

Modèle linéaire généralisé

Noémie Chardin

2024-06-24

Exploration des données

Description statistique

Nous allons faire une description statistique des données du dataset d'entraînement :

```
summary(data_train)
```

```

##          X           Year        Month       Day        Hour
## Min.   : 2.0   Min.   :2010   Min.   : 1.000   Min.   : 1.0   Min.   :0
## 1st Qu.: 721.5 1st Qu.:2012  1st Qu.: 3.000   1st Qu.: 8.0   1st Qu.:0
## Median :1451.0 Median :2014  Median : 6.000   Median :16.0   Median :0
## Mean    :1459.8 Mean    :2014  Mean    : 6.436   Mean    :15.8   Mean    :0
## 3rd Qu.:2189.0 3rd Qu.:2016  3rd Qu.: 9.000   3rd Qu.:23.0   3rd Qu.:0
## Max.   :2940.0  Max.   :2018  Max.   :12.000   Max.   :31.0   Max.   :0
##      Minute     Temperature   Relative.Humidity Sea.Level.Pressure
## Min.   :0       Min.   :-7.63   Min.   :38.33    Min.   : 978.9
## 1st Qu.:0       1st Qu.: 6.71   1st Qu.:64.82    1st Qu.:1012.4
## Median :0       Median :12.08   Median :72.21    Median :1017.0
## Mean    :0       Mean    :12.23   Mean    :71.40    Mean    :1017.0
## 3rd Qu.:0       3rd Qu.:17.54   3rd Qu.:78.63    3rd Qu.:1022.0
## Max.   :0       Max.   :29.45   Max.   :95.54    Max.   :1042.4
##      Total.Precipitation   Snowfall      Total.Cloud.Cover High.Cloud.Cover
## Min.   : 0.000   Min.   :0.00000   Min.   : 0.00   Min.   : 0.000
## 1st Qu.: 0.000   1st Qu.:0.00000   1st Qu.: 23.80   1st Qu.: 1.657
## Median : 0.100   Median :0.00000   Median : 51.67   Median : 11.880
## Mean    : 2.085   Mean    :0.04965   Mean    : 50.76   Mean    : 20.284
## 3rd Qu.: 2.300   3rd Qu.:0.00000   3rd Qu.: 78.53   3rd Qu.: 33.260
## Max.   :31.500   Max.   :8.61000   Max.   :100.00   Max.   :100.000
##      Medium.Cloud.Cover Low.Cloud.Cover Sunshine.Duration Shortwave.Radiation
## Min.   : 0.00   Min.   : 0.00   Min.   : 0.0   Min.   : 265.2
## 1st Qu.: 1.83   1st Qu.: 9.42   1st Qu.:114.3   1st Qu.:2096.2
## Median : 24.98   Median : 36.35   Median :366.8   Median :3675.3
## Mean    : 31.50   Mean    :39.34   Mean    :373.1   Mean    :3984.6
## 3rd Qu.: 54.21   3rd Qu.: 65.76   3rd Qu.:587.7   3rd Qu.:5723.6
## Max.   :100.00   Max.   :100.00   Max.   :1015.8   Max.   :8363.3
##      Wind.Speed.10   Wind.Direction.10 Wind.Speed.80   Wind.Direction.80
## Min.   : 1.260   Min.   : 11.19   Min.   : 1.34   Min.   : 12.18
## 1st Qu.: 6.428   1st Qu.:152.40   1st Qu.: 8.68   1st Qu.:157.42
## Median : 9.195   Median :206.36   Median :12.41   Median :213.78
## Mean    :10.707   Mean    :201.82   Mean    :14.28   Mean    :206.23
## 3rd Qu.:12.977   3rd Qu.:254.19   3rd Qu.:17.61   3rd Qu.:259.06
## Max.   :42.210   Max.   :331.67   Max.   :54.03   Max.   :333.43
##      Wind.Speed.900   Wind.Direction.900 Wind.Gust   Temperature.max
## Min.   : 2.25   Min.   : 17.37   Min.   : 2.25   Min.   : -3.84
## 1st Qu.:13.02   1st Qu.:144.02   1st Qu.: 9.48   1st Qu.:10.58
## Median :19.57   Median :233.47   Median :14.06   Median :16.54
## Mean    :24.57   Mean    :206.22   Mean    :16.69   Mean    :16.54
## 3rd Qu.:32.10   3rd Qu.:265.93   3rd Qu.:21.15   3rd Qu.:22.36
## Max.   :97.06   Max.   :344.82   Max.   :79.38   Max.   :35.77
##      Temperature.min   Relative.Humidity.max Relative.Humidity.min
## Min.   : -12.520   Min.   : 59.00   Min.   :19.00
## 1st Qu.: 3.350   1st Qu.: 83.00   1st Qu.:45.00
## Median : 8.005   Median : 89.00   Median :54.00
## Mean    : 8.062   Mean    : 87.69   Mean    :54.04
## 3rd Qu.:13.092   3rd Qu.: 94.00   3rd Qu.:63.00
## Max.   :23.940   Max.   :100.00   Max.   :92.00
##      Sea.Level.Pressure.max Sea.Level.Pressure.min Total.Cloud.Cover.max
## Min.   : 981.9   Min.   : 977   Min.   : 0.00
## 1st Qu.:1015.4   1st Qu.:1009   1st Qu.:100.00
## Median :1019.5   Median :1015   Median :100.00
## Mean    :1019.9   Mean    :1014   Mean    : 88.23
## 3rd Qu.:1024.7   3rd Qu.:1019   3rd Qu.:100.00
## Max.   :1045.4   Max.   :1039   Max.   :100.00
##      Total.Cloud.Cover.min High.Cloud.Cover.max High.Cloud.Cover.min

```

```

## Min. : 0.000      Min. : 0.00      Min. : 0.0000
## 1st Qu.: 0.000    1st Qu.: 15.00    1st Qu.: 0.0000
## Median : 0.000    Median : 97.00    Median : 0.0000
## Mean   : 8.692    Mean   : 60.17    Mean   : 0.9432
## 3rd Qu.: 2.400    3rd Qu.:100.00    3rd Qu.: 0.0000
## Max.   :100.000   Max.   :100.00    Max.   :100.0000
## Medium.Cloud.Cover.max Medium.Cloud.Cover.min Low.Cloud.Cover.max
## Min. : 0.00      Min. : 0.000      Min. : 0
## 1st Qu.: 22.75    1st Qu.: 0.000    1st Qu.:100
## Median :100.00    Median : 0.000    Median :100
## Mean   : 70.94    Mean   : 2.097    Mean   : 80
## 3rd Qu.:100.00    3rd Qu.: 0.000    3rd Qu.:100
## Max.   :100.00    Max.   :100.000    Max.   :100
## Low.Cloud.Cover.min Wind.Speed.10.max Wind.Speed.10.min Wind.Speed.80.max
## Min. : 0.000      Min. : 2.52      Min. : 0.00      Min. : 3.98
## 1st Qu.: 0.000    1st Qu.:12.32    1st Qu.: 1.14    1st Qu.:18.27
## Median : 0.000    Median :17.36    Median : 2.41    Median :23.85
## Mean   : 3.879    Mean   :19.06    Mean   : 3.57    Mean   :25.35
## 3rd Qu.: 0.000    3rd Qu.:23.44    3rd Qu.: 4.45    3rd Qu.:29.92
## Max.   :100.000   Max.   :79.99    Max.   :27.73    Max.   :93.84
## Wind.Speed.80.min Wind.Speed.900.max Wind.Speed.900.min Wind.Gust.max
## Min. : 0.000      Min. : 4.02      Min. : 0.00      Min. : 4.32
## 1st Qu.: 1.140    1st Qu.:24.54    1st Qu.: 3.05    1st Qu.:19.08
## Median : 2.600    Median :37.12    Median : 6.73    Median :26.10
## Mean   : 4.727    Mean   :41.82    Mean   :11.09    Mean   :29.31
## 3rd Qu.: 5.830    3rd Qu.:54.37    3rd Qu.:15.31    3rd Qu.:37.08
## Max.   :37.700    Max.   :136.25   Max.   :76.13    Max.   :95.04
## Wind.Gust.min     pluie.demain
## Min. : 0.000      Min. :0.0000
## 1st Qu.: 2.160    1st Qu.:0.0000
## Median : 3.960    Median :1.0000
## Mean   : 6.502    Mean   :0.5093
## 3rd Qu.: 8.280    3rd Qu.:1.0000
## Max.   :57.960    Max.   :1.0000

```

Nous observons :

- la variable pluie.demain qui est sous forme de bouléen, et qui est la variable à expliquer
- 46 variables variables numériques

Exclusion des variables non pertinentes

Nous allons tout d'abord exclure de notre analyse les variables suivantes :

- X qui est une variable à valeur unique pour chaque ligne
- Les variables Hour et Minute qui sont des variables identifiant l'heure de mesure, elles n'ont donc pas d'impact sur la probabilité de pleuvoir le lendemain
- La variable Year: dans notre cas, nous devons prédire la pluie du lendemain pour des données passées, nous pourrions donc conserver la variable. En revanche dans le but d'utiliser ce modèle pour prédire la météo sur les prochains jours dans le futur, cette variable ne nous permettra pas de prédire le risque de pluie du lendemain quand nous changerons d'année car nous n'aurons pas de point de données d'apprentissage sur cette nouvelle année. Nous observons une corrélation très faible de Year avec la variable pluie.demain (corrélation de 0.07), nous ne risquons pas de perdre beaucoup d'information en la retirant. Nous allons donc l'exclure.

```
round(cor(d$pluie.demain, d$Year),2)
```

```
## [1] 0.07
```

Traitement des variables temporelles

Nous conservons les variables Day et Month :

- La variable Month pourrait représenter un indicateur de risque de pluie car celle-ci nous indique la saison
- Nous nous attendons en revanche à ce que la variable Day ne soit pas pertinente dans le modèle car elle n'est pas un marqueur de saison ; nous verrons cela lors de la phase de sélection de modèle

Ces deux variables sont numériques, mais représentent en réalité des catégories. Nous allons donc les transformer.

```
# Transformation des variables Day et Month
d$Day <- as.character(d$Day)
d$Month <- as.character(d$Month)
d <- d %>%
  mutate(Month = case_when(
    Month == "1" ~ "January",
    Month == "2" ~ "February",
    Month == "3" ~ "March",
    Month == "4" ~ "April",
    Month == "5" ~ "May",
    Month == "6" ~ "June",
    Month == "7" ~ "July",
    Month == "8" ~ "August",
    Month == "9" ~ "September",
    Month == "10" ~ "October",
    Month == "11" ~ "November",
    Month == "12" ~ "December"
  ))
)
```

Analyse des données manquantes

```
colSums(is.na(data_train))
```

```

##          X           Year        Month
##          0            0           0
##      Day          Hour       Minute
##          0            0           0
## Temperature   Relative.Humidity  Sea.Level.Pressure
##          0            0           0
## Total.Precipitation   Snowfall  Total.Cloud.Cover
##          0            0           0
## High.Cloud.Cover   Medium.Cloud.Cover  Low.Cloud.Cover
##          0            0           0
## Sunshine.Duration   Shortwave.Radiation  Wind.Speed.10
##          0            0           0
## Wind.Direction.10   Wind.Speed.80  Wind.Direction.80
##          0            0           0
## Wind.Speed.900   Wind.Direction.900  Wind.Gust
##          0            0           0
## Temperature.max   Temperature.min  Relative.Humidity.max
##          0            0           0
## Relative.Humidity.min  Sea.Level.Pressure.max  Sea.Level.Pressure.min
##          0            0           0
## Total.Cloud.Cover.max  Total.Cloud.Cover.min  High.Cloud.Cover.max
##          0            0           0
## High.Cloud.Cover.min  Medium.Cloud.Cover.max  Medium.Cloud.Cover.min
##          0            0           0
## Low.Cloud.Cover.max  Low.Cloud.Cover.min  Wind.Speed.10.max
##          0            0           0
## Wind.Speed.10.min   Wind.Speed.80.max  Wind.Speed.80.min
##          0            0           0
## Wind.Speed.900.max   Wind.Speed.900.min  Wind.Gust.max
##          0            0           0
## Wind.Gust.min     pluie.demain
##          0            0           0

```

Nous n'observons aucune donnée manquante.

Analyse de la variable à expliquer

La variable à expliquer est une variable sous forme de bouléen que nous avons transformée en variable binaire prenant la valeur 0 ou 1. Nous sommes donc ici dans le cadre d'une regression logistique : nous cherchons à prédire si le jour suivant risque d'être pluvieux ou non.

```
table(data_train$pluie.demain)
```

```

## 
##   0   1
## 579 601

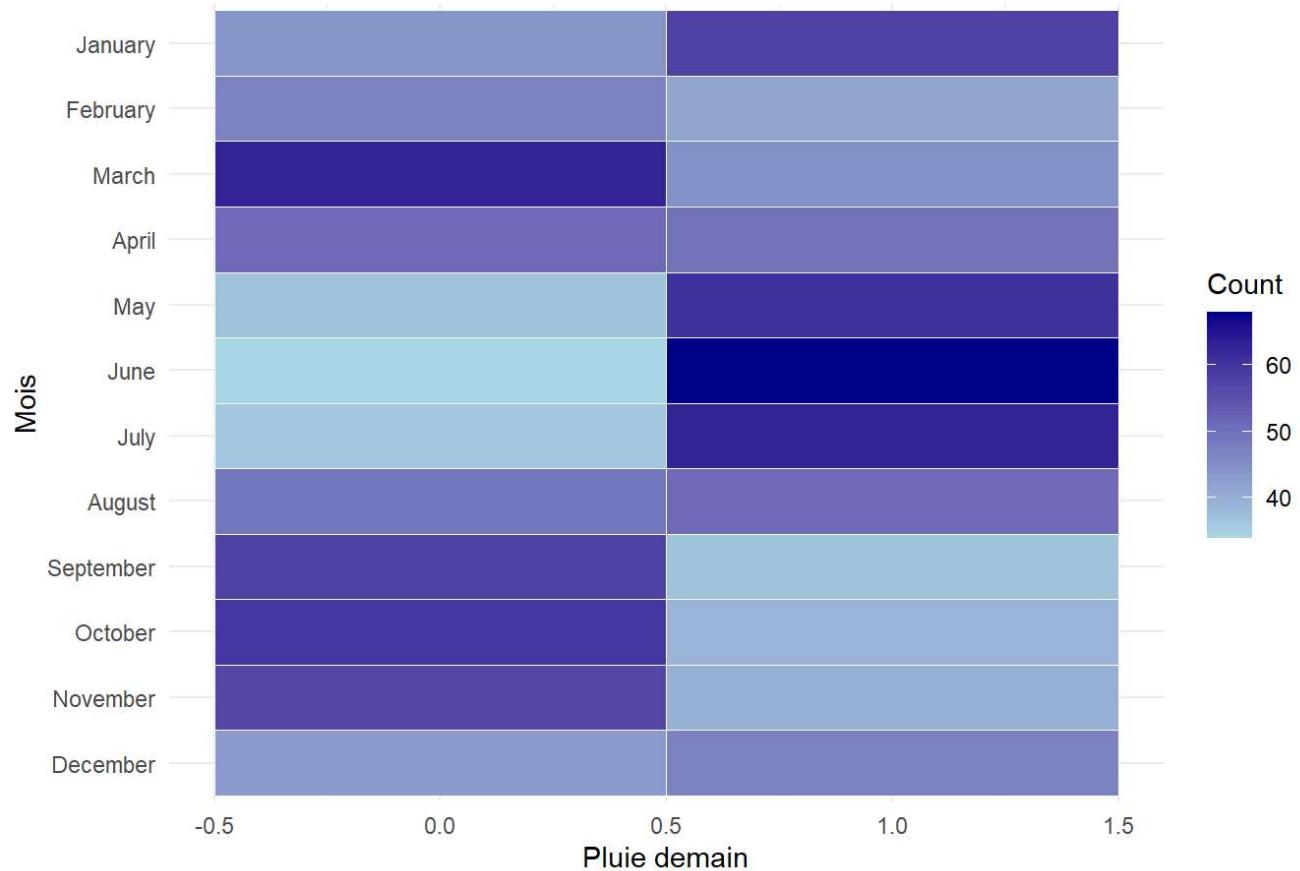
```

Nos classes sont équitablement réparties, avec 50,9% des observations étant des jours suivis d'un jour pluvieux. Nous n'allons donc pas rencontrer de problème de sous-représentation de l'une des catégories.

Analyse des variables explicatives temporelles

Nous allons tout d'abord analyser la variable Month et sa relation avec la variable pluie.demain à l'aide d'un tableau de contingence.

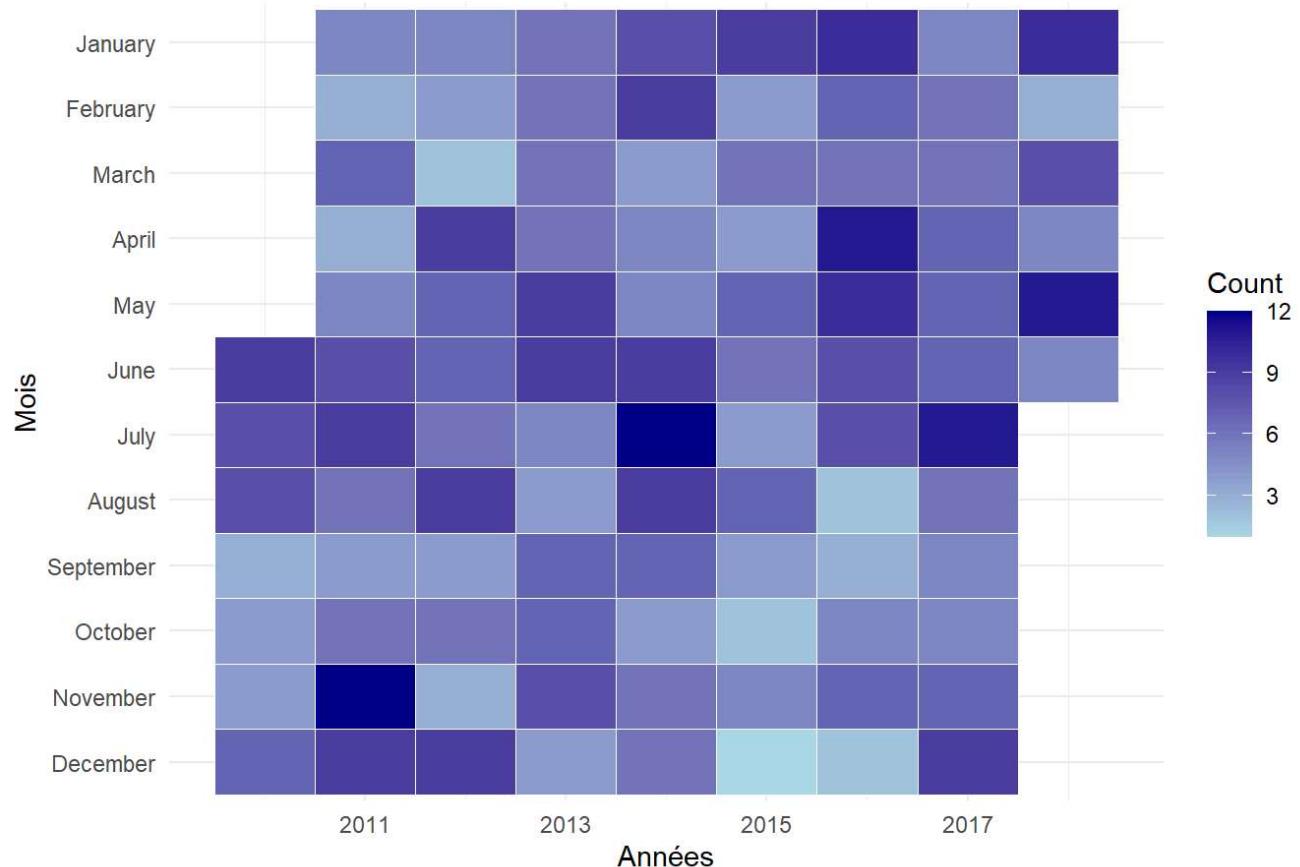
Table de contingence



Il semble y avoir un risque de pluie le lendemain plus important en janvier et entre mai et juillet. A contrario, nous observons des lendemains en moyenne moins pluvieux entre septembre et novembre et en mars.

Nous allons observer si cette tendance est similaire par année.

Pluie le lendemain par mois et par année

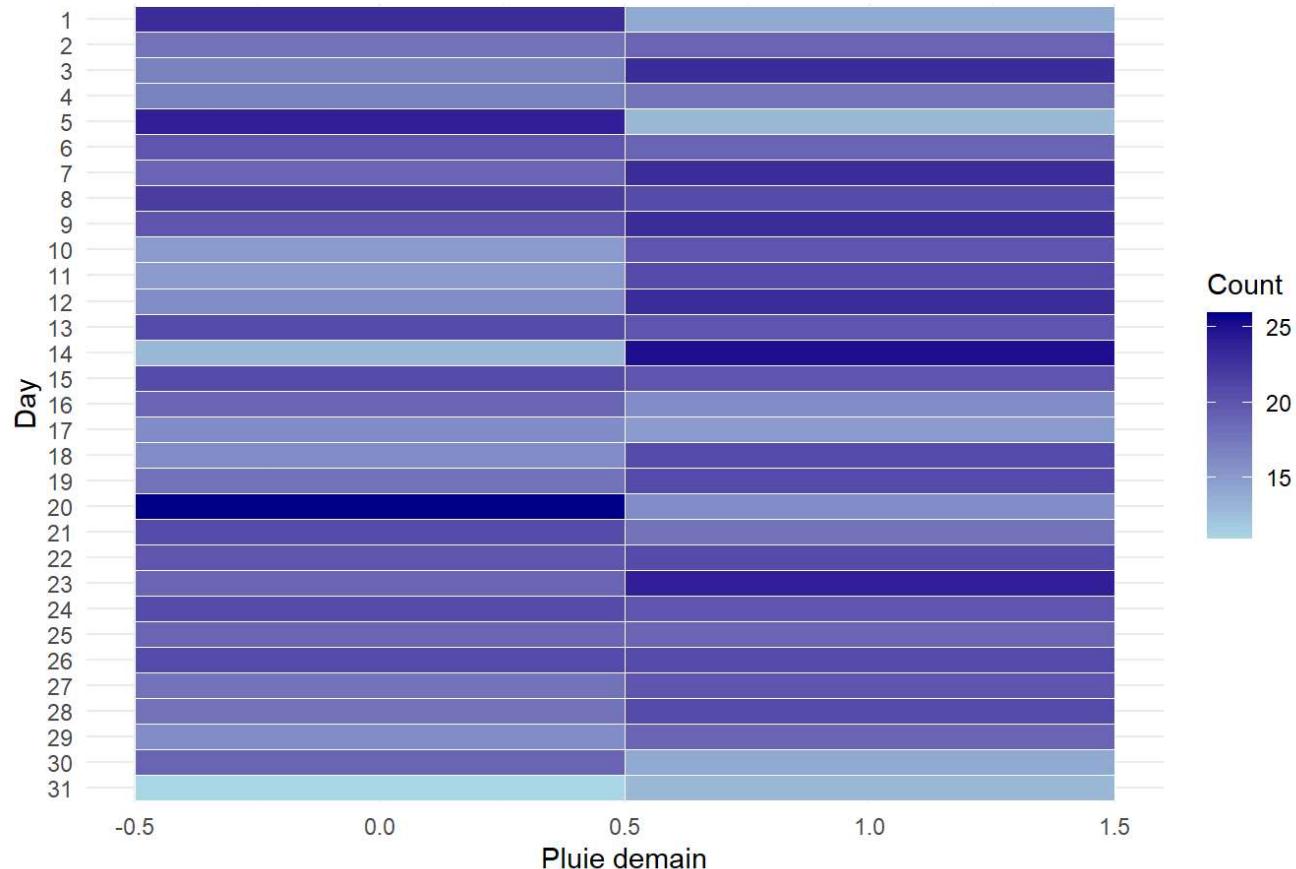


Nous observons des comportements qui varient d'une année à l'autre.

Cette analyse nous laisse penser qu'une certaine relation existe entre la variable Month et la variable pluie.demain, mais qu'elle n'est pas forcément stable d'une année à l'autre. Nous verrons dans la phase de sélection de modèle si cette variable apporte ou non de l'information pertinante au modèle.

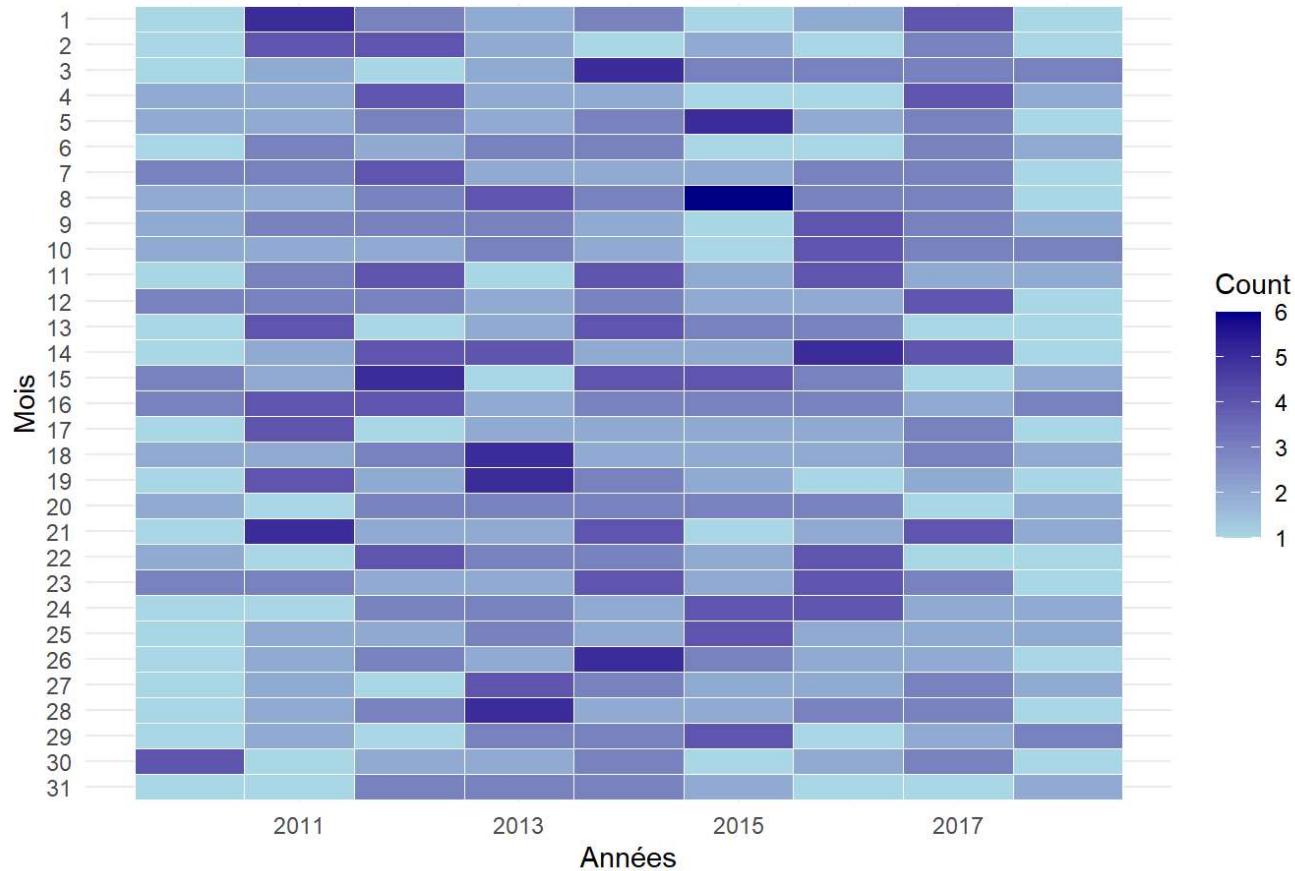
Nous allons à présent analyser la variable Day et sa relation avec la variable pluie.demain.

Table de contingence



Nous pourrions penser ici que le risque de pluie le lendemain serait plus élevé les 3, 14 et 23 du mois. En revanche, cela ne paraît que peu plausible. Nous allons analyser le comportement par année.

Pluie le lendemain par mois et par année



Nous observons des comportements très différents par année; l'information du jour ne semble donc que peu pertinente pour le modèle. Nous allons tout de même conserver cette variable et valider ou infirmer notre hypothèse lors de la phase de sélection de modèle.

Analyse des variables explicatives quantitatives

Nous allons à présent analyser les variables explicatives quantitatives, leur relation entre elles, et leur relation avec la variable pluie.demain.

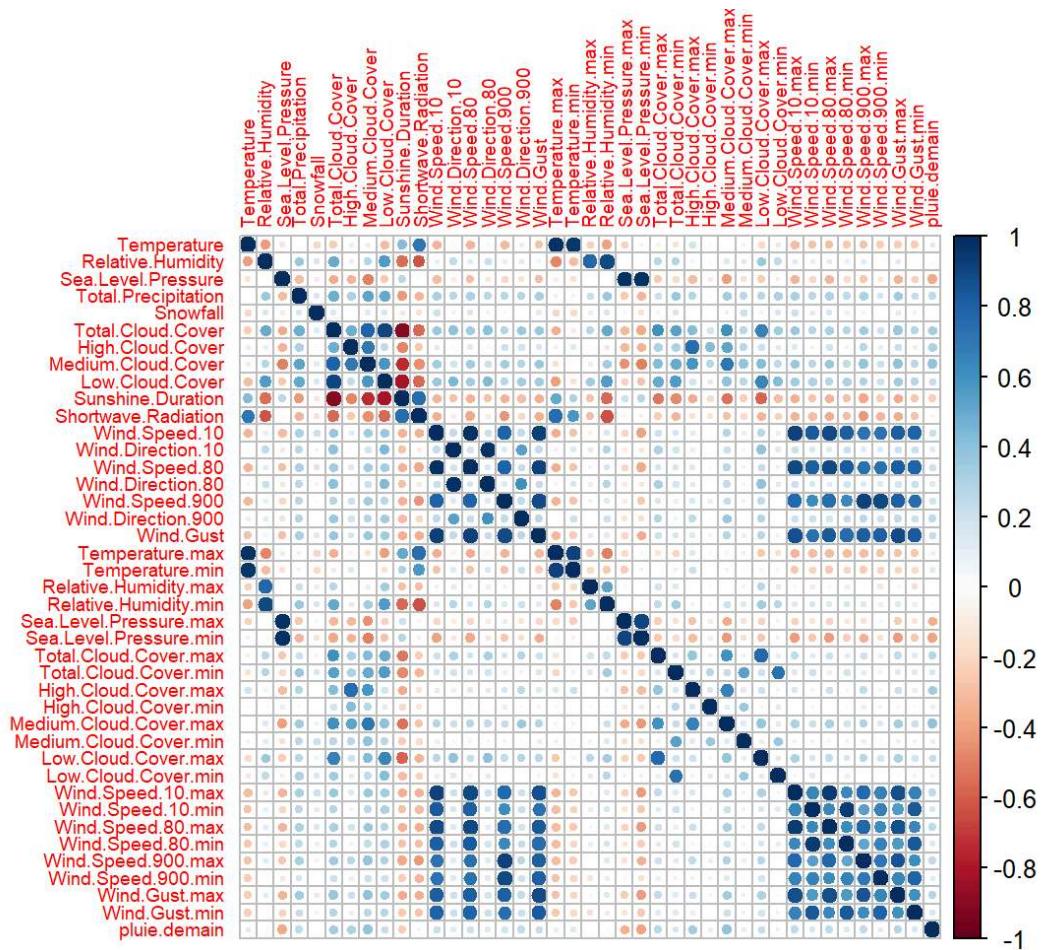
Ces variables quantitatives représentent des indicateurs de mesures journaliers. Pour les indicateurs pour lesquels cela est pertinent, nous avons trois variables de mesure : la moyenne, le maximum et le minimum.

Nous retrouvons finalement les variables suivantes :

- La température (min, max, mean)
- L'humidité relative (min, max, mean)
- La pression atmosphérique (min, max, mean)
- Les précipitations
- Les chutes de neige
- La couverture nuageuse, à bas, moyenne et haute altitude (min, max, mean)
- La durée d'ensoleillement
- Les ondes courtes
- La vitesse du vent à différentes altitudes (min, max, mean)
- La direction du vent à différentes altitudes
- Les rafales de vent (min, max, mean)

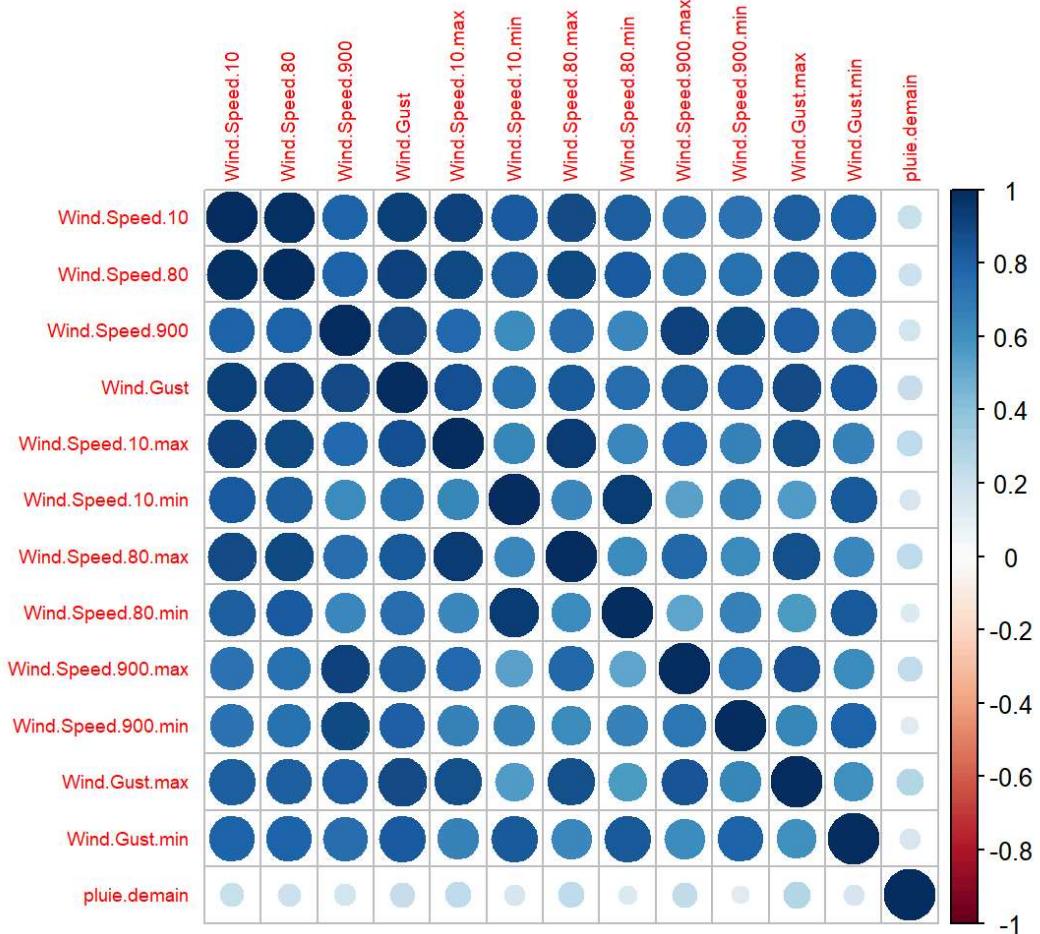
De part la nature de ces variables, nous pouvons donc nous attendre à observer des corrélations importantes entre certaines d'entre elles. Nous allons sélectionner nos variables pour limiter les corrélations trop élevées entre les variables. Cela permettra également de réduire le nombre de variables afin d'être plus efficace lors de notre phase de sélection de modèle.

Nous allons analyser les corrélations entre les variables.



Nous observons des corrélations très fortes entre certaines variables, notamment entre des variables basées sur des indicateurs météorologiques similaires. Nous allons donc faire une sélection au sein de nos variables.

Variables de vitesse du vent

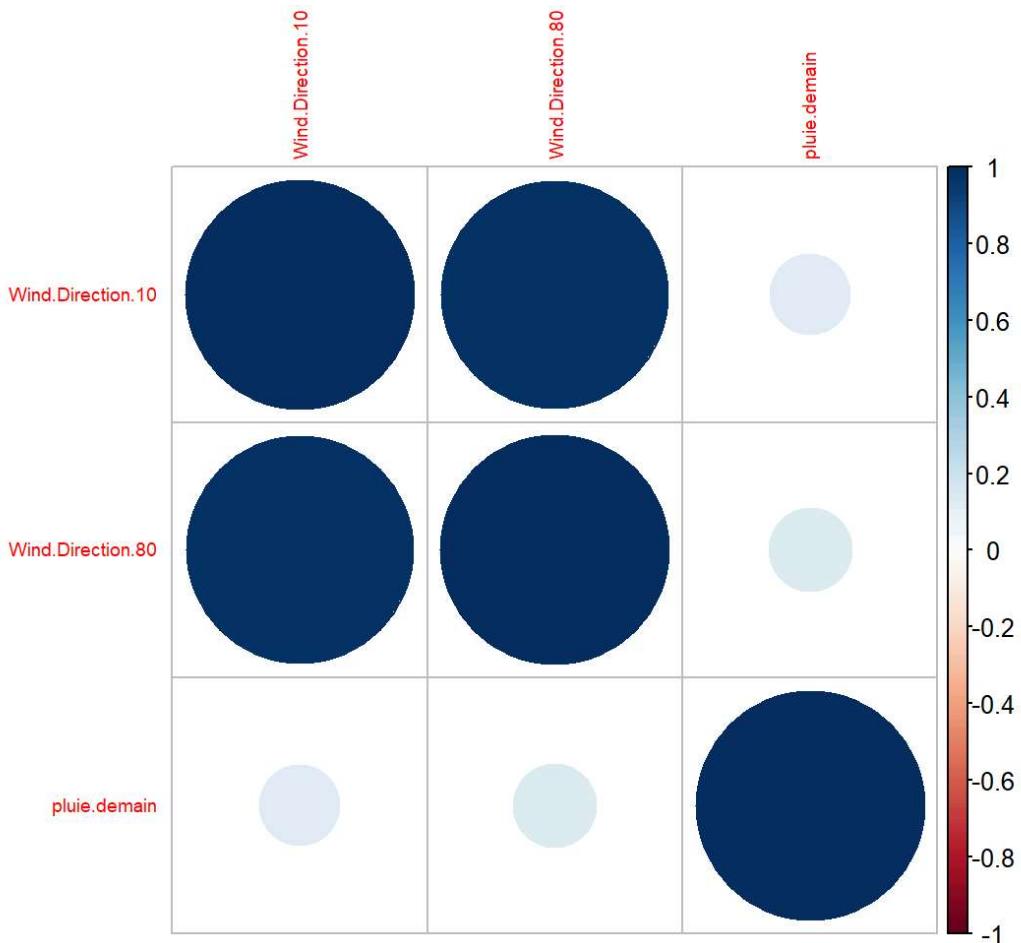


Nous observons des corrélations assez fortes entre les différents indicateurs liés à la vitesse du vent. Afin de simplifier le modèle, nous n'allons donc conserver qu'une seule variable.

Nous allons conserver la variable Wind.Gust.max qui a la corrélation la plus forte avec la variable pluie.demain avec une corrélation positive de 0.276. Cette corrélation n'est en revanche pas très forte.

```
d <- subset(d, select = -c(Wind.Speed.10,
  Wind.Speed.80,
  Wind.Speed.900,
  Wind.Gust,
  Wind.Speed.10.max,
  Wind.Speed.10.min,
  Wind.Speed.80.max,
  Wind.Speed.80.min,
  Wind.Speed.900.max,
  Wind.Speed.900.min,
  Wind.Gust.min))
```

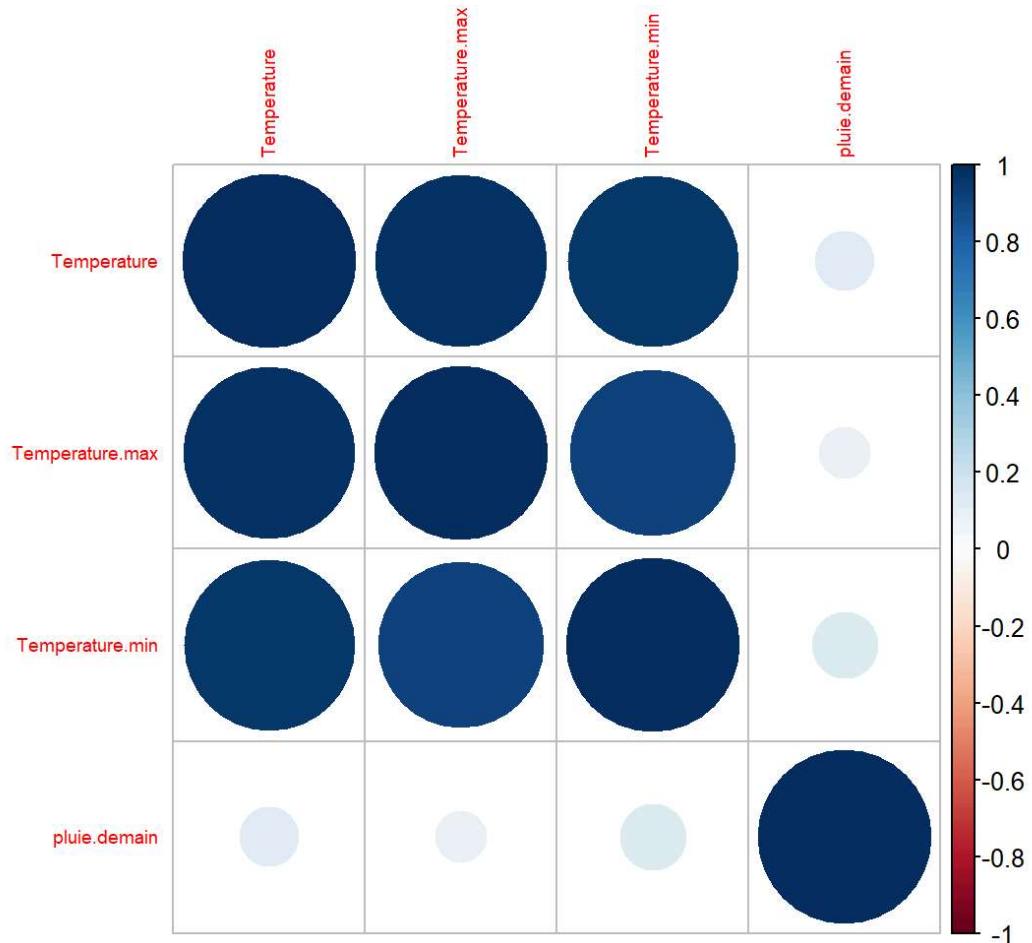
Variables de direction du vent



Nous observons des corrélations assez fortes entre les différents indicateurs liés à la direction du vent. Nous observons une corrélation de 0.97 entre wind.direction.10 et wind.direction.80. Nous n'allons donc conserver qu'une seule variable : la variable wind.direction.80 qui a la corrélation la plus forte avec la variable pluie.demain avec une corrélation positive de 0.133. Cette corrélation est faible.

```
d <- subset(d, select = -c(Wind.Direction.10))
```

Variables de température

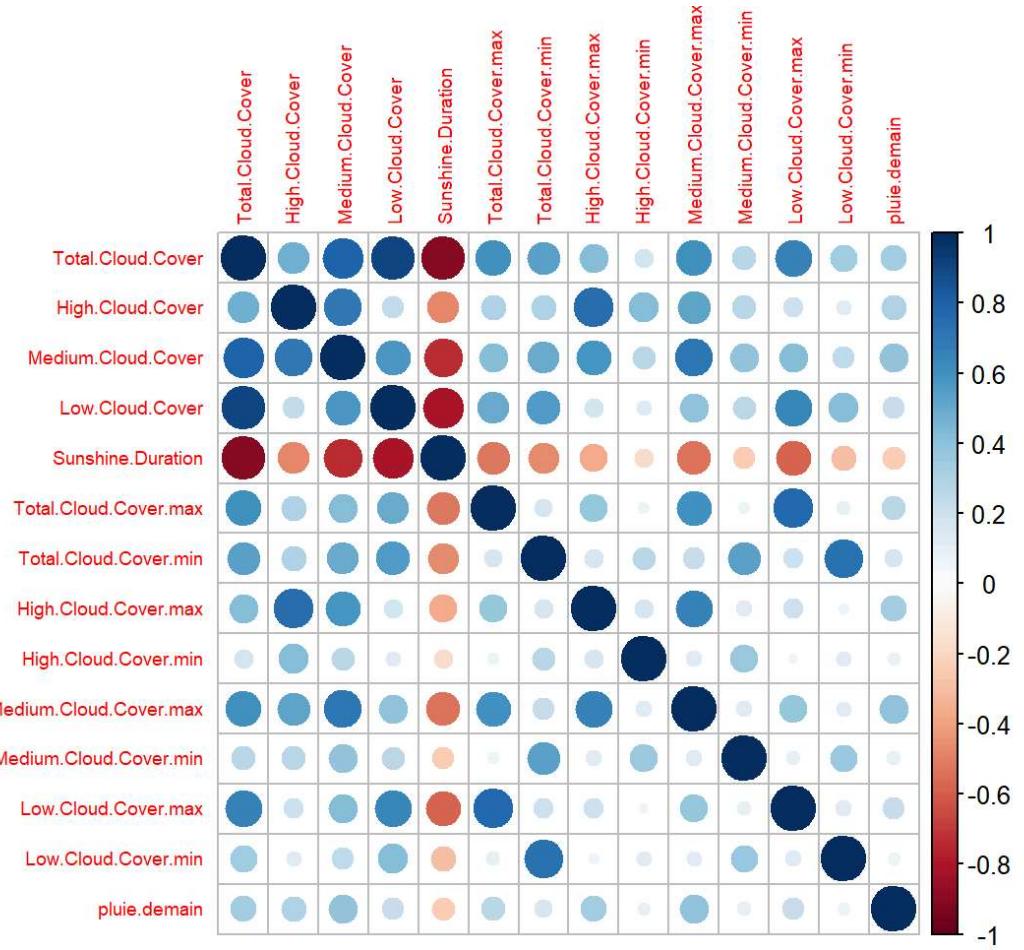


Nous allons conserver la variable Temperature.min qui a la corrélation la plus forte avec la variable pluie.demain avec une corrélation positive de 0.146. Cette corrélation est également faible.

```
d <- subset(d, select = -c(Temperature, Temperature.max))
```

Variables d'ensoleillement et de couverture nuageuse

Nous observons une forte corrélation entre la variable d'ensoleillement et de couverture nuageuse.



Nous observons deux corrélations très fortes :

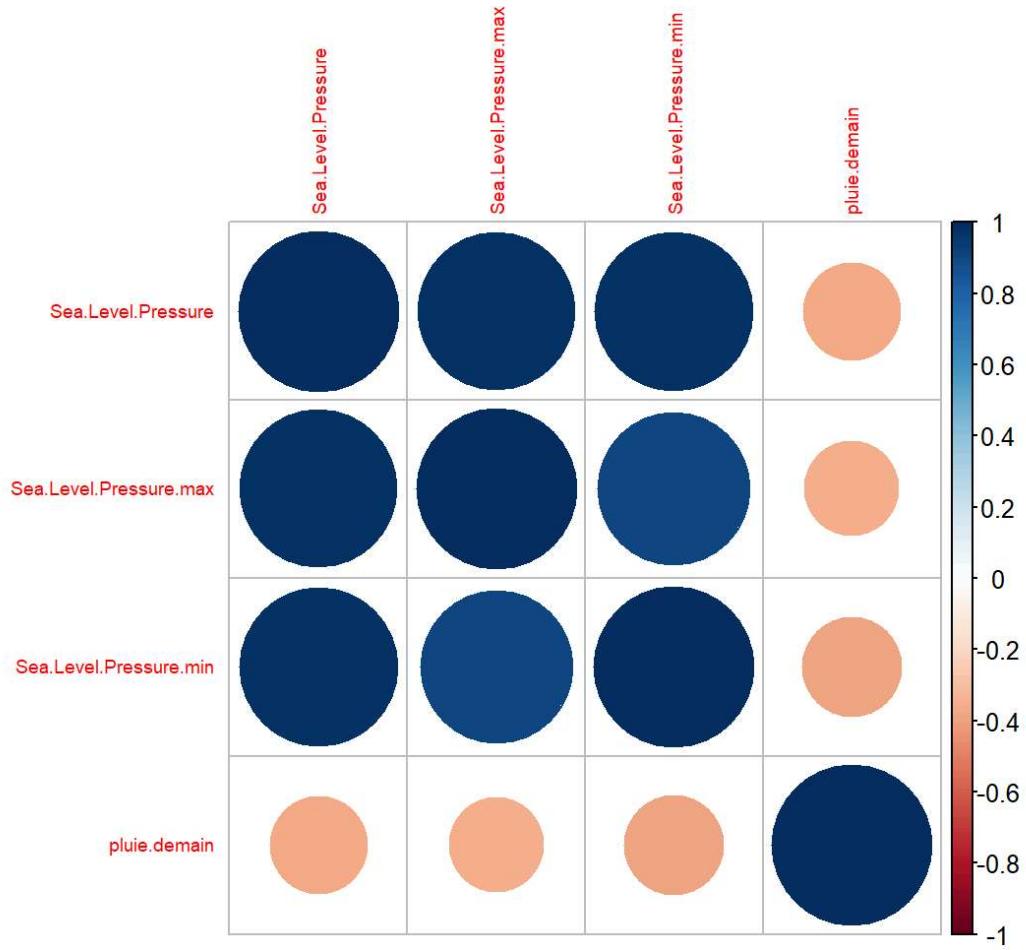
- La corrélation entre sunshine.duration et Total.Cloud.Cover avec une corrélation de -0.91
- La corrélation entre Low.Cloud.Cover et Total.Cloud.Cover avec une corrélation de 0.90

Nous conservons la variable total.cloud.cover qui présente une corrélation plus importante avec la variable pluie.demain avec une corrélation de 0.32.

```
d <- subset(d, select = -c(Sunshine.Duration, Low.Cloud.Cover))
```

Variables de pression atmosphérique

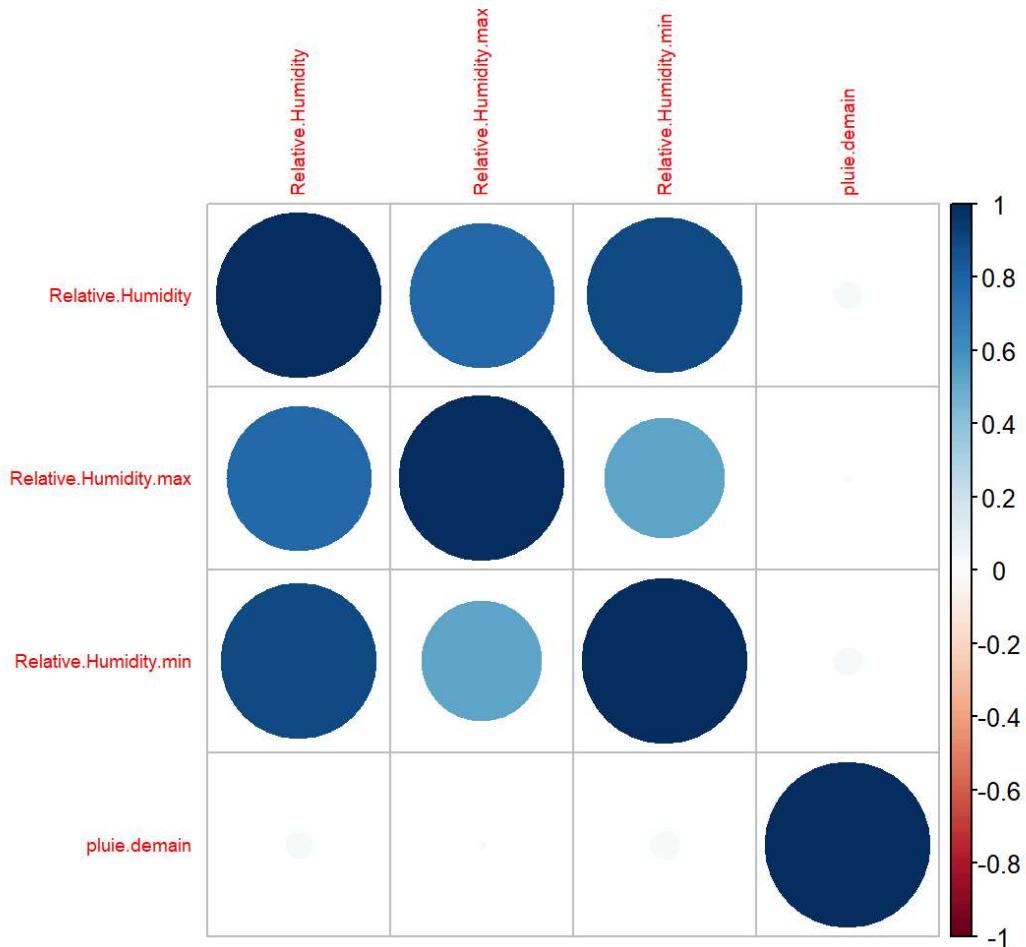
Nous observons des corrélations assez fortes entre les différents indicateurs liés à la direction du vent.



Nous observons des corrélations élevées de 0.97 entre les différentes variables de pression atmosphérique. Nous n'allons donc conserver que la variable Sea.Level.Pressure.min qui a la corrélation la plus forte avec la variable pluie.demain avec une corrélation négative de -0.387. Cette corrélation est assez importante en comparaison aux autres variables.

```
d <- subset(d, select = -c(Sea.Level.Pressure,
                           Sea.Level.Pressure.max))
```

Variables d'humidité relative



Nous observons une corrélation élevée de 0.89 entre relative.humidity et relative.humidity.min Nous allons donc exclure la variable relative.humidity qui a la corrélation la plus faible avec la variable pluie.demain.

En revanche, les coefficient de corrélation des trois variable avec la variable pluie.demain sont très faibles, laissant penser qu'elles ne sont pas correlées. Nous allons faire un test de corrélation entre la variable Relative.Humidity.min et pluie.demain.

```
cor.test(data_train$pluie.demain, data_train$Relative.Humidity.min)
```

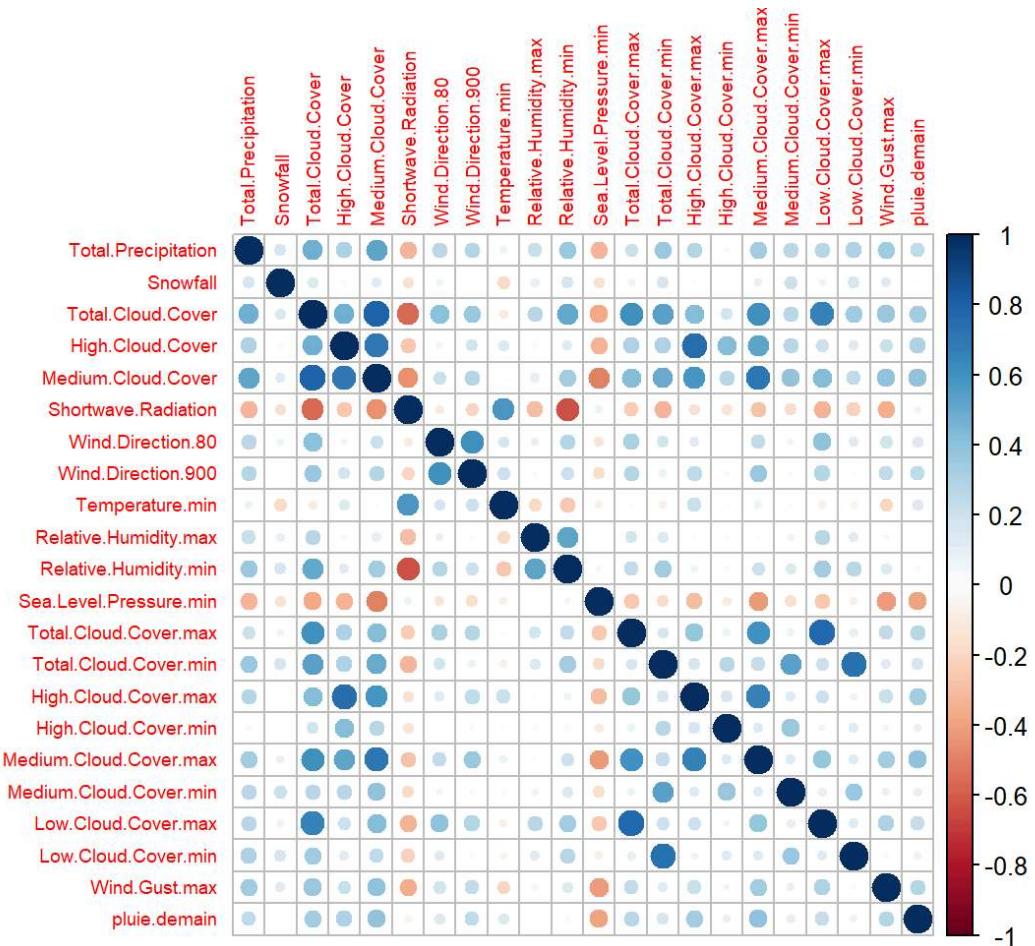
```
## 
## Pearson's product-moment correlation
##
## data: data_train$pluie.demain and data_train$Relative.Humidity.min
## t = 1.0267, df = 1178, p-value = 0.3048
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.02721437 0.08681864
## sample estimates:
##       cor
## 0.02989942
```

Ici nous observons une p-value élevée signifiant que nous acceptons l'hypothèse 0, et que les deux variables ne sont pas correlées. Nous allons tout de même conserver les variables Relative.Humidity.min et Relative.Humidity.max pour la phase de sélection de modèle, mais nous pouvons nous attendre à ce que leur effet ne soit pas significatif.

```
d <- subset(d, select = -c(Relative.Humidity))
```

Nouveau dataset

Notre nouveau dataset contient un total de 24 variables.



Modélisation et choix de modèles

Nous sommes ici dans un cas de classification, nous allons donc utiliser une regression logistique pour modéliser nos données. Le nombre de variables étant élevé, nous allons opter pour une méthode pas à pas. Nous gardons en tête la limite de cette méthode, qui ne teste pas toutes les combinaisons possibles, et qui ne permet donc pas d'obtenir un minimum global.

Nous allons tester les différentes méthodes pas à pas et en comparer les résultats.

Séparation de la base de donnée d'entraînement et de validation

Nous allons couper notre dataset afin de pouvoir évaluer les performances de notre modèle sur des données qu'il n'aura pas vu. Nous allons garder environ 80% des observations dans notre dataset d'entraînement et 20% des observations dans notre dataset d'évaluation.

```
d_train <- d[1:940,]  
d_val <- d[941:1180,]
```

Sélection de modèle à l'aide de la fonction step

La fonction step est une fonction très efficace pour faire de la prédiction. Nous allons utiliser cette fonction pour sélectionner les variables de notre modèle, en testant les trois méthodes : backward, forward et stepwise.

Nous allons chercher à minimiser l'AIC car nous nous cherchons à maximiser l'efficacité de notre prédiction et n'avons pas besoin d'être parcimonieux. Nous masquons ici les sorties des fonctions pour plus de lisibilité.

Fonction step - backward selection

```
# Fit du modèle complet  
mod <- glm(pluie.demain ~ . , family = binomial, data = d_train)  
  
# Sélection de modèle avec la fonction step backward  
stepbackw <- step(mod, direction="backward")
```

```
# Afficher le modèle choisi  
stepbackw
```

```
##  
## Call: glm(formula = pluie.demain ~ Month + Snowfall + High.Cloud.Cover +  
##           Wind.Direction.80 + Wind.Direction.900 + Relative.Humidity.min +  
##           Sea.Level.Pressure.min + Total.Cloud.Cover.max + Total.Cloud.Cover.min +  
##           Medium.Cloud.Cover.max + Wind.Gust.max, family = binomial,  
##           data = d_train)  
##  
## Coefficients:  
##              (Intercept)          MonthNovember          MonthOctober  
##              80.344672             -1.023820            -0.457549  
##              MonthSeptember          MonthAugust          MonthJuly  
##              -0.668068             -0.005253             0.547761  
##              MonthJune              MonthMay          MonthApril  
##              0.713040             -0.132297            -0.457078  
##              MonthMarch          MonthFebruary          MonthJanuary  
##              -0.587384             -0.224843            -0.032891  
##              Snowfall            High.Cloud.Cover          Wind.Direction.80  
##              -0.466335             0.013840            -0.003145  
##              Wind.Direction.900      Relative.Humidity.min      Sea.Level.Pressure.min  
##              0.004722             -0.014928            -0.080598  
##              Total.Cloud.Cover.max    Total.Cloud.Cover.min      Medium.Cloud.Cover.max  
##              0.007534              0.009712             0.006559  
##              Wind.Gust.max  
##              0.019453  
##  
## Degrees of Freedom: 939 Total (i.e. Null);  918 Residual  
## Null Deviance:      1303  
## Residual Deviance: 992   AIC: 1036
```

Fonction step - forward selection

```
# Fit du modèle le plus simple  
mod1 <- glm(pluie.demain ~ 1, family = binomial, data = d_train )  
  
# Sélection de modèle avec la fonction step forward  
stepfwd <- step(mod1, scope = list(lower = mod1, upper = mod), data = d_train, direction="forward")
```

```
# Afficher le modèle choisi  
stepfwd
```

```

## 
## Call: glm(formula = pluie.demain ~ Medium.Cloud.Cover + Sea.Level.Pressure.min +
##           Month + High.Cloud.Cover.max + Wind.Direction.900 + Wind.Gust.max +
##           Snowfall + Total.Cloud.Cover.max + Wind.Direction.80 + Relative.Humidity.min +
##           Total.Cloud.Cover.min, family = binomial, data = d_train)
##
## Coefficients:
## (Intercept) Medium.Cloud.Cover Sea.Level.Pressure.min
##             78.774309          0.009986         -0.078973
## MonthNovember MonthOctober MonthSeptember
##            -0.981486          -0.476392         -0.674128
## MonthAugust MonthJuly MonthJune
##            -0.038220          0.530301          0.681955
## MonthMay MonthApril MonthMarch
##            -0.179931          -0.483985         -0.641233
## MonthFebruary MonthJanuary High.Cloud.Cover.max
##            -0.267414          -0.029996          0.006345
## Wind.Direction.900 Wind.Gust.max Snowfall
##            0.004879          0.018413         -0.495531
## Total.Cloud.Cover.max Wind.Direction.80 Relative.Humidity.min
##            0.009387          -0.003402         -0.016750
## Total.Cloud.Cover.min
##            0.008068
##
## Degrees of Freedom: 939 Total (i.e. Null);  918 Residual
## Null Deviance:      1303
## Residual Deviance: 991.5      AIC: 1036

```

Fonction step - stepwise selection

```

# Sélection de modèle avec la fonction step stepwise en partant du modèle le plus simple
stepboth1 <- step(mod1, scope = list(lower = mod1, upper = mod), data =d_train, direction="both")

```

```

# Afficher le modèle choisi
stepboth1

```

```

## Call: glm(formula = pluie.demain ~ Medium.Cloud.Cover + Sea.Level.Pressure.min +
##           Month + High.Cloud.Cover.max + Wind.Direction.900 + Wind.Gust.max +
##           Snowfall + Total.Cloud.Cover.max + Wind.Direction.80 + Relative.Humidity.min +
##           Total.Cloud.Cover.min, family = binomial, data = d_train)
##
## Coefficients:
## (Intercept) Medium.Cloud.Cover Sea.Level.Pressure.min
## 78.774309      0.009986      -0.078973
## MonthNovember MonthOctober MonthSeptember
## -0.981486     -0.476392     -0.674128
## MonthAugust    MonthJuly   MonthJune
## -0.038220      0.530301      0.681955
## MonthMay       MonthApril MonthMarch
## -0.179931     -0.483985     -0.641233
## MonthFebruary MonthJanuary High.Cloud.Cover.max
## -0.267414     -0.029996      0.006345
## Wind.Direction.900 Wind.Gust.max Snowfall
## 0.004879      0.018413     -0.495531
## Total.Cloud.Cover.max Wind.Direction.80 Relative.Humidity.min
## 0.009387     -0.003402     -0.016750
## Total.Cloud.Cover.min
## 0.008068
##
## Degrees of Freedom: 939 Total (i.e. Null);  918 Residual
## Null Deviance:      1303
## Residual Deviance: 991.5      AIC: 1036

```

Sélection de modèle avec la fonction step stepwise en partant du modèle complet

```

mod <- glm(pluie.demain ~ . , family = binomial, data = d_train)
stepboth2 <- step(mod, direction = "both")

```

Afficher le modèle choisi

```

stepboth2

```

```

## 
## Call: glm(formula = pluie.demain ~ Month + Snowfall + High.Cloud.Cover +
##           Wind.Direction.80 + Wind.Direction.900 + Relative.Humidity.min +
##           Sea.Level.Pressure.min + Total.Cloud.Cover.max + Total.Cloud.Cover.min +
##           Medium.Cloud.Cover.max + Wind.Gust.max, family = binomial,
##           data = d_train)
##
## Coefficients:
## (Intercept) MonthNovember MonthOctober
## 80.344672   -1.023820   -0.457549
## MonthSeptember MonthAugust MonthJuly
## -0.668068   -0.005253   0.547761
## MonthJune MonthMay MonthApril
## 0.713040   -0.132297   -0.457078
## MonthMarch MonthFebruary MonthJanuary
## -0.587384   -0.224843   -0.032891
## Snowfall High.Cloud.Cover Wind.Direction.80
## -0.466335   0.013840   -0.003145
## Wind.Direction.900 Relative.Humidity.min Sea.Level.Pressure.min
## 0.004722   -0.014928   -0.080598
## Total.Cloud.Cover.max Total.Cloud.Cover.min Medium.Cloud.Cover.max
## 0.007534   0.009712   0.006559
## Wind.Gust.max
## 0.019453
##
## Degrees of Freedom: 939 Total (i.e. Null);  918 Residual
## Null Deviance:      1303
## Residual Deviance: 992    AIC: 1036

```

Les fonctions step forward et step both en partant du modèle le plus simple sélectionne le même modèle.

Comparaison des modèles sélectionnés

Nous allons à présent comparer la performance des modèles sélectionnés en comparant leurs AIC.

```

df <- data.frame(
  Metric = "AIC",
  Farward = round(stepfwd$aic),
  Backward = round(stepbackw$aic),
  Both1 = round(stepboth1$aic),
  Both2 = round(stepboth2$aic))
print(df)

```

```

##   Metric Farward Backward Both1 Both2
## 1     AIC     1036     1036   1036   1036

```

Le modèle sélectionné par les fonctions step forward et step both en partant du modèle le plus simple montre le meilleur AIC.

Simplification du modèle

```
# Modèle choisi par step forward
mod2 <- glm(formula = stepfwd$formula,
            family = binomial,
            data = d_train)

summary(mod2)
```

```
##
## Call:
## glm(formula = stepfwd$formula, family = binomial, data = d_train)
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                78.7774309  13.356353   5.898 3.68e-09 ***
## Medium.Cloud.Cover        0.009986   0.004237   2.357 0.018445 *
## Sea.Level.Pressure.min   -0.078973   0.013017  -6.067 1.31e-09 ***
## MonthNovember              -0.981486   0.426451  -2.302 0.021362 *
## MonthOctober               -0.476392   0.409296  -1.164 0.244452
## MonthSeptember             -0.674128   0.415339  -1.623 0.104572
## MonthAugust                -0.038220   0.421283  -0.091 0.927714
## MonthJuly                  0.530301   0.426958   1.242 0.214221
## MonthJune                  0.681955   0.429027   1.590 0.111939
## MonthMay                   -0.179931   0.428149  -0.420 0.674301
## MonthApril                 -0.483985   0.449515  -1.077 0.281622
## MonthMarch                 -0.641233   0.445479  -1.439 0.150031
## MonthFebruary              -0.267414   0.435511  -0.614 0.539200
## MonthJanuary               -0.029996   0.413184  -0.073 0.942126
## High.Cloud.Cover.max      0.006345   0.002266   2.800 0.005112 **
## Wind.Direction.900         0.004879   0.001446   3.375 0.000738 ***
## Wind.Gust.max              0.018413   0.006644   2.772 0.005579 **
## Snowfall                   -0.495531   0.227437  -2.179 0.029350 *
## Total.Cloud.Cover.max     0.009387   0.003552   2.643 0.008223 **
## Wind.Direction.80          -0.003402   0.001721  -1.976 0.048116 *
## Relative.Humidity.min    -0.016750   0.008227  -2.036 0.041751 *
## Total.Cloud.Cover.min     0.008068   0.004340   1.859 0.063020 .
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 1303.11 on 939 degrees of freedom
## Residual deviance: 991.54 on 918 degrees of freedom
## AIC: 1035.5
##
## Number of Fisher Scoring iterations: 4
```

Nous observons que certaines variables ne sont pas significatives, avec des p-value supérieurs à 0.05. Nous allons donc retirer les variables les moins significatives.

Nous observons notamment que certaines modalités de la variable Month ne sont pas significatives. Nous allons donc exclure les variables des mois n'étant pas significatives. Pour valider la pertinence de retirer ces variables nous allons faire un test de anova entre l'ancien et le nouveau modèle.

Nous effectuons cela en plusieurs itérations en retirant les variables au fur et à mesure, et en réalisant des tests entre chaque itérations pour nous assurer de la pertinance ou non des variables.

Nous conservons finalement que les variables June, July and November.

```
mod3 <- glm(formula = pluie.demain ~
              June +
              July +
              November +
              Snowfall +
              High.Cloud.Cover +
              Wind.Direction.80 +
              Wind.Direction.900 +
              Relative.Humidity.min +
              Sea.Level.Pressure.min +
              Total.Cloud.Cover.max +
              Total.Cloud.Cover.min +
              Medium.Cloud.Cover.max +
              Wind.Gust.max,
              family = binomial,
              data = d_train)

summary(mod3)
```

```

## 
## Call:
## glm(formula = pluie.demain ~ June + July + November + Snowfall +
##      High.Cloud.Cover + Wind.Direction.80 + Wind.Direction.900 +
##      Relative.Humidity.min + Sea.Level.Pressure.min + Total.Cloud.Cover.max +
##      Total.Cloud.Cover.min + Medium.Cloud.Cover.max + Wind.Gust.max,
##      family = binomial, data = d_train)
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)           75.783512  12.246770  6.188 6.09e-10 ***
## June                  1.033206   0.278460  3.710 0.000207 ***
## July                  0.865462   0.273490  3.165 0.001553 **
## November              -0.737889   0.310694 -2.375 0.017550 *
## Snowfall              -0.426262   0.208703 -2.042 0.041108 *
## High.Cloud.Cover     0.013129   0.004391  2.990 0.002791 **
## Wind.Direction.80    -0.003351   0.001669 -2.008 0.044645 *
## Wind.Direction.900    0.005081   0.001443  3.522 0.000428 ***
## Relative.Humidity.min -0.012886   0.007175 -1.796 0.072483 .
## Sea.Level.Pressure.min -0.076563   0.012028 -6.365 1.95e-10 ***
## Total.Cloud.Cover.max  0.007017   0.003836  1.829 0.067338 .
## Total.Cloud.Cover.min  0.009890   0.003996  2.475 0.013319 *
## Medium.Cloud.Cover.max 0.006915   0.002687  2.574 0.010067 *
## Wind.Gust.max         0.021591   0.006216  3.474 0.000513 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 1303.11 on 939 degrees of freedom
## Residual deviance: 999.14 on 926 degrees of freedom
## AIC: 1027.1
##
## Number of Fisher Scoring iterations: 4

```

Toutes les variables de mois sont bien significatives.

```
anova(mod3,mod2)
```

	Resid. Df <dbl>	Resid. Dev <dbl>	Df <dbl>	Deviance <dbl>	Pr(>Chi) <dbl>
1	926	999.1357	NA	NA	NA
2	918	991.5415	8	7.594216	0.4740767
2 rows					

On ne rejette pas l'hypothèse nulle, les variables que nous avons exclues n'apportent donc pas d'information supplémentaire.

Nous observons en revanche que d'autres variables ne sont pas significatives: les variables Relative.Humidity.min et Total.Cloud.Cover.max. Nous allons itérer en retirant ces variables une à une, en faisant un test anova pour mesurer l'effet des variables, puis en s'assurant de la significativité des variables restantes. Nous nous arrêtons quand toutes les variables sont significatives.

Voici un résumé de nos itérations :

```

# Calcul des p-value des tests anova entre les modèles emboités
pvalue3 <- anova(mod2,mod3)$`Pr(>Chi)`
pvalue4 <- anova(mod3,mod4)$`Pr(>Chi)`
pvalue5 <- anova(mod4,mod5)$`Pr(>Chi)`

# Création d'un dataframe consolidant les AIC et les résultats des test anova pour chaque modèle
df <- data.frame(
  Metric = c("AIC", "P-value from anova"),
  model2 = c(round(mod2$aic,1), ""),
  model3 = c(round(mod3$aic,1), round(pvalue3[2],2)),
  model4 = c(round(mod4$aic,1), round(pvalue4[2],2)),
  model5 = c(round(mod5$aic,1), round(pvalue5[2],2))
)
print(df)

```

```

##           Metric model2  model3  model4  model5
## 1          AIC  1035.5 1027.10 1028.60 1029.10
## 2 P-value from anova          0.47    0.06    0.11

```

- modèle 2 : modèle sélectionné par la fonction step backward
- modèle 3 : modèle avec sélection sur les variables months
- modèle 4 : modèle avec exclusion de la variable Total.Cloud.Cover.max
- modèle 5 : modèle avec exclusion de la variable Relative.Humidity.min

Le modèle avec le meilleur AIC est le modèle 3. Le test anova entre les modèles 3 et 4 et les modèles 4 et 5 montre des p_values supérieurs mais proches de 0.05. Cela montre que l'effet des variables incluent dans le modèle 3 n'apportent pas forcément d'information supplémentaire, mais peuvent tout de même en apporter.

Nous cherchons ici à faire les meilleures prédictions; nous ne voulons pas perdre d'information donc nous allons conserver le modèle 3.

Concernant nos hypothèses initiales sur les variables, la variables Day n'est effectivement pas pertinente pour le modèle. En revanche, la variable Relative.Humidity.min apporte finalement de l'information pertinente.

Analyse des résidus

Nous construisons une dataframe avec les y, les y prédit et les résidus afin de les analyser.

```

df_mod3 <- data.frame(
  y = mod3$data$pluie.demain,
  y_pred = mod3$fitted.values,
  residuals_deviance = residuals(mod3, type = "deviance"),
  residuals_pearson = residuals(mod3, type = "pearson")
)

```

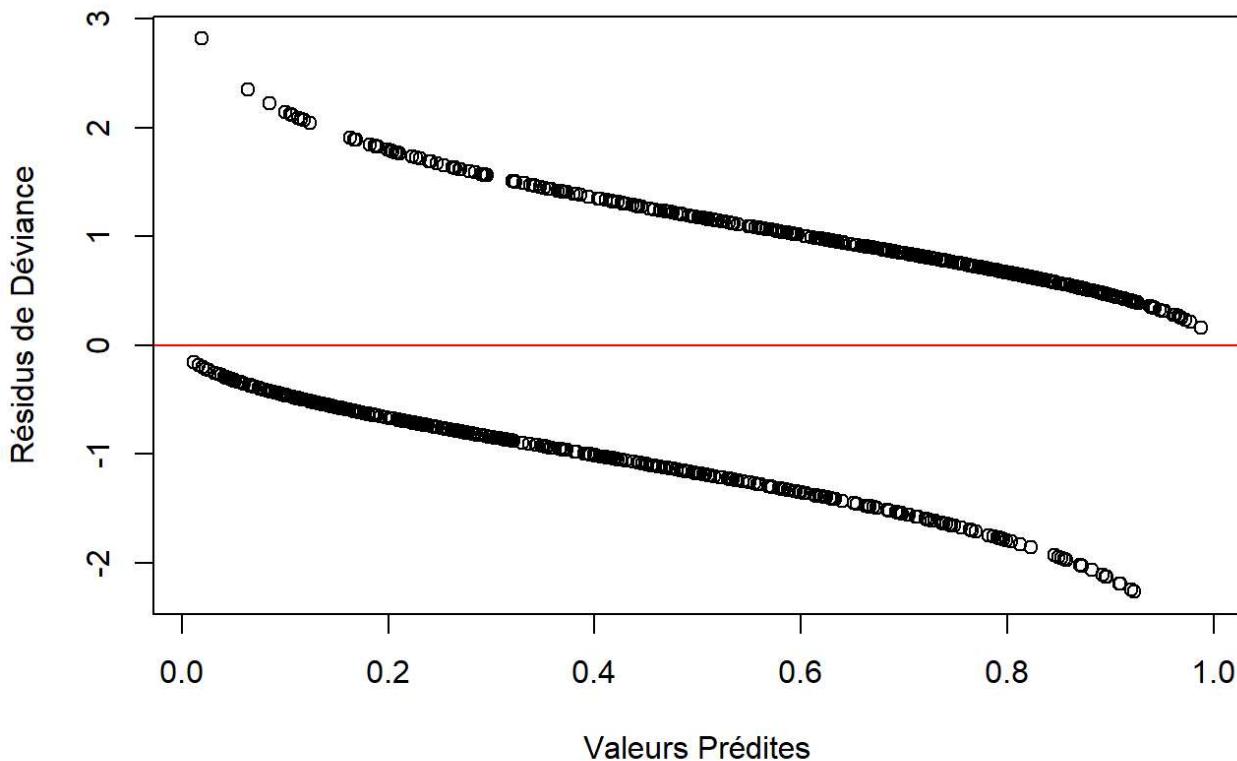
Nous allons à présent analyser la distributions des résidus en fonction des valeurs prédites.

```

# Diagramme de dispersion des résidus de déviance vs. valeurs prédites
plot(predict(mod3, type = "response"), df_mod3$residuals_deviance,
      xlab = "Valeurs Prédites", ylab = "Résidus de Déviance",
      main = "Résidus de Déviance vs. Valeurs Prédites")
abline(h = 0, col = "red")

```

Résidus de Déviance vs. Valeurs Prédites



Comme nous sommes ici dans un cas de prédiction binaire, l'analyse des résidus en fonction des valeurs prédites n'est pas très utile car nous observons deux lignes, l'une correspondant aux observations de la classe 1 et l'autre aux observations de la classe 0.

Afin d'analyser les résidus de manière plus comparable, nous allons créer des groupes de points ayant des probabilités prédites similaires, puis calculer un résidu moyen pour chaque groupe correspondant à la différence entre :

1. la part d'observations de la classe 1 au sein de chaque groupe
2. la probabilité prédictive moyenne de la classe 1 de chaque groupe

```

# Fonction pour créer des groupes par valeur de prédiction
binned_residuals_plot <- function(fitted, residuals, y, n_bins = 50) {
  bins <- cut(fitted, breaks = quantile(fitted, probs = seq(0, 1, length.out = n_bins + 1)), include.lowest = TRUE)
  binned_data <- data.frame(fitted, residuals, y, bins)

  # Agréger les résidus par groupe
  binned_summary <- aggregate(residuals ~ bins, data = binned_data, mean)
  binned_summary$n <- aggregate(fitted ~ bins, data = binned_data, length)$fitted
  binned_summary$y <- aggregate(y ~ bins, data = binned_data, sum)$y
  binned_summary$y_prop <- binned_summary$y / binned_summary$n
  binned_summary$fitted <- aggregate(fitted ~ bins, data = binned_data, mean)$fitted
  binned_summary$new_residuals <- binned_summary$y_prop - binned_summary$fitted

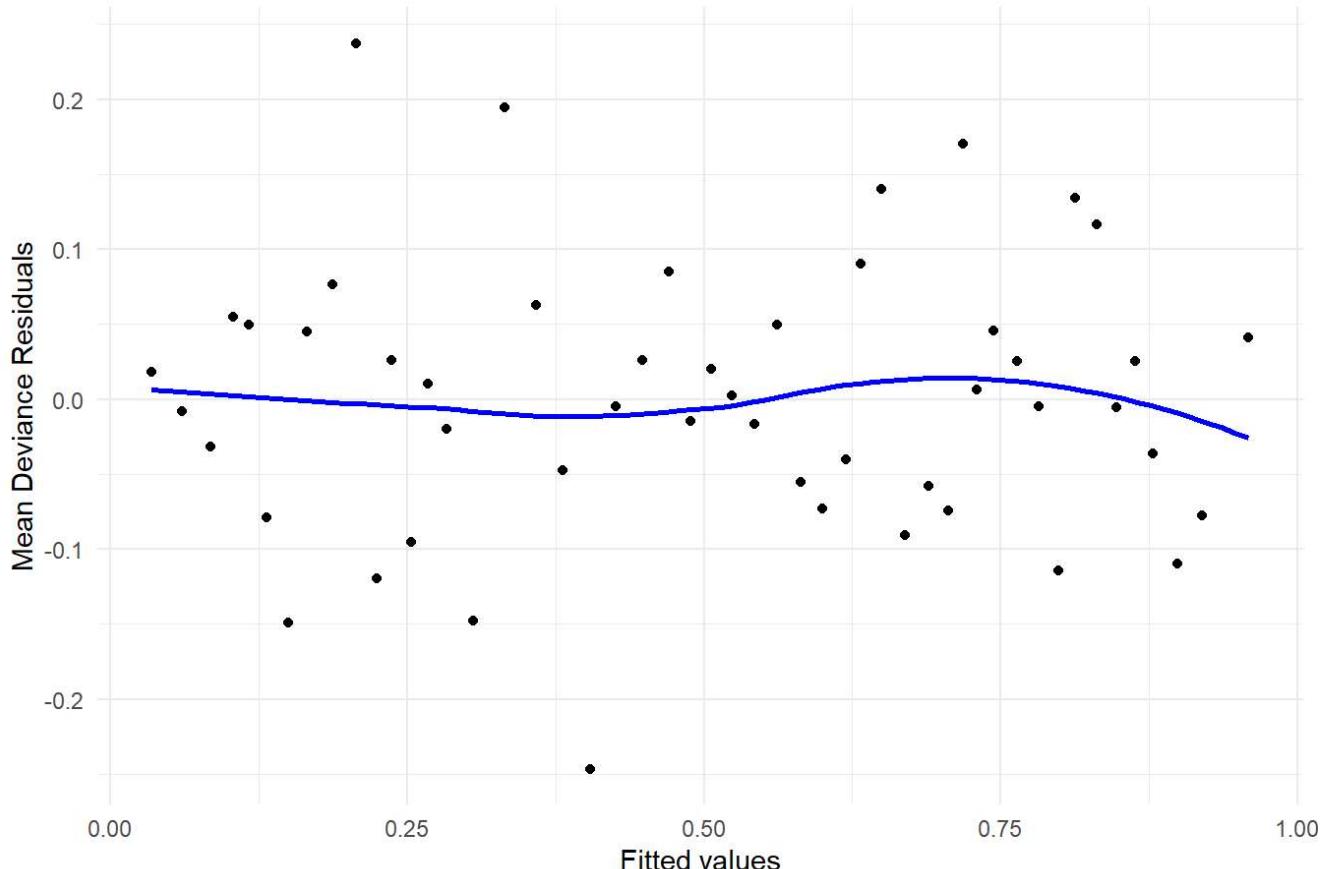
  # Plot des rédisus
  suppressWarnings(ggplot(binned_summary, aes(x = fitted, y = new_residuals)) +
    geom_point() +
    geom_smooth(method = "loess", se = FALSE, color = "blue") +
    labs(title = "Binned Residuals vs Fitted", x = "Fitted values", y = "Mean Deviance Residuals") +
    theme_minimal()
  )
}

# Calcul et plot des résidus sur nos données
binned_residuals_plot(df_mod3$y_pred, df_mod3$residuals_deviance, df_mod3$y)

```

```
## `geom_smooth()` using formula = 'y ~ x'
```

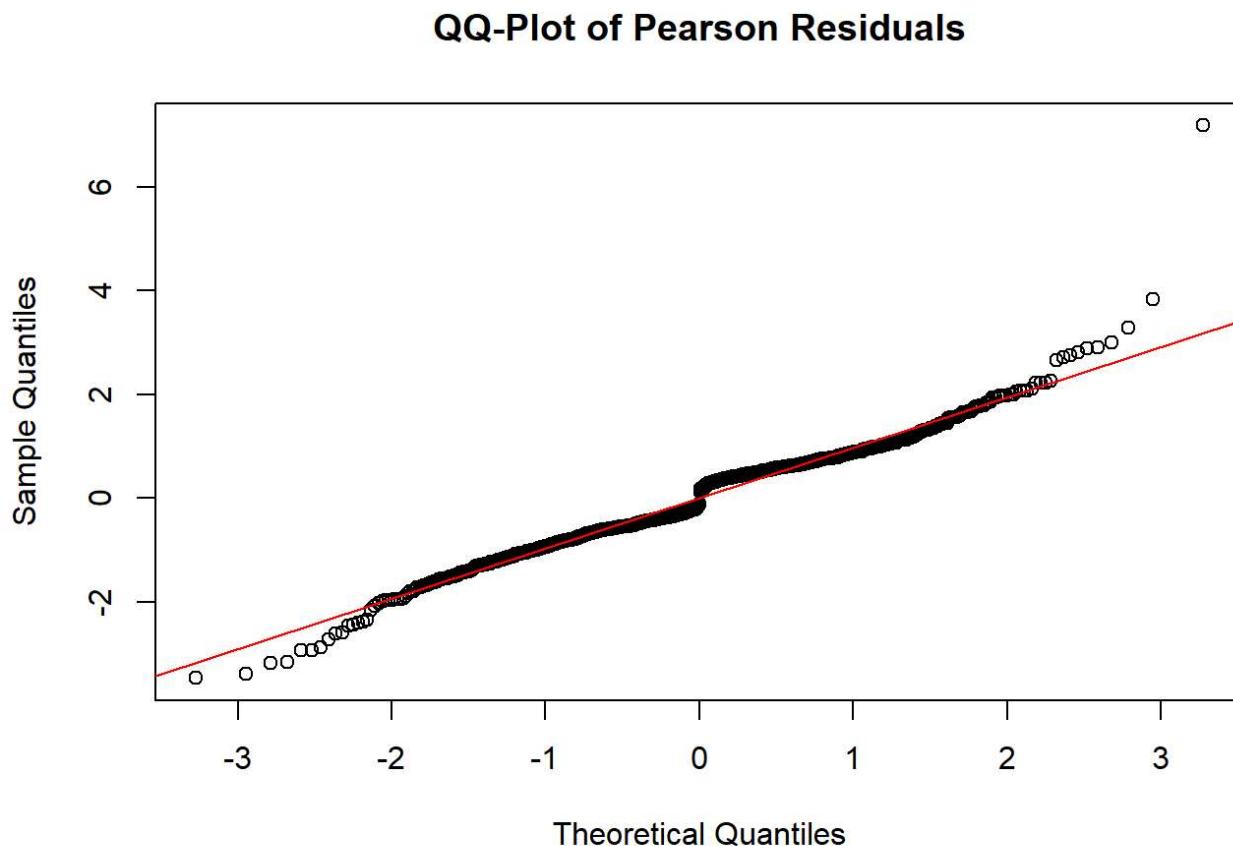
Binned Residuals vs Fitted



Nous observons que nos résidus sont centrés autour de 0 et que la tendance est globalement plate est lisse. Nous observons quelques mouvements dans la courbe et quelques outliers mais cela reste acceptable. Nos résultats sont ici plutôt satisfaisants.

Nous allons à présent analyser la distribution des résidus standardisés.

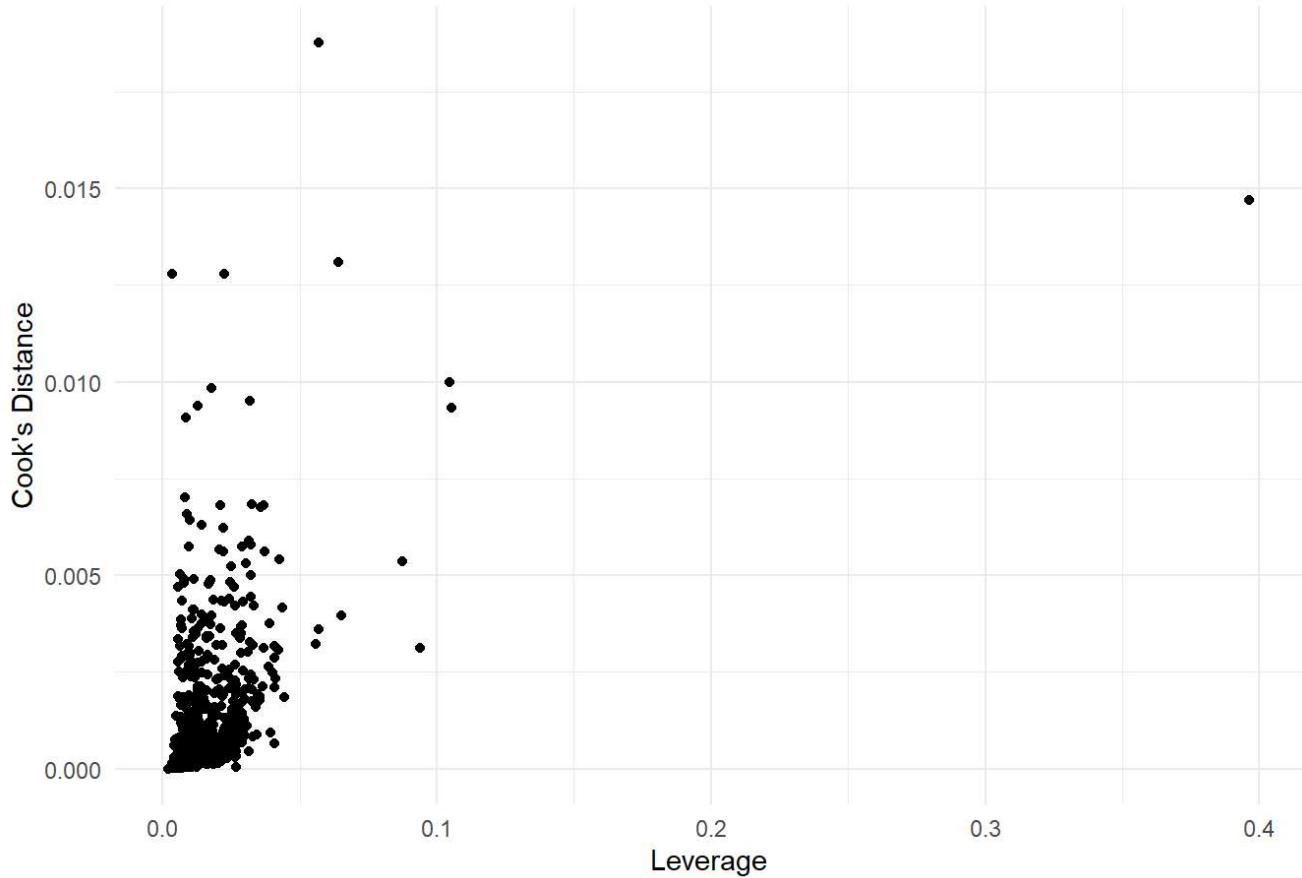
```
# QQ-Plot des résidus standardisés  
qqnorm(df_mod3$residuals_pearson, main = "QQ-Plot of Pearson Residuals")  
qqline(df_mod3$residuals_pearson, col = "red")
```



Nos résidus standardisés semblent être normalement distribués. Nous observons tout de même un outlier. Nous allons analyser la statistique de Cook pour observer si certains outliers sont problématiques.

```
# Leverage (Hat values)  
leverage <- hatvalues(mod3)  
  
# Cook's distance  
cooks_dist <- cooks.distance(mod3)  
  
# Plot du Leverage vs Cook's Distance  
ggplot(data.frame(Leverage = leverage, CooksDistance = cooks_dist), aes(x = Leverage, y = CooksDistance)) +  
  geom_point() +  
  labs(title = "Leverage vs Cook's Distance", x = "Leverage", y = "Cook's Distance") +  
  theme_minimal()
```

Leverage vs Cook's Distance



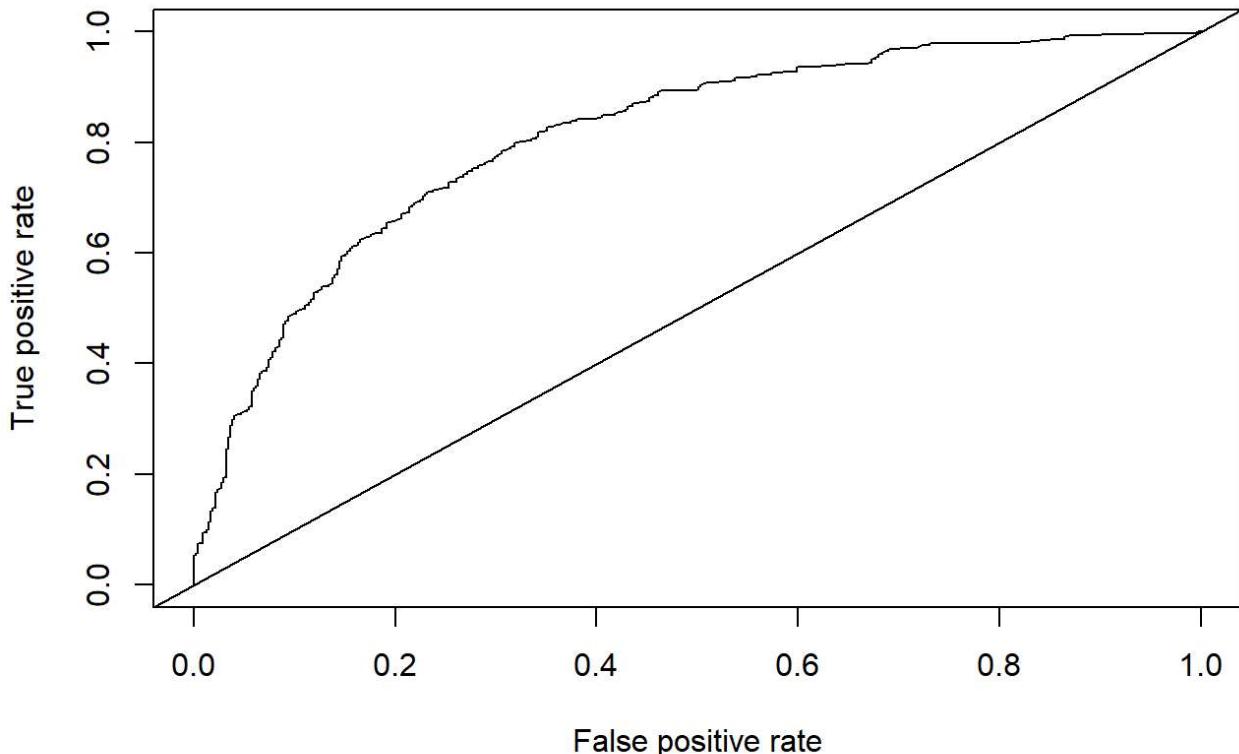
Toutes les statistiques de Cook ici sont inférieures à 0.02, ce qui est acceptable.

Analyse de la performance du modèle

Nous allons à présent analyser la courbe ROC du modèle afin de voir si notre modèle fait mieux que le hasard.

```
# Création et plot de la courbe ROC
p = prediction(df_mod3$y_pred, df_mod3$y)
plot(performance(p,"tpr","fpr"))

# Plot de la droite représentant le modèle aléatoire
abline(0,1)
```



Les prédictions que nous pourrions obtenir avec un modèle aléatoire sont représentées par la droite. Nous voyons que l'aire entre la courbe ROC et la droite, représentant l'AUC, est assez importante : notre modèle fait mieux que le modèle aléatoire.

```
## [1] "AUC du modèle: 0.81"
```

On obtient un AUC de 0.81, notre modèle prédit donc plutôt bien.

Choix du seuil et mesure de l'efficacité de prédiction

Le choix du seuil permet de définir la classe que nous allons attribuer à chaque observation en fonction de la probabilité prédite. Ici nous voulons pénaliser de la même manière les bonnes et les mauvaises prédictions.

Nous choisissons un seuil neutre à 0.5.

```

# Créer une colonne pred prenant la valeur 1 et 0 en fonction du seuil de probabilité de 0.5
df_mod3$pred <- df_mod3$y_pred > 0.5

# Créer une colonne error prenant la valeur 1 en cas d'erreur de prédiction, sinon 0
df_mod3$error <- df_mod3$y != df_mod3$pred

# Créer un tableau de contingence entre la classe réelle et la classe prédictive des observations
contingency_table <- table(df_mod3$y, df_mod3$pred)
contingency_table <- as.data.frame.matrix(contingency_table)

# Renommer les lignes et les colonnes
rownames(contingency_table) <- c("Real class 0", "Real class 1")
colnames(contingency_table) <- c("Predict class 0", "Predict class 1")

print(contingency_table)

```

```

##           Predict class 0 Predict class 1
## Real class 0          338            133
## Real class 1          113            356

```

```

## [1] "Part de bonnes prédictions : 74 %"

```

Nous observons :

- 356 vrais positifs
- 133 faux positifs
- 338 vrais négatifs
- 113 faux négatifs

Soit un total de 74% de bonne prédictions.

Nous allons à présent tester l'efficacité du modèle sur le groupe d'évaluation, afin de mesurer la capacité du modèle à se généraliser.

```

df_val <- data.frame(
  y = d_val$pluie.demain,
  y_pred = predict(mod3, d_val, type = "response")
)

# Créer une colonne pred prenant la valeur 1 et 0 en fonction du seuil de probabilité de 0.5
df_val$pred <- df_val$y_pred > 0.5

# Créer une colonne error prenant la valeur 1 en cas d'erreur de prédiction, sinon 0
df_val$error <- (df_val$y != df_val$pred) + 0

# Créer un tableau de contingence entre la classe réelle et la classe prédictive des observations
contingency_table <- table(df_val$y, df_val$pred)
contingency_table <- as.data.frame.matrix(contingency_table)

# Renommer les lignes et les colonnes
rownames(contingency_table) <- c("Real class 0", "Real class 1")
colnames(contingency_table) <- c("Predict class 0", "Predict class 1")

print(contingency_table)

```

```
##           Predict class 0 Predict class 1
## Real class 0            83             25
## Real class 1            38             94
```

```
## [1] "Part de bonnes prédictions : 74 %"
```

Nous observons :

- 94 vrais positifs
- 25 faux positifs
- 83 vrais négatifs
- 38 faux négatifs

Soit un total de 74% de bonne prédictions. Notre modèle se généralise donc correctement.

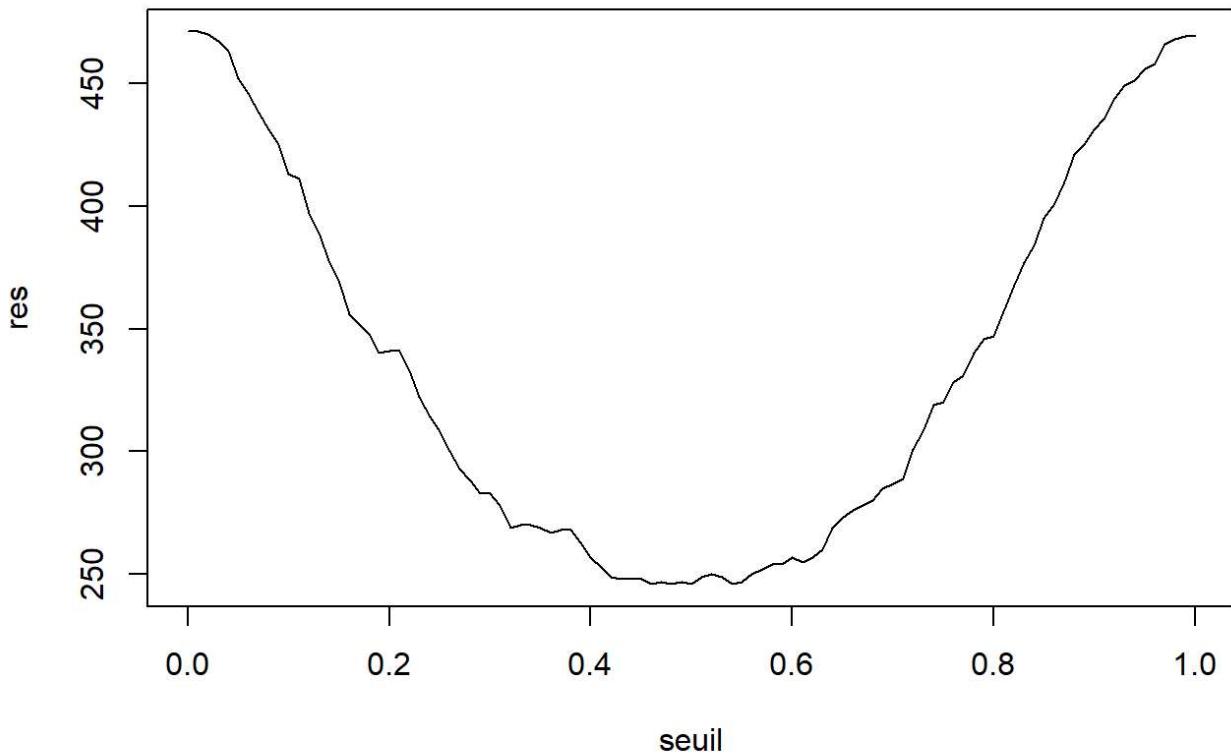
Nous allons chercher si un autre seuil ne permettrait pas de maximiser la performance du modèle sur le dataset d'entraînement.

```
# Créer une liste de seuils à tester
seuil <- seq(0,1,by=0.01)

# Créer un object pour stocker les valeurs des résidus
res <- rep(NA, length(seuil))

# Créer une boucle pour calculer des résidus pour chaque seuil
for (i in 1:length(seuil)){
  pred2 = ((df_mod3$y_pred >= seuil[i]) + 0)
  res[i] = sum(pred2 != df_mod3$y)
}

plot(seuil, res, type="l")
```



```
# Afficher la valeur de seuil qui minimise les résidus
seuil[which.min(res)]
```

```
## [1] 0.46
```

Nous observons que le minima n'est pas très net, avec des valeurs de résidus assez similaires pour des seuils autour de 0.5. Nous allons tout de même tester la performance du modèle en appliquant le seuil de 0.46.

```
##          Predict class 0 Predict class 1
## Real class 0            73             35
## Real class 1            34             98
```

```
## [1] "Part de bonnes prédictions : 71 %"
```

Nos obtenons de moins bons résultats avec ce seuil. Nous allons donc conserver le seuil à 0.5.

Prediction

Nous allons à présent effectuer des prédictions sur le dataset test. Nous avons préalablement encodé la variable Month de la même manière que sur le dataset train.

Nous allons récupérer les prédictions ainsi que les intervalles de confiance à 95% des prédictions.

```

# Prédiction sur les données du dataset test
test_prediction <- predict(mod3,dtest, type = "response", se.fit = T)

# Intervalle de confiance à 95% des prédictions
pred_proba <- data.frame(
  y_pred = test_prediction$fit,
  lower_ci = test_prediction$fit - 1.96 * test_prediction$se.fit,
  upper_ci = test_prediction$fit + 1.96 * test_prediction$se.fit
)

```

Nous choisissons le même seuil neutre de 0.5. Nous allons analyser le nombre de prédictions par classe.

```
pred_booleen <- data.frame(pred_proba > 0.5)
```

```
## [1] "Nombre de prédictions 'pluie demain' : 150"
```

```
## [1] "Nombre de prédictions 'pas de pluie demain' : 140"
```

Nous allons analyser les cas de prédiction dans lesquels nous sommes dans l'intervalle de confiance à 95%.

```

# Calcul de la part des intervalles de confiance ne chauvauchant pas le seuil de probabilité 0.5
pred_ci <- sum(pred_booleen$y_pred == pred_booleen$lower_ci & pred_booleen$y_pred == pred_booleen$upper_ci)
total_pred <- nrow(pred_booleen)
percent_pred_ci <- pred_ci/total_pred

## [1] "Part des prédictions dans l'intervalle de confiance à 95% : 72 %"

```

Dans le cas d'un seuil à 0.5, nous pouvons dire que les prédictions du modèle sont dans un intervalle de confiance de 95% pour 72% des observations. Pour le reste des observations, soit 28%, nous ne sommes pas dans l'intervalle de confiance de 95%.

Export des résultats

```

# Conversion du booleen en variable binaire
final_pred <- pred_booleen$y_pred + 0

# Export des prédictions au format csv
write.csv(final_pred,"glm_project_pred.csv")

```