)
- les points leviers (qui contribuent fortement à leur propre prévision)

II.5 Analyse des résidus

Avec les résidus de Pearson, notons bien que plus la valeur du résidu est grande en valeur absolue, moins le point a été correctement modélisé.

```
res.pearson = residuals(lg.FWDBicModel_acp,"pearson")
stat.pearson = sum(res.pearson^2)
print(pchisq(stat.pearson,nrow(tr.meteotrain_proj)-3,lower.tail = F))

# Les résidus standardisés #
res.std.pearson = rstandard(lg.FWDBicModel_acp,type = "pearson")

# Les contributions à la statistique de Pearson #
contrib.pearson = res.std.pearson^2
print(contrib.pearson)

khi2new = stat.pearson-sum(contrib.pearson[c(254,680,1097,1144)])
print(pchisq(khi2new,nrow(tr.meteotrain_proj)-7,lower.tail = F))

# Tracé des contributions de Pearson #
barplot(contrib.pearson,horiz=T,las=2,main="Contribution Pearson",cex.names=0.6)
```

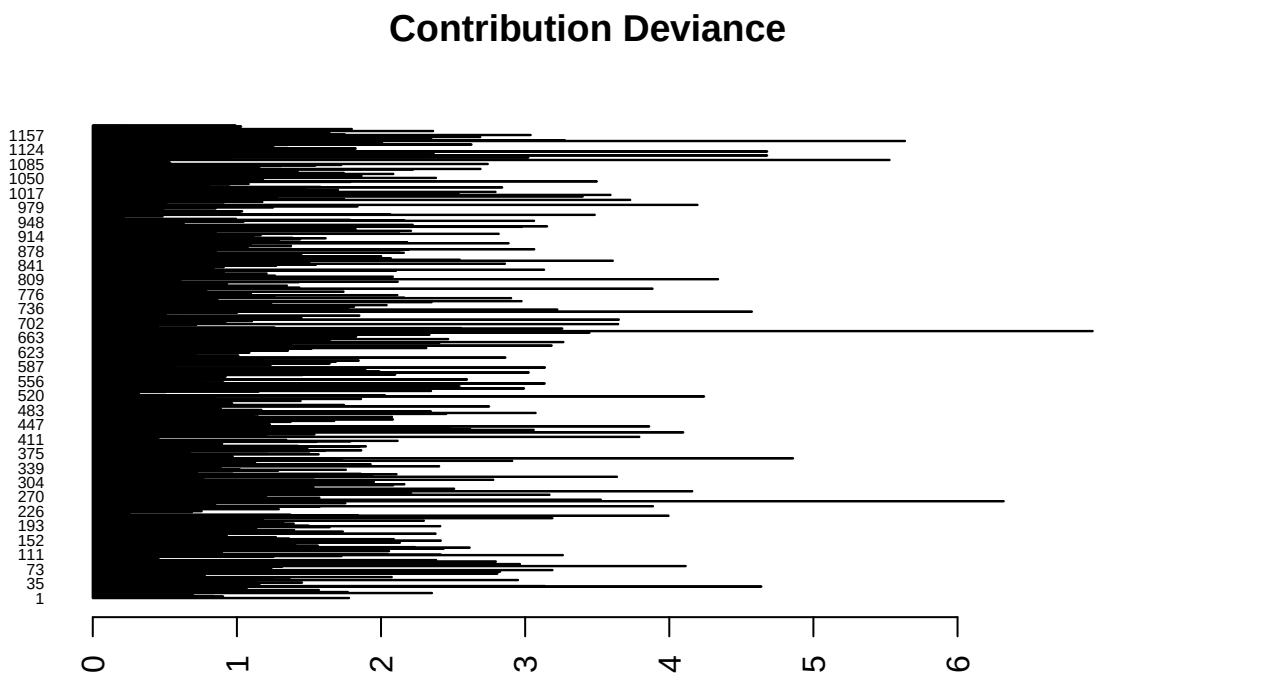


Nous voyons apparaître 4 points mal modélisés/aberrants avec une grande contribution qui sont : le 680 (à 30.9), 254 (à 22.4), 1097 (à 14.8),1144 (à 15.6).
Nous allons aller voir qui sont ces points dans la base de données de départ.

```
points.pearson = meteotrain[c(254,680,1097,1144),]
```

Avec les résidus de déviance cette fois :

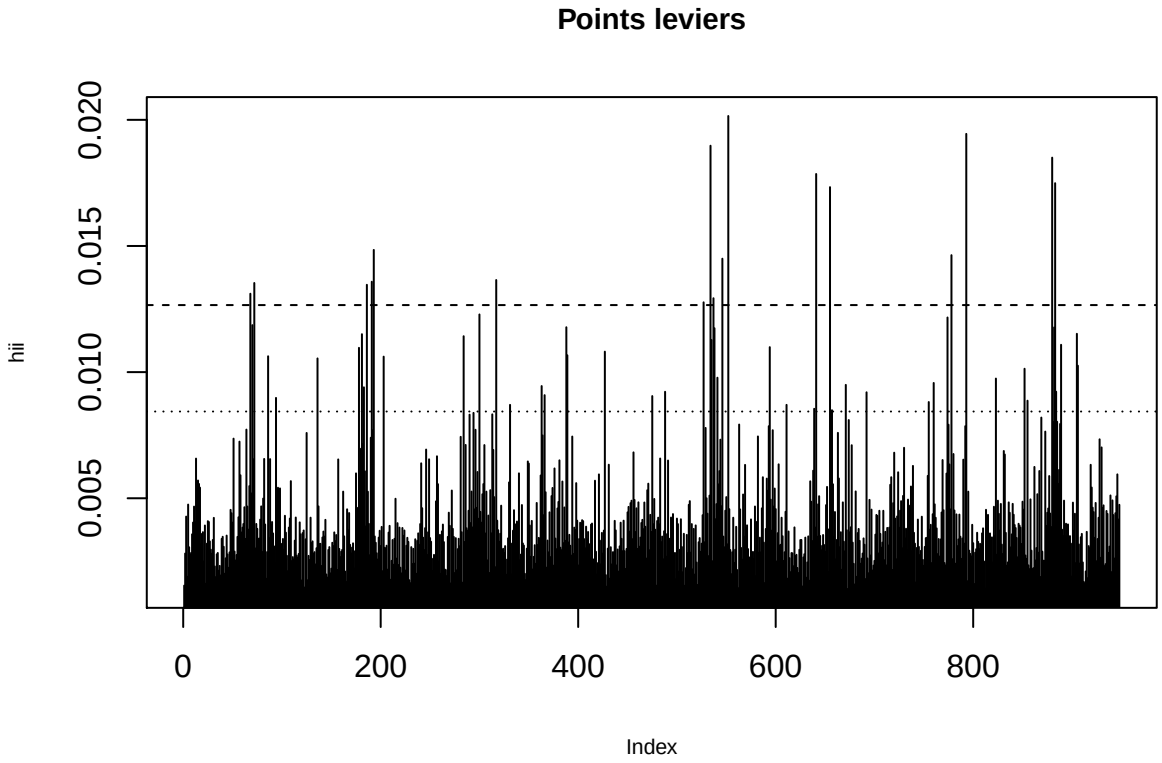
```
residuals(lg.FWDBicModel_acp,"deviance")

contrib.deviance = rstandard(lg.FWDBicModel_acp,type = "deviance")^2

barplot(contrib.deviance, horiz=T,las=2,main="Contribution Deviance",
        cex.names=0.5)
```



Les mêmes 4 observations aberrantes apparaîssent. Ces points méritent une analyse complémentaire sur les journées concernées (que nous ne ferons pas ici car ne font pas partie du cadre du cours).
En effet, peut-être s'agit-il de journées exceptionnelles avec un temps étrangement pluvieux ou sec, ou qu'il s'agit d'erreur de mesure...
En tous les cas, ils sont mal prédits par le modèle et perturbent la qualité de nos résultats.

Visualisons les points leviers, ceux qui déterminent fortement leur propre estimation.

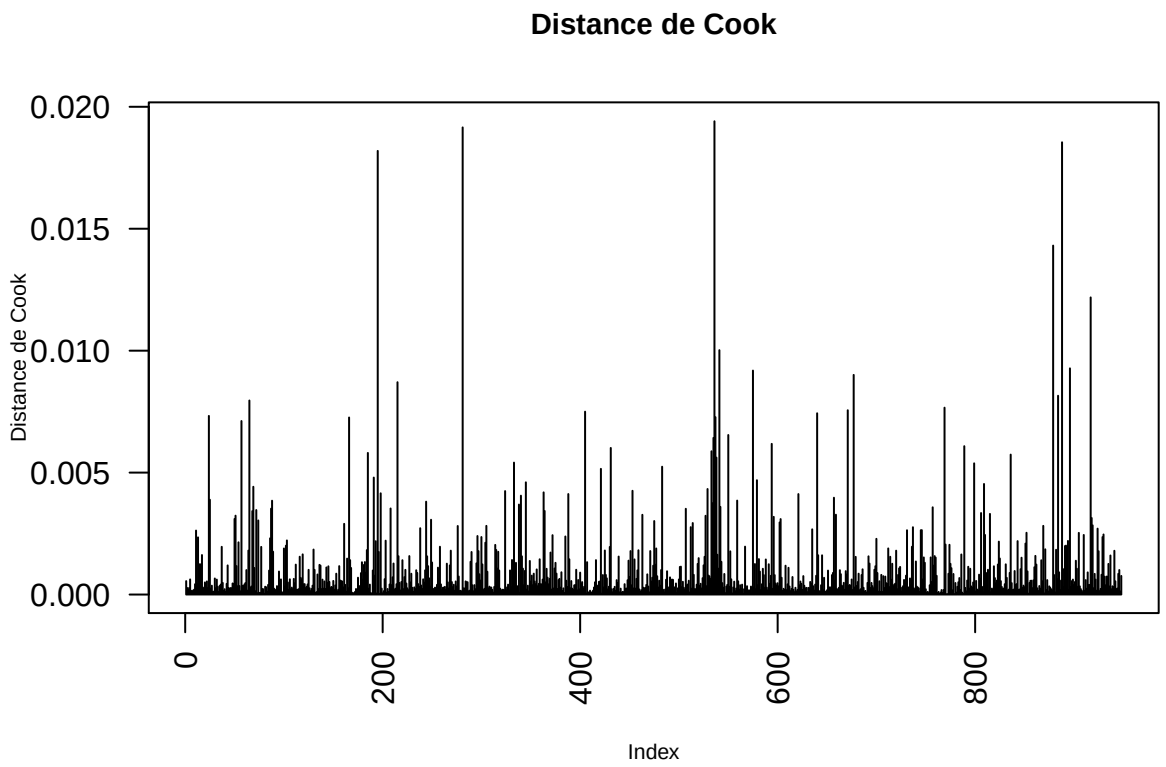
```
p <- length(lg.FWDBicModel_acp$coefficients)
n <- nrow(tr.meteotrain_proj)
plot(influence(lg.FWDBicModel_acp)$hat,type="h",ylab="hii",
     main = "Points leviers", cex.main = 0.9, cex.lab=0.7)
seuil1 = 3*p/n
abline(h=seuil1,col=1, lty=2)
seuil2 = 2*p/n
abline(h=seuil2,col=1,lty=3)
```



Sur le graphe, au moins 10 points dépassent le seuil indiquant des points leviers. Ceux-ci sont peut être également des points influents.

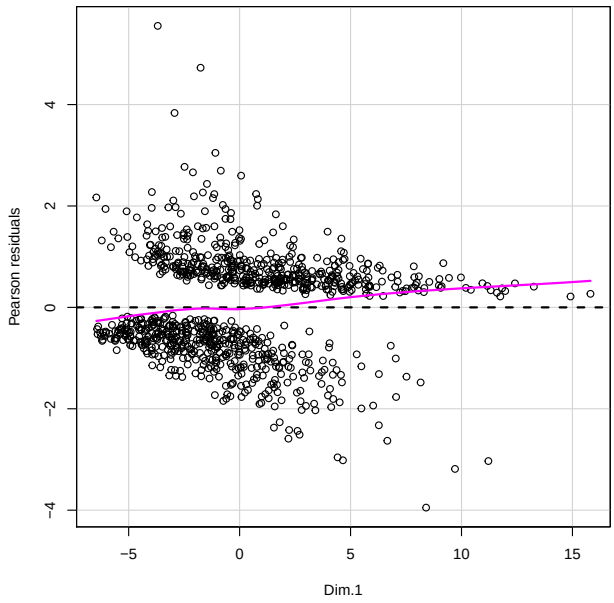
Voyons les points influents, c'est-à-dire les points qui ont le plus d'impact sur les coefficients estimés, et qui, s'ils étaient exclus, modifieraient complètement la structure du modèle :

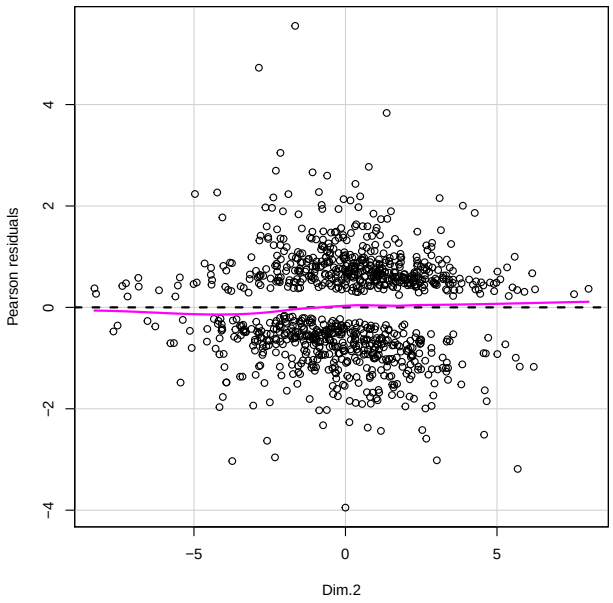
```
dcook = cooks.distance(lg.FWDBicModel_acp)
plot(dcook,type="h",ylab="Distance de Cook",las=2,
     main = "Distance de Cook",cex.lab=0.7, cex.main=0.9)
```

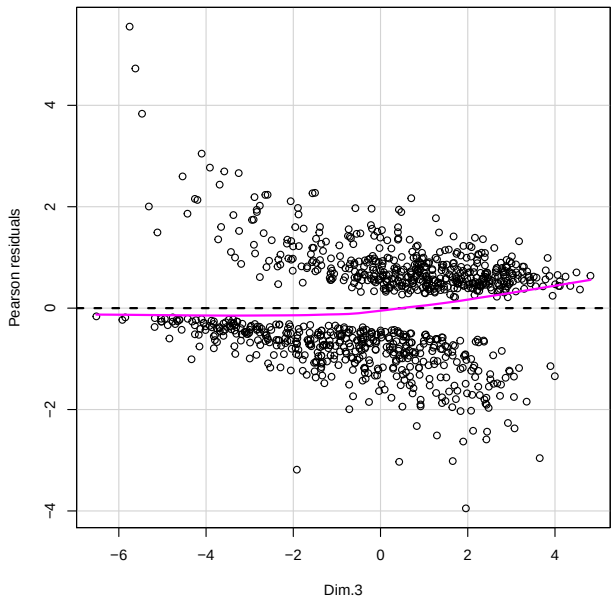


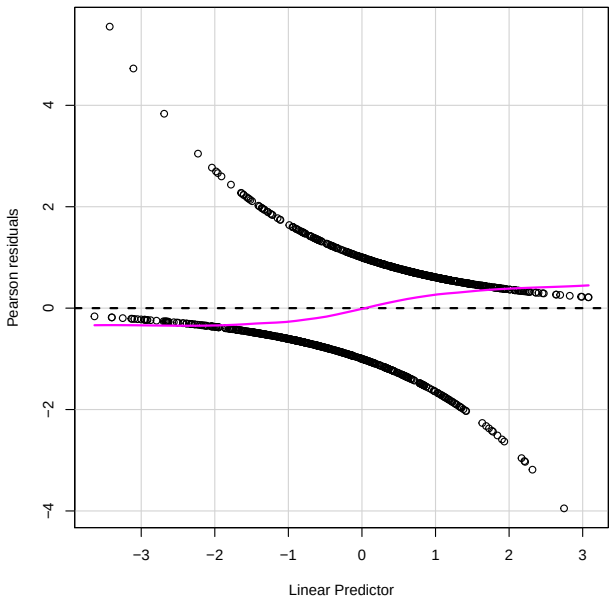
D'emblée, nous voyons réapparaître sur le graphe de la Distance de Cook les points aberrants et/ou leviers vus ci-dessus qui sont très influents.
Sans ces points, notre modèle aurait donc une tout autre tendance.
Nous n'allons pas approfondir l'analyse de ces observations cependant.

```
library(car)

# résidus par rapport aux dimensions #
residualPlots(lg.FWDBicModel_acp)
```

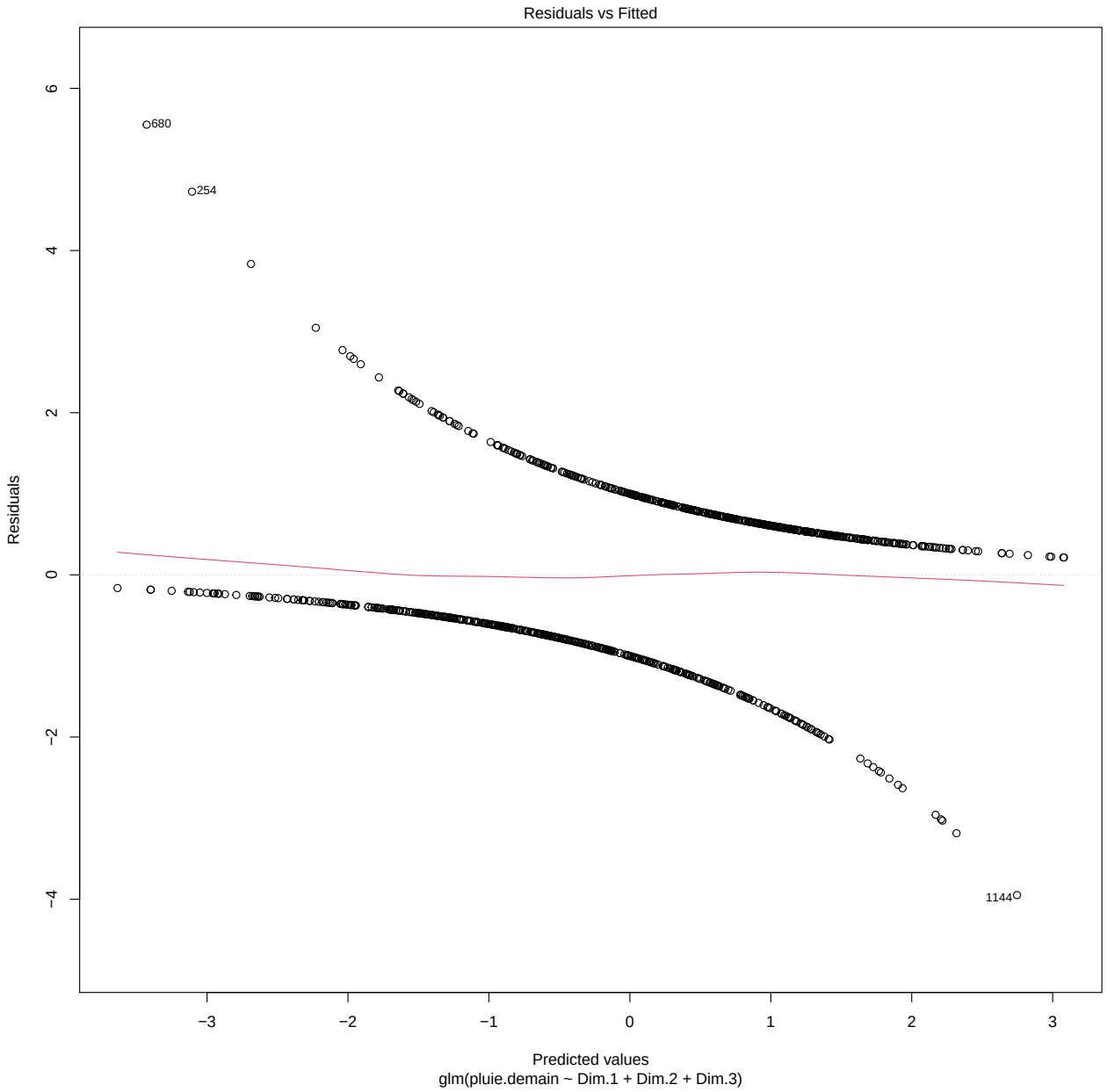




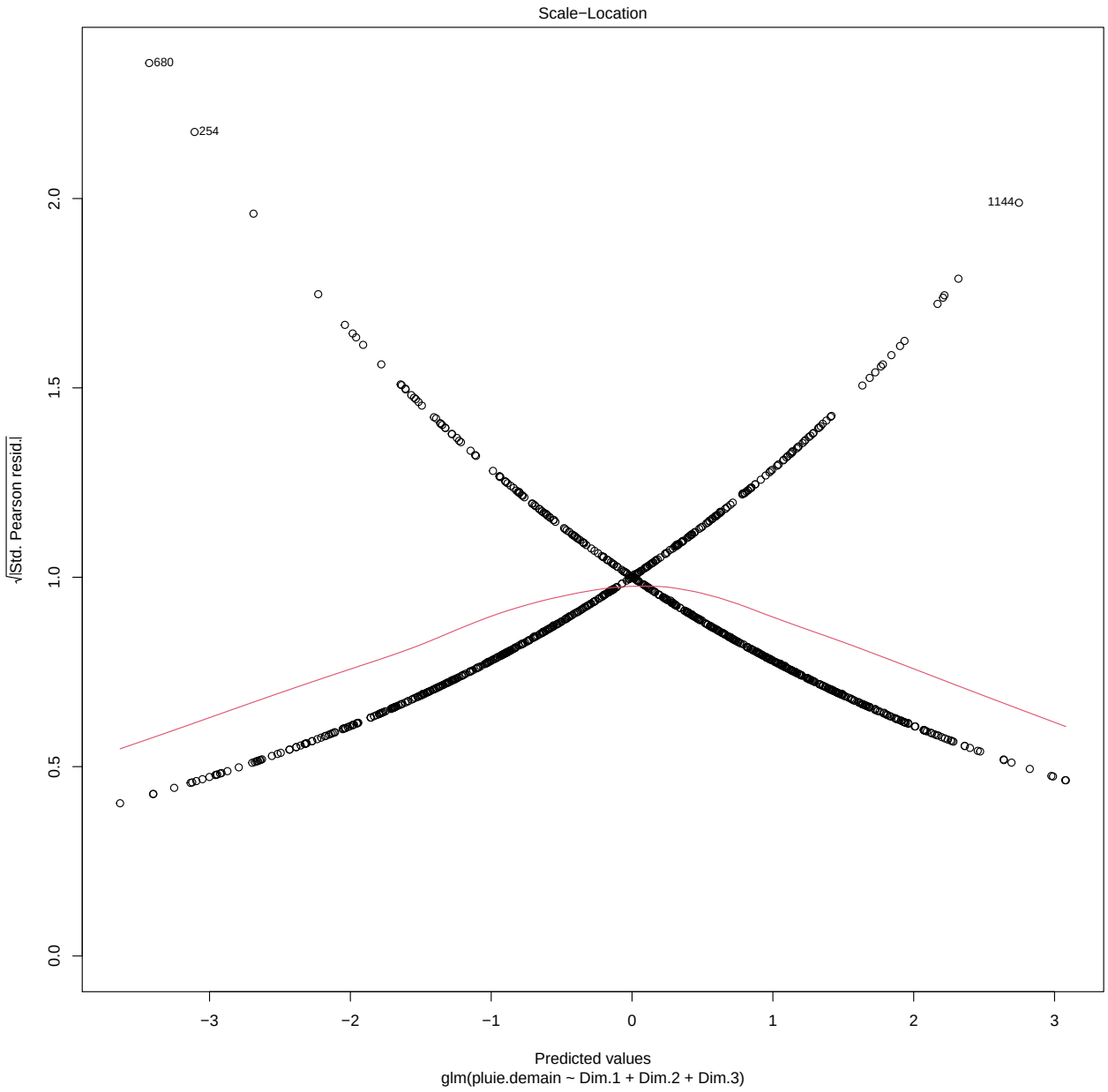




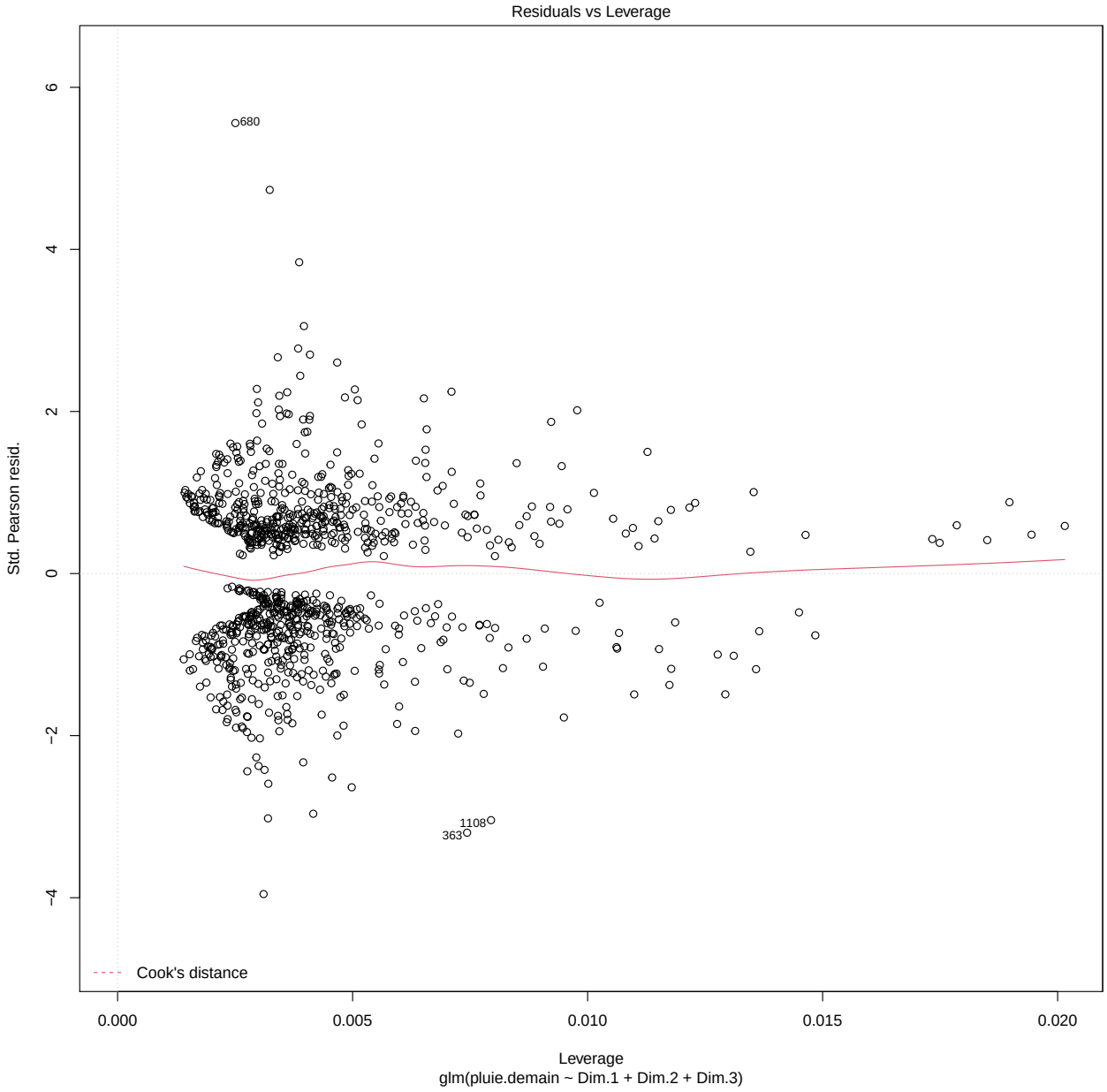

```r
# graphes des résidus  #
plot(lg.FWDBicModel_acp,which=1)
```



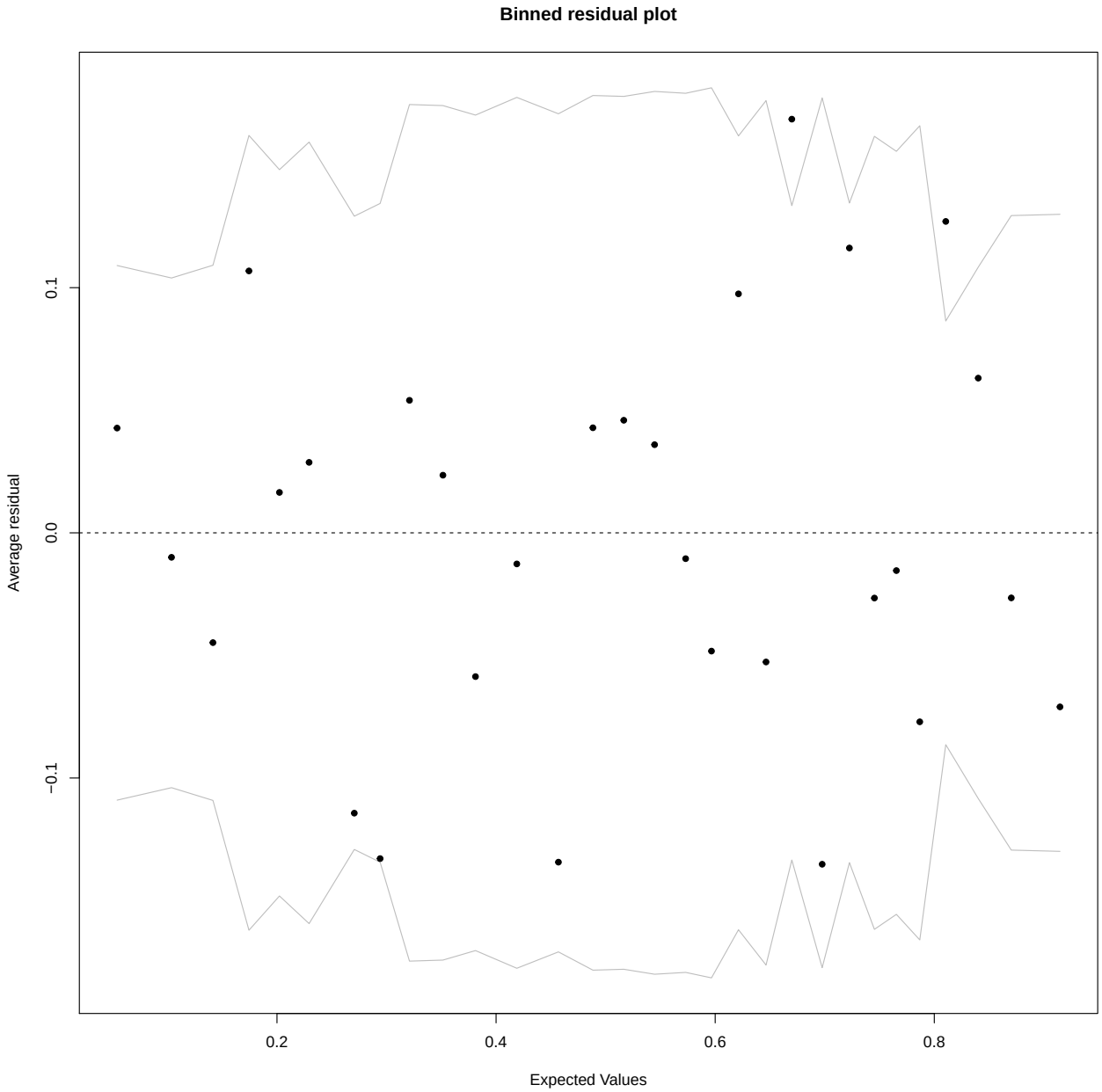

```
plot(lg.FWDBicModel_acp,which=3)
```




```
plot(lg.FWDBicModel_acp,which=5)
```




```
library(arm)
binnedplot(fitted(lg.FWDBicModel_acp), residuals(lg.FWDBicModel_acp, type = "response"))
```



Le 1er groupe de graphes montrent la linéarité des résidus par rapport aux dimensions. Il semble n'y avoir aucun tendance donc tout va bien. Les points sont plutôt alignés centrés en 0,exceptés quelques points aux extrémités (les mêmes points déjà repérés).

Le graphe QQ_plot ne nous sert à rien ici car la réponse ne répond pas à une distribution normale mais binomiale, nous ne l'affichons donc pas.

Le graphe suivant Residuals vs Fitted ne nous apporte pas beaucoup d'information, nous allons représenter un autre graphe montrant les résidus individuels par binnedplot. Nous y voyons l'intervalle de prédiction à 95%. Les résidus moyens devraient se situer à l'intérieur de cet intervalle pour environ 95%, ce qui est le cas ici, car seuls 2 résidus ne s'y trouvent pas. Notre modèle paraît bon.

Le graphe Residuals vs Leverage permet de détecter les points avec une grande influence, nous avons déjà vu plus haut les points influents et ils apparraissent dessus pour certains.

Nous n'analyserons pas plus les résidus.
Dans les 2 prochains paragraphes, nous mettons en oeuvre 2 méthodes de validations croisées qui auraient pu

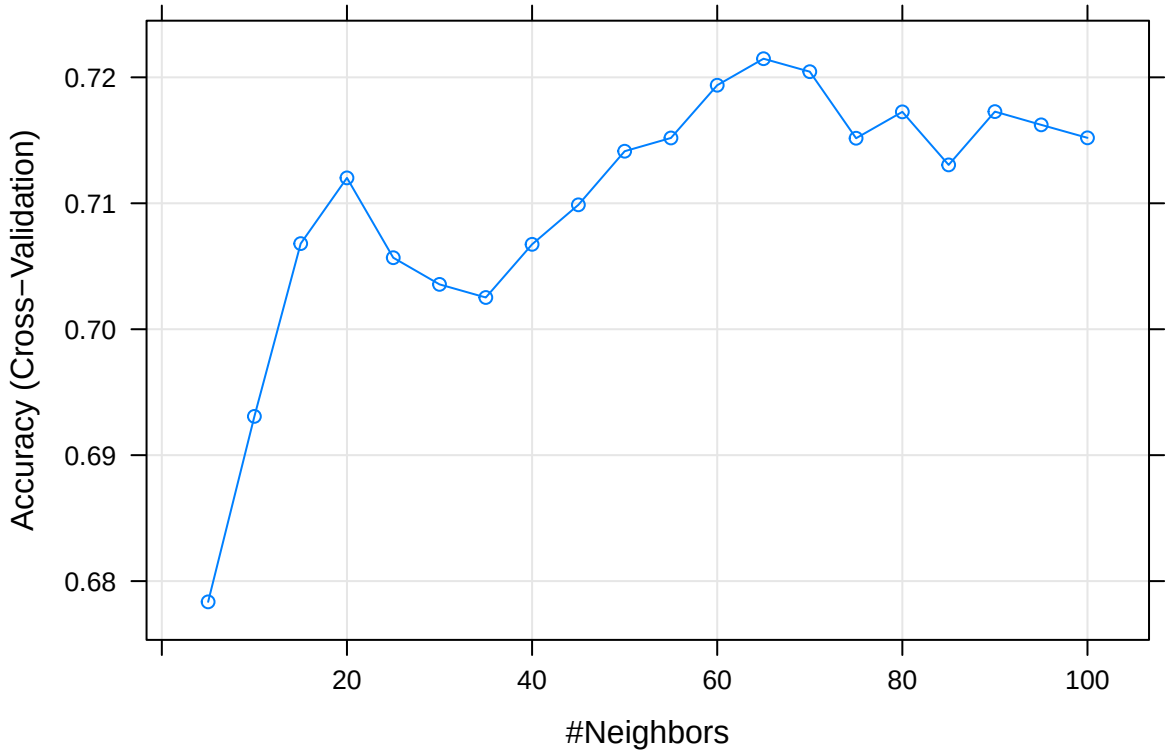
être appliquées.
Ici, nous les indiquons dans le simple objectif de nous rassurer sur l'ordre de grandeur des résultats précédemment obtenus.

III. Approche par validation croisée en K blocs

Notre échantillon se prête bien à cette méthode de validation, car cette dernière est la plus appropriée pour les jeux de taille en général < à 5000.

Nous allons utiliser la base tr.meteotrain1, qui sera donc divisée en 10 blocs, puis nous validerons avec la test.meteotrain. En théorie, il est recommandé de partitionner en 5 à 10 blocs. Nous allons partir sur 10 blocs.

```
library(caret)

# Fixons le seed pour que ce soit bien reproductible
set.seed(123)

# Spécifions la méthode d'apprentissage
train.method = trainControl(method = "cv",number =10)

# Régression logistique par méthode CV
lg.KCVmodel <- train(pluie.demain~.,
                     data = tr.meteotrain1,
                     method="knn",
                    # family = "binomial",
                     preProcess = c("center","scale"),
                     trControl = train.method,
                     tuneGrid=data.frame(k=seq(5,100, by=5))
                      )
print(lg.KCVmodel)


# Application à test.meteotrain_proj #
pred.KCVmodel = predict(lg.KCVmodel,newdata = test.meteotrain)

# Matrice de confusion #
confusionMatrix(pred.KCVmodel,test.meteotrain$pluie.demain)

plot(lg.KCVmodel)
```



Le meilleur modèle proposé par cette validation croisée a un accuracy = 0.72, sensitivity = 0.73, specificity = 0.71 et kappa = 0.44.

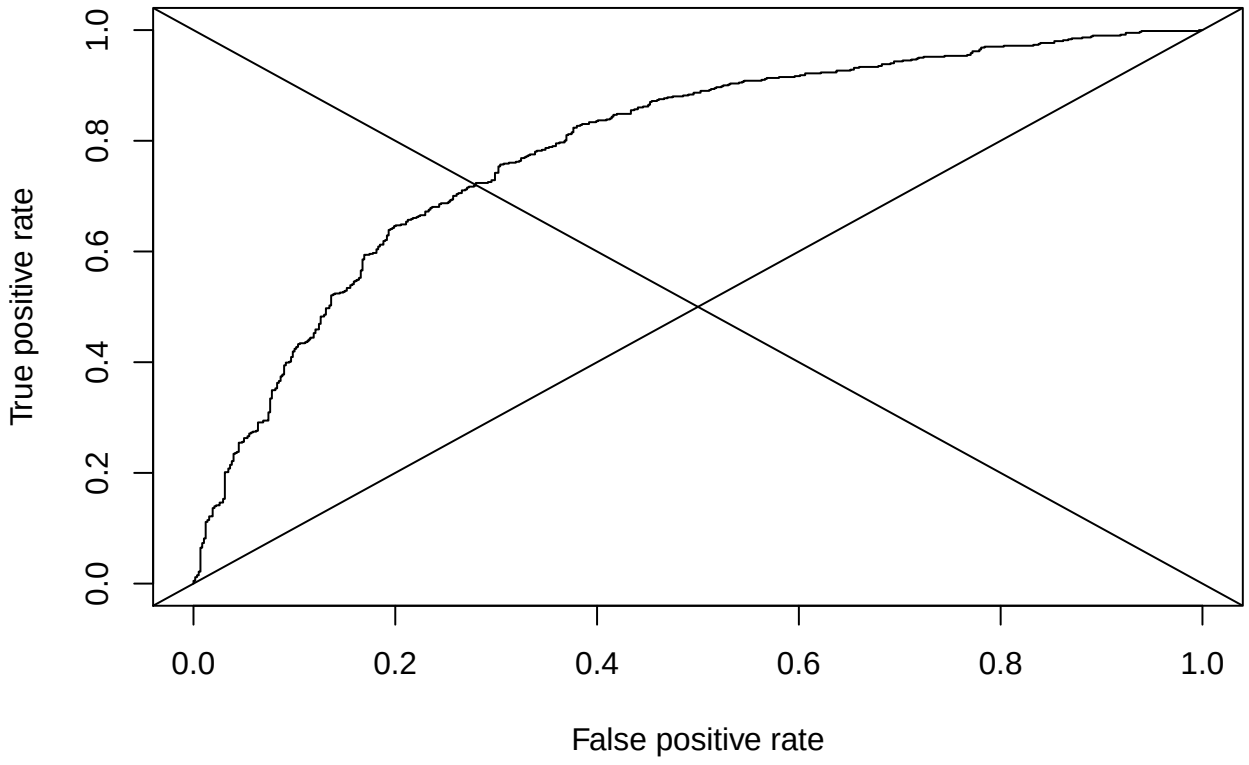
La matrice de confusion résultante ne diffère pas de celles que nous avions obtenues par la méthode de l'approche par échantillon de validation. Ceci nous conforte dans les démarches réalisées.

De même, nous pouvons mettre en oeuvre l'autre méthode de validation croisée.

IV. Approche par validation croisée Leave-One-Out (LOOCV)

Pour rappel, cette méthode consiste à ajuster le modèle sur les n-1 observations (journées ici), et à effectuer la prédiction sur l'observation mise de côté. Nous répétons ensuite cette procédure n fois.
Nous calculerons les scores d'affectation respectifs et verrons la courbe ROC (et l'Auc) qui s'ensuivent.
Nous utiliserons ici la base meteotrain d'origine sans avoir à la diviser : en effet, c'est la même dataset qui sert à la fois d'apprentissage (sur le n-1 échantillon) et de validation (sur l'observation i).

```
# Préparation d'un vecteur pour collecter les scores #
pred.LOOCVmodel <- rep(0,nrow(meteotrain))

# Mise en oeuvre LOOCV #
for ( i in 1:nrow(meteotrain)) {
  # modèle
    lg.LOOCVmodel = glm(pluie.demain~.,data = meteotrain[-i,],
                        family = binomial)
  # prédiction
    pred.LOOCVmodel[i] = predict(lg.LOOCVmodel,newdata=meteotrain[i,],type="response")
}
print(summary(pred.LOOCVmodel))
```



L'Auc qui en résulte est à 0.788, ce qui est très satisfaisant comme valeur.
Nous avons vu plus haut que tous nos modèles ont un Auc dans cette fourchette [0.7,0.8].

V. Decision du modèle retenu et application sur l'échantillon meteotest

Nous avons élu le modèle **lg.FWDBicModel_acp** comme meilleur modèle après nos diverses analyses.
En ultime étape, nous allons enfin prédire sur la nouvelle dataset Meteotest la variable pluie.demain, qui indiquera si le temps sera pluvieux ou sec pour chaque journée mentionnée.

```
# Standardisation de la base sans les variables supplémentaires de l'acp #
meteotest.std <- as.data.frame(scale(meteotest[,-c(1:4)],
                center = TRUE,scale = TRUE))

#Projection de la matrice test.meteotrain sur les axes factoriels de ACP#
pdtmat = as.matrix(meteotest.std) %*% as.matrix(acp.res.var.coord)

meteotest.proj <- matrix (rep(0, nrow(meteotest.std)*
                              ncol(acp.res$ind$coord)),
                          nrow(meteotest.std),
                          ncol(acp.res$ind$coord))
for (i in 1:8){
  meteotest.proj[,i] <- pdtmat[,i]/sqrt(acp.eig.val[i,1])
}

new.meteotest_proj = as.data.frame(meteotest.proj)

# Renommage des colonnes
colnames(new.meteotest_proj) = colnames(acp.res.var.coord)
```

```
pred.meteotest = predict(lg.FWDBicModel_acp,new.meteotest_proj,
                         type = "response")

# Conversion en binaire 0/1 au seuil de proba = 0/5 #
prd.meteotest = as.factor(ifelse(pred.meteotest>0.5,1,0))

# Fusion de meteotest et prd.meteotest #
fichier_predictions = as.data.frame(cbind(meteotest,prd.meteotest))
colnames(fichier_predictions)[45] <- "pluie.demain"
```

Voici le fichier de nos prédictions sur la base Meteotest.

```
write.csv(fichier_predictions,"/Users/Feno/Desktop/MeteotestPrevisions.csv", quote=FALSE)
```

Conclusion

Pour conclure, nous avons mené à bien notre mission de prédire la pluie sur l'échantillon meteo.test. Rappelons que nous avons effectué la validation selon 3 approches : approche par l'échantillon de validation (80/20 train-test sur la meteotrain), approche par K-folds cross validation et l'approche par la Leave-One-Out cross validation.

Avec la 1ère approche (par échantillon de validation), nous avons dû procéder à la division de la base Meteotrain en 2 (80/20) et mettre en oeuvre les recherches à partir de la base d'apprentissage de celle-ci. (c'est-à-dire sur les 80% de la meteo.train).
Au départ nous avions plus de 40 variables, nous avons opté en premier lieu pour la construction de modèles sur la totalité de ces variables (avec un premier modèle saturé en guise de référence) et malgré la complexité de la tâche et la lourdeur des calculs parfois, nous avons obtenu 6 modèles satisfaisants.

Dans un deuxième temps, la très grande taille de la base nous a amené à réfléchir sur la solution pour la traiter au mieux.
C'est pourquoi il nous a paru judicieux de procéder à une analyse en composantes principales (ACP) qui se prête bien à cette étude car elle permet de réduire au mieux la dimension de la base tout en gardant le maximum d'informations. Ainsi, nous avons réussi à construire 3 autres modèles avec les composantes principales comme régresseurs cette fois-ci.

Les 9 modèles ainsi construits ont été testés pour leur qualité prédictive et leur capacité à s'ajuster au mieux aux données.

Le modèle que nous avons alors décidé de retenir est le modèle issu de l'Acp que nous avons obtenu par une méthode de recherche Forward avec le critère Bic (au minimum). Celui-ci réunissait globalement les meilleurs scores en terme de performance prédictive et de qualité d'ajustement.

En seconde et troisième vue, nous avons essayé les 2 approches par K-folds cross validation et Leave-one-out cross validation. Les résultats nous ont conforté dans l'ordre de grandeur attendu des chiffres pour l'approche par l'échantillon de validation.

Enfin, nous avons visualisé les résidus sans approfondir les résultats.

Le fichier des résultats de prédiction de la Meteo.test se trouve dans le répertoire créé sur le lien de dépot Github : https://github.com/mioraandriantseheno/-GLM-Select-Models-Meteo-Project-