Rapport Modèle Linéaire Généralisé

Olivier Berthier

2024-06-30

L'objet de ce rapport consiste à étudier un jeu de données d'entraînement de données météorologiques (de Bâle, en Suisse) et de faire des prédictions (classification) sur un jeu de test.

```
library(readr)
library (ggplot2)
library(ggcorrplot)
library(dplyr)
library(pROC)
library(tidyr)
library(boot)
library(caret)
library(formattable)
library(rstanarm)
library(bayesplot)
library(car)
```

Importation et exploration des données

Importation du jeu de données d'entraînement dans le dataframe **d_train**. (Nous récapitulons en fin de devoir tous les noms et caractéristiques des dataframes et modèles utilisé dans le rapport, (ici).

```
d_train <- read_csv("C:/Users/olivi/OneDrive - Université Paris-Dauphine/Documents ODD gram/Formation/C
    col_types = cols(...1 = col_skip(), Year = col_skip(),
        Month = col_skip(), Day = col_skip(),
        Hour = col_skip(), Minute = col_skip()))
```

Nous n'affichons pas un **summary** du jeu de données ici à cause du nombre important de variables et de la place qu'occuperait la sortie.
Les données comportent **1180** observations qui correspondent à autant de jours du 2 juin 2010 au 18 juin 2018.
Les données couvrent un jour sur deux dans cet intervalle (moins les 290 jours supprimés pour notre jeu de test).
D'après le cahier des charges de l'exercice, l'objectif est de construire un modèle de classification permettant de prévoir s'il pleut le jour suivant en utilisant uniquement les données météo du jour. Nous écartons donc de nos variables explicatives la numérotation (**...1**), et les variables de date et d'heure (**Year**, **Month**, **Day**, **Hour** et **Minute**).
Il reste 40 variables explicatives de type météorologique (humidité, couverture nuageuse, vent, température, etc.).
Toutes les variables sont de type numérique. La valeur cible, **"pluie.demain"**, est de type binaire (logique).
Le nombre de jours de pluie est supérieur au nombre de jours sans pluie, 601 avec pluie et 579 sans, soit un ratio d'environ **0.51%** de jour avec pluie.

Recherche données manquantes:

```
missing_values <- colSums(is.na(d_train))
missing_columns <- missing_values[missing_values > 0]
print(names(missing_columns))
```

```
## character(0)
```

Aucune données manquantes.

Modèle naïf

Commençons par un model incluant toutes les variables (**model_total**):

```
model_total <- glm(pluie.demain ~ ., data = d_train, family = binomial)
summary(model_total)
```

```
##
## Call:
## glm(formula = pluie.demain ~ ., family = binomial, data = d_train)
##
## Coefficients:
##                                             Estimate Std. Error z value
## (Intercept)                                6.227e+01  1.235e+01   5.041
## Temperature.daily.mean..2.m.above.gnd.      1.547e-01  1.630e-01   0.949
## Relative.Humidity.daily.mean..2.m.above.gnd. 1.767e-02  3.217e-02   0.549
## Mean.Sea.Level.Pressure.daily.mean..MSL.    5.136e-01  1.400e-01   3.669
## Total.Precipitation.daily.sum..sfc.         3.008e-02  2.804e-02   1.073
## Snowfall.amount.raw.daily.sum..sfc.        -3.303e-01  2.442e-01  -1.353
## Total.Cloud.Cover.daily.mean..sfc.          1.173e-02  1.194e-02   0.982
## High.Cloud.Cover.daily.mean..high.cld.lay. -3.609e-03  6.805e-03  -0.530
## Medium.Cloud.Cover.daily.mean..mid.cld.lay. 6.278e-03  6.665e-03   0.942
## Low.Cloud.Cover.daily.mean..low.cld.lay.   -3.101e-03  8.050e-03  -0.385
## Sunshine.Duration.daily.sum..sfc.           4.502e-04  8.751e-04   0.514
## Shortwave.Radiation.daily.sum..sfc.         3.870e-05  9.362e-05   0.413
## Wind.Speed.daily.mean..10.m.above.gnd.     -5.819e-02  9.643e-02  -0.603
## Wind.Direction.daily.mean..10.m.above.gnd.  4.522e-03  5.693e-03   0.794
## Wind.Speed.daily.mean..80.m.above.gnd.     -9.080e-02  6.924e-02  -1.311
## Wind.Direction.daily.mean..80.m.above.gnd. -8.113e-03  5.884e-03  -1.379
## Wind.Speed.daily.mean..900.mb.              1.544e-02  2.586e-02   0.597
## Wind.Direction.daily.mean..900.mb.          5.302e-03  1.445e-03   3.668
## Wind.Gust.daily.mean..sfc.                  2.628e-02  3.661e-02   0.718
## Temperature.daily.max..2.m.above.gnd.      -1.669e-04  9.565e-02  -0.002
## Temperature.daily.min..2.m.above.gnd.      -1.203e-01  8.574e-02  -1.403
## Relative.Humidity.daily.max..2.m.above.gnd. -1.886e-03  2.043e-02  -0.092
## Relative.Humidity.daily.min..2.m.above.gnd. -1.013e-02  1.832e-02  -0.553
## Mean.Sea.Level.Pressure.daily.max..MSL.    -2.564e-01  7.522e-02  -3.409
## Mean.Sea.Level.Pressure.daily.min..MSL.    -3.230e-01  7.608e-02  -4.246
## Total.Cloud.Cover.daily.max..sfc.           3.835e-03  4.835e-03   0.793
## Total.Cloud.Cover.daily.min..sfc.           7.665e-03  6.291e-03   1.218
## High.Cloud.Cover.daily.max..high.cld.lay.   3.444e-03  2.884e-03   1.194
## High.Cloud.Cover.daily.min..high.cld.lay.   9.175e-03  2.094e-02   0.438
```

```
## Medium.Cloud.Cover.daily.max..mid.cld.lay.    6.131e-03  3.150e-03   1.947
## Medium.Cloud.Cover.daily.min..mid.cld.lay.   -5.969e-03  9.323e-03  -0.640
## Low.Cloud.Cover.daily.max..low.cld.lay.       2.413e-03  3.369e-03   0.716
## Low.Cloud.Cover.daily.min..low.cld.lay.       1.598e-04  7.023e-03   0.023
## Wind.Speed.daily.max..10.m.above.gnd.         6.017e-02  3.441e-02   1.749
## Wind.Speed.daily.min..10.m.above.gnd.         1.703e-01  6.378e-02   2.669
## Wind.Speed.daily.max..80.m.above.gnd.         1.004e-02  2.821e-02   0.356
## Wind.Speed.daily.min..80.m.above.gnd.        -6.240e-02  4.183e-02  -1.492
## Wind.Speed.daily.max..900.mb.                -1.175e-02  1.210e-02  -0.971
## Wind.Speed.daily.min..900.mb.                -5.781e-03  1.896e-02  -0.305
## Wind.Gust.daily.max..sfc.                     1.197e-02  1.665e-02   0.719
## Wind.Gust.daily.min..sfc.                     1.610e-02  2.739e-02   0.588
##                                              Pr(>|z|)
## (Intercept)                                  4.62e-07 ***
## Temperature.daily.mean..2.m.above.gnd.       0.342547
## Relative.Humidity.daily.mean..2.m.above.gnd. 0.582816
## Mean.Sea.Level.Pressure.daily.mean..MSL.     0.000244 ***
## Total.Precipitation.daily.sum..sfc.          0.283294
## Snowfall.amount.raw.daily.sum..sfc.          0.176155
## Total.Cloud.Cover.daily.mean..sfc.           0.325982
## High.Cloud.Cover.daily.mean..high.cld.lay.   0.595914
## Medium.Cloud.Cover.daily.mean..mid.cld.lay.  0.346188
## Low.Cloud.Cover.daily.mean..low.cld.lay.     0.700106
## Sunshine.Duration.daily.sum..sfc.            0.606960
## Shortwave.Radiation.daily.sum..sfc.          0.679326
## Wind.Speed.daily.mean..10.m.above.gnd.       0.546180
## Wind.Direction.daily.mean..10.m.above.gnd.   0.427025
## Wind.Speed.daily.mean..80.m.above.gnd.       0.189701
## Wind.Direction.daily.mean..80.m.above.gnd.   0.167897
## Wind.Speed.daily.mean..900.mb.               0.550472
## Wind.Direction.daily.mean..900.mb.           0.000244 ***
## Wind.Gust.daily.mean..sfc.                   0.472907
## Temperature.daily.max..2.m.above.gnd.        0.998607
## Temperature.daily.min..2.m.above.gnd.        0.160666
## Relative.Humidity.daily.max..2.m.above.gnd.  0.926455
## Relative.Humidity.daily.min..2.m.above.gnd.  0.580373
## Mean.Sea.Level.Pressure.daily.max..MSL.      0.000653 ***
## Mean.Sea.Level.Pressure.daily.min..MSL.      2.18e-05 ***
## Total.Cloud.Cover.daily.max..sfc.            0.427620
## Total.Cloud.Cover.daily.min..sfc.            0.223101
## High.Cloud.Cover.daily.max..high.cld.lay.    0.232445
## High.Cloud.Cover.daily.min..high.cld.lay.    0.661319
## Medium.Cloud.Cover.daily.max..mid.cld.lay.   0.051568 .
## Medium.Cloud.Cover.daily.min..mid.cld.lay.   0.522034
## Low.Cloud.Cover.daily.max..low.cld.lay.      0.473915
## Low.Cloud.Cover.daily.min..low.cld.lay.      0.981841
## Wind.Speed.daily.max..10.m.above.gnd.        0.080300 .
## Wind.Speed.daily.min..10.m.above.gnd.        0.007597 **
## Wind.Speed.daily.max..80.m.above.gnd.        0.721897
## Wind.Speed.daily.min..80.m.above.gnd.        0.135819
## Wind.Speed.daily.max..900.mb.                0.331630
## Wind.Speed.daily.min..900.mb.                0.760490
## Wind.Gust.daily.max..sfc.                    0.472148
## Wind.Gust.daily.min..sfc.                    0.556642
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1635.4  on 1179  degrees of freedom
## Residual deviance: 1239.5  on 1139  degrees of freedom
## AIC: 1321.5
##
## Number of Fisher Scoring iterations: 4
```

Nous constatons que beaucoup de variables ont des p-valeurs >5%. Ce modèle pourrait être largement amélioré. Nous le conservons cependant tel quel afin de comparer les performances d'un modèle "brut" avec des modèles plus sophistiqués.

Évaluation de **model_total** par matrice de confusion:

```
pred_prob <- predict(model_total, newdata = d_train, type = "response")
pred <- ifelse(pred_prob >= 0.5, "TRUE", "FALSE") # seuil de proba de 0.5
pred <- as.factor(pred)
pluie.demain <- as.factor(d_train$pluie.demain)
# matrice de confusion
conf_matrix <- confusionMatrix(pred, pluie.demain, positive ="TRUE")
print(conf_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE   412  140
##      TRUE    167  461
##
##                Accuracy : 0.7398
##                  95% CI : (0.7138, 0.7647)
##     No Information Rate : 0.5093
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.479
##
##  Mcnemar's Test P-Value : 0.1378
##
##             Sensitivity : 0.7671
##             Specificity : 0.7116
##          Pos Pred Value : 0.7341
##          Neg Pred Value : 0.7464
##              Prevalence : 0.5093
##          Detection Rate : 0.3907
##    Detection Prevalence : 0.5322
##       Balanced Accuracy : 0.7393
##
##        'Positive' Class : TRUE
##
```

Le modèle total prédit la bonne réponse dans 0.7398% das cas, c'est la précision (Accuracy): (vrai positif + vrai négatif) / total). Pour avoir un ordre de grandeur, il prédit 873 bonnes réponses contre 307 erreurs.

La sensibilité (Sensitivity) représente la proportion de jours pluvieux réels (cas positifs) que le modèle a correctement prédits: VP / (VP + FN). Le modèle prédit correctement un jour pluvieux dans 0.7671% des cas.
(Note: cette mesure est particulièrement importante si la prédiction de la pluie est cruciale, par exemple pour éviter les inondations).
La spécificité (0,7116%) représente la proportion de jours sans pluie réels (cas négatifs) que le modèle a correctement prédits (VN / (VN + FP)).

Nous évaluons maintenant notre modèle total par cross-validation (10 plis) en calculant la valeur moyenne des erreurs de prédictions.
Le modèle est entraîné 10 fois. À chaque itération, un pli différent est utilisé comme ensemble de test et les 9 plis restants comme ensemble d'entraînement.
Cette méthode est intéressante dans notre étude car nous ne connaissons pas les valeurs de la variable cible de notre jeu de test, et ne pouvons donc tester notre modèle sur ce jeu de données. De plus, elle permet d'utiliser l'ensemble des valeurs du jeu d'entraînement.
Prendre K=10 offre un bon compromis entre biais et variance. Un K plus petit peut introduire un biais plus élevé, et un K plus grand peut augmenter la variance et le coût computationnel (un grand K réduit également les risques de surapprentissage).
(ici et pour la suite nous utilisons une "graine" afin d'obtenir des résultats "stables" qui faliciteront les comparaisons.)

```
set.seed(100)      # Fixer la graine pour la reproductibilité
cv_model <- cv.glm(d_train, model_total, K = 10)  # K = nombre de plis pour la validation croisée
delta_rounded <- round(cv_model$delta, 3)  # Arrondir les valeurs à trois chiffres après la virgule
print(delta_rounded)  # Afficher les valeurs
```

```
## [1] 0.190 0.189
```

La sortie donne deux valeurs, la première est l'estimation brute de l'erreur, la deuxième est l'estimation corrigée du biais introduit en n'utilisant pas la méthode "leave one out" (où K=n).
Nous retenons ici et pour la suite la valeur corrigée, soit 0.189% d'erreur de prédiction.

Calculons l'AUC de **model_total**.
l'AUC (Area Under the Curve), évalue la performance du modèle en calculant l'aire sous la courbe ROC (Receiver Operating Characteristic). Nous reviendrons sur la courbe ROC plus loin.

```
pred_prob <- predict(model_total, newdata = d_train, type = "response") # Prédictions de probabilités s
roc_model_total <- roc(d_train$pluie.demain, pred_prob)  # Calculer le ROC
auc_model_total <- auc(roc_model_total)   # Calculer l'AUC
print(auc_model_total)
```

```
## Area under the curve: 0.8176
```

Sachant que **0** correspond à un modèle qui prédit toujours incorrectement et **1** correspond à un modèle qui prédit toujours correctement, l'AUC de **model_total** semble assez élevé. Ces mesures de performances vont surtout nous servir ici à comparer les différents modèles entre-eux.

Élaboration automatique via la fonction step()

Selon le critère AIC

Nous utilisons la fonction **step()** dans le but de réduire automatiquement le nombre de variables dans le modèle et éviter le surapprentissage et ainsi améliorer la généralisation sur de nouvelles données. Nous

utilisons en pemière approche le critère de sélection AIC (Akaike Information Criterion) qui vise à trouver un équilibre entre la précision de la modélisation et la complexité du modèle.

```
model_step_AIC <- step(glm(pluie.demain ~ ., data = d_train, family = binomial), direction = "both", tra
summary (model_step_AIC)
```

```
##
## Call:
## glm(formula = pluie.demain ~ Temperature.daily.mean..2.m.above.gnd. +
##     Mean.Sea.Level.Pressure.daily.mean..MSL. + Snowfall.amount.raw.daily.sum..sfc. +
##     Total.Cloud.Cover.daily.mean..sfc. + Wind.Speed.daily.mean..80.m.above.gnd. +
##     Wind.Direction.daily.mean..80.m.above.gnd. + Wind.Direction.daily.mean..900.mb. +
##     Wind.Gust.daily.mean..sfc. + Temperature.daily.min..2.m.above.gnd. +
##     Mean.Sea.Level.Pressure.daily.max..MSL. + Mean.Sea.Level.Pressure.daily.min..MSL. +
##     Total.Cloud.Cover.daily.min..sfc. + High.Cloud.Cover.daily.max..high.cld.lay. +
##     Medium.Cloud.Cover.daily.max..mid.cld.lay. + Wind.Speed.daily.max..10.m.above.gnd. +
##     Wind.Speed.daily.min..10.m.above.gnd. + Wind.Speed.daily.min..80.m.above.gnd.,
##     family = binomial, data = d_train)
##
## Coefficients:
##                                          Estimate Std. Error z value
## (Intercept)                             67.956250  11.550841   5.883
## Temperature.daily.mean..2.m.above.gnd.      0.161140   0.053523   3.011
## Mean.Sea.Level.Pressure.daily.mean..MSL.    0.494376   0.131650   3.755
## Snowfall.amount.raw.daily.sum..sfc.        -0.365181   0.225588  -1.619
## Total.Cloud.Cover.daily.mean..sfc.          0.011696   0.003960   2.953
## Wind.Speed.daily.mean..80.m.above.gnd.     -0.114252   0.035771  -3.194
## Wind.Direction.daily.mean..80.m.above.gnd. -0.003040   0.001539  -1.976
## Wind.Direction.daily.mean..900.mb.          0.004453   0.001285   3.467
## Wind.Gust.daily.mean..sfc.                  0.036115   0.019443   1.857
## Temperature.daily.min..2.m.above.gnd.      -0.117107   0.056616  -2.068
## Mean.Sea.Level.Pressure.daily.max..MSL.    -0.248507   0.070969  -3.502
## Mean.Sea.Level.Pressure.daily.min..MSL.    -0.316098   0.071793  -4.403
## Total.Cloud.Cover.daily.min..sfc.           0.005741   0.003858   1.488
## High.Cloud.Cover.daily.max..high.cld.lay.   0.003200   0.002148   1.490
## Medium.Cloud.Cover.daily.max..mid.cld.lay.  0.008195   0.002522   3.249
## Wind.Speed.daily.max..10.m.above.gnd.       0.069313   0.021606   3.208
## Wind.Speed.daily.min..10.m.above.gnd.       0.156011   0.054642   2.855
## Wind.Speed.daily.min..80.m.above.gnd.      -0.056769   0.038577  -1.472
##                                          Pr(>|z|)
## (Intercept)                              4.02e-09 ***
## Temperature.daily.mean..2.m.above.gnd.    0.002607 **
## Mean.Sea.Level.Pressure.daily.mean..MSL.  0.000173 ***
## Snowfall.amount.raw.daily.sum..sfc.       0.105491
## Total.Cloud.Cover.daily.mean..sfc.        0.003143 **
## Wind.Speed.daily.mean..80.m.above.gnd.    0.001403 **
## Wind.Direction.daily.mean..80.m.above.gnd. 0.048132 *
## Wind.Direction.daily.mean..900.mb.        0.000526 ***
## Wind.Gust.daily.mean..sfc.                0.063248 .
## Temperature.daily.min..2.m.above.gnd.     0.038597 *
## Mean.Sea.Level.Pressure.daily.max..MSL.   0.000462 ***
## Mean.Sea.Level.Pressure.daily.min..MSL.   1.07e-05 ***
## Total.Cloud.Cover.daily.min..sfc.         0.136716
## High.Cloud.Cover.daily.max..high.cld.lay. 0.136188
```

```
## Medium.Cloud.Cover.daily.max..mid.cld.lay. 0.001158 **
## Wind.Speed.daily.max..10.m.above.gnd.      0.001336 **
## Wind.Speed.daily.min..10.m.above.gnd.      0.004302 **
## Wind.Speed.daily.min..80.m.above.gnd.      0.141136
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1635.4  on 1179  degrees of freedom
## Residual deviance: 1249.6  on 1162  degrees of freedom
## AIC: 1285.6
##
## Number of Fisher Scoring iterations: 4
```

Le modèle **model_step_AIC** retient 17 variables (23 ont été éliminées). Notons que certaines variables présentes affichent des p-vlaeurs >5% et que le modèle pourrait sans doute être amélioré.
Nous le conservons tel quel afin de pouvoir de pouvoir comparer les différentes méthodes.
Sans surprise, car c'est le critère de sélection, son AIC (1285.6) est meilleur que celui du modèle total (1321.5).
Remarque: nous avons aussi essayer (toujours selon le critère AIC) la fonction **train()** du package **caret** qui utilise la cross-validation pour optimiser son modèle. Le modèle proposé était identique, les même variables retenues et les mêmes coefficients.

Évaluation de **model_step_AIC** par matrice de confusion:
(Pour ne pas surcharger le rapport, nous ne faisons pas apparaître les codes qui sont redondants, ici seul le nom du modèle diffère par rapport au calcul de matrice de confusion précedent.)

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE   409  134
##      TRUE    170  467
##
##                Accuracy : 0.7424
##                  95% CI : (0.7164, 0.7671)
##     No Information Rate : 0.5093
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.484
##
##  Mcnemar's Test P-Value : 0.04471
##
##             Sensitivity : 0.7770
##             Specificity : 0.7064
##          Pos Pred Value : 0.7331
##          Neg Pred Value : 0.7532
##              Prevalence : 0.5093
##          Detection Rate : 0.3958
##    Detection Prevalence : 0.5398
##       Balanced Accuracy : 0.7417
##
##        'Positive' Class : TRUE
##
```

876 bonnes réponses.

Évaluation de **model_step_AIC** par cross-validation (10 plis):

```
## [1] 0.182 0.182
```

Calculons l'AUC de **model_step_AIC**:

```
## Area under the curve: 0.8138
```

le **model_step_AIC** fait un peu mieux que le modèle total pour 4 critères sur 6 (il est moins bon pour la spécificité et l'AUC mais les différences sont négligeables).

Selon le critère BIC

Modèle avec sélection BIC (Bayesian Information Criterion), obtenu avec k = log(n).
Ce critère est plus parcimonieux que l'AIC. On considère généralement qu'il facilite l'interprétation (moins de variables) mais peut être moins bon en prédiction (perte d'information possible par élimination des variables).

```
model_step_BIC <- step(glm(pluie.demain ~ . , data = d_train, family = binomial), direction = "both", t
summary (model_step_BIC)
```

```
##
## Call:
## glm(formula = pluie.demain ~ Temperature.daily.mean..2.m.above.gnd. +
##     Mean.Sea.Level.Pressure.daily.mean..MSL. + Total.Cloud.Cover.daily.mean..sfc. +
##     Wind.Speed.daily.mean..80.m.above.gnd. + Wind.Direction.daily.mean..900.mb. +
##     Temperature.daily.min..2.m.above.gnd. + Mean.Sea.Level.Pressure.daily.max..MSL. +
##     Mean.Sea.Level.Pressure.daily.min..MSL. + Medium.Cloud.Cover.daily.max..mid.cld.lay. +
##     Wind.Speed.daily.max..10.m.above.gnd. + Wind.Speed.daily.min..10.m.above.gnd.,
##     family = binomial, data = d_train)
##
## Coefficients:
##                                          Estimate Std. Error z value
## (Intercept)                             67.673828  11.426292   5.923
## Temperature.daily.mean..2.m.above.gnd.      0.199770   0.052759   3.786
## Mean.Sea.Level.Pressure.daily.mean..MSL.    0.498667   0.130184   3.830
## Total.Cloud.Cover.daily.mean..sfc.          0.013794   0.003482   3.962
## Wind.Speed.daily.mean..80.m.above.gnd.     -0.091661   0.028551  -3.210
## Wind.Direction.daily.mean..900.mb.          0.003237   0.001062   3.049
## Temperature.daily.min..2.m.above.gnd.      -0.153529   0.055681  -2.757
## Mean.Sea.Level.Pressure.daily.max..MSL.    -0.256665   0.069129  -3.713
## Mean.Sea.Level.Pressure.daily.min..MSL.    -0.312440   0.071951  -4.342
## Medium.Cloud.Cover.daily.max..mid.cld.lay.  0.010433   0.002109   4.946
## Wind.Speed.daily.max..10.m.above.gnd.       0.076553   0.019595   3.907
## Wind.Speed.daily.min..10.m.above.gnd.       0.097036   0.034757   2.792
##                                          Pr(>|z|)
## (Intercept)                             3.17e-09 ***
## Temperature.daily.mean..2.m.above.gnd.     0.000153 ***
## Mean.Sea.Level.Pressure.daily.mean..MSL.   0.000128 ***
## Total.Cloud.Cover.daily.mean..sfc.         7.45e-05 ***
```

```
## Wind.Speed.daily.mean..80.m.above.gnd.     0.001326 **
## Wind.Direction.daily.mean..900.mb.         0.002294 **
## Temperature.daily.min..2.m.above.gnd.      0.005828 **
## Mean.Sea.Level.Pressure.daily.max..MSL.    0.000205 ***
## Mean.Sea.Level.Pressure.daily.min..MSL.    1.41e-05 ***
## Medium.Cloud.Cover.daily.max..mid.cld.lay. 7.56e-07 ***
## Wind.Speed.daily.max..10.m.above.gnd.      9.35e-05 ***
## Wind.Speed.daily.min..10.m.above.gnd.      0.005241 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1635.4  on 1179  degrees of freedom
## Residual deviance: 1268.1  on 1168  degrees of freedom
## AIC: 1292.1
##
## Number of Fisher Scoring iterations: 4
```

Sans surprise, son AIC est moins bon que celui de **model_step_AIC**.
Seulement 11 variables sont retenues, toutes avec une p-valeur <0.01%.
9 d'entre-elles étaient dèjà présentes dans **model_step_BIC**.

Les variables communes au deux modèles sont:

- Temperature.daily.mean..2.m.above.gnd.
- Mean.Sea.Level.Pressure.daily.mean..MSL.
- Temperature.daily.min..2.m.above.gnd.
- Mean.Sea.Level.Pressure.daily.max..MSL.
- Mean.Sea.Level.Pressure.daily.min..MSL.
- Wind.Speed.daily.max..10.m.above.gnd.
- Wind.Speed.daily.min..10.m.above.gnd.
- Wind.Direction.daily.mean..900.mb.
- Total.Cloud.Cover.daily.mean..sfc

Évaluation de **model_step_BIC** par matrice de confusion:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE   405  127
##      TRUE    174  474
##
##                Accuracy : 0.7449
##                  95% CI : (0.719, 0.7696)
```

```
##     No Information Rate : 0.5093
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4889
##
##  Mcnemar's Test P-Value : 0.008016
##
##             Sensitivity : 0.7887
##             Specificity : 0.6995
##          Pos Pred Value : 0.7315
##          Neg Pred Value : 0.7613
##              Prevalence : 0.5093
##          Detection Rate : 0.4017
##    Detection Prevalence : 0.5492
##       Balanced Accuracy : 0.7441
##
##        'Positive' Class : TRUE
##
```

879 bonnes réponses.

Évaluation de **model_step_BIC** par cross-validation (10 plis):

```
## [1] 0.183 0.182
```

Calculons l'AUC de **model_step_BIC**:

```
## Setting levels: control = FALSE, case = TRUE
```

```
## Setting direction: controls < cases
```

```
## Area under the curve: 0.8082
```

Les performances de **model_step_BIC** sont très proches des deux autres (notons qu'il a la meilleure précision des trois). Il prédit 4 faux positifs de plus que **model_step_AIC** mais 7 faux négatifs de moins.

Sélection de variables par algorithme

Nous proposons ici un algorithme de sélection automatique afin de palier au problème de multicolinéarité.
Nous créons un algorithme qui élimine automatiquement les variables trop corrélées entre elles.
Nous retenons 0.8 comme seuil de de corrélation (en valeur absolue).
Ce code identifie les paires de variables dans le jeu de données au dessus de ce seuil et élimine la variable avec la plus faible corrélation absolue avec **pluie.demain**.
La boucle est répétée jusqu'à ce qu'aucune paire de variables avec une corrélation >0.8 ne soit trouvée.
Nous copions **d_train** dans un nouveau dataframe **d_train_algo** pour cet usage.

```
d_train_algo <- d_train  # nouveau dataframe pour effectuer la sélection
# Initialiser la boucle
variables_supprimees <- c()

repeat {
  # Calculer la matrice de corrélation
```

```r
  cor_matrix <- cor(d_train_algo, use = "complete.obs")

  # Identifier les paires de variables avec une corrélation supérieure à 0.8
  high_cor_pairs <- which(abs(cor_matrix) > 0.8 & abs(cor_matrix) < 1, arr.ind = TRUE)
  high_cor_pairs <- high_cor_pairs[high_cor_pairs[, 1] < high_cor_pairs[, 2], , drop = FALSE]

  # Sortir de la boucle si aucune paire trouvée
  if (is.null(high_cor_pairs) || nrow(high_cor_pairs) == 0 || all(is.na(high_cor_pairs))) break

  # Trier les paires par corrélation décroissante
  cor_values <- cor_matrix[high_cor_pairs]
  cor_pairs <- data.frame(
    var1 = rownames(cor_matrix)[high_cor_pairs[, 1]],
    var2 = colnames(cor_matrix)[high_cor_pairs[, 2]],
    cor_value = cor_values
  )
  cor_pairs <- cor_pairs[order(abs(cor_pairs$cor_value), decreasing = TRUE), ]

  # Sélectionner la paire avec la corrélation la plus forte
  strongest_pair <- cor_pairs[1, ]
  var1 <- strongest_pair$var1
  var2 <- strongest_pair$var2

  # Calculer les corrélations avec pluie.demain
  cor_var1 <- cor(d_train_algo[[var1]], d_train_algo$pluie.demain)
  cor_var2 <- cor(d_train_algo[[var2]], d_train_algo$pluie.demain)

  # Afficher les corrélation
  cat("Corrélation entre", var1, "et", var2, ":", strongest_pair$cor_value, "\n")
  cat("Corrélation de", var1, "avec pluie.demain:", cor_var1, "\n")
  cat("Corrélation de", var2, "avec pluie.demain:", cor_var2, "\n")

  if (abs(cor_var1) < abs(cor_var2)) {
    d_train_algo <- d_train_algo %>% select(-var1)
    variables_supprimees <- c(variables_supprimees, var1)
    cat("Suppression de la variable:", var1, "\n\n")
  }
  else {
    d_train_algo <- d_train_algo %>% select(-var2)
    variables_supprimees <- c(variables_supprimees, var2)
    cat("Suppression de la variable:", var2, "\n\n")
  }
}
```

```
## Corrélation entre Wind.Speed.daily.mean..10.m.above.gnd. et Wind.Speed.daily.mean..80.m.above.gnd. :
## Corrélation de Wind.Speed.daily.mean..10.m.above.gnd. avec pluie.demain: 0.2118871
## Corrélation de Wind.Speed.daily.mean..80.m.above.gnd. avec pluie.demain: 0.1970128
## Suppression de la variable: Wind.Speed.daily.mean..80.m.above.gnd.
##
## Corrélation entre Temperature.daily.mean..2.m.above.gnd. et Temperature.daily.max..2.m.above.gnd. : 0
## Corrélation de Temperature.daily.mean..2.m.above.gnd. avec pluie.demain: 0.117959
## Corrélation de Temperature.daily.max..2.m.above.gnd. avec pluie.demain: 0.08931491
## Suppression de la variable: Temperature.daily.max..2.m.above.gnd.
```

```
##
## Corrélation entre Wind.Direction.daily.mean..10.m.above.gnd. et Wind.Direction.daily.mean..80.m.above
## Corrélation de Wind.Direction.daily.mean..10.m.above.gnd. avec pluie.demain: 0.1245994
## Corrélation de Wind.Direction.daily.mean..80.m.above.gnd. avec pluie.demain: 0.1328056
## Suppression de la variable: Wind.Direction.daily.mean..10.m.above.gnd.
##
## Corrélation entre Mean.Sea.Level.Pressure.daily.mean..MSL. et Mean.Sea.Level.Pressure.daily.min..MSL
## Corrélation de Mean.Sea.Level.Pressure.daily.mean..MSL. avec pluie.demain: -0.3721108
## Corrélation de Mean.Sea.Level.Pressure.daily.min..MSL. avec pluie.demain: -0.3873982
## Suppression de la variable: Mean.Sea.Level.Pressure.daily.mean..MSL.
##
## Corrélation entre Temperature.daily.mean..2.m.above.gnd. et Temperature.daily.min..2.m.above.gnd. : 0
## Corrélation de Temperature.daily.mean..2.m.above.gnd. avec pluie.demain: 0.117959
## Corrélation de Temperature.daily.min..2.m.above.gnd. avec pluie.demain: 0.1455173
## Suppression de la variable: Temperature.daily.mean..2.m.above.gnd.
##
## Corrélation entre Wind.Speed.daily.max..10.m.above.gnd. et Wind.Speed.daily.max..80.m.above.gnd. : 0
## Corrélation de Wind.Speed.daily.max..10.m.above.gnd. avec pluie.demain: 0.2483077
## Corrélation de Wind.Speed.daily.max..80.m.above.gnd. avec pluie.demain: 0.2444658
## Suppression de la variable: Wind.Speed.daily.max..80.m.above.gnd.
##
## Corrélation entre Wind.Speed.daily.min..10.m.above.gnd. et Wind.Speed.daily.min..80.m.above.gnd. : 0
## Corrélation de Wind.Speed.daily.min..10.m.above.gnd. avec pluie.demain: 0.1677079
## Corrélation de Wind.Speed.daily.min..80.m.above.gnd. avec pluie.demain: 0.1341332
## Suppression de la variable: Wind.Speed.daily.min..80.m.above.gnd.
##
## Corrélation entre Wind.Speed.daily.mean..10.m.above.gnd. et Wind.Gust.daily.mean..sfc. : 0.921728
## Corrélation de Wind.Speed.daily.mean..10.m.above.gnd. avec pluie.demain: 0.2118871
## Corrélation de Wind.Gust.daily.mean..sfc. avec pluie.demain: 0.2291145
## Suppression de la variable: Wind.Speed.daily.mean..10.m.above.gnd.
##
## Corrélation entre Wind.Speed.daily.mean..900.mb. et Wind.Speed.daily.max..900.mb. : 0.9168068
## Corrélation de Wind.Speed.daily.mean..900.mb. avec pluie.demain: 0.1858268
## Corrélation de Wind.Speed.daily.max..900.mb. avec pluie.demain: 0.2379114
## Suppression de la variable: Wind.Speed.daily.mean..900.mb.
##
## Corrélation entre Total.Cloud.Cover.daily.mean..sfc. et Sunshine.Duration.daily.sum..sfc. : -0.90646
## Corrélation de Total.Cloud.Cover.daily.mean..sfc. avec pluie.demain: 0.3220703
## Corrélation de Sunshine.Duration.daily.sum..sfc. avec pluie.demain: -0.2444878
## Suppression de la variable: Sunshine.Duration.daily.sum..sfc.
##
## Corrélation entre Mean.Sea.Level.Pressure.daily.max..MSL. et Mean.Sea.Level.Pressure.daily.min..MSL.
## Corrélation de Mean.Sea.Level.Pressure.daily.max..MSL. avec pluie.demain: -0.3513344
## Corrélation de Mean.Sea.Level.Pressure.daily.min..MSL. avec pluie.demain: -0.3873982
## Suppression de la variable: Mean.Sea.Level.Pressure.daily.max..MSL.
##
## Corrélation entre Total.Cloud.Cover.daily.mean..sfc. et Low.Cloud.Cover.daily.mean..low.cld.lay. : 0
## Corrélation de Total.Cloud.Cover.daily.mean..sfc. avec pluie.demain: 0.3220703
## Corrélation de Low.Cloud.Cover.daily.mean..low.cld.lay. avec pluie.demain: 0.2280604
## Suppression de la variable: Low.Cloud.Cover.daily.mean..low.cld.lay.
##
## Corrélation entre Relative.Humidity.daily.mean..2.m.above.gnd. et Relative.Humidity.daily.min..2.m.ab
## Corrélation de Relative.Humidity.daily.mean..2.m.above.gnd. avec pluie.demain: 0.02591274
## Corrélation de Relative.Humidity.daily.min..2.m.above.gnd. avec pluie.demain: 0.02989942
```

```
## Suppression de la variable: Relative.Humidity.daily.mean..2.m.above.gnd.
##
## Corrélation entre Wind.Gust.daily.mean..sfc. et Wind.Gust.daily.max..sfc. : 0.885367
## Corrélation de Wind.Gust.daily.mean..sfc. avec pluie.demain: 0.2291145
## Corrélation de Wind.Gust.daily.max..sfc. avec pluie.demain: 0.2761492
## Suppression de la variable: Wind.Gust.daily.mean..sfc.
##
## Corrélation entre Wind.Speed.daily.max..10.m.above.gnd. et Wind.Gust.daily.max..sfc. : 0.8713995
## Corrélation de Wind.Speed.daily.max..10.m.above.gnd. avec pluie.demain: 0.2483077
## Corrélation de Wind.Gust.daily.max..sfc. avec pluie.demain: 0.2761492
## Suppression de la variable: Wind.Speed.daily.max..10.m.above.gnd.
##
## Corrélation entre Wind.Speed.daily.max..900.mb. et Wind.Gust.daily.max..sfc. : 0.8412173
## Corrélation de Wind.Speed.daily.max..900.mb. avec pluie.demain: 0.2379114
## Corrélation de Wind.Gust.daily.max..sfc. avec pluie.demain: 0.2761492
## Suppression de la variable: Wind.Speed.daily.max..900.mb.
##
## Corrélation entre Wind.Speed.daily.min..10.m.above.gnd. et Wind.Gust.daily.min..sfc. : 0.8343751
## Corrélation de Wind.Speed.daily.min..10.m.above.gnd. avec pluie.demain: 0.1677079
## Corrélation de Wind.Gust.daily.min..sfc. avec pluie.demain: 0.1696535
## Suppression de la variable: Wind.Speed.daily.min..10.m.above.gnd.
```

```
cat("Nombre de variables supprimées :", length(variables_supprimees), "\n")
```

```
## Nombre de variables supprimées : 17
```

```
cat("Variables supprimées :", paste(variables_supprimees, collapse = ", "), "\n")
```

```
## Variables supprimées : Wind.Speed.daily.mean..80.m.above.gnd., Temperature.daily.max..2.m.above.gnd.
```

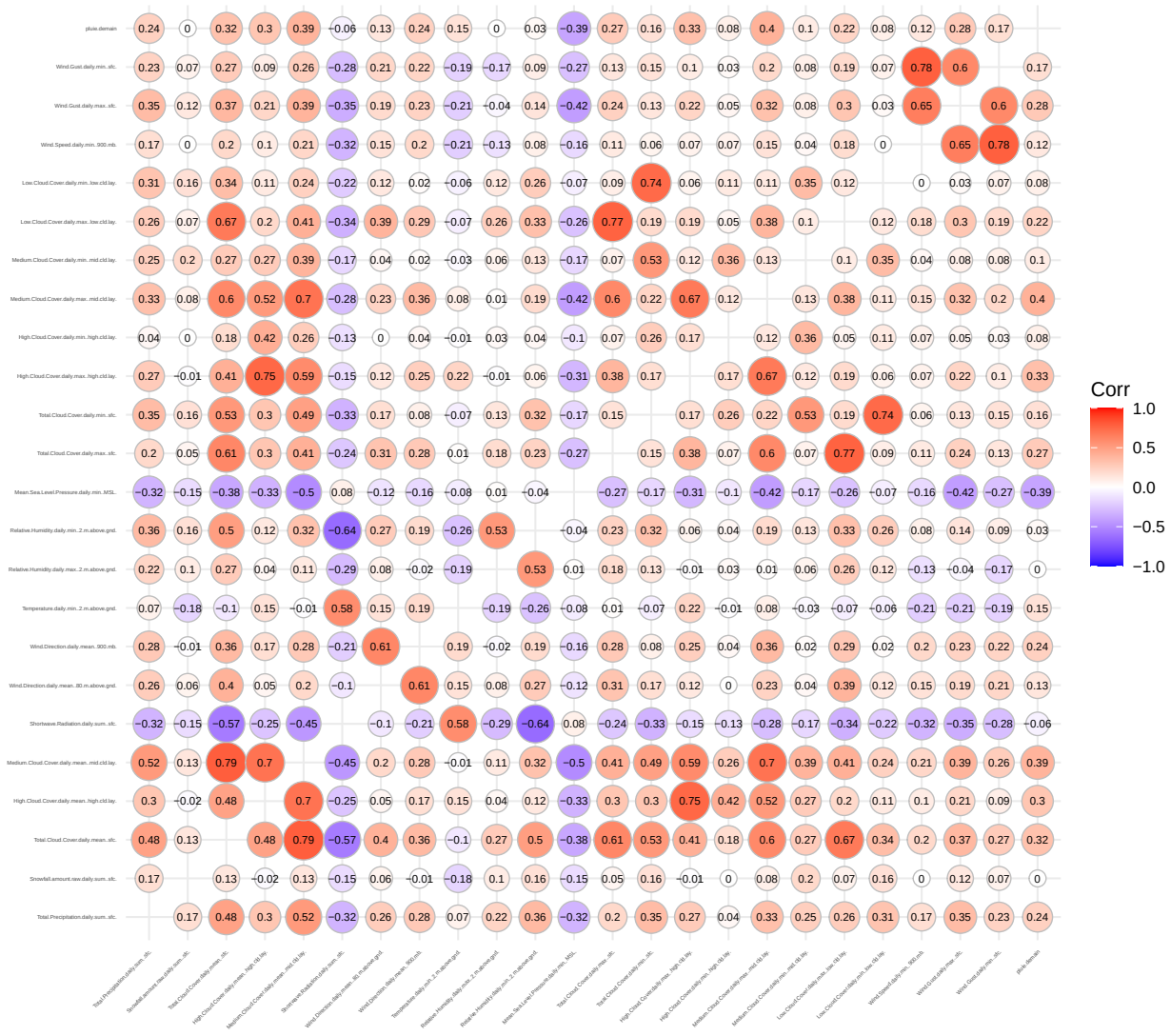
17 variables ont été supprimées selon nos critères, il reste donc 23 variables explicatives dans **d__train__algo**.

Nous constatons que notre méthode de sélection a écarté des variables retenues par la méthode step avec critère AIC. Variables présentes dans **model__step__AIC** mais éliminées par notre algorithme:

- Wind.Speed.daily.mean..80.m.above.gnd.
- Wind.Gust.daily.mean..sfc.
- Mean.Sea.Level.Pressure.daily.max..MSL.
- Total.Cloud.Cover.daily.min..sfc.
- Wind.Speed.daily.max..10.m.above.gnd.
- Wind.Speed.daily.min..10.m.above.gnd.
- Wind.Speed.daily.min..80.m.above.gnd.

Afichons à le corrélogramme des variables restantes:

```
corr <- round(cor(d_train_algo), 2)
ggcorrplot(corr,
           method = "circle",
           lab = TRUE,
           lab_size = 2,
           show.diag = FALSE,
           tl.cex = 3)
```



Il ne reste en effet aucune corrélation supérieure à 0.8.

Modèle utilisant toutes les variabes restantes et calcul du VIF:

```
model_algo <- glm(pluie.demain ~ ., data = d_train_algo, family = binomial)
summary (model_algo)
```

```
##
## Call:
## glm(formula = pluie.demain ~ ., family = binomial, data = d_train_algo)
##
## Coefficients:
##                                          Estimate Std. Error z value
## (Intercept)                             5.700e+01  1.121e+01   5.085
## Total.Precipitation.daily.sum..sfc.     1.379e-02  2.467e-02   0.559
## Snowfall.amount.raw.daily.sum..sfc.    -3.235e-01  1.921e-01  -1.684
## Total.Cloud.Cover.daily.mean..sfc.      4.795e-03  5.444e-03   0.881
## High.Cloud.Cover.daily.mean..high.cld.lay.  -3.303e-03  6.368e-03  -0.519
## Medium.Cloud.Cover.daily.mean..mid.cld.lay.  5.813e-03  5.932e-03   0.980
## Shortwave.Radiation.daily.sum..sfc.     1.208e-04  6.396e-05   1.889
## Wind.Direction.daily.mean..80.m.above.gnd.  -3.916e-03  1.596e-03  -2.453
## Wind.Direction.daily.mean..900.mb.      4.714e-03  1.307e-03   3.607
## Temperature.daily.min..2.m.above.gnd.   2.956e-02  1.666e-02   1.774
## Relative.Humidity.daily.max..2.m.above.gnd.  1.743e-02  1.161e-02   1.501
## Relative.Humidity.daily.min..2.m.above.gnd. -1.269e-02  8.991e-03  -1.412
## Mean.Sea.Level.Pressure.daily.min..MSL.    -6.014e-02  1.092e-02  -5.507
## Total.Cloud.Cover.daily.max..sfc.       3.482e-03  4.643e-03   0.750
## Total.Cloud.Cover.daily.min..sfc.       6.630e-03  6.135e-03   1.081
## High.Cloud.Cover.daily.max..high.cld.lay.   4.200e-03  2.801e-03   1.499
## High.Cloud.Cover.daily.min..high.cld.lay.   9.833e-03  1.962e-02   0.501
## Medium.Cloud.Cover.daily.max..mid.cld.lay.  6.605e-03  3.056e-03   2.161
## Medium.Cloud.Cover.daily.min..mid.cld.lay. -4.122e-03  8.812e-03  -0.468
## Low.Cloud.Cover.daily.max..low.cld.lay.     1.906e-03  3.166e-03   0.602
## Low.Cloud.Cover.daily.min..low.cld.lay.     1.712e-03  6.906e-03   0.248
## Wind.Speed.daily.min..900.mb.              -9.461e-03  1.098e-02  -0.862
## Wind.Gust.daily.max..sfc.                   2.215e-02  7.345e-03   3.016
## Wind.Gust.daily.min..sfc.                   1.727e-02  1.757e-02   0.983
##                                          Pr(>|z|)
## (Intercept)                             3.68e-07 ***
## Total.Precipitation.daily.sum..sfc.      0.57625
## Snowfall.amount.raw.daily.sum..sfc.      0.09222 .
## Total.Cloud.Cover.daily.mean..sfc.       0.37845
## High.Cloud.Cover.daily.mean..high.cld.lay.  0.60394
## Medium.Cloud.Cover.daily.mean..mid.cld.lay.  0.32712
## Shortwave.Radiation.daily.sum..sfc.      0.05883 .
## Wind.Direction.daily.mean..80.m.above.gnd.  0.01415 *
## Wind.Direction.daily.mean..900.mb.       0.00031 ***
## Temperature.daily.min..2.m.above.gnd.    0.07599 .
## Relative.Humidity.daily.max..2.m.above.gnd.  0.13334
## Relative.Humidity.daily.min..2.m.above.gnd.  0.15798
## Mean.Sea.Level.Pressure.daily.min..MSL.  3.65e-08 ***
## Total.Cloud.Cover.daily.max..sfc.        0.45327
## Total.Cloud.Cover.daily.min..sfc.        0.27985
## High.Cloud.Cover.daily.max..high.cld.lay.   0.13375
## High.Cloud.Cover.daily.min..high.cld.lay.   0.61623
## Medium.Cloud.Cover.daily.max..mid.cld.lay.  0.03069 *
## Medium.Cloud.Cover.daily.min..mid.cld.lay.  0.63999
## Low.Cloud.Cover.daily.max..low.cld.lay.     0.54712
## Low.Cloud.Cover.daily.min..low.cld.lay.     0.80423
## Wind.Speed.daily.min..900.mb.               0.38888
## Wind.Gust.daily.max..sfc.                   0.00256 **
```

```
## Wind.Gust.daily.min..sfc.                    0.32569
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1635.4  on 1179  degrees of freedom
## Residual deviance: 1276.3  on 1156  degrees of freedom
## AIC: 1324.3
##
## Number of Fisher Scoring iterations: 4
```

```
vif_values <- vif(model_algo)  # calculer les VIF
print(vif_values)
```

```
##         Total.Precipitation.daily.sum..sfc.
##                                    1.757675
##         Snowfall.amount.raw.daily.sum..sfc.
##                                    1.214137
##          Total.Cloud.Cover.daily.mean..sfc.
##                                    5.793072
##  High.Cloud.Cover.daily.mean..high.cld.lay.
##                                    3.851611
## Medium.Cloud.Cover.daily.mean..mid.cld.lay.
##                                    5.933701
##         Shortwave.Radiation.daily.sum..sfc.
##                                    4.190843
##  Wind.Direction.daily.mean..80.m.above.gnd.
##                                    2.146590
##          Wind.Direction.daily.mean..900.mb.
##                                    1.947596
##       Temperature.daily.min..2.m.above.gnd.
##                                    2.293402
## Relative.Humidity.daily.max..2.m.above.gnd.
##                                    1.674569
## Relative.Humidity.daily.min..2.m.above.gnd.
##                                    2.721944
##     Mean.Sea.Level.Pressure.daily.min..MSL.
##                                    1.348434
##           Total.Cloud.Cover.daily.max..sfc.
##                                    2.912179
##           Total.Cloud.Cover.daily.min..sfc.
##                                    3.704032
##   High.Cloud.Cover.daily.max..high.cld.lay.
##                                    3.017734
##   High.Cloud.Cover.daily.min..high.cld.lay.
##                                    1.405868
##  Medium.Cloud.Cover.daily.max..mid.cld.lay.
##                                    3.188593
##  Medium.Cloud.Cover.daily.min..mid.cld.lay.
##                                    1.661298
##     Low.Cloud.Cover.daily.max..low.cld.lay.
##                                    2.907086
##     Low.Cloud.Cover.daily.min..low.cld.lay.
```

```
##                                  2.643417
##               Wind.Speed.daily.min..900.mb.
##                                  3.261107
##                   Wind.Gust.daily.max..sfc.
##                                  2.152603
##                   Wind.Gust.daily.min..sfc.
##                                  2.863481
```

Aucune valeur de VIF supérieur à 10 constatée sur le modèle utilisant toutes les variables retenues.
Le VIF (Variance Inflation Factor) est une mesure utilisée pour détecter la présence de multicolinéarité entre les variables explicatives.
(On aurait pu aussi sélectionner nos variables avec ce critère plutôt qu'avec une corrélation de 0.8, en prenant par exemple 10 comme valeur seuil du VIF).

Notre modèle avec 23 variables explicatives peut-être largement améioloré (présence de trop fortes p-valeurs).

Nous créons une boucle pour sélectionner automatiquement la variable du modèle qui a la plus forte p-valeur et vérifions par ANOVA que le modèle mis à jour sans cette variable n'affecte pas significativement l'ajustement du modèle.
Nous arrètons la boucle quand aucune variable restante n'a une p-valeur >0.1.

```r
repeat {
  # Obtenir un résumé du modèle
  summary_model <- summary(model_algo)

  # Extraire les p-values des coefficients et leurs noms de variable
  coef_data <- data.frame(
    Variable = rownames(summary_model$coefficients),
    P_Value = summary_model$coefficients[, 4]
  )

  # Trouver la plus forte p-value
  max_p_value <- max(coef_data$P_Value)

  # Sortir de la boucle si la plus forte p-value est inférieure à 0.1
  if (max_p_value < 0.10) {
    break
  }

  # Extraire le nom de la variable correspondant à la plus forte p-value
  variable_max_p <- coef_data$Variable[which.max(coef_data$P_Value)]

  # Afficher la plus forte p-value et le nom de la variable
  cat("\nLa plus forte p-value parmi les coefficients du modèle est :", max_p_value, "\n")
  cat("Variable correspondante :", variable_max_p, "\n")

  # Mettre à jour le modèle en supprimant la variable avec la plus forte p-value
  model_algo <- update(model_algo, paste(". ~ . -", variable_max_p))

  # Comparaison des deux modèles avec l'ANOVA
  anova_result <- anova(update(model_algo, paste(". ~ . +", variable_max_p)), model_algo, test = "LRT")

  # Afficher les résultats de l'ANOVA
  print(anova_result)
}
```

```
##
## La plus forte p-value parmi les coefficients du modèle est : 0.8042286
## Variable correspondante : Low.Cloud.Cover.daily.min..low.cld.lay.
## Analysis of Deviance Table
##
## Model 1: pluie.demain ~ Total.Precipitation.daily.sum..sfc. + Snowfall.amount.raw.daily.sum..sfc. +
##     Total.Cloud.Cover.daily.mean..sfc. + High.Cloud.Cover.daily.mean..high.cld.lay. +
##     Medium.Cloud.Cover.daily.mean..mid.cld.lay. + Shortwave.Radiation.daily.sum..sfc. +
##     Wind.Direction.daily.mean..80.m.above.gnd. + Wind.Direction.daily.mean..900.mb. +
##     Temperature.daily.min..2.m.above.gnd. + Relative.Humidity.daily.max..2.m.above.gnd. +
##     Relative.Humidity.daily.min..2.m.above.gnd. + Mean.Sea.Level.Pressure.daily.min..MSL. +
##     Total.Cloud.Cover.daily.max..sfc. + Total.Cloud.Cover.daily.min..sfc. +
##     High.Cloud.Cover.daily.max..high.cld.lay. + High.Cloud.Cover.daily.min..high.cld.lay. +
##     Medium.Cloud.Cover.daily.max..mid.cld.lay. + Medium.Cloud.Cover.daily.min..mid.cld.lay. +
##     Low.Cloud.Cover.daily.max..low.cld.lay. + Wind.Speed.daily.min..900.mb. +
##     Wind.Gust.daily.max..sfc. + Wind.Gust.daily.min..sfc. + Low.Cloud.Cover.daily.min..low.cld.lay.
## Model 2: pluie.demain ~ Total.Precipitation.daily.sum..sfc. + Snowfall.amount.raw.daily.sum..sfc. +
##     Total.Cloud.Cover.daily.mean..sfc. + High.Cloud.Cover.daily.mean..high.cld.lay. +
##     Medium.Cloud.Cover.daily.mean..mid.cld.lay. + Shortwave.Radiation.daily.sum..sfc. +
##     Wind.Direction.daily.mean..80.m.above.gnd. + Wind.Direction.daily.mean..900.mb. +
##     Temperature.daily.min..2.m.above.gnd. + Relative.Humidity.daily.max..2.m.above.gnd. +
##     Relative.Humidity.daily.min..2.m.above.gnd. + Mean.Sea.Level.Pressure.daily.min..MSL. +
##     Total.Cloud.Cover.daily.max..sfc. + Total.Cloud.Cover.daily.min..sfc. +
##     High.Cloud.Cover.daily.max..high.cld.lay. + High.Cloud.Cover.daily.min..high.cld.lay. +
##     Medium.Cloud.Cover.daily.max..mid.cld.lay. + Medium.Cloud.Cover.daily.min..mid.cld.lay. +
##     Low.Cloud.Cover.daily.max..low.cld.lay. + Wind.Speed.daily.min..900.mb. +
##     Wind.Gust.daily.max..sfc. + Wind.Gust.daily.min..sfc.
##   Resid. Df Resid. Dev Df  Deviance Pr(>Chi)
## 1      1156     1276.3
## 2      1157     1276.4 -1 -0.061252   0.8045
##
## La plus forte p-value parmi les coefficients du modèle est : 0.628661
## Variable correspondante : Medium.Cloud.Cover.daily.min..mid.cld.lay.
## Analysis of Deviance Table
##
## Model 1: pluie.demain ~ Total.Precipitation.daily.sum..sfc. + Snowfall.amount.raw.daily.sum..sfc. +
##     Total.Cloud.Cover.daily.mean..sfc. + High.Cloud.Cover.daily.mean..high.cld.lay. +
##     Medium.Cloud.Cover.daily.mean..mid.cld.lay. + Shortwave.Radiation.daily.sum..sfc. +
##     Wind.Direction.daily.mean..80.m.above.gnd. + Wind.Direction.daily.mean..900.mb. +
##     Temperature.daily.min..2.m.above.gnd. + Relative.Humidity.daily.max..2.m.above.gnd. +
##     Relative.Humidity.daily.min..2.m.above.gnd. + Mean.Sea.Level.Pressure.daily.min..MSL. +
##     Total.Cloud.Cover.daily.max..sfc. + Total.Cloud.Cover.daily.min..sfc. +
##     High.Cloud.Cover.daily.max..high.cld.lay. + High.Cloud.Cover.daily.min..high.cld.lay. +
##     Medium.Cloud.Cover.daily.max..mid.cld.lay. + Low.Cloud.Cover.daily.max..low.cld.lay. +
##     Wind.Speed.daily.min..900.mb. + Wind.Gust.daily.max..sfc. +
##     Wind.Gust.daily.min..sfc. + Medium.Cloud.Cover.daily.min..mid.cld.lay.
## Model 2: pluie.demain ~ Total.Precipitation.daily.sum..sfc. + Snowfall.amount.raw.daily.sum..sfc. +
##     Total.Cloud.Cover.daily.mean..sfc. + High.Cloud.Cover.daily.mean..high.cld.lay. +
##     Medium.Cloud.Cover.daily.mean..mid.cld.lay. + Shortwave.Radiation.daily.sum..sfc. +
##     Wind.Direction.daily.mean..80.m.above.gnd. + Wind.Direction.daily.mean..900.mb. +
##     Temperature.daily.min..2.m.above.gnd. + Relative.Humidity.daily.max..2.m.above.gnd. +
##     Relative.Humidity.daily.min..2.m.above.gnd. + Mean.Sea.Level.Pressure.daily.min..MSL. +
##     Total.Cloud.Cover.daily.max..sfc. + Total.Cloud.Cover.daily.min..sfc. +
##     High.Cloud.Cover.daily.max..high.cld.lay. + High.Cloud.Cover.daily.min..high.cld.lay. +
```

```
##     Medium.Cloud.Cover.daily.max..mid.cld.lay. + Low.Cloud.Cover.daily.max..low.cld.lay. +
##     Wind.Speed.daily.min..900.mb. + Wind.Gust.daily.max..sfc. +
##     Wind.Gust.daily.min..sfc.
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      1157     1276.4
## 2      1158     1276.6 -1 -0.22809   0.6329
##
## La plus forte p-value parmi les coefficients du modèle est : 0.6924559
## Variable correspondante : High.Cloud.Cover.daily.min..high.cld.lay.
## Analysis of Deviance Table
##
## Model 1: pluie.demain ~ Total.Precipitation.daily.sum..sfc. + Snowfall.amount.raw.daily.sum..sfc. +
##     Total.Cloud.Cover.daily.mean..sfc. + High.Cloud.Cover.daily.mean..high.cld.lay. +
##     Medium.Cloud.Cover.daily.mean..mid.cld.lay. + Shortwave.Radiation.daily.sum..sfc. +
##     Wind.Direction.daily.mean..80.m.above.gnd. + Wind.Direction.daily.mean..900.mb. +
##     Temperature.daily.min..2.m.above.gnd. + Relative.Humidity.daily.max..2.m.above.gnd. +
##     Relative.Humidity.daily.min..2.m.above.gnd. + Mean.Sea.Level.Pressure.daily.min..MSL. +
##     Total.Cloud.Cover.daily.max..sfc. + Total.Cloud.Cover.daily.min..sfc. +
##     High.Cloud.Cover.daily.max..high.cld.lay. + Medium.Cloud.Cover.daily.max..mid.cld.lay. +
##     Low.Cloud.Cover.daily.max..low.cld.lay. + Wind.Speed.daily.min..900.mb. +
##     Wind.Gust.daily.max..sfc. + Wind.Gust.daily.min..sfc. + High.Cloud.Cover.daily.min..high.cld.lay
## Model 2: pluie.demain ~ Total.Precipitation.daily.sum..sfc. + Snowfall.amount.raw.daily.sum..sfc. +
##     Total.Cloud.Cover.daily.mean..sfc. + High.Cloud.Cover.daily.mean..high.cld.lay. +
##     Medium.Cloud.Cover.daily.mean..mid.cld.lay. + Shortwave.Radiation.daily.sum..sfc. +
##     Wind.Direction.daily.mean..80.m.above.gnd. + Wind.Direction.daily.mean..900.mb. +
##     Temperature.daily.min..2.m.above.gnd. + Relative.Humidity.daily.max..2.m.above.gnd. +
##     Relative.Humidity.daily.min..2.m.above.gnd. + Mean.Sea.Level.Pressure.daily.min..MSL. +
##     Total.Cloud.Cover.daily.max..sfc. + Total.Cloud.Cover.daily.min..sfc. +
##     High.Cloud.Cover.daily.max..high.cld.lay. + Medium.Cloud.Cover.daily.max..mid.cld.lay. +
##     Low.Cloud.Cover.daily.max..low.cld.lay. + Wind.Speed.daily.min..900.mb. +
##     Wind.Gust.daily.max..sfc. + Wind.Gust.daily.min..sfc.
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      1158     1276.6
## 2      1159     1276.8 -1 -0.16717   0.6826
##
## La plus forte p-value parmi les coefficients du modèle est : 0.7174757
## Variable correspondante : High.Cloud.Cover.daily.mean..high.cld.lay.
## Analysis of Deviance Table
##
## Model 1: pluie.demain ~ Total.Precipitation.daily.sum..sfc. + Snowfall.amount.raw.daily.sum..sfc. +
##     Total.Cloud.Cover.daily.mean..sfc. + Medium.Cloud.Cover.daily.mean..mid.cld.lay. +
##     Shortwave.Radiation.daily.sum..sfc. + Wind.Direction.daily.mean..80.m.above.gnd. +
##     Wind.Direction.daily.mean..900.mb. + Temperature.daily.min..2.m.above.gnd. +
##     Relative.Humidity.daily.max..2.m.above.gnd. + Relative.Humidity.daily.min..2.m.above.gnd. +
##     Mean.Sea.Level.Pressure.daily.min..MSL. + Total.Cloud.Cover.daily.max..sfc. +
##     Total.Cloud.Cover.daily.min..sfc. + High.Cloud.Cover.daily.max..high.cld.lay. +
##     Medium.Cloud.Cover.daily.max..mid.cld.lay. + Low.Cloud.Cover.daily.max..low.cld.lay. +
##     Wind.Speed.daily.min..900.mb. + Wind.Gust.daily.max..sfc. +
##     Wind.Gust.daily.min..sfc. + High.Cloud.Cover.daily.mean..high.cld.lay.
## Model 2: pluie.demain ~ Total.Precipitation.daily.sum..sfc. + Snowfall.amount.raw.daily.sum..sfc. +
##     Total.Cloud.Cover.daily.mean..sfc. + Medium.Cloud.Cover.daily.mean..mid.cld.lay. +
##     Shortwave.Radiation.daily.sum..sfc. + Wind.Direction.daily.mean..80.m.above.gnd. +
##     Wind.Direction.daily.mean..900.mb. + Temperature.daily.min..2.m.above.gnd. +
##     Relative.Humidity.daily.max..2.m.above.gnd. + Relative.Humidity.daily.min..2.m.above.gnd. +
```

```
##     Mean.Sea.Level.Pressure.daily.min..MSL. + Total.Cloud.Cover.daily.max..sfc. +
##     Total.Cloud.Cover.daily.min..sfc. + High.Cloud.Cover.daily.max..high.cld.lay. +
##     Medium.Cloud.Cover.daily.max..mid.cld.lay. + Low.Cloud.Cover.daily.max..low.cld.lay. +
##     Wind.Speed.daily.min..900.mb. + Wind.Gust.daily.max..sfc. +
##     Wind.Gust.daily.min..sfc.
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      1159     1276.8
## 2      1160     1276.9 -1 -0.13087   0.7175
##
## La plus forte p-value parmi les coefficients du modèle est : 0.5469809
## Variable correspondante : Total.Precipitation.daily.sum..sfc.
## Analysis of Deviance Table
##
## Model 1: pluie.demain ~ Snowfall.amount.raw.daily.sum..sfc. + Total.Cloud.Cover.daily.mean..sfc. +
##     Medium.Cloud.Cover.daily.mean..mid.cld.lay. + Shortwave.Radiation.daily.sum..sfc. +
##     Wind.Direction.daily.mean..80.m.above.gnd. + Wind.Direction.daily.mean..900.mb. +
##     Temperature.daily.min..2.m.above.gnd. + Relative.Humidity.daily.max..2.m.above.gnd. +
##     Relative.Humidity.daily.min..2.m.above.gnd. + Mean.Sea.Level.Pressure.daily.min..MSL. +
##     Total.Cloud.Cover.daily.max..sfc. + Total.Cloud.Cover.daily.min..sfc. +
##     High.Cloud.Cover.daily.max..high.cld.lay. + Medium.Cloud.Cover.daily.max..mid.cld.lay. +
##     Low.Cloud.Cover.daily.max..low.cld.lay. + Wind.Speed.daily.min..900.mb. +
##     Wind.Gust.daily.max..sfc. + Wind.Gust.daily.min..sfc. + Total.Precipitation.daily.sum..sfc.
## Model 2: pluie.demain ~ Snowfall.amount.raw.daily.sum..sfc. + Total.Cloud.Cover.daily.mean..sfc. +
##     Medium.Cloud.Cover.daily.mean..mid.cld.lay. + Shortwave.Radiation.daily.sum..sfc. +
##     Wind.Direction.daily.mean..80.m.above.gnd. + Wind.Direction.daily.mean..900.mb. +
##     Temperature.daily.min..2.m.above.gnd. + Relative.Humidity.daily.max..2.m.above.gnd. +
##     Relative.Humidity.daily.min..2.m.above.gnd. + Mean.Sea.Level.Pressure.daily.min..MSL. +
##     Total.Cloud.Cover.daily.max..sfc. + Total.Cloud.Cover.daily.min..sfc. +
##     High.Cloud.Cover.daily.max..high.cld.lay. + Medium.Cloud.Cover.daily.max..mid.cld.lay. +
##     Low.Cloud.Cover.daily.max..low.cld.lay. + Wind.Speed.daily.min..900.mb. +
##     Wind.Gust.daily.max..sfc. + Wind.Gust.daily.min..sfc.
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      1160     1276.9
## 2      1161     1277.3 -1 -0.37056   0.5427
##
## La plus forte p-value parmi les coefficients du modèle est : 0.5458859
## Variable correspondante : Low.Cloud.Cover.daily.max..low.cld.lay.
## Analysis of Deviance Table
##
## Model 1: pluie.demain ~ Snowfall.amount.raw.daily.sum..sfc. + Total.Cloud.Cover.daily.mean..sfc. +
##     Medium.Cloud.Cover.daily.mean..mid.cld.lay. + Shortwave.Radiation.daily.sum..sfc. +
##     Wind.Direction.daily.mean..80.m.above.gnd. + Wind.Direction.daily.mean..900.mb. +
##     Temperature.daily.min..2.m.above.gnd. + Relative.Humidity.daily.max..2.m.above.gnd. +
##     Relative.Humidity.daily.min..2.m.above.gnd. + Mean.Sea.Level.Pressure.daily.min..MSL. +
##     Total.Cloud.Cover.daily.max..sfc. + Total.Cloud.Cover.daily.min..sfc. +
##     High.Cloud.Cover.daily.max..high.cld.lay. + Medium.Cloud.Cover.daily.max..mid.cld.lay. +
##     Wind.Speed.daily.min..900.mb. + Wind.Gust.daily.max..sfc. +
##     Wind.Gust.daily.min..sfc. + Low.Cloud.Cover.daily.max..low.cld.lay.
## Model 2: pluie.demain ~ Snowfall.amount.raw.daily.sum..sfc. + Total.Cloud.Cover.daily.mean..sfc. +
##     Medium.Cloud.Cover.daily.mean..mid.cld.lay. + Shortwave.Radiation.daily.sum..sfc. +
##     Wind.Direction.daily.mean..80.m.above.gnd. + Wind.Direction.daily.mean..900.mb. +
##     Temperature.daily.min..2.m.above.gnd. + Relative.Humidity.daily.max..2.m.above.gnd. +
##     Relative.Humidity.daily.min..2.m.above.gnd. + Mean.Sea.Level.Pressure.daily.min..MSL. +
##     Total.Cloud.Cover.daily.max..sfc. + Total.Cloud.Cover.daily.min..sfc. +
```

```
##     High.Cloud.Cover.daily.max..high.cld.lay. + Medium.Cloud.Cover.daily.max..mid.cld.lay. +
##     Wind.Speed.daily.min..900.mb. + Wind.Gust.daily.max..sfc. +
##     Wind.Gust.daily.min..sfc.
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      1161     1277.3
## 2      1162     1277.6 -1 -0.36447    0.546
##
## La plus forte p-value parmi les coefficients du modèle est : 0.3851067
## Variable correspondante : Wind.Speed.daily.min..900.mb.
## Analysis of Deviance Table
##
## Model 1: pluie.demain ~ Snowfall.amount.raw.daily.sum..sfc. + Total.Cloud.Cover.daily.mean..sfc. +
##     Medium.Cloud.Cover.daily.mean..mid.cld.lay. + Shortwave.Radiation.daily.sum..sfc. +
##     Wind.Direction.daily.mean..80.m.above.gnd. + Wind.Direction.daily.mean..900.mb. +
##     Temperature.daily.min..2.m.above.gnd. + Relative.Humidity.daily.max..2.m.above.gnd. +
##     Relative.Humidity.daily.min..2.m.above.gnd. + Mean.Sea.Level.Pressure.daily.min..MSL. +
##     Total.Cloud.Cover.daily.max..sfc. + Total.Cloud.Cover.daily.min..sfc. +
##     High.Cloud.Cover.daily.max..high.cld.lay. + Medium.Cloud.Cover.daily.max..mid.cld.lay. +
##     Wind.Gust.daily.max..sfc. + Wind.Gust.daily.min..sfc. + Wind.Speed.daily.min..900.mb.
## Model 2: pluie.demain ~ Snowfall.amount.raw.daily.sum..sfc. + Total.Cloud.Cover.daily.mean..sfc. +
##     Medium.Cloud.Cover.daily.mean..mid.cld.lay. + Shortwave.Radiation.daily.sum..sfc. +
##     Wind.Direction.daily.mean..80.m.above.gnd. + Wind.Direction.daily.mean..900.mb. +
##     Temperature.daily.min..2.m.above.gnd. + Relative.Humidity.daily.max..2.m.above.gnd. +
##     Relative.Humidity.daily.min..2.m.above.gnd. + Mean.Sea.Level.Pressure.daily.min..MSL. +
##     Total.Cloud.Cover.daily.max..sfc. + Total.Cloud.Cover.daily.min..sfc. +
##     High.Cloud.Cover.daily.max..high.cld.lay. + Medium.Cloud.Cover.daily.max..mid.cld.lay. +
##     Wind.Gust.daily.max..sfc. + Wind.Gust.daily.min..sfc.
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      1162     1277.6
## 2      1163     1278.4 -1 -0.75118   0.3861
##
## La plus forte p-value parmi les coefficients du modèle est : 0.5388822
## Variable correspondante : Wind.Gust.daily.min..sfc.
## Analysis of Deviance Table
##
## Model 1: pluie.demain ~ Snowfall.amount.raw.daily.sum..sfc. + Total.Cloud.Cover.daily.mean..sfc. +
##     Medium.Cloud.Cover.daily.mean..mid.cld.lay. + Shortwave.Radiation.daily.sum..sfc. +
##     Wind.Direction.daily.mean..80.m.above.gnd. + Wind.Direction.daily.mean..900.mb. +
##     Temperature.daily.min..2.m.above.gnd. + Relative.Humidity.daily.max..2.m.above.gnd. +
##     Relative.Humidity.daily.min..2.m.above.gnd. + Mean.Sea.Level.Pressure.daily.min..MSL. +
##     Total.Cloud.Cover.daily.max..sfc. + Total.Cloud.Cover.daily.min..sfc. +
##     High.Cloud.Cover.daily.max..high.cld.lay. + Medium.Cloud.Cover.daily.max..mid.cld.lay. +
##     Wind.Gust.daily.max..sfc. + Wind.Gust.daily.min..sfc.
## Model 2: pluie.demain ~ Snowfall.amount.raw.daily.sum..sfc. + Total.Cloud.Cover.daily.mean..sfc. +
##     Medium.Cloud.Cover.daily.mean..mid.cld.lay. + Shortwave.Radiation.daily.sum..sfc. +
##     Wind.Direction.daily.mean..80.m.above.gnd. + Wind.Direction.daily.mean..900.mb. +
##     Temperature.daily.min..2.m.above.gnd. + Relative.Humidity.daily.max..2.m.above.gnd. +
##     Relative.Humidity.daily.min..2.m.above.gnd. + Mean.Sea.Level.Pressure.daily.min..MSL. +
##     Total.Cloud.Cover.daily.max..sfc. + Total.Cloud.Cover.daily.min..sfc. +
##     High.Cloud.Cover.daily.max..high.cld.lay. + Medium.Cloud.Cover.daily.max..mid.cld.lay. +
##     Wind.Gust.daily.max..sfc.
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      1163     1278.4
## 2      1164     1278.8 -1 -0.37933    0.538
```

```
##
## La plus forte p-value parmi les coefficients du modèle est : 0.3888811
## Variable correspondante : Medium.Cloud.Cover.daily.mean..mid.cld.lay.
## Analysis of Deviance Table
##
## Model 1: pluie.demain ~ Snowfall.amount.raw.daily.sum..sfc. + Total.Cloud.Cover.daily.mean..sfc. +
##     Shortwave.Radiation.daily.sum..sfc. + Wind.Direction.daily.mean..80.m.above.gnd. +
##     Wind.Direction.daily.mean..900.mb. + Temperature.daily.min..2.m.above.gnd. +
##     Relative.Humidity.daily.max..2.m.above.gnd. + Relative.Humidity.daily.min..2.m.above.gnd. +
##     Mean.Sea.Level.Pressure.daily.min..MSL. + Total.Cloud.Cover.daily.max..sfc. +
##     Total.Cloud.Cover.daily.min..sfc. + High.Cloud.Cover.daily.max..high.cld.lay. +
##     Medium.Cloud.Cover.daily.max..mid.cld.lay. + Wind.Gust.daily.max..sfc. +
##     Medium.Cloud.Cover.daily.mean..mid.cld.lay.
## Model 2: pluie.demain ~ Snowfall.amount.raw.daily.sum..sfc. + Total.Cloud.Cover.daily.mean..sfc. +
##     Shortwave.Radiation.daily.sum..sfc. + Wind.Direction.daily.mean..80.m.above.gnd. +
##     Wind.Direction.daily.mean..900.mb. + Temperature.daily.min..2.m.above.gnd. +
##     Relative.Humidity.daily.max..2.m.above.gnd. + Relative.Humidity.daily.min..2.m.above.gnd. +
##     Mean.Sea.Level.Pressure.daily.min..MSL. + Total.Cloud.Cover.daily.max..sfc. +
##     Total.Cloud.Cover.daily.min..sfc. + High.Cloud.Cover.daily.max..high.cld.lay. +
##     Medium.Cloud.Cover.daily.max..mid.cld.lay. + Wind.Gust.daily.max..sfc.
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      1164     1278.8
## 2      1165     1279.5 -1 -0.74356   0.3885
##
## La plus forte p-value parmi les coefficients du modèle est : 0.3357505
## Variable correspondante : Total.Cloud.Cover.daily.max..sfc.
## Analysis of Deviance Table
##
## Model 1: pluie.demain ~ Snowfall.amount.raw.daily.sum..sfc. + Total.Cloud.Cover.daily.mean..sfc. +
##     Shortwave.Radiation.daily.sum..sfc. + Wind.Direction.daily.mean..80.m.above.gnd. +
##     Wind.Direction.daily.mean..900.mb. + Temperature.daily.min..2.m.above.gnd. +
##     Relative.Humidity.daily.max..2.m.above.gnd. + Relative.Humidity.daily.min..2.m.above.gnd. +
##     Mean.Sea.Level.Pressure.daily.min..MSL. + Total.Cloud.Cover.daily.min..sfc. +
##     High.Cloud.Cover.daily.max..high.cld.lay. + Medium.Cloud.Cover.daily.max..mid.cld.lay. +
##     Wind.Gust.daily.max..sfc. + Total.Cloud.Cover.daily.max..sfc.
## Model 2: pluie.demain ~ Snowfall.amount.raw.daily.sum..sfc. + Total.Cloud.Cover.daily.mean..sfc. +
##     Shortwave.Radiation.daily.sum..sfc. + Wind.Direction.daily.mean..80.m.above.gnd. +
##     Wind.Direction.daily.mean..900.mb. + Temperature.daily.min..2.m.above.gnd. +
##     Relative.Humidity.daily.max..2.m.above.gnd. + Relative.Humidity.daily.min..2.m.above.gnd. +
##     Mean.Sea.Level.Pressure.daily.min..MSL. + Total.Cloud.Cover.daily.min..sfc. +
##     High.Cloud.Cover.daily.max..high.cld.lay. + Medium.Cloud.Cover.daily.max..mid.cld.lay. +
##     Wind.Gust.daily.max..sfc.
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      1165     1279.5
## 2      1166     1280.4 -1 -0.93612   0.3333
##
## La plus forte p-value parmi les coefficients du modèle est : 0.1588103
## Variable correspondante : Relative.Humidity.daily.min..2.m.above.gnd.
## Analysis of Deviance Table
##
## Model 1: pluie.demain ~ Snowfall.amount.raw.daily.sum..sfc. + Total.Cloud.Cover.daily.mean..sfc. +
##     Shortwave.Radiation.daily.sum..sfc. + Wind.Direction.daily.mean..80.m.above.gnd. +
##     Wind.Direction.daily.mean..900.mb. + Temperature.daily.min..2.m.above.gnd. +
##     Relative.Humidity.daily.max..2.m.above.gnd. + Mean.Sea.Level.Pressure.daily.min..MSL. +
```

```
##     Total.Cloud.Cover.daily.min..sfc. + High.Cloud.Cover.daily.max..high.cld.lay. +
##     Medium.Cloud.Cover.daily.max..mid.cld.lay. + Wind.Gust.daily.max..sfc. +
##     Relative.Humidity.daily.min..2.m.above.gnd.
## Model 2: pluie.demain ~ Snowfall.amount.raw.daily.sum..sfc. + Total.Cloud.Cover.daily.mean..sfc. +
##     Shortwave.Radiation.daily.sum..sfc. + Wind.Direction.daily.mean..80.m.above.gnd. +
##     Wind.Direction.daily.mean..900.mb. + Temperature.daily.min..2.m.above.gnd. +
##     Relative.Humidity.daily.max..2.m.above.gnd. + Mean.Sea.Level.Pressure.daily.min..MSL. +
##     Total.Cloud.Cover.daily.min..sfc. + High.Cloud.Cover.daily.max..high.cld.lay. +
##     Medium.Cloud.Cover.daily.max..mid.cld.lay. + Wind.Gust.daily.max..sfc.
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      1166     1280.4
## 2      1167     1282.4 -1  -1.9937    0.158
##
## La plus forte p-value parmi les coefficients du modèle est : 0.2580406
## Variable correspondante : Relative.Humidity.daily.max..2.m.above.gnd.
## Analysis of Deviance Table
##
## Model 1: pluie.demain ~ Snowfall.amount.raw.daily.sum..sfc. + Total.Cloud.Cover.daily.mean..sfc. +
##     Shortwave.Radiation.daily.sum..sfc. + Wind.Direction.daily.mean..80.m.above.gnd. +
##     Wind.Direction.daily.mean..900.mb. + Temperature.daily.min..2.m.above.gnd. +
##     Mean.Sea.Level.Pressure.daily.min..MSL. + Total.Cloud.Cover.daily.min..sfc. +
##     High.Cloud.Cover.daily.max..high.cld.lay. + Medium.Cloud.Cover.daily.max..mid.cld.lay. +
##     Wind.Gust.daily.max..sfc. + Relative.Humidity.daily.max..2.m.above.gnd.
## Model 2: pluie.demain ~ Snowfall.amount.raw.daily.sum..sfc. + Total.Cloud.Cover.daily.mean..sfc. +
##     Shortwave.Radiation.daily.sum..sfc. + Wind.Direction.daily.mean..80.m.above.gnd. +
##     Wind.Direction.daily.mean..900.mb. + Temperature.daily.min..2.m.above.gnd. +
##     Mean.Sea.Level.Pressure.daily.min..MSL. + Total.Cloud.Cover.daily.min..sfc. +
##     High.Cloud.Cover.daily.max..high.cld.lay. + Medium.Cloud.Cover.daily.max..mid.cld.lay. +
##     Wind.Gust.daily.max..sfc.
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      1167     1282.4
## 2      1168     1283.7 -1  -1.2818   0.2576
##
## La plus forte p-value parmi les coefficients du modèle est : 0.1195761
## Variable correspondante : Temperature.daily.min..2.m.above.gnd.
## Analysis of Deviance Table
##
## Model 1: pluie.demain ~ Snowfall.amount.raw.daily.sum..sfc. + Total.Cloud.Cover.daily.mean..sfc. +
##     Shortwave.Radiation.daily.sum..sfc. + Wind.Direction.daily.mean..80.m.above.gnd. +
##     Wind.Direction.daily.mean..900.mb. + Mean.Sea.Level.Pressure.daily.min..MSL. +
##     Total.Cloud.Cover.daily.min..sfc. + High.Cloud.Cover.daily.max..high.cld.lay. +
##     Medium.Cloud.Cover.daily.max..mid.cld.lay. + Wind.Gust.daily.max..sfc. +
##     Temperature.daily.min..2.m.above.gnd.
## Model 2: pluie.demain ~ Snowfall.amount.raw.daily.sum..sfc. + Total.Cloud.Cover.daily.mean..sfc. +
##     Shortwave.Radiation.daily.sum..sfc. + Wind.Direction.daily.mean..80.m.above.gnd. +
##     Wind.Direction.daily.mean..900.mb. + Mean.Sea.Level.Pressure.daily.min..MSL. +
##     Total.Cloud.Cover.daily.min..sfc. + High.Cloud.Cover.daily.max..high.cld.lay. +
##     Medium.Cloud.Cover.daily.max..mid.cld.lay. + Wind.Gust.daily.max..sfc.
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      1168     1283.7
## 2      1169     1286.1 -1  -2.4278   0.1192
```

Les ANOVA successives confirment que le retrait de la variable à chaque étape n'affecte pas significativement l'ajustement du modèle.

Affichons le modèle ainsi retenu:

```
summary(model_algo)
```

```
##
## Call:
## glm(formula = pluie.demain ~ Snowfall.amount.raw.daily.sum..sfc. +
##     Total.Cloud.Cover.daily.mean..sfc. + Shortwave.Radiation.daily.sum..sfc. +
##     Wind.Direction.daily.mean..80.m.above.gnd. + Wind.Direction.daily.mean..900.mb. +
##     Mean.Sea.Level.Pressure.daily.min..MSL. + Total.Cloud.Cover.daily.min..sfc. +
##     High.Cloud.Cover.daily.max..high.cld.lay. + Medium.Cloud.Cover.daily.max..mid.cld.lay. +
##     Wind.Gust.daily.max..sfc., family = binomial, data = d_train_algo)
##
## Coefficients:
##                                          Estimate Std. Error z value
## (Intercept)                             6.222e+01  1.067e+01   5.834
## Snowfall.amount.raw.daily.sum..sfc.    -3.461e-01  1.845e-01  -1.876
## Total.Cloud.Cover.daily.mean..sfc.      1.168e-02  3.903e-03   2.994
## Shortwave.Radiation.daily.sum..sfc.     2.080e-04  4.406e-05   4.722
## Wind.Direction.daily.mean..80.m.above.gnd. -3.881e-03  1.510e-03  -2.570
## Wind.Direction.daily.mean..900.mb.      5.164e-03  1.250e-03   4.132
## Mean.Sea.Level.Pressure.daily.min..MSL. -6.450e-02  1.040e-02  -6.199
## Total.Cloud.Cover.daily.min..sfc.       6.708e-03  3.797e-03   1.766
## High.Cloud.Cover.daily.max..high.cld.lay.  4.741e-03  2.023e-03   2.344
## Medium.Cloud.Cover.daily.max..mid.cld.lay.  7.981e-03  2.471e-03   3.230
## Wind.Gust.daily.max..sfc.               2.098e-02  5.707e-03   3.675
##                                         Pr(>|z|)
## (Intercept)                             5.41e-09 ***
## Snowfall.amount.raw.daily.sum..sfc.     0.060657 .
## Total.Cloud.Cover.daily.mean..sfc.      0.002757 **
## Shortwave.Radiation.daily.sum..sfc.     2.34e-06 ***
## Wind.Direction.daily.mean..80.m.above.gnd. 0.010178 *
## Wind.Direction.daily.mean..900.mb.      3.60e-05 ***
## Mean.Sea.Level.Pressure.daily.min..MSL. 5.68e-10 ***
## Total.Cloud.Cover.daily.min..sfc.       0.077320 .
## High.Cloud.Cover.daily.max..high.cld.lay.  0.019088 *
## Medium.Cloud.Cover.daily.max..mid.cld.lay. 0.001239 **
## Wind.Gust.daily.max..sfc.               0.000237 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1635.4  on 1179  degrees of freedom
## Residual deviance: 1286.1  on 1169  degrees of freedom
## AIC: 1308.1
##
## Number of Fisher Scoring iterations: 4
```

Notre méthode ne retient que 10 variables explicatives.

Évaluation de **model_algo** par matrice de confusion:

```
## Confusion Matrix and Statistics
```

```
##
##           Reference
## Prediction FALSE TRUE
##      FALSE   403  131
##      TRUE    176  470
##
##                Accuracy : 0.7398
##                  95% CI : (0.7138, 0.7647)
##     No Information Rate : 0.5093
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.4787
##
##  Mcnemar's Test P-Value : 0.01203
##
##             Sensitivity : 0.7820
##             Specificity : 0.6960
##          Pos Pred Value : 0.7276
##          Neg Pred Value : 0.7547
##              Prevalence : 0.5093
##          Detection Rate : 0.3983
##    Detection Prevalence : 0.5475
##       Balanced Accuracy : 0.7390
##
##        'Positive' Class : TRUE
##
```

873 bonnes réponses.

Évaluation de **model_algo** par cross-validation (10 plis):

```
## [1] 0.186 0.186
```

Calculons l'AUC de **model_algo**:

```
## Area under the curve: 0.8005
```

Le **model_algo** obtient le même nombre de bonnes réponses que le modèle total avec quatre fois moins de variables explicatives. Sur les autres cirtères de performance, il est assez comparable avec les autres modèles développés jusqu'ici.

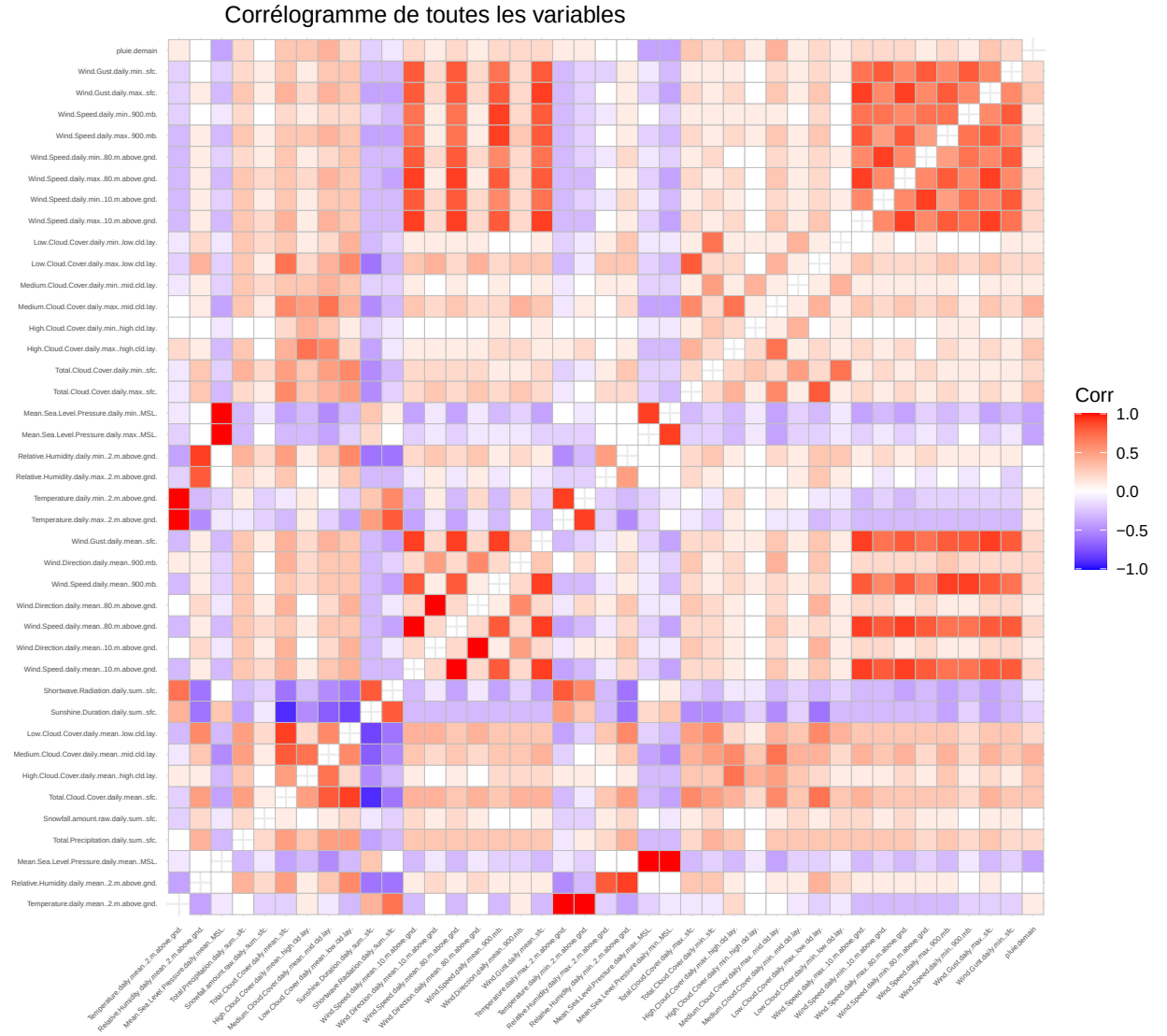
Sélection variables "à la main"

Pour le moment, nos choix se sont faits "à l'aveugle", c'est à dire sans analyser directement les variables et les corréllations entre-elles.
Voyons si nous pouvons faire mieux en les analysant plus précisément.

Commençons par afficher le correlogramme de l'ensemble des données.

```
corr <- round(cor(d_train), 1)
ggcorrplot(corr,
           lab = FALSE,
           show.diag=FALSE,
```

```
            tl.cex = 4,
            title= "          Corrélogramme de toutes les variables")
```

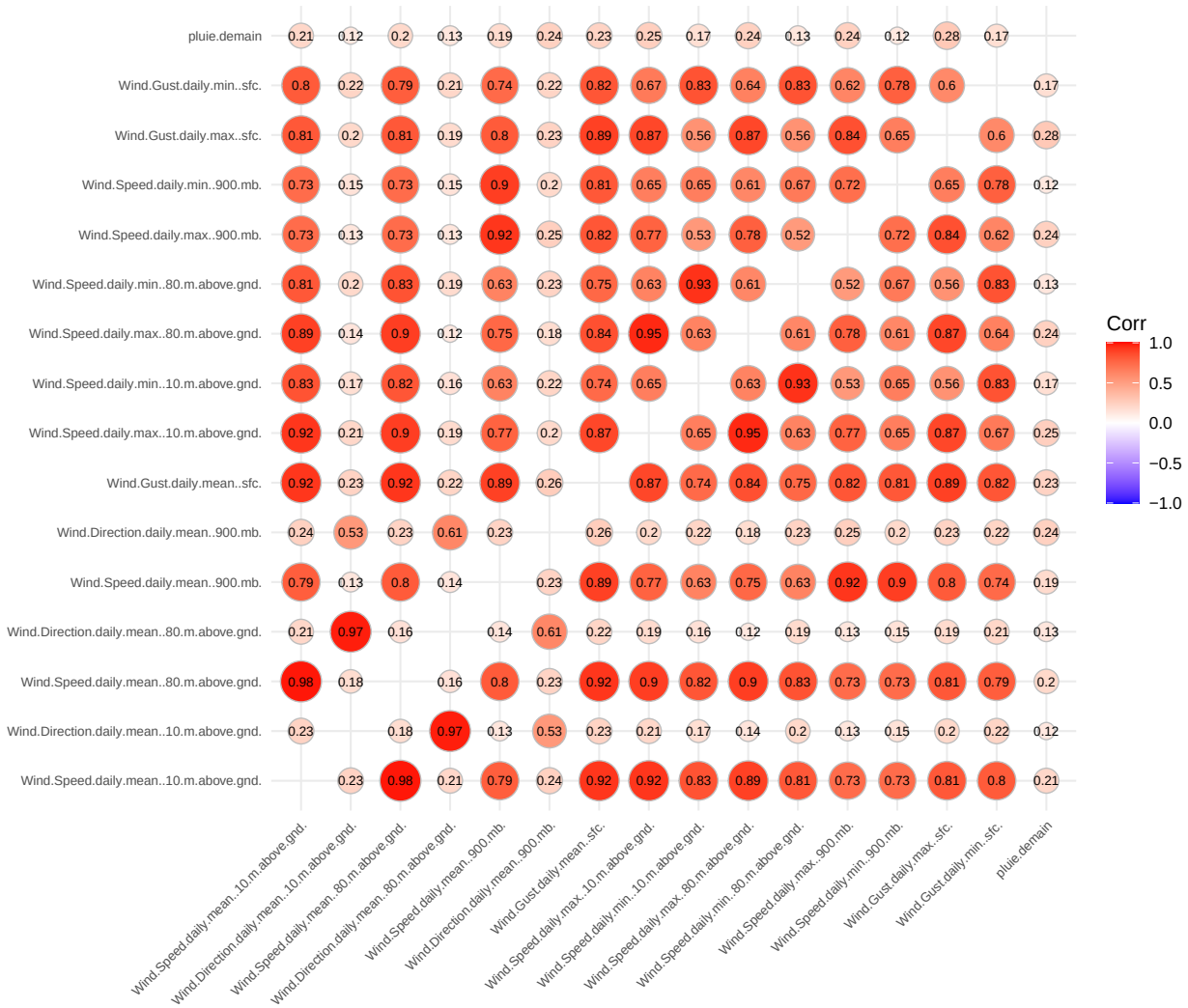


Le trop grand nombre de variable rend le corrélogramme difficilement exploitable. Il permet cependant de constater que nombre de variables sont très corrélées entre-elles. Nous voyons également que la variable d'intérêt (dernière colonne) n'a de forte corrélation avec aucune des variables explicatives (ce qui peut expliquer que la fonction **step()** a retourné des modèles assez différents selon le critère demandé et non des modèles emboîtés).

Nous allons analyser les variables par groupes et les sélectionner "manuellement". Cette méthode est plus laborieuse mais elle serait sans doute à privilégier pourr un spécialiste en météorologie afin de garder la main sur l'information retenue ou rejetée.
Nous divisons notre analyse en trois groupes, les variables de type **"Wind"**, celles de type **"Cloud"** et le groupe des variables n'appartenant à aucun des deux groupes précédant que nous nommons **"Reliquat"**.

Variables “Wind”

```
d_wind <- select(d_train, contains("Wind"), pluie.demain)

corr <- round(cor(d_wind), 2)

ggcorrplot(corr,
           method = "circle",
           lab_size = 2.5,
           lab = TRUE,
           show.diag=FALSE,
           tl.cex = 7)
```

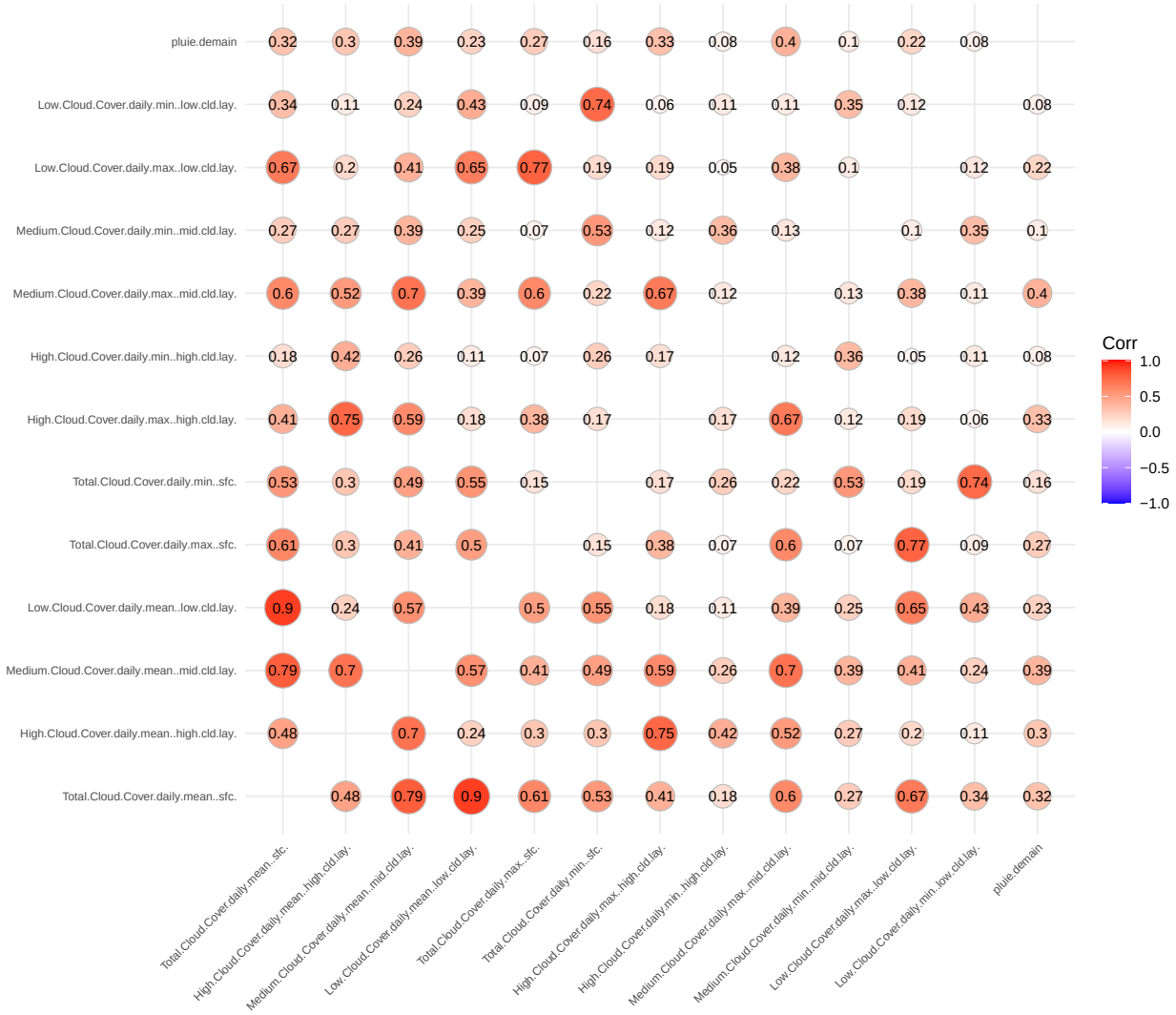


Premier constat, toutes les corrélations sont positives! Plus de vent tend à augmenter en général la probabilité de pluie le jour suivant.
Nous voyons aussi que les plus fortes corrélations sont largement entre les variables explicatives elles-mêmes, beaucoup plus qu'avec notre variable cible (malheureusement).

Après analyse du corrélogramme, nous retenons la variable **"Wind.Gust.daily.max..sfc."** avec correlation de 0.28 avec la variable d'intêret et très corrélée avec d'autres variables qui ont elles-mêmes une forte corrélation avec **"pluidemain"** . Nous retenons ausi **"Wind.Direction.daily.mean..900.mb."** de corrélation 0.24 et peu corrélée avec avec **"Wind.Gust.daily.max..sfc."** (0.23).

Variables "Cloud"

```
d_cloud <- d_train %>%
  select(contains("Cloud"), pluie.demain)

corr <- round(cor(d_cloud), 2)
ggcorrplot(corr,
           method = "circle",
           lab_size = 3,
           lab = TRUE,
           show.diag=FALSE,
           tl.cex = 7)
```

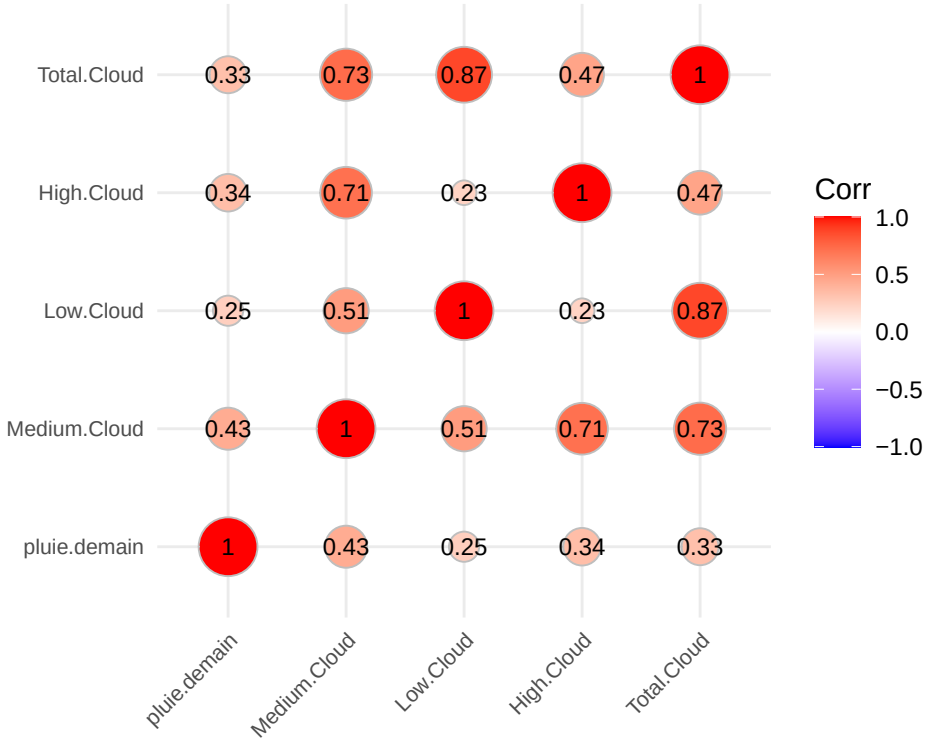


Les corrélation avec la variable d'intêret sont globalement plus fortes qu'elles ne l'étaient avec le vent (cec qui est assez intuitif).

Nous réunissons certaines variables au sein de nouvelles afin de simplifier et tenter d'augmenter les corrélations avec la variable cible:

```
d_cloud_2 <- d_cloud
d_cloud_2 <- d_cloud_2 %>%
  mutate(Medium.Cloud = rowMeans(select(., Medium.Cloud.Cover.daily.max..mid.cld.lay., Medium.Cloud.Cov
 select(-Medium.Cloud.Cover.daily.max..mid.cld.lay., -Medium.Cloud.Cover.daily.mean..mid.cld.lay.) %>%
  mutate(Low.Cloud = rowMeans(select(., Low.Cloud.Cover.daily.max..low.cld.lay., Low.Cloud.Cover.daily.
 select(-Low.Cloud.Cover.daily.max..low.cld.lay., -Low.Cloud.Cover.daily.mean..low.cld.lay.)%>%
  mutate(High.Cloud = rowMeans(select(., High.Cloud.Cover.daily.mean..high.cld.lay., High.Cloud.Cover.da
 select(-High.Cloud.Cover.daily.mean..high.cld.lay., -High.Cloud.Cover.daily.max..high.cld.lay.)%>%
  mutate(Total.Cloud = rowMeans(select(., Total.Cloud.Cover.daily.max..sfc., Total.Cloud.Cover.daily.mea
```

```r
select(-Total.Cloud.Cover.daily.max..sfc., -Total.Cloud.Cover.daily.mean..sfc.)

d_cloud_2 <- d_cloud_2[, !grepl("min", names(d_cloud_2))] # Elimine les variables de type contenant "mi

corr <- round(cor(d_cloud_2), 2)
ggcorrplot(corr,
           method = "circle",
           lab_size = 3,
           lab = TRUE,
           tl.cex = 8)
```

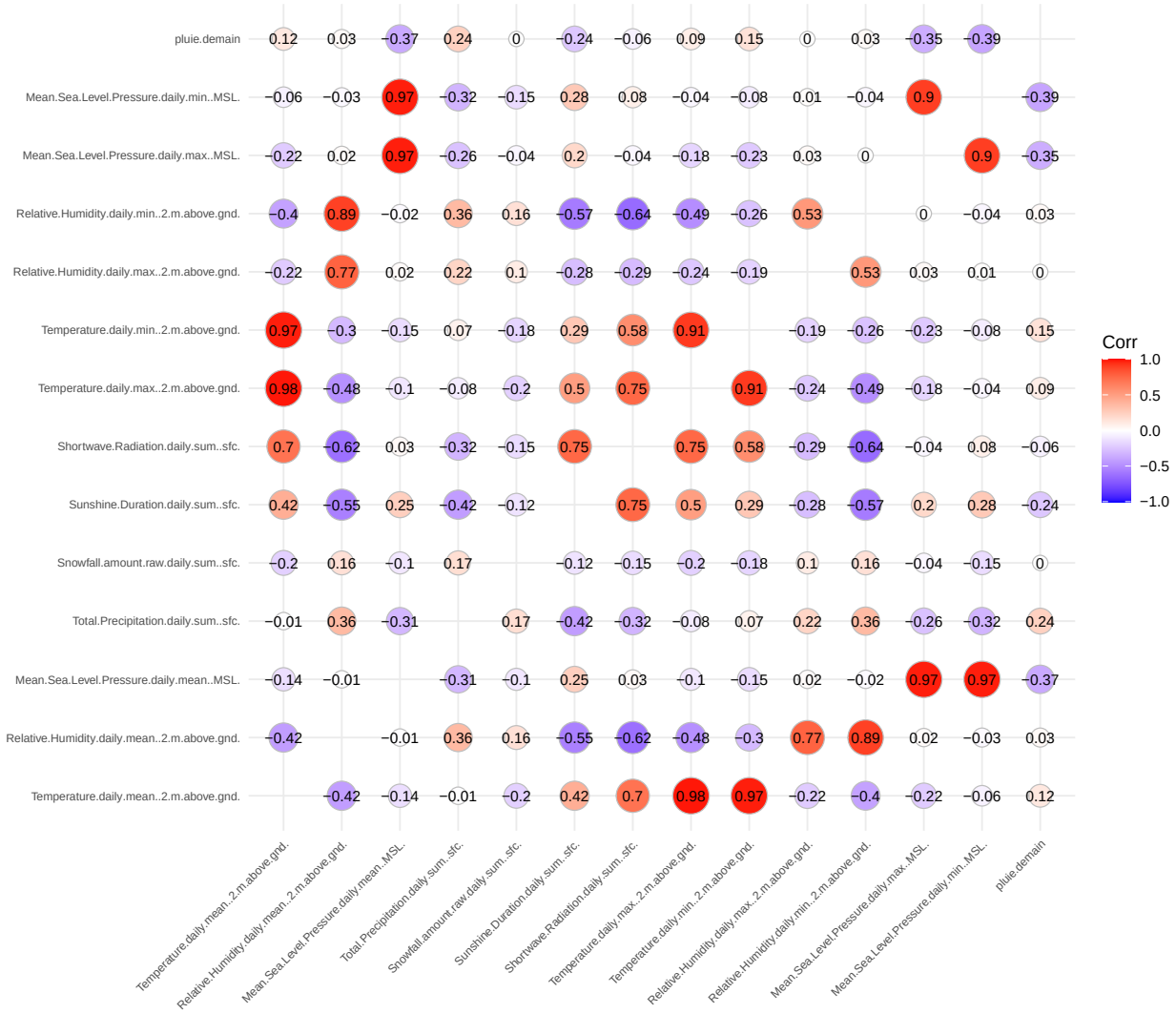


Nous avons supprimé toutes les valeurs “min” (peu corrélées) et moyennées les valeurs “high” au sein de **“High.Cloud”**, les valeurs “total” au sien de **“Total.Cloud”** , les variables “low” dans **“Low.Cloud”** et les variables “medium” dans **“Medium.cloud”**. Cette démarche vise à simplifier le groupe mais aussi à augmenter la corrélation avec la variable d’intêret.

Après analyse du corrélogramme mis à jour, nous ne retenons pas **“Low.Cloud”** la moins corrélée avec notre variable cible (0.25) et très corrélée avec **“Total.Cloud”** (0.87 entre elles).

Variables restantes

```r
d_reliq <- d_train %>%
  select( -contains(c("Wind","Cloud","numb","Year","Month", "Day")),pluie.demain)

corr <- round(cor(d_reliq), 2)
ggcorrplot(corr,
           method = "circle",
           lab_size = 3,
           lab = TRUE,
           show.diag=FALSE,
           tl.cex = 7)
```



Nous effectuons le même travail de sélection (favoriser les meilleures corrélations avec **pluie.demain** et éviter une trop forte multicolinéarité).

Liste finale des variables retenues:

- Wind.Direction.daily.mean..900.mb.
- Wind.Gust.daily.max..sfc.
- Medium.Cloud
- High.Cloud
- Total.Cloud
- Total.Precipitation.daily.sum..sfc.
- Sunshine.Duration.daily.sum..sfc.
- Mean.Sea.Level.Pressure.daily.min..MSL.
- Temperature.daily.min..2.m.above.gnd.

modèle avec les variables retenues

Créons un nouveau dataframe **d__train__new** qui ajoute nos nouvelles variables retenues à **d__train**:

```
d_train_new <- d_train %>%
  mutate(Medium.Cloud = rowMeans(select(., Medium.Cloud.Cover.daily.max..mid.cld.lay., Medium.Cloud.Cov
 select(-Medium.Cloud.Cover.daily.max..mid.cld.lay., -Medium.Cloud.Cover.daily.mean..mid.cld.lay.) %>%
  mutate(Low.Cloud = rowMeans(select(., Low.Cloud.Cover.daily.max..low.cld.lay., Low.Cloud.Cover.daily.
 select(-Low.Cloud.Cover.daily.max..low.cld.lay., -Low.Cloud.Cover.daily.mean..low.cld.lay.)%>%
  mutate(High.Cloud = rowMeans(select(., High.Cloud.Cover.daily.mean..high.cld.lay., High.Cloud.Cover.da
 select(-High.Cloud.Cover.daily.mean..high.cld.lay., -High.Cloud.Cover.daily.max..high.cld.lay.)%>%
  mutate(Total.Cloud = rowMeans(select(., Total.Cloud.Cover.daily.max..sfc., Total.Cloud.Cover.daily.mea
select(-Total.Cloud.Cover.daily.max..sfc., -Total.Cloud.Cover.daily.mean..sfc.)
```

Après essai du modèle (non représenté ici) sur l'ensemble des variables de **d__train__new**, nous ne retenons pas **Total.Precipitation.daily.sum..sfc.** (p-valeur de 0.84107).

Nous obtenons finalement le modèle suivant:

```
model_select <- glm(pluie.demain ~  Wind.Gust.daily.max..sfc.+ Wind.Direction.daily.mean..900.mb.+Mediu
summary (model_select)
```

```
##
## Call:
## glm(formula = pluie.demain ~ Wind.Gust.daily.max..sfc. + Wind.Direction.daily.mean..900.mb. +
##     Medium.Cloud + High.Cloud + Total.Cloud + Sunshine.Duration.daily.sum..sfc. +
##     Mean.Sea.Level.Pressure.daily.min..MSL. + Temperature.daily.min..2.m.above.gnd.,
##     family = binomial, data = d_train_new)
##
## Coefficients:
##                                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)                         61.1037609 10.6328156   5.747  9.1e-09
## Wind.Gust.daily.max..sfc.            0.0186640  0.0057308   3.257  0.00113
## Wind.Direction.daily.mean..900.mb.   0.0023502  0.0010125   2.321  0.02027
## Medium.Cloud                         0.0129145  0.0039406   3.277  0.00105
## High.Cloud                           0.0050895  0.0031276   1.627  0.10367
## Total.Cloud                          0.0119029  0.0053601   2.221  0.02637
## Sunshine.Duration.daily.sum..sfc.    0.0008304  0.0004828   1.720  0.08547
## Mean.Sea.Level.Pressure.daily.min..MSL. -0.0635755  0.0103864  -6.121  9.3e-10
## Temperature.daily.min..2.m.above.gnd.    0.0428477  0.0132656   3.230  0.00124
##
```

```
## (Intercept)                              ***
## Wind.Gust.daily.max..sfc.                **
## Wind.Direction.daily.mean..900.mb.       *
## Medium.Cloud                             **
## High.Cloud
## Total.Cloud                              *
## Sunshine.Duration.daily.sum..sfc.        .
## Mean.Sea.Level.Pressure.daily.min..MSL. ***
## Temperature.daily.min..2.m.above.gnd.    **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1635.4  on 1179  degrees of freedom
## Residual deviance: 1297.7  on 1171  degrees of freedom
## AIC: 1315.7
##
## Number of Fisher Scoring iterations: 4
```

Évaluation de **model_select** par matrice de confusion:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE   398  129
##      TRUE    181  472
##
##                Accuracy : 0.7373
##                  95% CI : (0.7112, 0.7622)
##     No Information Rate : 0.5093
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4735
##
##  Mcnemar's Test P-Value : 0.003772
##
##             Sensitivity : 0.7854
##             Specificity : 0.6874
##          Pos Pred Value : 0.7228
##          Neg Pred Value : 0.7552
##              Prevalence : 0.5093
##          Detection Rate : 0.4000
##    Detection Prevalence : 0.5534
##       Balanced Accuracy : 0.7364
##
##        'Positive' Class : TRUE
##
```

Évaluation de **model_select** par cross-validation (10 plis):

```
## [1] 0.188 0.188
```

Calculons l'AUC de **model_step_AIC**:

```
## Area under the curve: 0.7953
```

Avec 870 bonnes réponses, **model_select** obtient globalement de moins bonnes performances que les autres étudiés jusque là. Mais il ne démérite pas sur les autres critères de performance malgrè sa parcimonie assez poussée (8 variables).
Cette parcimonie rend plus facile l'ajout d'interactions, d'un point de vue computationnel mais aussi en cas de recherche de modèle explicatif.

model_select avec interactions

Sélectionnons les interactions avec la fonction **step()**:

```
formula <- as.formula("pluie.demain ~ (Wind.Gust.daily.max..sfc. + Wind.Direction.daily.mean..900.mb. +
    Medium.Cloud + High.Cloud + Total.Cloud + Sunshine.Duration.daily.sum..sfc. +
    Mean.Sea.Level.Pressure.daily.min..MSL. + Temperature.daily.min..2.m.above.gnd.)^2")

# Créer un modèle provisoire avec toutes les interactions
model_prov <- glm(formula, data = d_train_new, family = binomial)

# Sélectionner modèle par méthode step selon critère AIC
model_select_int <- step(model_prov, direction = "both", trace = 0)
summary (model_select_int)
```

```
##
## Call:
## glm(formula = pluie.demain ~ Wind.Gust.daily.max..sfc. + Wind.Direction.daily.mean..900.mb. +
##     Medium.Cloud + High.Cloud + Total.Cloud + Sunshine.Duration.daily.sum..sfc. +
##     Mean.Sea.Level.Pressure.daily.min..MSL. + Temperature.daily.min..2.m.above.gnd. +
##     Wind.Gust.daily.max..sfc.:Temperature.daily.min..2.m.above.gnd. +
##     Wind.Direction.daily.mean..900.mb.:Temperature.daily.min..2.m.above.gnd. +
##     Medium.Cloud:Temperature.daily.min..2.m.above.gnd. + High.Cloud:Total.Cloud +
##     High.Cloud:Temperature.daily.min..2.m.above.gnd. + Total.Cloud:Sunshine.Duration.daily.sum..sfc.
##     Mean.Sea.Level.Pressure.daily.min..MSL.:Temperature.daily.min..2.m.above.gnd.,
##     family = binomial, data = d_train_new)
##
## Coefficients:
##                                                                    Estimate
## (Intercept)                                                       9.402e+00
## Wind.Gust.daily.max..sfc.                                         4.890e-02
## Wind.Direction.daily.mean..900.mb.                                6.985e-03
## Medium.Cloud                                                     -7.672e-03
## High.Cloud                                                        5.236e-02
## Total.Cloud                                                       5.422e-02
## Sunshine.Duration.daily.sum..sfc.                                 4.648e-03
## Mean.Sea.Level.Pressure.daily.min..MSL.                          -1.711e-02
## Temperature.daily.min..2.m.above.gnd.                             1.127e+01
## Wind.Gust.daily.max..sfc.:Temperature.daily.min..2.m.above.gnd.  -4.567e-03
## Wind.Direction.daily.mean..900.mb.:Temperature.daily.min..2.m.above.gnd. -5.756e-04
## Medium.Cloud:Temperature.daily.min..2.m.above.gnd.                2.652e-03
## High.Cloud:Total.Cloud                                           -4.671e-04
```

```
## High.Cloud:Temperature.daily.min..2.m.above.gnd.                                -1.596e-03
## Total.Cloud:Sunshine.Duration.daily.sum..sfc.                                   -4.546e-05
## Mean.Sea.Level.Pressure.daily.min..MSL.:Temperature.daily.min..2.m.above.gnd. -1.091e-02
##                                                                                 Std. Error
## (Intercept)                                                                      1.398e+01
## Wind.Gust.daily.max..sfc.                                                        9.154e-03
## Wind.Direction.daily.mean..900.mb.                                               1.610e-03
## Medium.Cloud                                                                     5.966e-03
## High.Cloud                                                                       1.318e-02
## Total.Cloud                                                                      1.369e-02
## Sunshine.Duration.daily.sum..sfc.                                                1.471e-03
## Mean.Sea.Level.Pressure.daily.min..MSL.                                          1.352e-02
## Temperature.daily.min..2.m.above.gnd.                                            2.001e+00
## Wind.Gust.daily.max..sfc.:Temperature.daily.min..2.m.above.gnd.                  9.867e-04
## Wind.Direction.daily.mean..900.mb.:Temperature.daily.min..2.m.above.gnd.         1.656e-04
## Medium.Cloud:Temperature.daily.min..2.m.above.gnd.                               6.176e-04
## High.Cloud:Total.Cloud                                                           1.455e-04
## High.Cloud:Temperature.daily.min..2.m.above.gnd.                                 5.771e-04
## Total.Cloud:Sunshine.Duration.daily.sum..sfc.                                    1.567e-05
## Mean.Sea.Level.Pressure.daily.min..MSL.:Temperature.daily.min..2.m.above.gnd.    1.959e-03
##                                                                                 z value
## (Intercept)                                                                       0.672
## Wind.Gust.daily.max..sfc.                                                         5.342
## Wind.Direction.daily.mean..900.mb.                                                4.339
## Medium.Cloud                                                                     -1.286
## High.Cloud                                                                        3.971
## Total.Cloud                                                                       3.961
## Sunshine.Duration.daily.sum..sfc.                                                 3.160
## Mean.Sea.Level.Pressure.daily.min..MSL.                                          -1.265
## Temperature.daily.min..2.m.above.gnd.                                             5.634
## Wind.Gust.daily.max..sfc.:Temperature.daily.min..2.m.above.gnd.                  -4.628
## Wind.Direction.daily.mean..900.mb.:Temperature.daily.min..2.m.above.gnd.         -3.475
## Medium.Cloud:Temperature.daily.min..2.m.above.gnd.                                4.294
## High.Cloud:Total.Cloud                                                           -3.211
## High.Cloud:Temperature.daily.min..2.m.above.gnd.                                 -2.766
## Total.Cloud:Sunshine.Duration.daily.sum..sfc.                                    -2.901
## Mean.Sea.Level.Pressure.daily.min..MSL.:Temperature.daily.min..2.m.above.gnd.    -5.568
##                                                                                 Pr(>|z|)
## (Intercept)                                                                      0.50134
## Wind.Gust.daily.max..sfc.                                                       9.17e-08
## Wind.Direction.daily.mean..900.mb.                                              1.43e-05
## Medium.Cloud                                                                     0.19847
## High.Cloud                                                                      7.15e-05
## Total.Cloud                                                                     7.45e-05
## Sunshine.Duration.daily.sum..sfc.                                                0.00158
## Mean.Sea.Level.Pressure.daily.min..MSL.                                          0.20586
## Temperature.daily.min..2.m.above.gnd.                                           1.76e-08
## Wind.Gust.daily.max..sfc.:Temperature.daily.min..2.m.above.gnd.                 3.70e-06
## Wind.Direction.daily.mean..900.mb.:Temperature.daily.min..2.m.above.gnd.         0.00051
## Medium.Cloud:Temperature.daily.min..2.m.above.gnd.                              1.76e-05
## High.Cloud:Total.Cloud                                                           0.00132
## High.Cloud:Temperature.daily.min..2.m.above.gnd.                                 0.00567
## Total.Cloud:Sunshine.Duration.daily.sum..sfc.                                    0.00372
## Mean.Sea.Level.Pressure.daily.min..MSL.:Temperature.daily.min..2.m.above.gnd. 2.58e-08
```

```
##
## (Intercept)
## Wind.Gust.daily.max..sfc.                                                    ***
## Wind.Direction.daily.mean..900.mb.                                           ***
## Medium.Cloud
## High.Cloud                                                                   ***
## Total.Cloud                                                                  ***
## Sunshine.Duration.daily.sum..sfc.                                            **
## Mean.Sea.Level.Pressure.daily.min..MSL.
## Temperature.daily.min..2.m.above.gnd.                                        ***
## Wind.Gust.daily.max..sfc.:Temperature.daily.min..2.m.above.gnd.              ***
## Wind.Direction.daily.mean..900.mb.:Temperature.daily.min..2.m.above.gnd.     ***
## Medium.Cloud:Temperature.daily.min..2.m.above.gnd.                           ***
## High.Cloud:Total.Cloud                                                       **
## High.Cloud:Temperature.daily.min..2.m.above.gnd.                             **
## Total.Cloud:Sunshine.Duration.daily.sum..sfc.                                **
## Mean.Sea.Level.Pressure.daily.min..MSL.:Temperature.daily.min..2.m.above.gnd. ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1635.4  on 1179  degrees of freedom
## Residual deviance: 1215.6  on 1164  degrees of freedom
## AIC: 1247.6
##
## Number of Fisher Scoring iterations: 4
```

7 Interactions ont été retenues. L'AIC a baissé assez largement 1247.6 contre 1315.7 sans interactions. Il est même le plus faible rencontré jusque-ici.

Évaluation de **model_select_int** par matrice de confusion:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE   413  137
##      TRUE    166  464
##
##                Accuracy : 0.7432
##                  95% CI : (0.7173, 0.7679)
##     No Information Rate : 0.5093
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.4858
##
##  Mcnemar's Test P-Value : 0.1077
##
##             Sensitivity : 0.7720
##             Specificity : 0.7133
##          Pos Pred Value : 0.7365
##          Neg Pred Value : 0.7509
##              Prevalence : 0.5093
```

```
##          Detection Rate : 0.3932
##    Detection Prevalence : 0.5339
##       Balanced Accuracy : 0.7427
##
##        'Positive' Class : TRUE
##
```

Évaluation de **model_select_int** par cross-validation (10 plis):

```
## [1] 0.178 0.177
```

Calculons l'AUC de **model_select_int**:

```
## Area under the curve: 0.8241
```

Le **model_select_int** donne 877 fois sur 1180 la bonne réponse (le meilleut score jusqu'ici) et obtient de bonnes performances, comparativement aux autres modèles, sur tous nos critères.

Composantes principales

Essayons une approche par composante principale. L'avantage est qu'ici nous n'avons pas à nous soucier de la multicolinéarité, les composantes principales étant indépendantes entre elles (corrélation égale à 0 pour chaque paire de composantes). Effectuons l'ACP sur toutes les variables et évaluons un modèle complet:

```r
# Effectuer l'ACP en normalisant les données (scale = TRUE)
pca_result <- prcomp(d_train[, -which(names(d_train) == "pluie.demain")], scale. = TRUE)
summary(pca_result)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     3.600 2.3622 2.1319 1.59800 1.39197 1.35867 1.15212
## Proportion of Variance 0.324 0.1395 0.1136 0.06384 0.04844 0.04615 0.03318
## Cumulative Proportion  0.324 0.4635 0.5771 0.64095 0.68939 0.73554 0.76872
##                           PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     1.03971 1.02231 0.99654 0.93877 0.80361 0.74181 0.73100
## Proportion of Variance 0.02702 0.02613 0.02483 0.02203 0.01614 0.01376 0.01336
## Cumulative Proportion  0.79575 0.82188 0.84670 0.86874 0.88488 0.89864 0.91200
##                           PC15   PC16    PC17    PC18    PC19    PC20    PC21
## Standard deviation     0.71113 0.6871 0.61704 0.58388 0.46934 0.44306 0.43427
## Proportion of Variance 0.01264 0.0118 0.00952 0.00852 0.00551 0.00491 0.00471
## Cumulative Proportion  0.92464 0.9364 0.94596 0.95448 0.95999 0.96490 0.96961
##                           PC22   PC23    PC24    PC25    PC26    PC27    PC28
## Standard deviation     0.42181 0.4000 0.37358 0.36036 0.32797 0.29873 0.27182
## Proportion of Variance 0.00445 0.0040 0.00349 0.00325 0.00269 0.00223 0.00185
## Cumulative Proportion  0.97406 0.9781 0.98155 0.98479 0.98748 0.98971 0.99156
##                           PC29    PC30    PC31   PC32   PC33    PC34    PC35
## Standard deviation     0.24724 0.24249 0.21691 0.1998 0.1787 0.15924 0.15024
## Proportion of Variance 0.00153 0.00147 0.00118 0.0010 0.0008 0.00063 0.00056
## Cumulative Proportion  0.99309 0.99456 0.99574 0.9967 0.9975 0.99817 0.99873
##                           PC36    PC37    PC38    PC39    PC40
## Standard deviation     0.13605 0.13418 0.08892 0.06145 0.05104
## Proportion of Variance 0.00046 0.00045 0.00020 0.00009 0.00007
## Cumulative Proportion  0.99919 0.99964 0.99984 0.99993 1.00000
```

```r
# Utiliser toutes les composantes principales pour la régression logistique
X_pca <- predict(pca_result, newdata = d_train[, -which(names(d_train) == "pluie.demain")])

# Combiner les composantes principales avec la variable cible
data_pca <- data.frame(X_pca, pluie.demain = d_train$pluie.demain)

# Modèle de régression logistique avec toutes les composantes principales
model_ACP_complet <- glm(pluie.demain ~ ., data = data_pca, family = binomial)

summary(model_ACP_complet)
```

```
##
## Call:
## glm(formula = pluie.demain ~ ., family = binomial, data = data_pca)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.030143   0.072085   0.418 0.675826
## PC1          0.253446   0.022837  11.098  < 2e-16 ***
## PC2          0.159632   0.032178   4.961 7.02e-07 ***
## PC3         -0.438894   0.037803 -11.610  < 2e-16 ***
## PC4          0.052729   0.043883   1.202 0.229524
## PC5         -0.118131   0.055435  -2.131 0.033090 *
## PC6          0.027433   0.053008   0.518 0.604784
## PC7          0.017067   0.061827   0.276 0.782509
## PC8         -0.086951   0.073139  -1.189 0.234501
## PC9         -0.003351   0.078731  -0.043 0.966047
## PC10        -0.042093   0.076154  -0.553 0.580443
## PC11        -0.059672   0.083373  -0.716 0.474165
## PC12         0.054507   0.089867   0.607 0.544169
## PC13        -0.024921   0.100259  -0.249 0.803695
## PC14        -0.026797   0.107793  -0.249 0.803674
## PC15        -0.333073   0.102076  -3.263 0.001102 **
## PC16         0.064805   0.109594   0.591 0.554305
## PC17        -0.193540   0.114047  -1.697 0.089693 .
## PC18        -0.256815   0.118522  -2.167 0.030249 *
## PC19        -0.033904   0.151303  -0.224 0.822694
## PC20         0.252550   0.160369   1.575 0.115301
## PC21        -0.153711   0.162590  -0.945 0.344460
## PC22        -0.092768   0.168733  -0.550 0.582460
## PC23         0.082388   0.176992   0.465 0.641581
## PC24         0.329512   0.196753   1.675 0.093982 .
## PC25         0.018261   0.199301   0.092 0.926994
## PC26         0.053035   0.216831   0.245 0.806771
## PC27         0.732994   0.250780   2.923 0.003468 **
## PC28         0.078864   0.255351   0.309 0.757437
## PC29         0.323434   0.299503   1.080 0.280187
## PC30         0.675800   0.296680   2.278 0.022734 *
## PC31        -0.302745   0.324255  -0.934 0.350478
## PC32        -0.159646   0.361778  -0.441 0.659009
## PC33         0.473530   0.396322   1.195 0.232160
## PC34         0.239379   0.443994   0.539 0.589784
## PC35        -0.391814   0.471120  -0.832 0.405599
```

```
## PC36        -1.000586   0.515255  -1.942 0.052147 .
## PC37        -0.054067   0.547513  -0.099 0.921337
## PC38        -0.644583   0.800841  -0.805 0.420887
## PC39         4.406189   1.307428   3.370 0.000751 ***
## PC40         3.021998   1.465455   2.062 0.039193 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1635.4  on 1179  degrees of freedom
## Residual deviance: 1239.5  on 1139  degrees of freedom
## AIC: 1321.5
##
## Number of Fisher Scoring iterations: 4
```

Évaluation de **model_ACP_complet** par matrice de confusion:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE   412  140
##      TRUE    167  461
##
##                Accuracy : 0.7398
##                  95% CI : (0.7138, 0.7647)
##     No Information Rate : 0.5093
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.479
##
##  Mcnemar's Test P-Value : 0.1378
##
##             Sensitivity : 0.7671
##             Specificity : 0.7116
##          Pos Pred Value : 0.7341
##          Neg Pred Value : 0.7464
##              Prevalence : 0.5093
##          Detection Rate : 0.3907
##    Detection Prevalence : 0.5322
##       Balanced Accuracy : 0.7393
##
##        'Positive' Class : TRUE
##
```

Évaluation de **model_ACP_complet** par cross-validation (10 plis):

```
## [1] 0.190 0.189
```

Calculons l'AUC de **model_ACP_complet**:

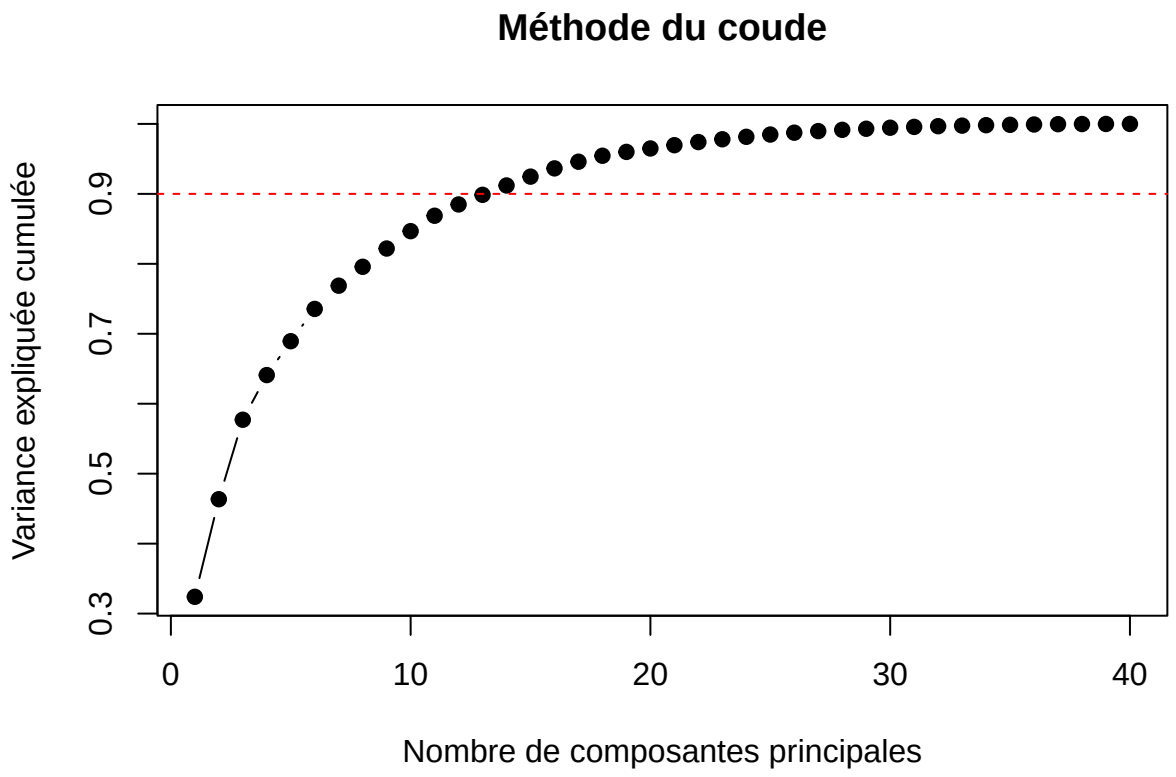
```
## Area under the curve: 0.8176
```

Nous constatons que pour tous nos critères, les résultats sont rigoureusement identiques à ceux du modèle total sur les variables "brutes" (**model_total**). Nous l'expliquons par le fait nous avons dans ces deux modèles la même "quantité d'information".

Avec séléction des composantes principales

Si les deux modèles avec toutes les variables (avec et sans ACP) sont identiques, nous pouvons utiliser nos composantes principales en sélectionnant les premières, celles qui expliquent le mieux Utilisons la méthode du coude pour sélectionner nos CP, avec 90% de la variance expliquée comme seuil:

```
explained_variance <- pca_result$sdev^2 / sum(pca_result$sdev^2)
cumulative_explained_variance <- cumsum(explained_variance)

# Tracer le graphique
plot(cumulative_explained_variance, type = "b", pch = 19, xlab = "Nombre de composantes principales", yl

# ligne horizontale 0.9
abline(h = 0.9, col = "red", lty = 2)
```



Nous retenons les 13 premières composantes principales. Élaborons un modèle à partir de ces 13 variables en utilisant la fonction **step()** avec interactions.

Note: nous avons essayer un modèle utilisant directement les 13 composantes dans un modèle sans sélection de composantes ni interactions, les performances sont décevantes: 0.7263 en précision et AIC de 1406.6. Nous ne le présentons pas ici.

ACP avec interactions

```r
# Capturer l'horodatage de début
start_time <- Sys.time()

# Sélectionner les 13 premières CP
X_pca_13 <- X_pca[, 1:13]

# Créer un nouveau data.frame avec les composantes principales et la variable cible
data_pca_13 <- data.frame(X_pca_13, pluie.demain = d_train$pluie.demain)

# Définir la formule initiale avec les interactions entre les 13 premières composantes principales
formula <- as.formula(paste("pluie.demain ~ (", paste(names(data_pca_13)[-ncol(data_pca_13)], collapse =

# Créer le modèle initial avec toutes les interactions
model_initial <- glm(formula, data = data_pca_13, family = binomial)

# Sélectionner le meilleur modèle par méthode step avec critère AIC, en incluant les interactions
model_ACP_13_int <- step(model_initial, direction = "both", trace = 0)

# Afficher un résumé du modèle sélectionné
summary(model_ACP_13_int)
```

```
##
## Call:
## glm(formula = pluie.demain ~ PC1 + PC2 + PC3 + PC4 + PC5 + PC6 +
##     PC7 + PC8 + PC9 + PC10 + PC11 + PC12 + PC13 + PC1:PC2 + PC1:PC3 +
##     PC1:PC4 + PC1:PC8 + PC1:PC9 + PC1:PC11 + PC1:PC12 + PC2:PC3 +
##     PC2:PC4 + PC2:PC6 + PC2:PC8 + PC2:PC9 + PC2:PC10 + PC2:PC11 +
##     PC2:PC13 + PC3:PC4 + PC3:PC5 + PC3:PC6 + PC3:PC7 + PC3:PC8 +
##     PC3:PC9 + PC3:PC12 + PC3:PC13 + PC4:PC5 + PC4:PC6 + PC4:PC7 +
##     PC4:PC8 + PC4:PC11 + PC4:PC12 + PC4:PC13 + PC5:PC9 + PC6:PC7 +
##     PC6:PC8 + PC6:PC10 + PC6:PC12 + PC7:PC8 + PC7:PC9 + PC7:PC13 +
##     PC8:PC10 + PC9:PC10 + PC9:PC12 + PC10:PC11 + PC10:PC12, family = binomial,
##     data = data_pca_13)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.02153    0.08290   0.260 0.795132
## PC1          0.29526    0.04006   7.370 1.70e-13 ***
## PC2          0.26138    0.06129   4.265 2.00e-05 ***
## PC3         -0.41695    0.04942  -8.437  < 2e-16 ***
## PC4          0.27984    0.07678   3.645 0.000268 ***
## PC5         -0.08405    0.12070  -0.696 0.486207
## PC6          0.40526    0.09018   4.494 7.00e-06 ***
## PC7         -0.10372    0.08147  -1.273 0.202985
## PC8          0.09370    0.15673   0.598 0.549938
## PC9         -1.08322    0.23717  -4.567 4.94e-06 ***
## PC10        -0.02461    0.12913  -0.191 0.848870
## PC11         0.61696    0.22486   2.744 0.006073 **
## PC12        -0.45266    0.15904  -2.846 0.004424 **
## PC13         0.31229    0.14175   2.203 0.027590 *
## PC1:PC2     -0.02481    0.01397  -1.775 0.075863 .
```

```
## PC1:PC3      0.03992    0.01692   2.360 0.018284 *
## PC1:PC4     -0.05029    0.02355  -2.135 0.032724 *
## PC1:PC8     -0.05733    0.03449  -1.662 0.096437 .
## PC1:PC9      0.13223    0.04210   3.141 0.001685 **
## PC1:PC11    -0.07399    0.03760  -1.968 0.049101 *
## PC1:PC12     0.07424    0.04013   1.850 0.064290 .
## PC2:PC3     -0.06003    0.02285  -2.628 0.008598 **
## PC2:PC4     -0.06001    0.02835  -2.117 0.034268 *
## PC2:PC6     -0.06243    0.03421  -1.825 0.067990 .
## PC2:PC8     -0.12233    0.04502  -2.717 0.006579 **
## PC2:PC9      0.20131    0.06981   2.884 0.003932 **
## PC2:PC10    -0.08465    0.04792  -1.767 0.077307 .
## PC2:PC11    -0.17191    0.05639  -3.049 0.002300 **
## PC2:PC13    -0.14482    0.05058  -2.863 0.004196 **
## PC3:PC4     -0.06274    0.03259  -1.925 0.054209 .
## PC3:PC5     -0.08122    0.04454  -1.823 0.068251 .
## PC3:PC6     -0.11049    0.03896  -2.836 0.004565 **
## PC3:PC7     -0.05430    0.03816  -1.423 0.154701
## PC3:PC8      0.07352    0.05102   1.441 0.149572
## PC3:PC9      0.12217    0.06410   1.906 0.056664 .
## PC3:PC12     0.19814    0.05807   3.412 0.000644 ***
## PC3:PC13    -0.14057    0.06084  -2.311 0.020859 *
## PC4:PC5      0.07778    0.05159   1.508 0.131620
## PC4:PC6     -0.07468    0.04355  -1.715 0.086350 .
## PC4:PC7      0.14991    0.04952   3.027 0.002468 **
## PC4:PC8      0.23205    0.07144   3.248 0.001162 **
## PC4:PC11    -0.11292    0.06887  -1.640 0.101095
## PC4:PC12     0.12594    0.06974   1.806 0.070937 .
## PC4:PC13    -0.17422    0.07957  -2.190 0.028556 *
## PC5:PC9     -0.14949    0.08924  -1.675 0.093885 .
## PC6:PC7      0.16335    0.06288   2.598 0.009383 **
## PC6:PC8     -0.21205    0.06666  -3.181 0.001468 **
## PC6:PC10     0.10097    0.06910   1.461 0.143949
## PC6:PC12     0.19048    0.08570   2.223 0.026245 *
## PC7:PC8      0.19040    0.08116   2.346 0.018978 *
## PC7:PC9     -0.21259    0.10466  -2.031 0.042231 *
## PC7:PC13     0.14551    0.09778   1.488 0.136690
## PC8:PC10     0.12567    0.08594   1.462 0.143638
## PC9:PC10    -0.19647    0.12779  -1.537 0.124196
## PC9:PC12    -0.25269    0.12949  -1.951 0.051002 .
## PC10:PC11    0.26558    0.11378   2.334 0.019590 *
## PC10:PC12   -0.18905    0.12831  -1.473 0.140653
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1635.4  on 1179  degrees of freedom
## Residual deviance: 1162.3  on 1123  degrees of freedom
## AIC: 1276.3
##
## Number of Fisher Scoring iterations: 6
```

```r
# Capturer l'horodatage de fin
end_time <- Sys.time()
# Calculer la différence en temps
time_taken <- end_time - start_time
# Afficher le temps de calcul
print(time_taken)
```

```
## Time difference of 4.514441 mins
```

Évaluation de **model_ACP_13_int** par matrice de confusion:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE   430  117
##      TRUE    149  484
##
##                Accuracy : 0.7746
##                  95% CI : (0.7496, 0.7981)
##     No Information Rate : 0.5093
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.5485
##
##  Mcnemar's Test P-Value : 0.05734
##
##             Sensitivity : 0.8053
##             Specificity : 0.7427
##          Pos Pred Value : 0.7646
##          Neg Pred Value : 0.7861
##              Prevalence : 0.5093
##          Detection Rate : 0.4102
##    Detection Prevalence : 0.5364
##       Balanced Accuracy : 0.7740
##
##        'Positive' Class : TRUE
##
```

Évaluation de **model_ACP_13_int** par cross-validation (10 plis):

```
## [1] 0.186 0.185
```

Calculons l'AUC de **model_ACP_13_int**:

```
## Area under the curve: 0.8418
```

Avec 914 bonnes réponses, **roc_model_ACP_13_int** obtient le meilleur score et améliore de 35 points celui de **model_step_BIC** (879). Il obtient la meilleure performance pour tous les critères sauf l'AIC et l'estimation de l'erreur de prédiction par cross-validation (respectivement 1276.3 et 0.185).

ACP avec citère AIC

Appliquons une sélection par la fonction **step()** (critère AIC) sur le dataframe complet des composantes principales:

```
model_ACP_AIC <- step(glm(pluie.demain ~ . , data = data_pca, family = binomial), direction = "both", t
summary(model_ACP_AIC)
```

```
##
## Call:
## glm(formula = pluie.demain ~ PC1 + PC2 + PC3 + PC5 + PC15 + PC17 +
##     PC18 + PC20 + PC24 + PC27 + PC30 + PC36 + PC39 + PC40, family = binomial,
##     data = data_pca)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.02417    0.07041   0.343 0.731412
## PC1          0.24971    0.02224  11.228  < 2e-16 ***
## PC2          0.15977    0.03156   5.063 4.12e-07 ***
## PC3         -0.43067    0.03697 -11.650  < 2e-16 ***
## PC5         -0.11131    0.05268  -2.113 0.034600 *
## PC15        -0.32806    0.10029  -3.271 0.001071 **
## PC17        -0.18732    0.11275  -1.661 0.096623 .
## PC18        -0.26400    0.11728  -2.251 0.024379 *
## PC20         0.26641    0.15937   1.672 0.094597 .
## PC24         0.31724    0.19439   1.632 0.102689
## PC27         0.70583    0.24344   2.899 0.003738 **
## PC30         0.67653    0.29176   2.319 0.020404 *
## PC36        -0.97618    0.51341  -1.901 0.057257 .
## PC39         4.36796    1.26632   3.449 0.000562 ***
## PC40         3.07499    1.44118   2.134 0.032870 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1635.4  on 1179  degrees of freedom
## Residual deviance: 1251.9  on 1165  degrees of freedom
## AIC: 1281.9
##
## Number of Fisher Scoring iterations: 4
```

14 composantes sont retenues.

Il est intéressant de noter que si 4 des 5 premières composantes sont retenues (celles expliquant le mieux la variance), des composantes principales parmi les dernières sont aussi retenues. Notamment PC40 qui a aussi le plus gros coefficient.
Testons ce modèle:

Évaluation de **model_ACP_AIC** par matrice de confusion:

```
## Confusion Matrix and Statistics
##
##           Reference
```

```
## Prediction FALSE TRUE
##      FALSE   412  131
##      TRUE    167  470
##
##                Accuracy : 0.7475
##                  95% CI : (0.7216, 0.772)
##     No Information Rate : 0.5093
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.4942
##
##  Mcnemar's Test P-Value : 0.04261
##
##             Sensitivity : 0.7820
##             Specificity : 0.7116
##          Pos Pred Value : 0.7378
##          Neg Pred Value : 0.7587
##              Prevalence : 0.5093
##          Detection Rate : 0.3983
##    Detection Prevalence : 0.5398
##       Balanced Accuracy : 0.7468
##
##        'Positive' Class : TRUE
##
```

Évaluation de **model_ACP_AIC** par cross-validation (10 plis):

```
## [1] 0.18 0.18
```

Calculons l'AUC de **model_ACP_AIC**:

```
set.seed(100)
# Faire des prédictions de probabilités sur les données d'entraînement (ou de test)
pred_prob <- predict(model_ACP_AIC, newdata = data_pca, type = "response")
# Calculer l'AUC
roc_model_ACP_AIC <- roc(data_pca$pluie.demain, pred_prob)
```

```
## Setting levels: control = FALSE, case = TRUE
```

```
## Setting direction: controls < cases
```

```
# Afficher l'AUC
auc_model_ACP_AIC <- auc(roc_model_ACP_AIC)
print(auc_model_ACP_AIC)
```

```
## Area under the curve: 0.8138
```

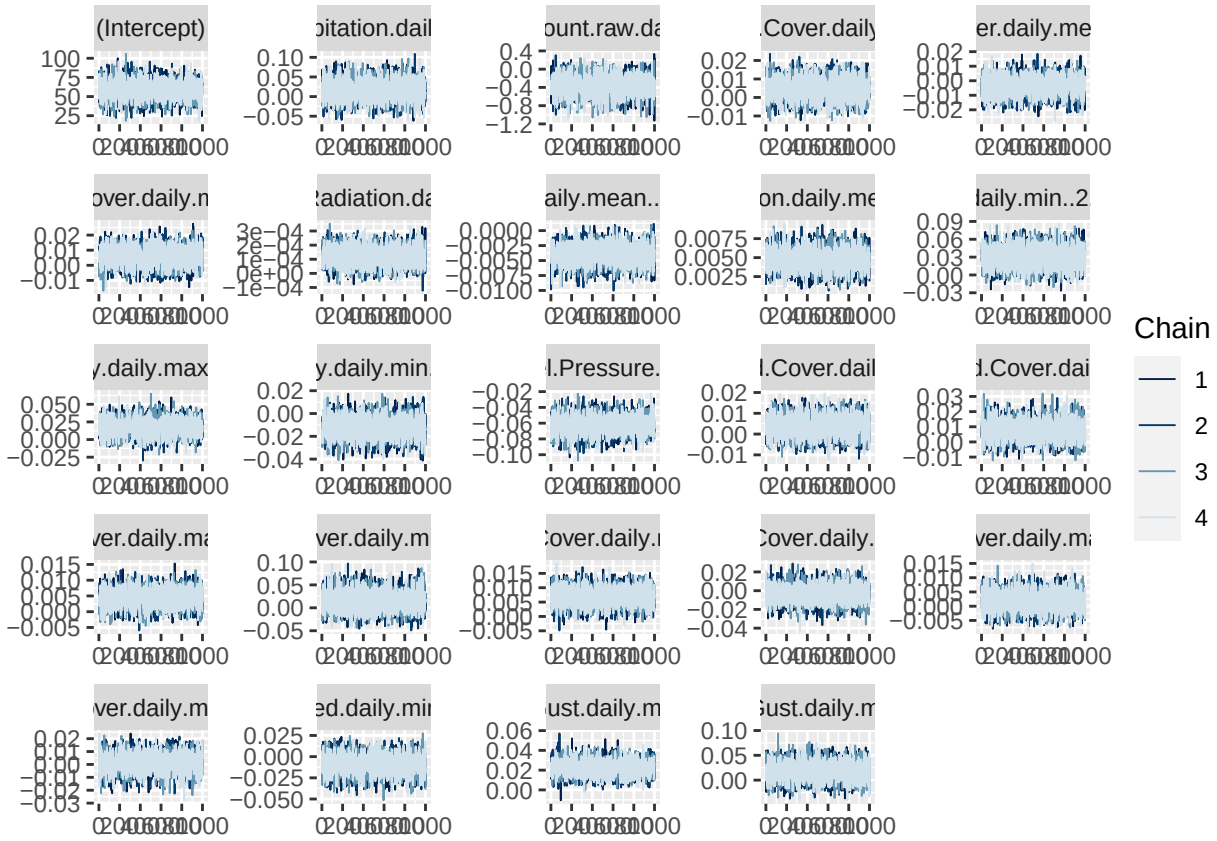
Avec 882 bonnes réponses, **model_ACP_AIC** obtient le deuxième meilleur score. Toutes les performances obtenues par **model_ACP_AIC** sont dans la moyenne haute des autres modèles.

Approche bayesienne

En raison du temps de calcul, nous utiliserons le dataframe **d_train_algo**. Nous laissons la fonction définir les “priors” par défaut.

```
# Capturer l'horodatage de début
start_time <- Sys.time()

# Ajuster le modèle de régression logistique bayésienne
model_bayes <- stan_glm(pluie.demain ~ .,
                        data = d_train_algo,
                        family = binomial(link = "logit"),
                        chains = 4,
                        iter = 2000,
                        seed = 100)

# Capturer l'horodatage de fin
end_time <- Sys.time()
# Calculer la différence en temps
time_taken <- end_time - start_time
# Afficher le temps de calcul
print(time_taken)
```

```
# Tracer les diagnostics du modèle
posterior <- as.array(model_bayes)
color_scheme_set("blue")
mcmc_trace(posterior)
```




```
# Extraire les coefficients
coef(model_bayes)
```

```
##                                    (Intercept)
##                                  58.6928729796
##          Total.Precipitation.daily.sum..sfc.
##                                   0.0156926683
##          Snowfall.amount.raw.daily.sum..sfc.
##                                  -0.3529132698
##           Total.Cloud.Cover.daily.mean..sfc.
##                                   0.0045728627
##   High.Cloud.Cover.daily.mean..high.cld.lay.
##                                  -0.0034591161
## Medium.Cloud.Cover.daily.mean..mid.cld.lay.
##                                   0.0057523289
##          Shortwave.Radiation.daily.sum..sfc.
##                                   0.0001264303
##   Wind.Direction.daily.mean..80.m.above.gnd.
##                                  -0.0039927872
##           Wind.Direction.daily.mean..900.mb.
##                                   0.0048447646
##        Temperature.daily.min..2.m.above.gnd.
##                                   0.0295924125
## Relative.Humidity.daily.max..2.m.above.gnd.
##                                   0.0178202721
## Relative.Humidity.daily.min..2.m.above.gnd.
```

```
##                                -0.0127203330
##     Mean.Sea.Level.Pressure.daily.min..MSL.
##                                -0.0618678955
##           Total.Cloud.Cover.daily.max..sfc.
##                                 0.0037124927
##           Total.Cloud.Cover.daily.min..sfc.
##                                 0.0075413330
##   High.Cloud.Cover.daily.max..high.cld.lay.
##                                 0.0043246948
##   High.Cloud.Cover.daily.min..high.cld.lay.
##                                 0.0128119366
##  Medium.Cloud.Cover.daily.max..mid.cld.lay.
##                                 0.0066567058
##  Medium.Cloud.Cover.daily.min..mid.cld.lay.
##                                -0.0039883634
##     Low.Cloud.Cover.daily.max..low.cld.lay.
##                                 0.0020122014
##     Low.Cloud.Cover.daily.min..low.cld.lay.
##                                 0.0014569107
##               Wind.Speed.daily.min..900.mb.
##                                -0.0095279417
##                   Wind.Gust.daily.max..sfc.
##                                 0.0226401944
##                   Wind.Gust.daily.min..sfc.
##                                 0.0173192480
```

Nous n'utilisons pas le critère AIC pour un modèle bayésien.
D'autres critères propres à cette approche existent (DIC, WAIC ou LOO par exemple) mais ne permettrait pas une comparaison avec nos modèles fréquentistes.

Les traces sont bonnes ce qui indique que les MCMC ont bien convergé (nombre d'itération suffisant), les R-hat à **1** le confirme. Si nous comparons avec les valeurs des coefficients obtenus avec **model_algo** sur le même dataframe, elles sont assez proches. Par exemple concernant la variable **Snowfall.amount.raw.daily.sum..sfc.** il obtient le coefficient **-0.3235** contre **-0.3529132698** pour **model_bayes**.

Évaluation de **model_bayes** par matrice de confusion:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE   405  138
##      TRUE    174  463
##
##                Accuracy : 0.7356
##                  95% CI : (0.7094, 0.7606)
##     No Information Rate : 0.5093
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.4704
##
##  Mcnemar's Test P-Value : 0.04754
##
##             Sensitivity : 0.7704
```

```
##             Specificity : 0.6995
##          Pos Pred Value : 0.7268
##          Neg Pred Value : 0.7459
##              Prevalence : 0.5093
##          Detection Rate : 0.3924
##    Detection Prevalence : 0.5398
##       Balanced Accuracy : 0.7349
##
##        'Positive' Class : TRUE
##
```

Évaluation de **model_bayes** par cross-validation (10 plis):

```
# Sortie cachée car très longue, affichage des résultats par le chunk suivant

# Capturer l'horodatage de début
start_time <- Sys.time()

set.seed(100)
# validation croisée
cv_model <- cv.glm(d_train_algo, model_bayes, K = 10)  # K = nombre de plis pour la validation croisée

# Capturer l'horodatage de fin
end_time <- Sys.time()
# Calculer la différence en temps
time_taken <- end_time - start_time
# Afficher le temps de calcul
print(time_taken)
```

```
# Arrondir les valeurs à trois chiffres après la virgule
delta_rounded <- round(cv_model$delta, 3)

# Afficher les valeurs arrondies
print(delta_rounded)
```

```
## [1] 0.319 0.178
```

Calculons l'AUC de **model_bayes**:

```
## Area under the curve: 0.8005
```

Comparaison des modèles

```
# Définir les noms des modèles et les colonnes
model_names <- c("model_total", "model_step_AIC", "model_step_BIC", "model_algo", "model_select", "model
metric_names <- c("AIC", "Accuracy", "Sensitivity", "Specificity", "erreur_CV", "AUC")

#Matrice des valeurs
metric_values <- matrix(
  c(
```

```
    1321.5, 0.7398, 0.7671, 0.7116, 0.189, 0.8176,
    1285.6, 0.7424, 0.7770, 0.7064, 0.182, 0.8138,
    1292.1, 0.7449, 0.7887, 0.6995, 0.182, 0.8082,
    1308.1, 0.7398, 0.7820, 0.6960, 0.186, 0.8005,
    1315.7, 0.7373, 0.7854, 0.6874, 0.188, 0.7953,
    1247.6, 0.7432, 0.7720, 0.7133, 0.177, 0.8241,
    1321.5, 0.7398, 0.7671, 0.7116, 0.189, 0.8176,
    1276.3, 0.7746, 0.8053, 0.7427, 0.185, 0.8418,
    1281.9, 0.7475, 0.7820, 0.7116, 0.180, 0.8138,
    "XXXX", 0.7356, 0.7704, 0.6995, 0.178, 0.8005),
  nrow = 10,
  byrow = TRUE)

# Convertir la matrice en data.frame
results_df <- data.frame(
  Model = model_names,
  AIC = metric_values[, 1],
  Accuracy = metric_values[, 2],
  Sensitivity = metric_values[, 3],
  Specificity = metric_values[, 4],
  erreur_CV = metric_values[, 5],
  AUC = metric_values[, 6])

# dégradés de couleur à chaque colonne
formattable_df <- formattable(results_df, list(
  AIC = color_tile("deepskyblue", "darksalmon"), # Le plus petit AIC est le plus bleu, le plus grand AI
  Accuracy = color_tile("darksalmon", "deepskyblue"), # Le plus petit Accuracy est le plus rouge, le pl
  Sensitivity = color_tile("darksalmon", "deepskyblue"), # Le plus petit Sensitivity est le plus rouge,
  Specificity = color_tile("darksalmon", "deepskyblue"), # Le plus petit Specificity est le plus rouge,
  erreur_CV = color_tile("deepskyblue", "darksalmon"), # Le plus grand Taux d'erreur CV est le plus rou
  AUC = color_tile("darksalmon", "deepskyblue") # Le plus petit AUC est le plus rouge, le plus grand AU
))

formattable_df
```

Model

AIC

Accuracy

Sensitivity

Specificity

erreur_CV

AUC

model_total

1321.5

0.7398

0.7671

0.7116

0.189

0.8176

model_step_AIC

1285.6

0.7424

0.777

0.7064

0.182

0.8138

model_step_BIC

1292.1

0.7449

0.7887

0.6995

0.182

0.8082

model_algo

1308.1

0.7398

0.782

0.696

0.186

0.8005

model_select

1315.7

0.7373

0.7854

0.6874

0.188

0.7953

model_select_int

1247.6

0.7432

0.772

0.7133

0.177

0.8241

model_ACP_complet

1321.5

0.7398

0.7671

0.7116

0.189

0.8176

model_ACP_13_int

1276.3

0.7746

0.8053

0.7427

0.185

0.8418

model_ACP_AIC

1281.9

0.7475

0.782

0.7116

0.18

0.8138

model_bayes

XXXX

0.7356

0.7704

0.6995

0.178

0.8005

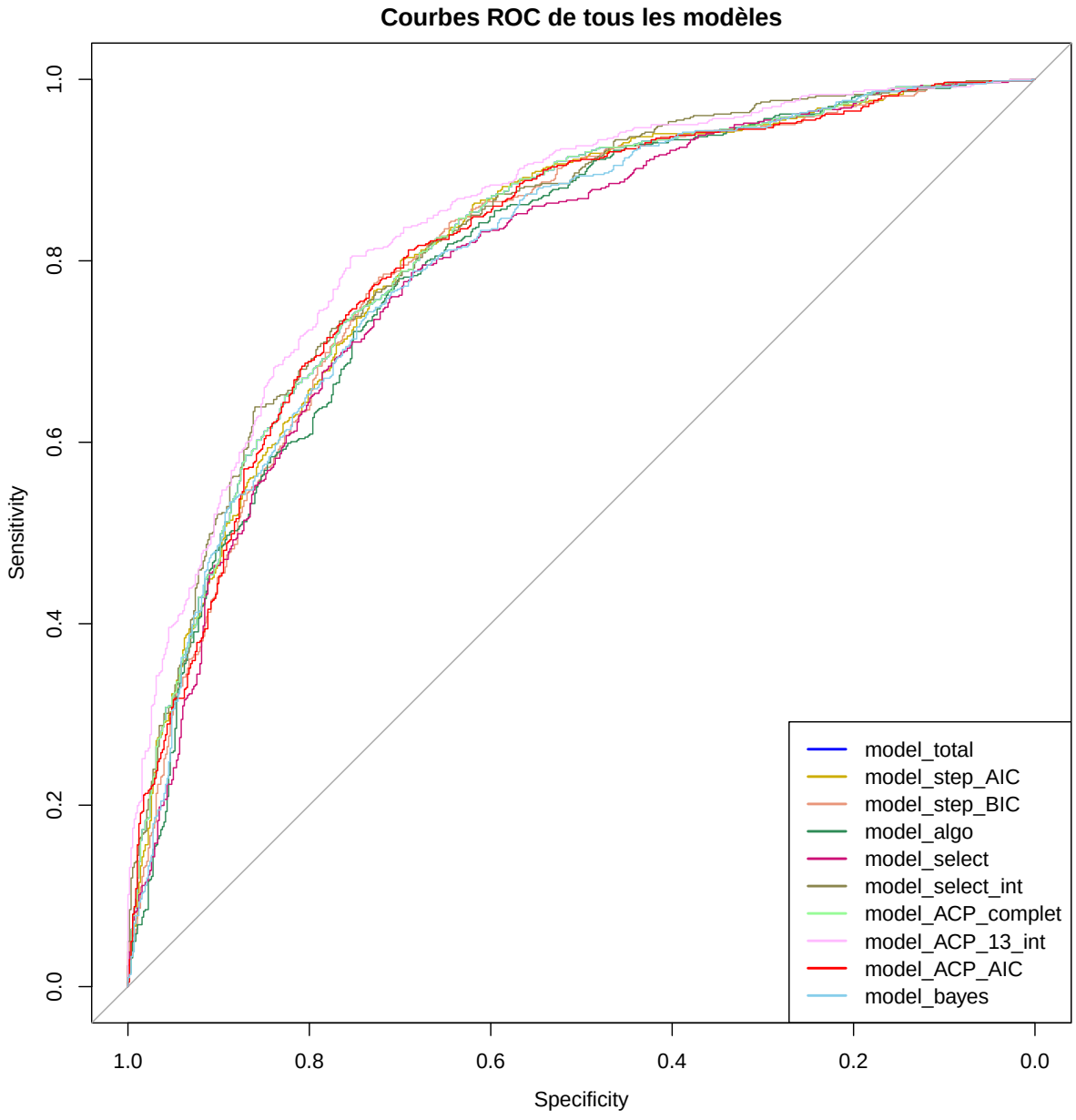
Le tableau confirme que c'est bien **model_ACP_13_int** qui obtient les meilleurs résultats dans l'ensemble.

Courbes ROC

Traçons les courbes ROC de tous les modèles sur un même graphique:

```
plot(roc_model_total, col = "blue", main = "Courbes ROC de tous les modèles", lwd = 1)
plot(roc_model_step_AIC, col = "gold3", add = TRUE, lwd = 1)
plot(roc_model_step_BIC, col = "darksalmon", add = TRUE, lwd = 1)
plot(roc_model_algo, col = "seagreen4", add = TRUE, lwd = 1)
plot(roc_model_select, col = "deeppink3", add = TRUE, lwd = 1)
```

```r
plot(roc_model_select_int, col = "khaki4", add = TRUE, lwd = 1)
plot(roc_model_ACP_complet, col = "palegreen", add = TRUE, lwd = 1)
plot(roc_model_ACP_13_int, col = "plum1", add = TRUE, lwd = 1)
plot(roc_model_ACP_AIC, col = "red", add = TRUE, lwd = 1)
plot(roc_model_bayes, col = "skyblue", add = TRUE, lwd = 1)

legend("bottomright", legend = c("model_total", "model_step_AIC", "model_step_BIC", "model_algo", "model
```



Toutes nos courbes ROC sont assez proches, seule celle d'**model_ACP_13_int** se détache légèrement.

Nous retenons 3 modèles: **model_ACP_13_int** pour ses performances générales, **model_ACP_13_int** pour son faible AIC et **model_step_AIC** pour... la sécurité.

Prédiction sur le jeu de test

Importation des données du jeu de test:

```
d_test <- read_csv("C:/Users/olivi/OneDrive - Université Paris-Dauphine/Documents ODD gram/Formation/Cou
    col_types = cols(...1 = col_skip(), Year = col_skip(),
        Month = col_skip(), Day = col_skip(),
        Hour = col_skip(), Minute = col_skip()))
```

Prédicitions avec model_step_AIC

```
pred_model_step_AIC <- predict(model_step_AIC, newdata = d_test, type = "response") > 0.5
```

Prédicitions avec model_select_int

Commençons par appliquer les mêmes modifications qu'à notre dataframe d'entraînement:

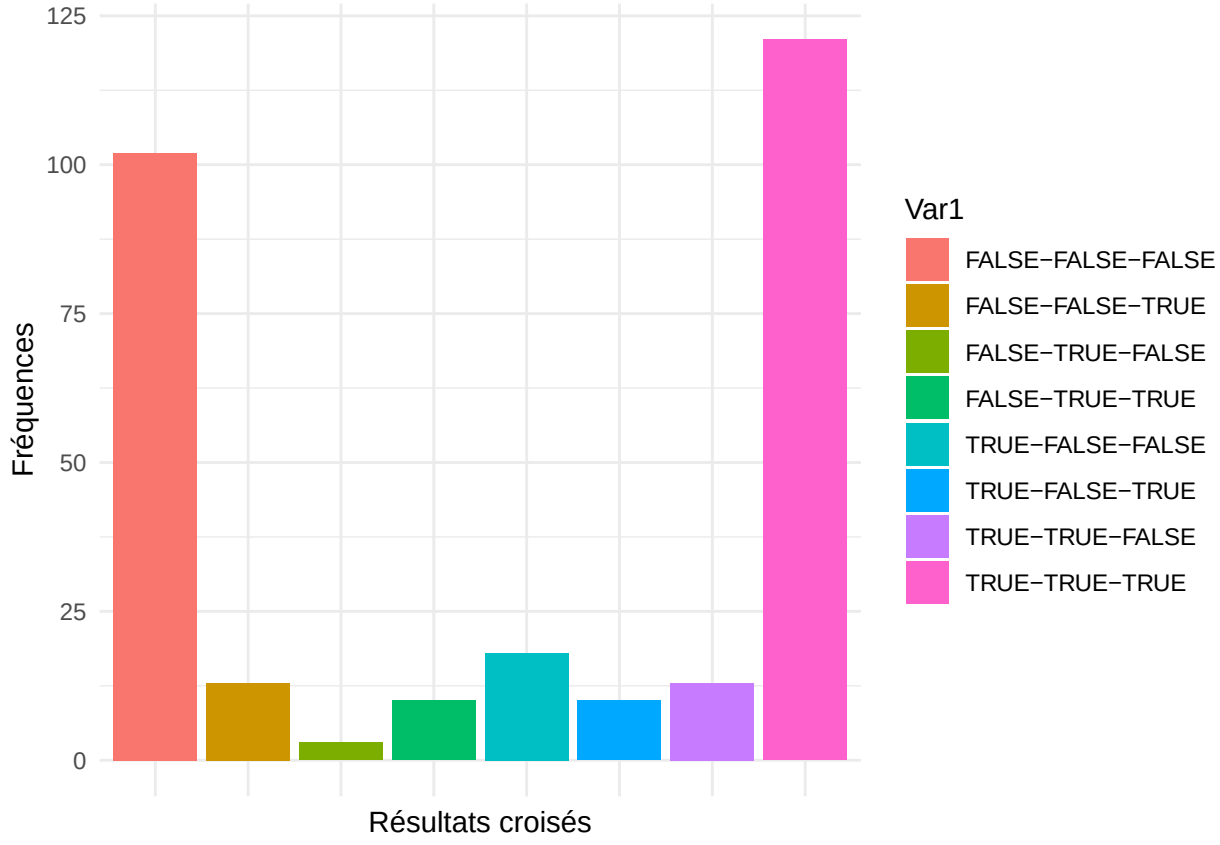
```
d_test_new <- d_test %>%
  mutate(Medium.Cloud = rowMeans(select(., Medium.Cloud.Cover.daily.max..mid.cld.lay., Medium.Cloud.Cove
 select(-Medium.Cloud.Cover.daily.max..mid.cld.lay., -Medium.Cloud.Cover.daily.mean..mid.cld.lay.) %>%
  mutate(Low.Cloud = rowMeans(select(., Low.Cloud.Cover.daily.max..low.cld.lay., Low.Cloud.Cover.daily.
 select(-Low.Cloud.Cover.daily.max..low.cld.lay., -Low.Cloud.Cover.daily.mean..low.cld.lay.)%>%
  mutate(High.Cloud = rowMeans(select(., High.Cloud.Cover.daily.mean..high.cld.lay., High.Cloud.Cover.da
 select(-High.Cloud.Cover.daily.mean..high.cld.lay., -High.Cloud.Cover.daily.max..high.cld.lay.)%>%
  mutate(Total.Cloud = rowMeans(select(., Total.Cloud.Cover.daily.max..sfc., Total.Cloud.Cover.daily.mea
select(-Total.Cloud.Cover.daily.max..sfc., -Total.Cloud.Cover.daily.mean..sfc.)

pred_model_select_int <- predict(model_select_int, newdata = d_test_new, type = "response") > 0.5
```

Prédiciton avec model_ACP_13_int

```
# Transformer les données de test avec PCA en utilisant les mêmes transformations que celles appliquées
X_test_pca <- predict(pca_result, newdata = d_test)

# Sélectionner les 13 premières CP
X_test_pca_13 <- X_test_pca[, 1:13]

# Créer un nouveau data.frame avec les composantes principales
data_test_pca_13 <- data.frame(X_test_pca_13)

# prédictions avec le modèle ACP
pred_model_ACP_13_int <- predict(model_ACP_13_int, newdata = data_test_pca_13, type = "response") > 0.5
```

Comparaisons des prédictions des trois modèles

```r
# data.frame avec les combinaisons de valeurs
df <- data.frame(
  Vector1 = pred_model_step_AIC,
  Vector2 = pred_model_select_int,
  Vector3 = pred_model_ACP_13_int
)
# ajouter une colonne pour les combinaisons
df$comb <- apply(df, 1, function(row) paste(row, collapse = "-"))

# Compter les occurrences de chaque combinaison
counts <- as.data.frame(table(df$comb))

ggplot(counts, aes(x = Var1, y = Freq, fill = Var1)) +
  geom_bar(stat = "identity") +
  labs(x = "Résultats croisés", y = "Fréquences") +
  scale_x_discrete(labels = NULL) +
  theme_minimal()
```



Dans une large majorité des cas, les réponses sont identiques sur les trois modèles.

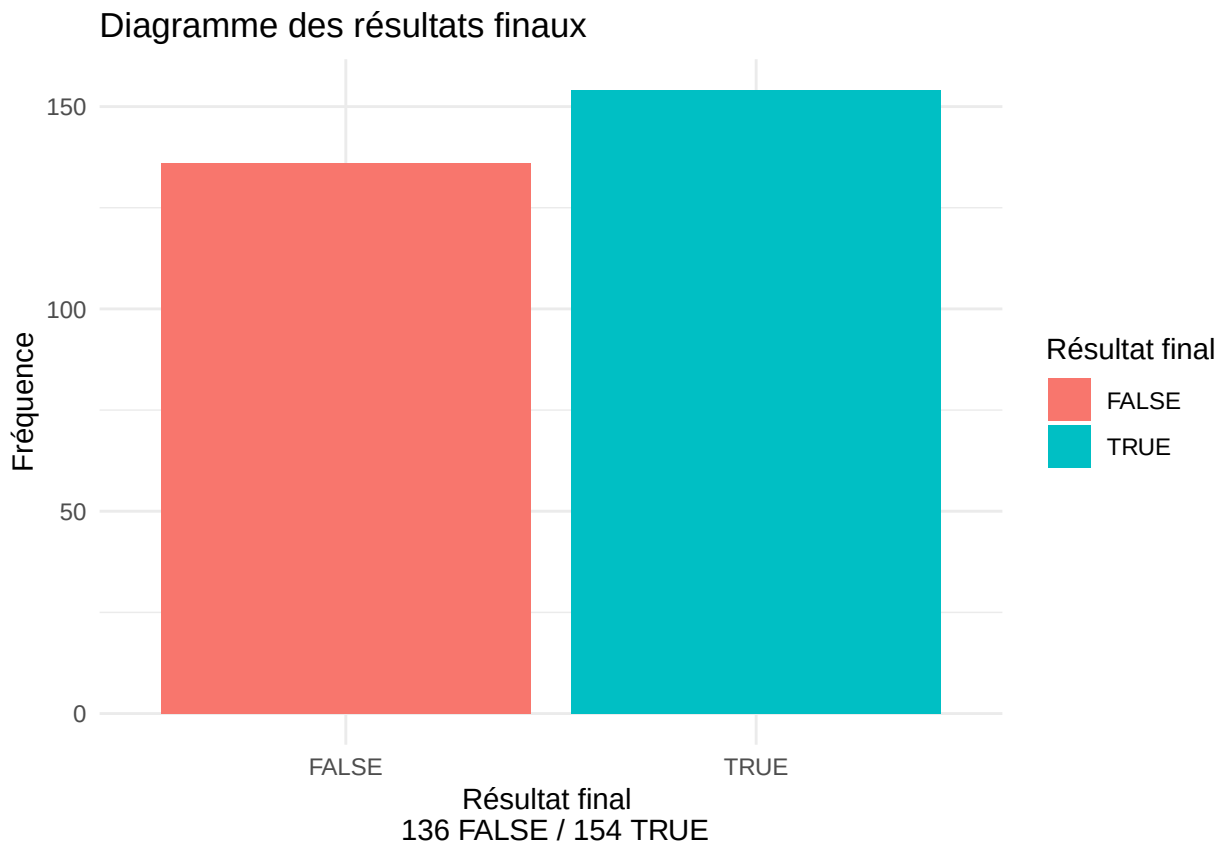
Prédicitons finales

Faisons les moyennes des réponses des trois modèles retenus ainsi:

- si FALSE apparait 3 fois -> réponse FALSE

- si FALSE apparait 2 fois -> réponse FALSE
- si TRUE apparait 3 fois -> réponse TRUE
- si TRUE apparait 2 fois -> réponse TRUE

```r
# matrice avec les vecteurs
mat <- cbind(pred_model_step_AIC, pred_model_select_int, pred_model_ACP_13_int)

# le vecteur combiné
pred_model_final <- rowSums(mat) == 2 | rowSums(mat) == 3

# Compte les occurrences de TRUE et FALSE
count_true <- sum(pred_model_final)
count_false <- length(pred_model_final) - count_true

# data frame avec le vecteur combiné
df_comb <- data.frame(pred_model_final)

# graphique
ggplot(df_comb, aes(x = factor(pred_model_final), fill = factor(pred_model_final))) +
  geom_bar(stat = "count") +
  labs(x = paste("Résultat final \n", count_false, "FALSE /", count_true, "TRUE"),
       y = "Fréquence", fill = "Résultat final") +
  ggtitle("Diagramme des résultats finaux") +
  theme_minimal()
```



Notre prédiction finale est de 154 jours avec pluie, 136 sans pluie.

Ajoutons nos prédictons dans un nouveau dataframe **d_test_pred** et créons un fichier **"meteo.test_pred.csv"**.

```
d_test_pred <- read_csv("C:/Users/olivi/OneDrive - Université Paris-Dauphine/Documents ODD gram/Formati
d_test_pred$pluie.demain <- pred_model_final
write.csv(d_test_pred, file = "meteo.test_pred.csv", row.names = FALSE)
```

Conclusion

Nous avons essayé plusieurs approches et nous constatons des différences de performances mais elles restent globalement faibles (0.7746 de précison pour le plus performant contre 0.7356 pour le moins performant). Nous avons pu constater l'importance d'incorporer les interactions dans les calculs (le "meilleur" modèle inclus des interactions). D'autres modèles pourraient être developpés en mixant plusieurs approches: comme appliquer le critère BIC sur les composantes principales par exemple. Une méthode efficace mais très couteuse en calculs et en temps d'analyse des résultat serait de faire une sélection de variables (par critère AIC par exemple) en incorporant toutes les interactions de toutes les variables. Cela maximiserait l'utilisation de l'information contenue dans le jeu de données. La meilleure démarche à adopter dépend de l'objectif que l'on se poursuit, ici nous avons privilégié la qualité des prédictions sans volonté explicative. Si on souhaitait au contraire comprendre pourquoi il pleuvra ou non le jour suivant et non plus chercher uniquement à le prédire, on privilégierait l'analyse des variables retenues et leur coefficient.
Autre remarque, nous avons choisi comme seuil la probabilité 0.5. Selon l'intentionnalité, on pourrait retenir d'autres valeurs (si par exemple on souhaitait prévoir l'opportunité d'une course en montagne le lendemain, nous chercherions surtout à éviter des faux négatifs et choisirions un seuil plus bas).

Pour l'approche bayesienne, Nous avons laissé la fonction définir des "priors" par défaut. On aurait pu définir les priors en fonction de la distribution observée des variables. Dans le cas d'une analyse similaire mais effectuée par un expert de la discipline que les données décrivent, l'approche bayesienne gagnerait d'autant en pertinence, les priors pourraient être définis en fonction des connaissances à priori de l'expert en question sur chaque variable disponible. Un modèle bayesien pourrait aussi sans doute obtenir des performances supérieures à une approche fréquentiste classique dans le cas d'un jeu de données limité en nombre d'observations (là encore, à condition de bien définir les priors).

Autre approche non développée ici car contraire au cahier des charges: nous aurions pu intégrer les variables de date dans la régression (mois, années et la numérotaion). Par curiosité, nous avons commencé à développer des modèles allant dans ce sens et les résultats étaient très prometteurs.

Nous avons essayé un modèle **probit** mais ne l'avons pas présenté ici, les performances obtenues étant en tout point comparables à celle du modèle **logit**.

Récapitulatif des dataframes utilisés:

- **d_train** : jeu de données d'entraînement sans les variables inutilisées (**Year**, **Month**, **Day**, **Hour** et **Minute**), soit 40 variables explicatives.
- **d_train_algo** : jeu de données avec les variables sélectionnées par colinéarité, p-valeurs et ANOVA succéssives, soit 23 variables explicatives.
- **d_train_new** : datframe incluant les variables modifiées "à la main".
- **data_pca** : dataframe de toutes les composantes principales, soit 40 CP.
- **data_pca_13** : dataframe des 13 premières composantes principales.

Récapitulatif des modèles utilisés:

- **model_total** : modèle complet avec toutes les données (dataframe **d_train**).

- **model_step_AIC** : modèle avec les variables sélectionnées par la fonction **step()** selon le critère **AIC** sur le dataframe **d_train**.
- **model_step_BIC** : modèle avec les variables sélectionnées par la fonction **step()** selon le critère **BIC** sur le dataframe **d_train**.
- **model_algo** : modèle sur **d_train_algo**.

- **model_select** : Modèle avec variables sélectionnées “à la main”.

- **model_select_int** : même variable que **model_select** incluant les intercations, sélection via **step()** critère AIC.

- **model_ACP_complet** : modèle avec toutes les composantes principales sur **data_pca**.

- **model_ACP_13_int** : modèle avec les 13 premières CP sur **data_pca** avec intéractions, sélection par la fonction **step()** critère **AIC**.
- **model_ACP_step** : modèle avec les CP sélectionnées par la fonction **step()** selon le critère **AIC** sur le dataframe **data_pca**.
- **model_bayes** : modèle bayésien sur **d_train_algo**.