

MÉTÉO

Anthony LEZIN

7/18/2020

I. La température (modèle explicatif)

Je dispose d'un certain nombre de relevés de diverses observations (température, pression, nébulosité, vent, etc..). L'objectif de cette partie est d'expliquer la variable d'intérêt *température du lendemain* en fonction des observations.

Quelques informations potentiellement utiles sur Bâle pour mieux apprécier les données (Hauteur : Bâle : 268 m Min. 244 m Max. 363 m).

A. Préparation des données

```
met0=read.csv("/Users/anthonylezin/Desktop/Projets Stats/Projet GLM/meteo.train.csv",header=TRUE,sep=",")  
  
Après avoir visualisé sommairement les données, je décide de renommer les variables.  
nom=data.frame("X","Year","Month","Day","Hour","Minute","Tmoy2","Hmoy2","Prmoy","Plmoy","Snow","TNebmoy")  
  
#suppression des colonnes inutiles  
met1=met0[,-c(1,5,6)]  
nom1=nom[,-c(1,5,6)]  
  
#équivalences entre anciens noms et nouveaux noms  
equivalences = cbind(names(met1),t(nom1))  
rownames(equivalences)=c()
```

voici le tableau des équivalences pour les noms des variables :

```
kable(equivalences)
```

Year	Year
Month	Month
Day	Day
Temperature.daily.mean..2.m.above.gnd.	Tmoy2
Relative.Humidity.daily.mean..2.m.above.gnd.	Hmoy2
Mean.Sea.Level.Pressure.daily.mean..MSL.	Prmoy
Total.Precipitation.daily.sum..sfc.	Plmoy
Snowfall.amount.raw.daily.sum..sfc.	Snow
Total.Cloud.Cover.daily.mean..sfc.	TNebmoy
High.Cloud.Cover.daily.mean..high.cld.lay.	HNebmoy
Medium.Cloud.Cover.daily.mean..mid.cld.lay.	MNebmoy
Low.Cloud.Cover.daily.mean..low.cld.lay.	LNebmoy
Sunshine.Duration.daily.sum..sfc.	Sun
Shortwave.Radiation.daily.sum..sfc.	Ray
Wind.Speed.daily.mean..10.m.above.gnd.	Wsmoy10

Wind.Direction.daily.mean..10.m.above.gnd.	Wdmoy10
Wind.Speed.daily.mean..80.m.above.gnd.	Wsmoy80
Wind.Direction.daily.mean..80.m.above.gnd.	Wdmoy80
Wind.Speed.daily.mean..900.mb.	Wsmoy900
Wind.Direction.daily.mean..900.mb.	Wdmoy900
Wind.Gust.daily.mean..sfc.	Wgmoy
Temperature.daily.max..2.m.above.gnd.	Tmax2
Temperature.daily.min..2.m.above.gnd.	Tmin2
Relative.Humidity.daily.max..2.m.above.gnd.	Hmax2
Relative.Humidity.daily.min..2.m.above.gnd.	Hmin2
Mean.Sea.Level.Pressure.daily.max..MSL.	Prmax
Mean.Sea.Level.Pressure.daily.min..MSL.	Prmin
Total.Cloud.Cover.daily.max..sfc.	TNebmax
Total.Cloud.Cover.daily.min..sfc.	TNebmin
High.Cloud.Cover.daily.max..high.cld.lay.	HNebmax
High.Cloud.Cover.daily.min..high.cld.lay.	HNebmin
Medium.Cloud.Cover.daily.max..mid.cld.lay.	MNebmax
Medium.Cloud.Cover.daily.min..mid.cld.lay.	MNebmin
Low.Cloud.Cover.daily.max..low.cld.lay.	LNebmax
Low.Cloud.Cover.daily.min..low.cld.lay.	LNebmin
Wind.Speed.daily.max..10.m.above.gnd.	Wsmax10
Wind.Speed.daily.min..10.m.above.gnd.	Wsmin10
Wind.Speed.daily.max..80.m.above.gnd.	Wsmax80
Wind.Speed.daily.min..80.m.above.gnd.	Wsmin80
Wind.Speed.daily.max..900.mb.	Wsmax900
Wind.Speed.daily.min..900.mb.	Wsmin900
Wind.Gust.daily.max..sfc.	Wgmax
Wind.Gust.daily.min..sfc.	Wgmin
pluie.demain	pluie.demain
temp.demain	temp.demain

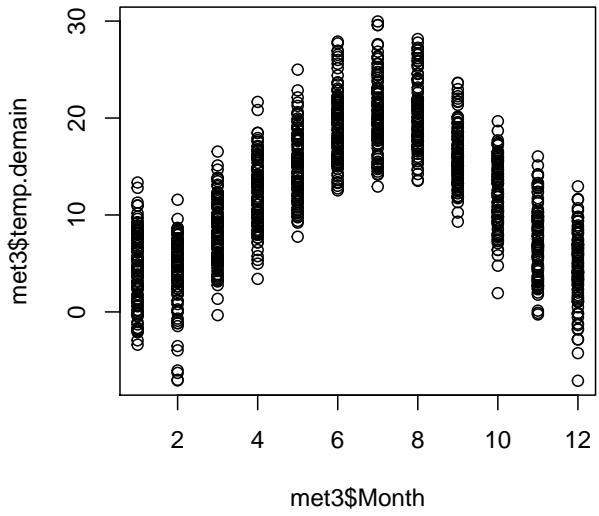
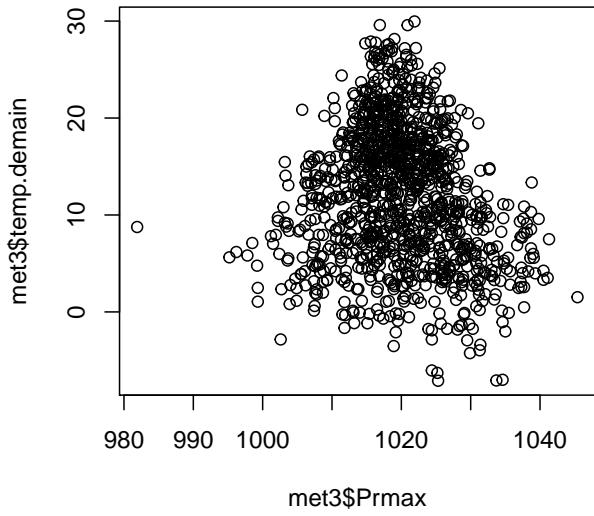
Je renomme les variables dans un nouveau tableau et réarrange les colonnes par catégorie.

```
#je renomme les varaiables
met2=met1
names(met2)=as.vector(nom1)

#je réarrange l'ordre des variables pour regrouper celles de même catégorie
met3 <- met2[, c(1:4,23,22,5,25,24,6,27,26,7:9,29,28,10,31,30,11,33,32,12,35,34,13:15,37,36,16,17,39,38)]
```

B. Premières visualisation de quelques dépendances potentielles

```
par(mfrow=c(1,2))
plot(met3$temp.demain~met3$Prmax)
plot(met3$temp.demain~met3$Month)
```

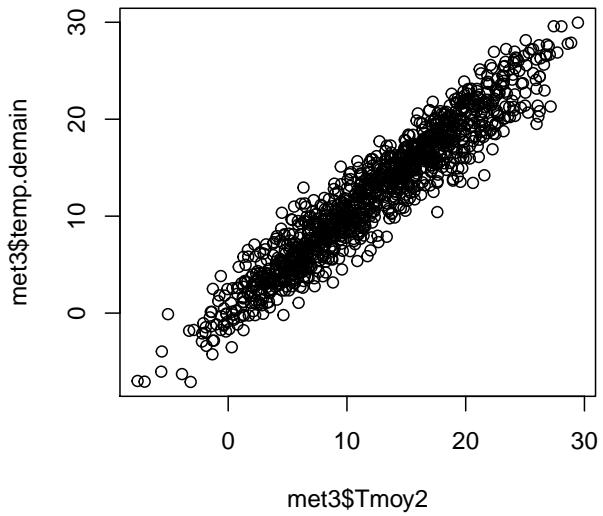
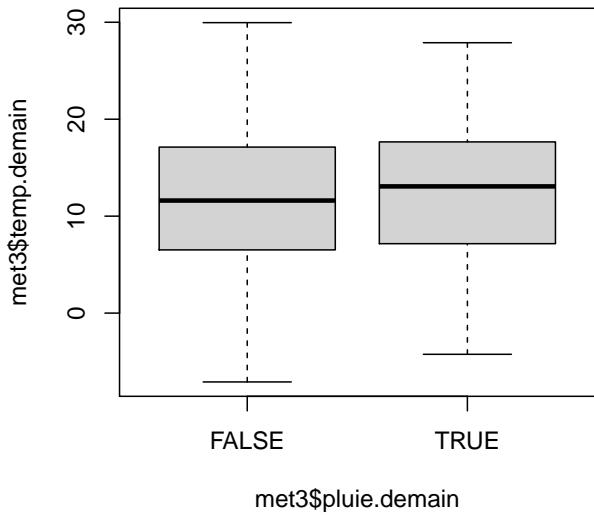


La covariable *temp.demain* semble liée (non linéairement) à la pression maximale de la veille. En effet, j'observe que la température est :

- croissante pour une pression allant de $1000mB$ à $1020mB$
- décroissante pour une pression allant de $1020mB$ à $1030mB$

Le second graphique montre que *temp.demain* varie également selon les mois de l'année. Il y a un effet "saisonnalité".

```
par(mfrow=c(1,2))
boxplot(met3$temp.demain~met3$pluie.demain)
plot(met3$temp.demain~met3$Tmoy2)
```

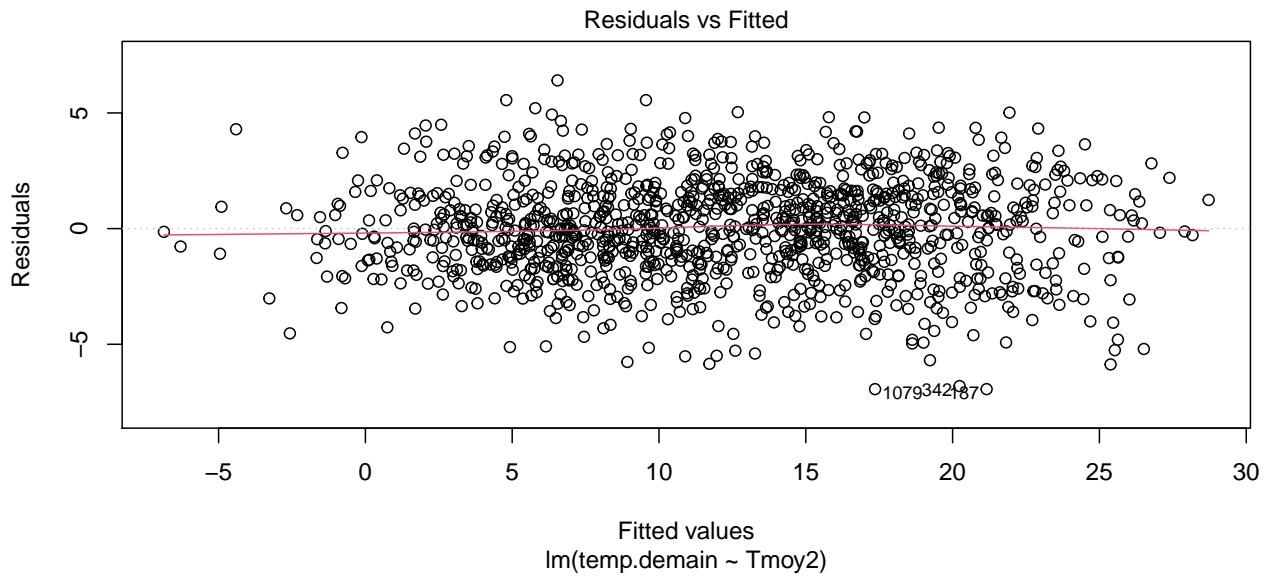


Le fait de prévoir de la pluie le lendemain semble peu influer sur *temp.demain*.

En revanche, la température moyenne du jour semble linéairement très influente sur la covariable *temp.demain*, car le nuage de points est resserré autour d'une droite.

Je le vérifie sur une régression locale :

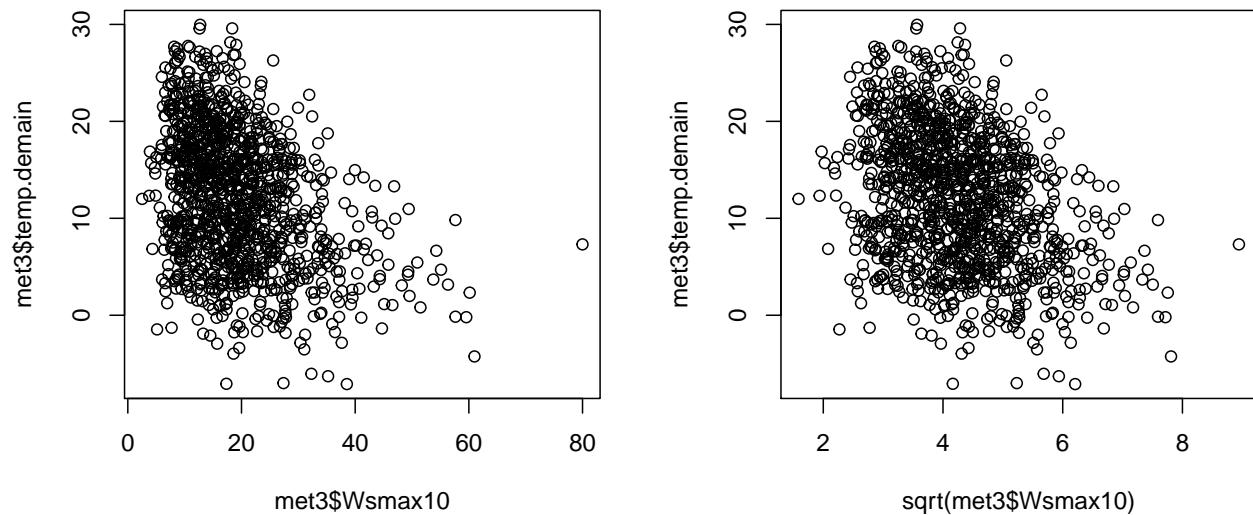
```
plot(lm(temp.demain~Tmoy2, data=met3), which=1)
```



La courbe est très proche de la droite d'abscisse 0 et la répartition des résidus de part et d'autre de la courbe est excellente.

Je cherche désormais à visualiser une éventuelle relation entre la covariable *temp.demain* et la vitesse du vent (à 10m).

```
par(mfrow=c(1,2))
plot(met3$temp.demain~met3$Wsmax10)
# un "essai exotique"
plot(met3$temp.demain~sqrt(met3$Wsmax10))
```



la transformation avec "✓" entraîne un "étalement du nuage" qui pourrait être intéressant.

C. Sélection d'un premier modèle

```
temp=lm(temp.demain~., data=met3)
summary(temp)
```

1. Le modèle global

```

##
## Call:
## lm(formula = temp.demain ~ ., data = met3)
##
## Residuals:
##      Min       1Q   Median      3Q      Max
## -5.7078 -1.1781  0.0198  1.1403  6.6058
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -1.073e+02 5.406e+01 -1.984 0.047492 *  
## Year         4.037e-02 2.672e-02  1.511 0.131112    
## Month        6.787e-02 1.862e-02  3.645 0.000280 *** 
## Day          -9.807e-03 6.260e-03 -1.567 0.117481    
## Tmoy2         9.712e-01 1.210e-01  8.027 2.48e-15 *** 
## Tmin2        -2.528e-01 6.491e-02 -3.895 0.000104 *** 
## Tmax2         1.160e-01 6.984e-02  1.661 0.096904 .  
## Hmoy2         2.325e-02 2.457e-02  0.946 0.344204    
## Hmin2         1.281e-02 1.376e-02  0.930 0.352319    
## Hmax2         -2.048e-02 1.570e-02 -1.304 0.192423    
## Prmoy         1.755e-01 9.292e-02  1.889 0.059135 .  
## Prmin         -4.247e-02 4.894e-02 -0.868 0.385681    
## Prmax         -1.085e-01 5.241e-02 -2.071 0.038592 *  
## Plmoy         2.880e-02 1.917e-02  1.503 0.133211    
## Snow          -1.999e-01 1.588e-01 -1.259 0.208170    
## TNebmoy       -1.042e-02 9.258e-03 -1.125 0.260752    
## TNebmin       -1.095e-03 4.532e-03 -0.242 0.809158    
## TNebmax       -4.268e-03 3.688e-03 -1.157 0.247391    
## HNebmoy       6.184e-03 5.144e-03  1.202 0.229545    
## HNebmin       -1.111e-02 1.299e-02 -0.855 0.392565    
## HNebmax       -2.719e-03 2.313e-03 -1.175 0.240050    
## MNebmoy       3.042e-03 5.261e-03  0.578 0.563201    
## MNebmin       1.570e-03 6.311e-03  0.249 0.803581    
## MNebmax       1.939e-03 2.511e-03  0.772 0.440163    
## LNebmoy       1.211e-02 6.293e-03  1.925 0.054492 .  
## LNebmin       9.421e-04 5.103e-03  0.185 0.853572    
## LNebmax       -8.051e-04 2.784e-03 -0.289 0.772534    
## Sun           -1.230e-03 6.910e-04 -1.780 0.075339 .  
## Ray           5.245e-04 7.627e-05  6.877 1.01e-11 *** 
## Wsmoy10       2.465e-02 7.367e-02  0.335 0.737932    
## Wsmin10       -2.274e-02 4.840e-02 -0.470 0.638606    
## Wsmax10       -6.002e-02 2.653e-02 -2.262 0.023877 *  
## Wdmoy10       -1.838e-03 4.437e-03 -0.414 0.678764    
## Wsmoy80       3.509e-02 5.273e-02  0.665 0.505870    
## Wsmin80       2.785e-02 3.276e-02  0.850 0.395433    
## Wsmax80       -2.338e-02 2.183e-02 -1.071 0.284437    
## Wdmoy80       -1.549e-04 4.579e-03 -0.034 0.973015    
## Wsmoy900      4.811e-02 1.894e-02  2.541 0.011196 *  
## Wsmin900      -4.184e-02 1.388e-02 -3.013 0.002643 ** 
## Wsmax900      -1.022e-02 9.019e-03 -1.133 0.257594    
## Wdmoy900      3.521e-03 1.115e-03  3.156 0.001640 ** 
## Wgmoy         4.156e-03 2.735e-02  0.152 0.879223    
## Wgmin         1.190e-03 2.070e-02  0.057 0.954163    
## Wgmax         -3.222e-03 1.290e-02 -0.250 0.802734

```

```

## pluie.demainTRUE -4.622e-01 1.285e-01 -3.597 0.000335 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.857 on 1135 degrees of freedom
## Multiple R-squared: 0.9321, Adjusted R-squared: 0.9294
## F-statistic: 354 on 44 and 1135 DF, p-value: < 2.2e-16
AIC(temp)

## [1] 4854.957

```

2. Interprétation des coefficients

- le coefficient de *Tmoy2* est positif (“0.971”) et fortement significatif. On retrouve bien le fait que *temp.demain* et *Tmoy2* sont “fortement liées”. Plus précisemment, quand la température moyenne augmente d’une unité (donc de 1°C), la température du lendemain augmente de 0.971 °C.
- c'est en revanche, le phénomène inverse pour *Wsmin900* puisque, cette fois-ci, le coefficient est négatif (“-0.0418”). Plus précisemment, quand la la vitesse minimale du vent (à une pression de 900 mB) augmente d’une unité, la température du lendemain diminue de -0.0418 °C.
- C'est un peu différent pour la covariable *pluie.demain* de part son caractère qualitatif et ses différentes modalités (“FALSE, TRUE”). Le coefficient de “*pluie.demainTRUE*” est un **effet différentiel** par rapport à la cellule de référence “*pluie.demainFALSE*” **représentée par l’intercept** (il suffit d’enlever l’intercept du modèle pour vérifier la concordance des coefficients).

Ainsi, prévoir de la pluie demain fait baisser la température du lendemain de 0.462 °C (par rapport au fait de prévoir une absence de pluie).

3. Un algorithme de sélection de modèle Beaucoup de variables ne sont pas significatives. J'élimine les moins pertinentes de manière progressive en me “laissant de la marge”.

Pour cela, j'effectue la procédure suivante :

- j’ôte que les covariables du modèle dont **la p-valeur est supérieure à 0,20**.
- j’effectue une régression linéaire sur les covariables restantes

J’obtiens successivement les modèles *temp1*, *temp2*, *temp3* (je synthétiserai les résultats essentiels dans un tableau).

```

temp1=lm(temp.demain~ Year+Month+Day+Tmoy2+Tmin2+Tmax2+Hmax2+Prmoy+Plmoy+H Nebmax+L Nebmoy+Sun+Ray+Wsmax10+Wsmin900+Wdmoy900+pluie.demain,data=met3)

```

```

temp2=lm(temp.demain~Month+Day+Tmoy2+Tmin2+Prmoy+Plmoy+H Nebmax+L Nebmoy+Ray+Wsmax10+Wsmin900+Wdmoy900+pluie.demain,data=met3)

```

```

temp3=lm(temp.demain~Month+Day+Tmoy2+Tmin2+Prmoy+Plmoy+L Nebmoy+Ray+Wsmax10+Wsmin900+Wdmoy900+pluie.demain,data=met3)

```

Il n'y a maintenant plus de covariables vérifiant mon critère de suppression.

Je l'adapte en supprimant désormais celles de **p-valeur supérieure à 0,10**.

```

temp4=lm(temp.demain~Month+Day+Tmoy2+Tmin2+Prmoy+Plmoy+L Nebmoy+Ray+Wsmax10+Wsmin900+Wdmoy900+pluie.demain,data=met3)

```

Seule la covariable *Day* est à la limite de la significativité. Je choisis de la conserver pour l'instant. Le test global de Fisher fournit une p-valeur très proche de 0 rejetant logiquement l'hypothèse de l'inutilité de la régression (et ce quelque soit les modèles testés).

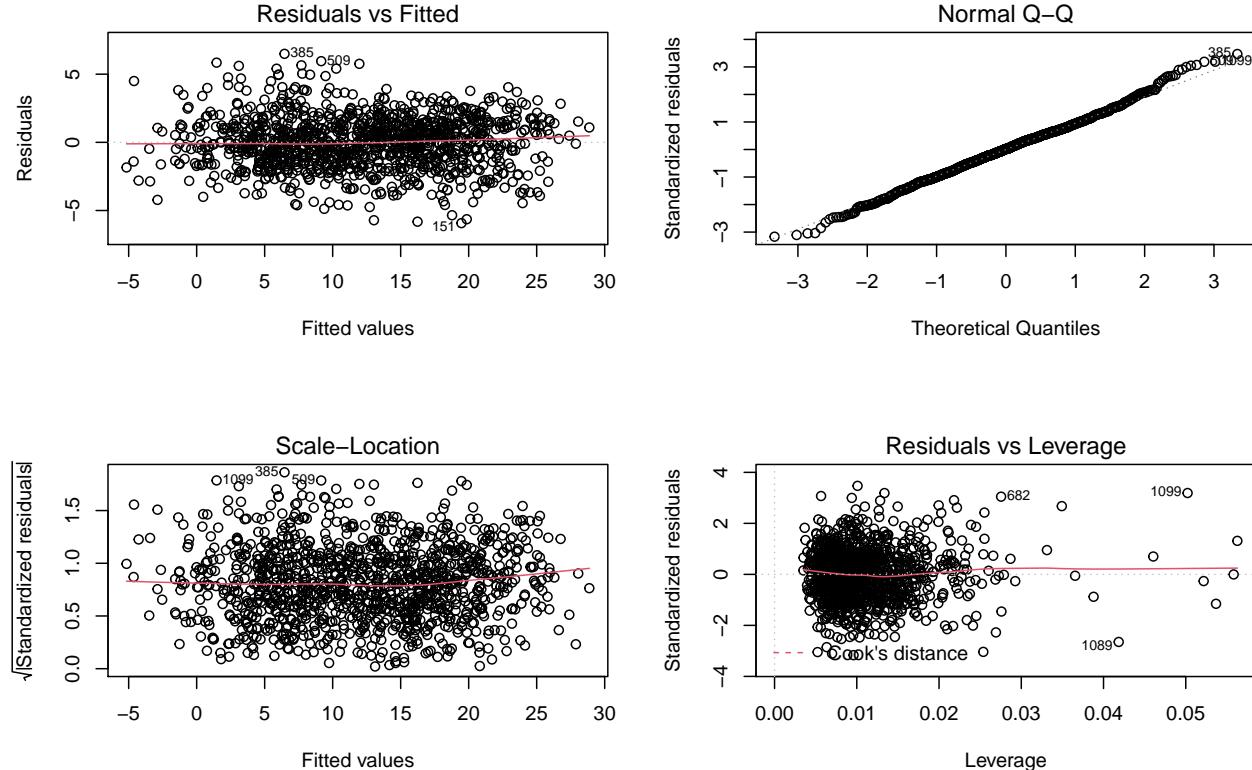
Pour résumer l'étude faite précédemment, voici un tableau contenant les covariables intervenant dans chacun des modèles ainsi que leurs p-valeurs respectives :

covariables	<i>temp1</i>	<i>temp2</i>	<i>temp3</i>	<i>temp4</i>
(Intercept)	6.52e-02	1.95e-03	1.36e-03	2.33e-03
Year	1.86e-01			
Month	1.52e-04	7.35e-04	6.08e-04	5.90e-04
Day	8.37e-02	7.42e-02	7.08e-02	6.61e-02
Tmoy2	1.28e-19	6.07e-96	6.14e-97	4.85e-100
Tmin2	4.63e-05	3.64e-09	4.19e-09	6.18e-10
Tmax2	2.63e-01			
Hmax2	5.85e-01			
Prmoy	2.46e-03	2.45e-03	1.76e-03	2.88e-03
Plmoy	6.43e-03	1.44e-02	1.56e-02	1.00e-02
HNebmax	2.12e-01	5.31e-01		
LNebmoy	1.62e-01	7.41e-03	6.79e-03	7.69e-03
Sun	1.29e-01			
Ray	1.77e-10	9.77e-11	6.84e-12	2.69e-11
Wsmax10	3.28e-10	8.28e-10	7.12e-10	1.75e-09
Wsmoy900	3.52e-03	3.24e-03	3.63e-03	1.33e-02
Wsmin900	7.53e-02	6.34e-02	7.07e-02	
Wdmoy900	3.00e-03	4.39e-03	4.88e-03	4.52e-03
pluie.demainTRUE	1.65e-04	3.27e-04	1.71e-04	2.35e-04

Le modèle *temp4* commence à être satisfaisant. J'étudie maintenant la structure de ses résidus.

D. Structure des résidus du modèle *temp4*

```
par(mfrow=c(2,2))
plot(temp4)
```



- l'hypothèse de linéarité du modèle (“residuals vs fitted plot”) est acceptable. En effet, lorsque les réponses prédictes par le modèle (fitted values) augmentent, les résidus restent globalement uniformément distribués de part et d'autre de 0. Cela montre, qu'en moyenne, la droite de régression est bien adaptée aux données.
- le QQ-plot traduit une normalité qui est assez bien respectée (sauf en quelques “points leviers”)
- l'hypothèse d'homogénéité des résidus (“scale location”) est acceptée. En effet, la courbe rouge (courbe de regression locale) est globalement plate. Les résidus ont tendance à être répartis de façon homogène tout le long du gradient des valeurs prédictes de *temp.demain*.
- pour les distances de Cook, les données 682, 1089 1099 semblent être des points levier. Néanmoins, les distances sont toutes inférieures à 0.5. L'influence de ces valeurs sur les paramètres du modèle n'est pas vraiment problématique.

La structure générale des résidus m'indique que le modèle de régression linéaire *temp4* est légitime.

E. Amélioration potentielle du modèle sans interaction

Que se passe-t-il si j'enlève la covariable *Day* ? (pour rappel, elle était à la limite de la significativité)

```
temp5=lm(temp.demain~Month+Tmoy2+Tmin2+Prmoy+Plmoy+L Nebmoy+Ray+Wsmoy900+Wdmoy900+ pluie.demain,
```

- toutes les covariables présentes sont significatives
- ce modèle contient 11 covariables
- la structure des résidus est sensiblement identique au modèle précédent

```

R_AIC=function(A){
  return(c(as.numeric(summary(A)$"adj.r.squared"),AIC(A)))
}

temp_list=paste('temp',1:5,sep=' ')
R_tab=NULL
for (i in 1:5){
R_tab=t(c(R_tab,R_AIC(get(temp_list[i]))))

R_tab=t(matrix(R_tab,ncol=5,nrow=2,byrow = F))
rownames(R_tab)=temp_list
colnames(R_tab)=c("R^2_adj","AIC")
kable(R_tab)

```

Comparaison des modèles précédents en fonction de R_{adj}^2 et de AIC

	R^2_adj	AIC
temp1	0.9278796	4855.653
temp2	0.9277388	4854.012
temp3	0.9277765	4852.409
temp4	0.9276359	4853.716
temp5	0.9274881	4855.133

Au vu de ces 2 critères, je choisis temp4 comme **premier modèle de référence** (sans interaction). En effet,

- l'AIC est parmi les plus basses
- le nombre de covariables est réduit
- le R_{adj}^2 (0.930) est sensiblement identique à tous les modèles

J'ai pris le parti de ne pas privilégier son concurrent direct (temp5), car la covariable enlevée est à la limite de la significativité (covariable que je souhaite conserver pour l'instant).

F. Procédure automatique de selection de modèle

Voici les codes respectifs pour les méthodes ascendantes, descendantes et Stepwise.

```

library(MASS)
modselect_f=stepAIC(temp,temp.deman~.,data=
met3,trace=TRUE,direction=c("forward"))
summary(modselect_f)
AIC(modselect_f)

modselect_f=stepAIC(temp,temp.deman~.,data=
met3,trace=TRUE,direction=c("backward"))
summary(modselect_f)
AIC(modselect_f)

modselect_f=stepAIC(temp,temp.deman~.,data=
met3,trace=TRUE,direction=c("both"))
summary(modselect_f)
AIC(modselect_f)

```

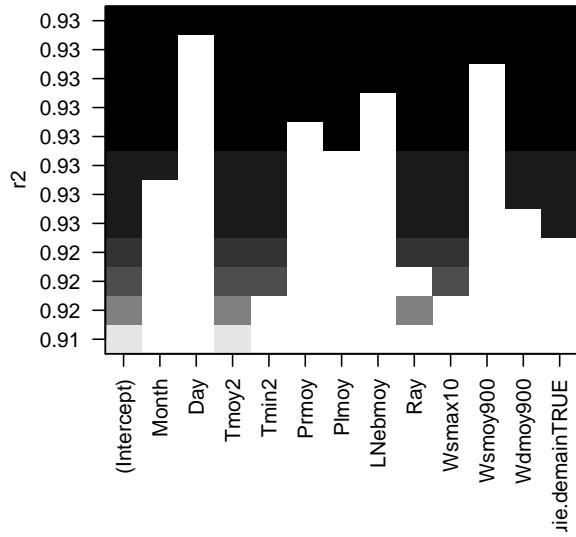
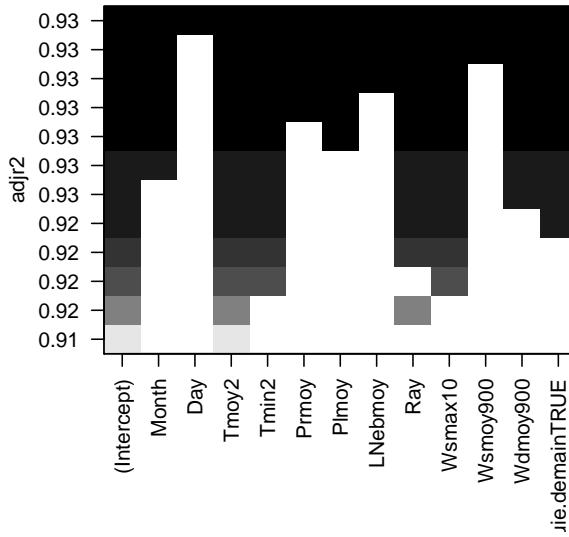
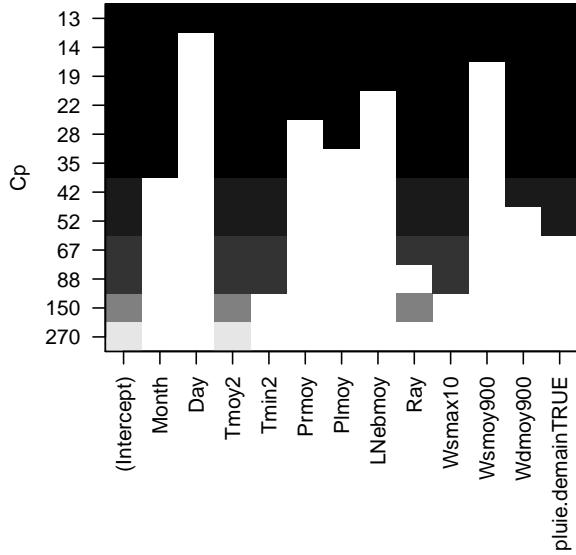
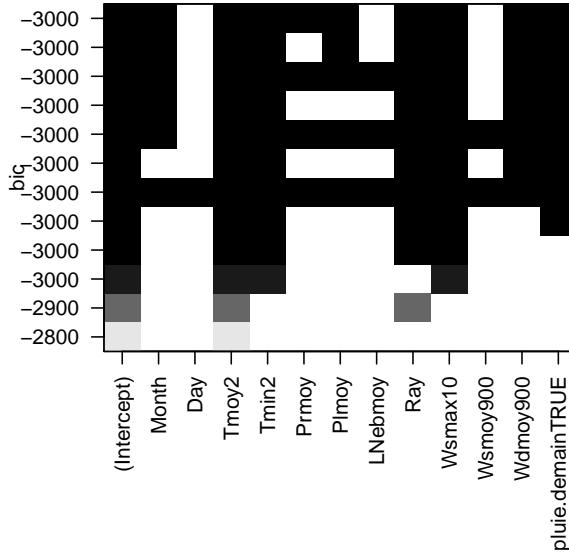
Les méthodes descendantes et stepwise fournissent une AIC un peu meilleure que le modèle de référence *temp4* ($AIC = 4820$) mais utilise 22 covariables (dont un certain nombre non significatives).

Je ne retiens pas le modèle obtenu et conserve le modèle *temp4*.

G. Procédure automatique de recherche d'un modèle plus petit que *temp4*

```
library(leaps)
par(mfrow=c(2,2))
recherche.ex=regsubsets(temp.demain~
                         Month+Day+Tmoy2+Tmin2+Prmoy+Plmoy+LNebmoy+Ray+Wsmoy900+Wdmoy900+ plui
                         int=T,nbest=1,nvmax=13,method="exhaustive",data=met3)

l=c("bic","Cp","adjr2","r2")
for (i in 1:4){
  plot(recherche.ex,scale=l[i])
}
```



La recherche de modèles plus petits par procédure automatique conforte mes choix pour 3 critères sur 4 :

- le R^2 (évidemment, car il augmente “mécaniquement” avec un nombre croissant de covariables)
- le R_{adj}^2 (déjà vérifié plus haut)
- le C_p de Mallows

Je retiens également le meilleur modèle fourni pour le critère du BIC (que j'appelle *temp6*) afin de tester ultérieurement ses qualités prédictives.

```
temp6=lm(temp.demain~Month+Tmoy2+Tmin2+Prmoy+Plmoy+L.Nebmoy+Ray+Wsmax10+Wdmoy900+ pluie.demain, data=met3)
```

H. Ajout des interactions

Je commence par effectuer un “scatterplot” pour y déceler d'éventuelles informations sur des corrélations potentielles.

Les corrélations élevées m'indiquent des pistes dans la recherche des interactions potentielles.

```

# préparation des données
met4 <- met3[,c(2:5,10,13,24,28,31,37,40,44)]


# Scatterplot

# Histogramme sur la diagonale
panel.hist=function(x,...){
  usr=par("usr"); on.exit(par(usr))
  par(usr=c(usr[1:2], 0, 1.5) )
  h=hist(x, plot = FALSE, col="lightblue")
  breaks=h$breaks; nb = length(breaks)
  y=h$counts ; y<-y/max(y)
  rect(breaks[nb], 0, breaks[-1], y , col="cyan", ...)
}

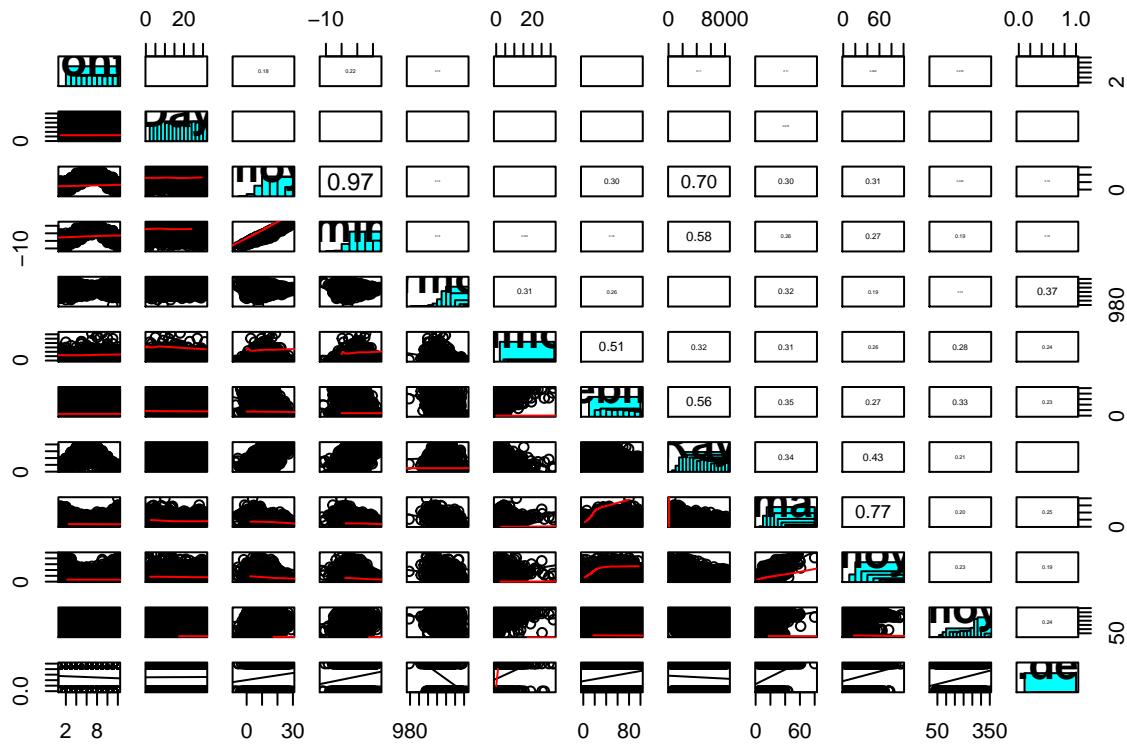
# coefficients de corrélation en valeur absolue sur la partie haute
# Taille de police proportionnelle au coefficient de corrélation

panel.cor = function(x,y,digits=2,prefix="",cex.cor,...){
  usr=par("usr"); on.exit(par(usr))
  par(usr=c(0,1,0,1 ))
  r=abs(cor(x,y))
  txt=format(c(r,0.123456789),digits=digits)[1]
  txt=paste(prefix, txt, sep="")
  if(missing(cex.cor)) cex.cor=0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex=cex.cor*r)
}

# regression linéaire lm sur la partie basse ;
# d'où le signe du coefficient de corrélation.

panel.lm = function(x,y)
{
  points(x,y)
  abline(lm(y~x))
  lines(lowess(y,x),col="red")
}
# on regroupe le tout
pairs(met4,diag.panel=panel.hist,cex.labels=2,font.labels = 2,
      upper.panel=panel.cor, lower.panel=panel.lm)

```



La figure est très chargée et difficilement lisible (de part le nombre important de covariables), néanmoins, quelques informations en ressort.

Il semble y avoir une corrélation importante en $Tmoy2$ et $Tmin$ ($r = 0.97$) d'une part, $Tmoy2$ et Ray ($r = 0.70$) d'autre part. J'étudie l'interaction entre ces covariables dans un nouveau modèle.

J'y ajoute également l'interaction entre $Tmoy2$ et $Wsmax10$, car j'ai l'intuition que la température moyenne dépend fortement de la vitesse du vent.

```
temp4b=lm(temp.demain~Month+Day+Tmoy2*(Tmin2+Wsmax10+Ray)+Prmoy+Plmoy+L Nebmoy+Wsmoy900+Wdmoy900
+ pluie.demain,data=met3)
```

```
AIC(temp4b)
```

```
## [1] 4816.45
```

L'AIC baisse de un peu. Avant de supprimer les interactions et covariables non significatives, je teste l'interaction entre $Tmoy2$ et $Prmoy$ (intuition sur un lien potentiel "température" et "pression")

```
temp4c=lm(temp.demain~Month+Day+Tmoy2*(Tmin2+Wsmax10+Ray+Prmoy)+Plmoy+L Nebmoy+Wsmoy900+Wdmoy900
+ pluie.demain,data=met3)
```

```
AIC(temp4c)
```

```
## [1] 4811.612
```

L'ajout de toutes ces interactions semblent déstabiliser le modèle. Beaucoup de covariables et d'interactions dépassent allègrement le seuil de significativité.

l'intercept, ainsi que la pression moyenne s'avèrent désormais totalement inutiles.

- Un essai "brutal"- Dans le modèle précédent, je décide d'enlever brutalement tout ce qui n'est pas significatif.

```

temp4c2=lm(temp.demain~-1+Month+Tmoy2:Prmoy+Tmoy2*Tmin2+Tmoy2:Wsmax10+Plmoy+LNebmoy+Ray+Ws moy900
+Wdmoy900+ pluie.demain,data=met3)
summary(temp4c2)

##
## Call:
## lm(formula = temp.demain ~ -1 + Month + Tmoy2:Prmoy + Tmoy2 +
##      Tmin2 + Tmoy2:Wsmax10 + Plmoy + LNebmoy + Ray + Wsmoy900 +
##      Wdmoy900 + pluie.demain, data = met3)
##
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -5.7951 -1.1859  0.0309  1.1499  6.5578 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## Month        5.144e-02  1.762e-02   2.920  0.00357 ** 
## Tmoy2       -2.631e+00  8.054e-01  -3.267  0.00112 ** 
## Tmin2       -2.839e-01  4.643e-02  -6.115 1.32e-09 *** 
## Plmoy        5.031e-02  1.657e-02   3.037  0.00244 ** 
## LNebmoy      6.210e-03  2.556e-03   2.430  0.01527 *  
## Ray          3.128e-04  5.093e-05   6.142 1.12e-09 *** 
## Wsmoy900     1.293e-02  4.942e-03   2.617  0.00898 ** 
## Wdmoy900     2.225e-03  8.434e-04   2.638  0.00845 ** 
## pluie.demainFALSE -1.971e+00  3.642e-01  -5.411 7.60e-08 *** 
## pluie.demainTRUE -2.388e+00  3.843e-01  -6.215 7.14e-10 *** 
## Tmoy2:Prmoy    3.813e-03  7.924e-04   4.812 1.69e-06 *** 
## Tmoy2:Wsmax10   -5.870e-03  7.499e-04  -7.828 1.10e-14 *** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.848 on 1168 degrees of freedom
## Multiple R-squared:  0.9829, Adjusted R-squared:  0.9827 
## F-statistic:  5590 on 12 and 1168 DF,  p-value: < 2.2e-16
AIC(temp4c2)

```

```
## [1] 4811.59
```

Ce modèle ne contient que des éléments significatifs, mais est peut-être un peu petit en taille pour bien décrire la variable d'intérêt. l'AIC est somme toute convenable.

Je retiens ce modèle pour le tester ultérieurement.

Je décide de repartir du modèle *temp4c* et d'enlever progressivement les éléments les moins significatifs :

- l'interaction entre *Tmoy2* et *Ray*
- les covariable *Prmoy*, *Wsmoy900* et *Wdmoy900* (je ne conserve que l'interaction entre ces 2 dernières)

```

temp4d=lm(temp.demain~Month+Day+Tmoy2:Prmoy+Tmoy2*(Tmin2+Wsmax10)+Plmoy+LNebmoy+Ray+Ws moy900 :Wdmoy900
+ pluie.demain,data=met3)

```

```
AIC(temp4d)
```

```
## [1] 4802.935
```

l'AIC (4803), tout comme le nombre de covariables continue de baisser.

Malgré la faible significativité, je conserve l'interaction entre $Tmoy2$ et $Tmin2$ (j'ai l'intuition que ces variables sont quand même liées) et enlève les covariables à la limite de la significativité Day et $Wsmax10$.

```
temp4e=lm(temp.demain~Month+Tmoy2:Prmoy+Tmoy2*Tmin2+Tmoy2:Wsmax10+Plmoy+LNebmoy+Ray+Ws moy900:Wdmoy900  
+ pluie.demain,data=met3)
```

```
AIC(temp4e)
```

```
## [1] 4805.584
```

l'AIC remonte un peu (4806). N'ayant pas réussi à la rendre significative, je me résous à finalement supprimer l'interaction entre $Tmoy2$ et $Tmin2$.

J'y ajoute "une touche exotique" car j'avais observé dans le scatterplot qu'il pourrait être intéressant de régresser $\sqrt{Wsmax10}$ (au lieu de $Wsmax10$) au vu des régressions 2 à 2.

```
temp4f=lm(temp.demain~Month+Tmoy2:(Prmoy+Wsmax10)+Tmoy2+Tmin2+Plmoy+LNebmoy+Ray+Ws moy900:Wdmoy900  
+sqrt(Wsmax10):LNebmoy+ pluie.demain,data=met3)
```

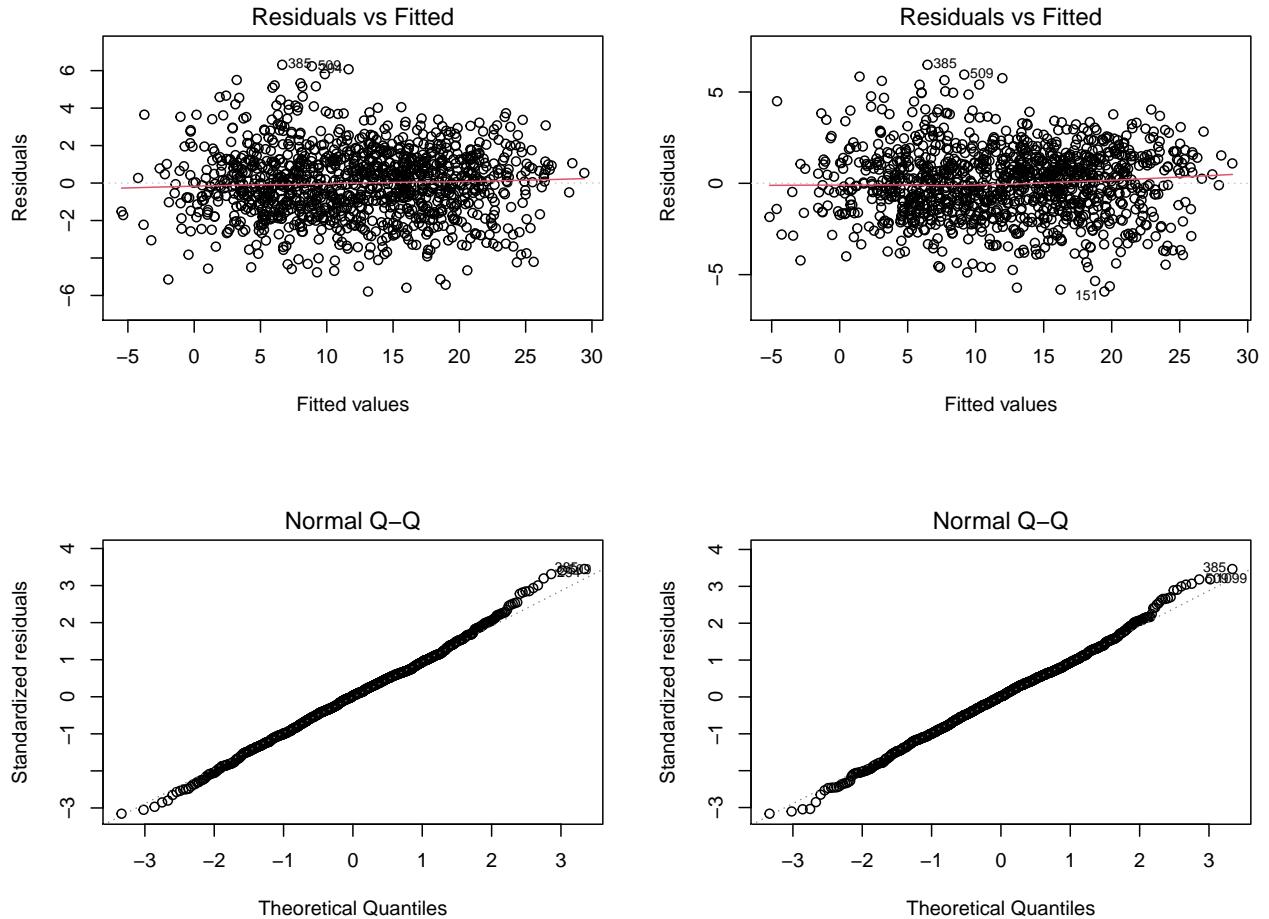
```
AIC(temp4f)
```

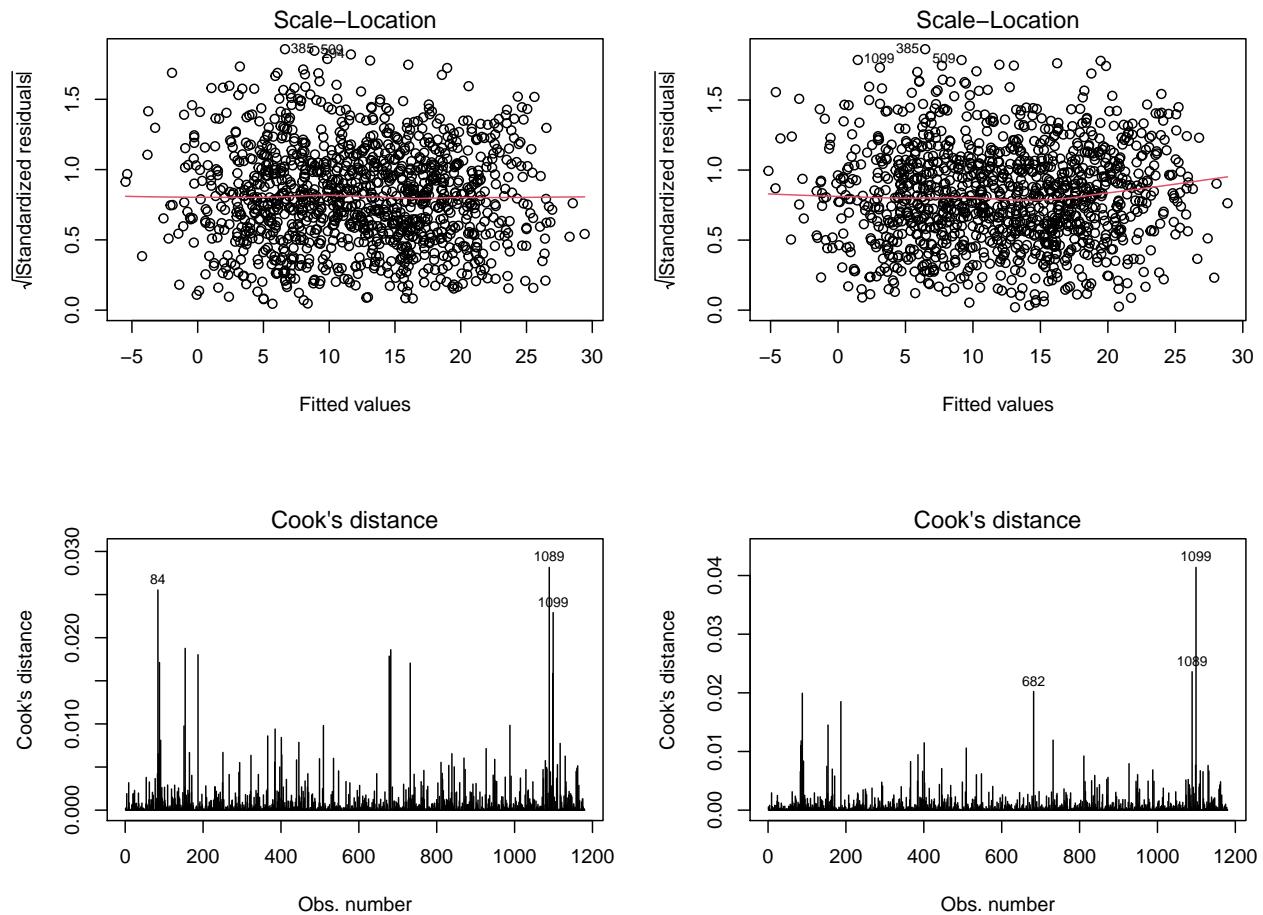
```
## [1] 4801.695
```

J'ai l'AIC la plus basse de tous mes essais (4801.6), un nombre réduit de covariables, d'interactions et tous les éléments sont très significatifs.

J'étudie la structure des résidus afin de potentiellement valider le modèle. Je décide de les comparer à ceux de *temp4* dont j'ai déjà effectué l'étude précédemment.

```
par(mfrow=c(2,2))
for (i in 1:4){
  plot(temp4f,i)
  plot(temp4,i)
}
```





Les modèles sont assez similaires. *temp4f* (à gauche) semble même légèrement meilleur que *temp4* (à droite) pour les éléments étudiés sur les 3 premiers graphiques (les interprétations ont été faites précédemment).

les distances de Cook traduisent un nombre plus important de points leviers sur le modèle *temp4f*, mais leur valeurs inférieures à 0.5 minimise leur influence sur le modèle.

Conclusion Ce modèle me semble intéressant. Il explique relativement bien la variable à expliquer.

La synthèse de l'ensemble de l'étude m'enclint à privilégier les modèles *temp4c2*, *temp4d*, *temp4f*.

Je vais maintenant désormais tester leurs qualités prédictives, ainsi que toutes celles de leurs concurrents.

II. La température (prédiction)

A. Préparation des données

```
# chargement des données test
metpr0=read.csv("/Users/anthonylezin/Desktop/Projets Stats/Projet GLM/meteo.test.csv",header=TRUE,sep="")

# suppression des colonnes inutiles
metpr1=metpr0[,-c(1,5,6)]
metpr2=metpr1

names(metpr2)=as.vector(nom1)

#réarrangement de l'ordre des variables afin de regrouper celles concernant la même catégorie
metpr2 <- metpr2[,c(1:4,23,22,5,25,24,6,27,26,7:9,29,28,10,31,30,11,33,32,12,35,34,13:15,37,36,16,17,39)]
```

```
# pour rappel, voici la liste des modèles :
mod=c("temp","temp1","temp2","temp3","temp4","temp5","temp6","temp4b","temp4c",
      "temp4c2","temp4d","temp4e","temp4f")
```

Je teste les qualités prédictives des modèles *temp4*, *temp4c2*, *temp4d*, *temp4f* en affichant l'**intervalle de confiance associé** (au seuil de 95%) sur le fichier “test”.

Par ailleurs, je les compare à celles du fichier original *temp*.

```
# choix des modèles : "temp","temp4","temp4c2","temp4d","temp4f"
temp_list2=mod[c(1,5,10,11,13)]

amp=function(model){
  predict(model, newdata=metpr2,interval="confidence")
}

df1=data.frame(head(amp(temp)),head(amp(temp4)))
df2=data.frame(head(amp(temp4c2)),head(amp(temp4d)),head(amp(temp4f)))

model.names=function(model){
  A=c("fit(model)","lwr(model)","upr(model)")
  return(A)
}

colnames(df1)=c(model.names(temp),model.names(temp4))
colnames(df2)=c(model.names(temp4c2),model.names(temp4d),model.names(temp4f))
kable(df1)
```

fit(model)	lwr(model)	upr(model)	fit(model)	lwr(model)	upr(model)
18.96115	18.38954	19.53275	19.15470	18.84112	19.46828
25.40556	24.73954	26.07158	25.24704	24.87154	25.62254
16.03218	15.42594	16.63841	15.91658	15.61326	16.21989
16.11441	15.65174	16.57708	16.39576	16.10254	16.68898
17.46388	16.88184	18.04592	17.07020	16.78224	17.35816
17.67435	17.11626	18.23244	18.52145	18.22675	18.81614

kable(df2)

fit(model)	lwr(model)	upr(model)	fit(model)	lwr(model)	upr(model)	fit(model)	lwr(model)	upr(model)
19.33404	19.03804	19.63005	19.36645	19.04897	19.68393	19.25914	18.96467	19.55360
24.23340	23.73769	24.72911	24.10472	23.55394	24.65549	24.28817	23.79154	24.78481
15.91088	15.61011	16.21165	15.73126	15.44743	16.01508	15.80711	15.54280	16.07142
16.59804	16.33623	16.85984	16.40697	16.10989	16.70405	16.55528	16.29446	16.81610
17.14461	16.86167	17.42755	17.24226	16.97704	17.50747	17.23639	16.97742	17.49535
18.96707	18.67506	19.25907	18.89228	18.59292	19.19165	18.99187	18.70930	19.27443

Le travail effectué ci-dessus répond donc à la question en fournissant une préduction continue de la température sur le fichier “test” selon mes meilleurs modèles. Pour en améliorer l’interprétation, je décide de poursuivre l’exploitation des résultats.

L’amplitude de l’intervalle de confiance dépend bien évidemment du modèle. Je calcule l’amplitude moyenne.

```

# amplitude moyenne de l'intervalle de prédiction
amp_moy=function(model){
  mean(amp(model)[,3]-amp(model)[,2])
}
df4=NULL
for (i in 1:5){
  df4[i]=amp_moy(get(temp_list2[i]))
}

df4=matrix(df4,ncol=5,nrow=1)
rownames(df4)="Amplitude moyenne"
colnames(df4)=temp_list2
df4

##           temp      temp4    temp4c2   temp4d    temp4f
## Amplitude moyenne 1.438601 0.7792343 0.7308236 0.789149 0.7309937

```

Comme attendu, c'est lorsque le modèle contient toutes les covariables que l'intervalle de prédiction est le plus large et donc le "plus permissif".

Pour un modèle possédant un nombre raisonnable de covariables, l'amplitude moyenne est comprise entre 0.7°C et 0.8°C.

B. Mesure de la qualité de prédiction (partie 1)

1. Une première étape Les intervalles de confiance des modèles raisonnables sont "un peu étroits" pour évaluer convenablement la qualité de la prédiction. En effet, une différence de 0.5°C entre la valeur réelle et la valeur prédictée classerait cette dernière comme une erreur de prédiction.

Voici une fonction permettant (entre autres) de vérifier ce fait.

```

qual_pred.temp=function(model,j){
  A= predict(model, newdata=metpr2,interval="confidence")
  tmp=cbind(A[,2]-j , A[,3]+j)

B=NULL
  for (i in (1:dim(tmp)[1])){
    B[i]=(metpr2[i,45]>tmp[i,1]) && (metpr2[i,45]<tmp[i,2])
  }
  return(c(sum(B),100*mean(B)))
}
qual_pred.temp(temp4,0)

## [1] 44.00000 15.17241

```

Voici le nombre de prédictions correctes (sur 290), ainsi que le taux de bonnes prédictions pour les modèles *temp*, *temp4*, *temp4c2*, *temp4d*, *temp4f*.

```

df5=NULL
for (i in 1:5){
  df5=t(c(df5,qual_pred.temp(get(temp_list2[i]),0)))
}

df5=t(matrix(df5,ncol=2,nrow=5,byrow = T))
colnames(df5)=temp_list2
rownames(df5)=c("bonnes prédictions","taux (en %)")
kable(df5)

```

	temp	temp4	temp4c2	temp4d	temp4f
bonnes prédictions	91.00000	44.00000	49.00000	51.00000	45.00000
taux (en %)	31.37931	15.17241	16.89655	17.58621	15.51724

Les résultats extrêmement décevants, néanmoins :

une différence de 1°C doit-elle obligatoirement être considérée comme une erreur de prédiction ?

Je propose une méthode “plus en accord” avec l’usage courant en fixant arbitrairement la règle suivante : je considère alors comme une erreur de prédiction, une température observée n’appartenant pas à l’intervalle de confiance dont “j’élargis chacune des bornes” de j°C.

Le cas échéant, je considère la température comme convenablement prédite.

Voici une fonction iniquant les qualités prédictives des modèles précédents suivant ce critère.

```
# renommage de la fonction
qpt=function(model){
  V=NULL
  for (i in (1:3)) {
    V[i]=qual_pred.temp(model,i-1)[2]
  }
  return(V)
}
df6=NULL
for (i in (1:length(mod))){
  df6=t(c(df6,(qpt(get(mod[i])))))
}
df6=matrix(df6,ncol=3,nrow=length(mod),byrow = T)
rownames(df6)=mod[1:length(mod)]
colnames(df6)=c("bornes +0°C", "bornes +1°C", "bornes +2°C")
kable(df6)
```

	bornes +0°C	bornes +1°C	bornes +2°C
temp	31.37931	71.72414	86.55172
temp1	17.58621	61.72414	84.82759
temp2	15.51724	59.31034	84.48276
temp3	14.48276	58.62069	83.79310
temp4	15.17241	59.31034	83.79310
temp5	14.13793	60.00000	83.44828
temp6	13.79310	57.93103	82.75862
temp4b	21.72414	61.03448	85.17241
temp4c	21.72414	61.37931	86.89655
temp4c2	16.89655	60.68966	84.13793
temp4d	17.58621	59.31034	85.86207
temp4e	17.93103	60.68966	84.13793
temp4f	15.51724	61.03448	84.13793

2. Résumé des informations et choix de modèle J’ai choisi *temp4* comme modèle de base, puis j’ai cherché à travailler sur les interactions entre les covariables explicatives afin :

- d’en minimiser l’AIC
- d’en maximiser le pouvoir prédictif.

Pour rappel, le modèle *temp4* est une régression linéaire contenant les covariables suivantes :

Month, Day, Tmoy2, Tmin2, Prmoy, Plmoy, LNebmoy, Ray, Wsmax10, Wsmoy900, Wdmoy900, pluie.demain

Voici un tableau résumant mes divers essais avec leurs caractéristiques.

modèles	caractéristiques du modèle	nbre de cov.	AIC	R^2_{adj}	qualité de prédiction (bornes +1°C)
temp	toutes les covariables	45	4855	0.929	71.7 %
temp4	aucune interaction	13	4854	0.928	59.3%
temp4b	(temp4) + Tmoy2*(Tmin2+Ray+Wsmax10)	16	4816	0.930	61.0%
temp4c	(temp4b) + Tmoy2:Prmoy	17	4812	0.930	61.4%
temp4d	(temp4c) + Wsmoy900:Wdmoy900 - (Prmoy + Tmoy2:Ray - Wdmoy900 - Wsmoy900)	14	4803	0.931	59.3%
temp4e	(temp4d) - Day - Wsmax10	12	4806	0.930	60.7%
temp4f	(temp4e) - Tmoy2:Tmin2 + LNebmoy:sqrt(Wsmax10)	12	4801.7	0.931	61%
temp4c2	(temp4c) - (Intercept + Wsmax10 + Day +Tmoy2:(Ray+Tmin2))	12	4812	0.983	60.7%

Lorsque l'on augmente les marges des bornes de l'intervalle à 2°C, la qualité de prédiction des modèles (située entre 82.8% et 86.6%) s'uniformise ayant pour conséquences de moins bien "séparer" les modèles.

Il est néanmoins intéressant de remarquer que dans plus de 82% des cas, la température observée se situe à moins de 2.4°C de la température prédictive.

C. Mesure de la qualité de prédiction (partie 2)

Dans cette partie je décide d'utiliser un autre critère pour mesurer la qualité prédictive du modèle.

En effet, il se peut très bien que :

- dans un modèle A, les données prédictives soient 0.6°C supérieure ou inférieure à leurs valeurs observées contenues dans le jeu test les excluant systématiquement de l'intervalle de confiance initial et par voie de fait, les classant en une erreur de prédiction ("au niveau + 0°C")
- dans un modèle B, certaines données soient dans l'intervalle, alors que d'autres sont distantes de plus de 5°C de la valeur observée.

La partie précédente m'aurait engagé à délaisser le modèle A au profit du modèle B, alors qu'intuitivement, il est assez proche de la réalité et a simplement le "mauvais goût de ne jamais tomber juste".

Afin de palier à ce problème, je vais désormais considérer l'**erreur moyenne de prédiction**

- en valeur absolue (en calculant $\frac{\sum_{i=1}^{n=45} |y_i - \hat{y}_i|}{n}$)
- quadratique (au carré) en calculant $\frac{\sum_{i=1}^{n=45} (y_i - \hat{y}_i)^2}{n}$. Cela aura pour effet "d'accentuer" les erreurs de prédiction éloignées.

```
erp=function(model){
  A=mean(abs(amp(model)-metpr2[,45]))
  B=mean(abs(amp(model)-metpr2[,45])^2)
  return(c(A,B))
}

df7=data.frame(erp(temp),erp(temp4),erp(temp4c2),erp(temp4d),erp(temp4f))
rownames(df7)=c("erreur moyenne (en valeur absolue)","erreur quadratique moyenne")
```

```
colnames(df7)=temp_list2
kable(df7)
```

	temp	temp4	temp4c2	temp4d	temp4f
erreur moyenne (en valeur absolue)	1.522670	1.465161	1.406967	1.413687	1.419368
erreur quadratique moyenne	4.072625	3.674128	3.375307	3.401780	3.414274

Avec ce critère, la qualité prédictive du modèle *temp* est **la plus mauvaise**, confirmant ainsi le fait que le modèle idéal ne peut pas être celui qui contient l'ensemble des covariables.

Voici la synthèse de l'ensemble des résultats de l'étude des qualités prédictives des modèles.

modèles	nbre de cov.	AIC	R^2_{adj}	qualité (+1°C)	$\frac{\sum_{i=1}^n y_i - \hat{y}_i }{n}$	$\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}$
temp	45	4855	0.929	71.7 %	1.522	4.073
temp4	13	4854	0.928	59.3%	1.465	3.674
temp4b	16	4816	0.930	61.0%	1.431	3.478
temp4c	17	4812	0.930	61.4%	1.413	3.413
temp4d	14	4803	0.931	59.3%	1.414	3.375
temp4e	12	4806	0.930	60.7%	1.404	3.378
temp4f	12	4801.7	0.931	61%	1.419	3.402
temp4c2	12	4812	0.983	60.7%	1.407	3.414

3 modèles ressortent du lot (*temp4e*, *temp4f* et *temp4c2*). En effet, ces modèles possèdent * un nombre minimal de covariables * une AIC assez basse (entre 4801.7 et 4812) * un bon pouvoir prédictif.

Parmi ces modèles, *temp4c2* “tire son épingle du jeu” avec son R^2_{adj} **5% supérieur aux autres**. Il explique mieux la température du lendemain tout en étant sensiblement au même niveau prédictif que ses concurrents.

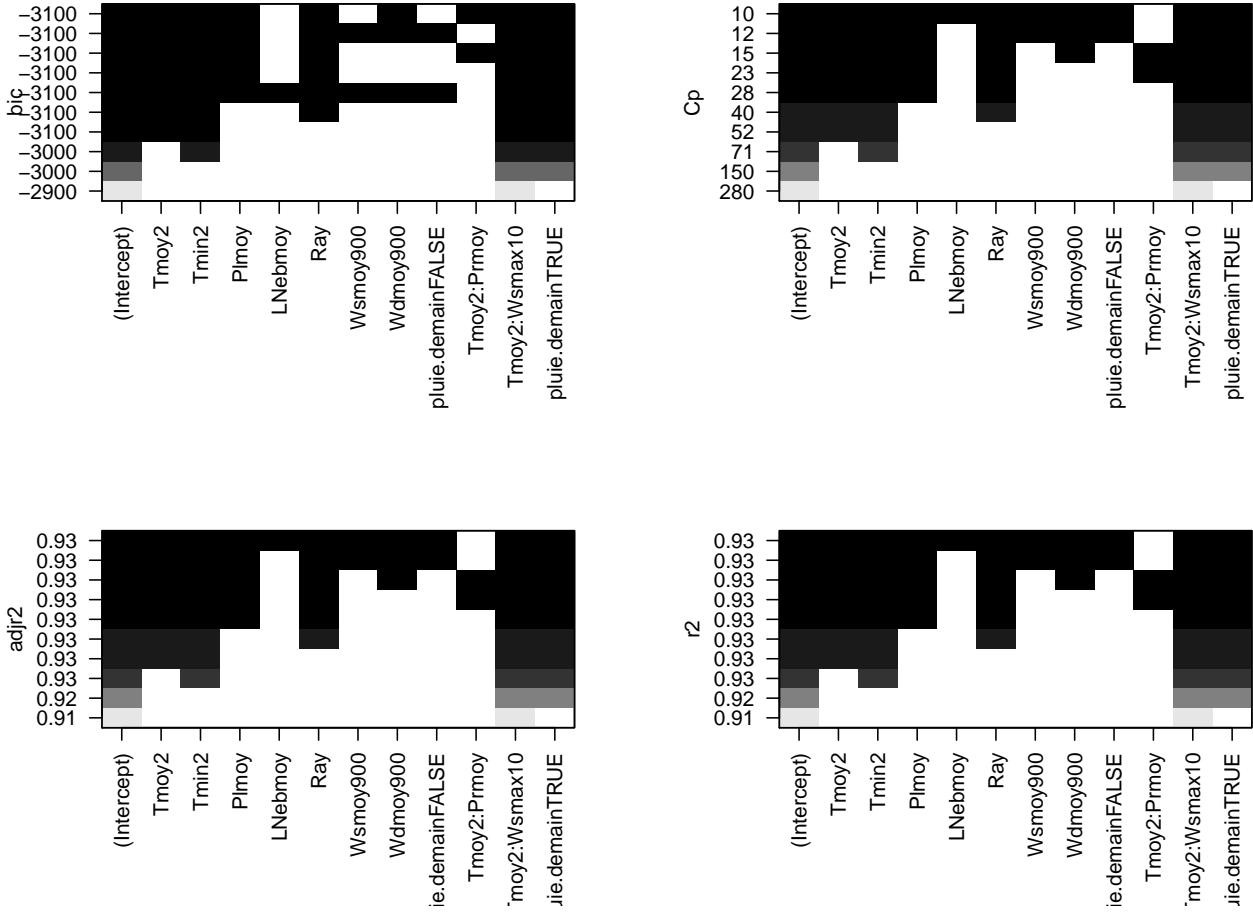
Pour répondre à la dualité (“explication/prédition”), je sélectionne donc modèle le modèle *temp4c2*.

Un meilleur modèle ?

Je regarde si le nombre des covariables intervenant dans le modèle *temp4c2* ne peut être réduit.

```
par(mfrow=c(2,2))
recherche.ex2=regsubsets(temp.demain~-1+Month+Tmoy2:Prmoy+Tmoy2+Tmin2+Tmoy2:Wsmax10+Plmoy+LNebmoy
+Ray+Ws moy900+Wdmoy900+ pluie.demain,
  int=T,nbest=1,nvmax=12,method="exhaustive",data=met3)

## Reordering variables and trying again:
for (i in 1:4){
  plot(recherche.ex2,scale=1[i])
}
```



Dans tous les critères, le modèle semble meilleur si j'enlève l'interaction *Tmoy2 : Prmoy*.

Je vais les comparer (au vu de l'AIC) et vérifier les qualités prédictives du nouveau modèle.

```
# modèle temp4c2 privé de "Tmoy2:Prmoy".
temp4c3=lm(temp.demain~-1+Month+Tmoy2+Tmin2+Tmoy2:Wsmax10+Plmoy+LNebmoy+Ray
           +Wsmoy900+Wdmoy900+ pluie.demain, data=met3)
```

```
AIC(temp4c3)
summary(temp4c3)$adj.r.squared
qpt(temp4c3)
erp(temp4c3)
```

modèles	nbre de cov.	<i>AIC</i>	R^2_{adj}	qualité (+1°C)	$\sum_{i=1}^{i=n} y_i - \hat{y}_i $	$\sum_{i=1}^{i=n} (y_i - \hat{y}_i)^2$
temp4c2	12	4812	0.983	60.7%	1.407	3.414
temp4c3	11	4830	0.983	60.0%	1.430	3.510

Au vu des résultats, je conserve le modèle *temp4c2* malgré la covariable en moins. En effet, au long de cette étude, j'ai choisi le *AIC* et le R^2_{adj} comme critères de sélection.

Mon **modèle idéal** est donc le modèle *temp4c2*

Enfin, parce qu'il le faut le faire à un moment, je m'arrête ici...

Voici l'export des prédictions associées au modèle *temp4c2*.

```
# Export des prédictions associées au modèle "temp4c2"

amp.temp4c2=round((amp(temp4c2)), digits=2)

amp.temp4c2=data.frame(cbind(round(amp(temp4c2)[,1],digits=2),
  data.frame(paste("[",amp.temp4c2[,2],":",amp.temp4c2[,3],"]",sep=""))))

colnames(amp.temp4c2)=c("temp. pred", "confiance (95%)")

kable(head(amp.temp4c2))
```

temp. pred	confiance (95%)
19.33	[19.04:19.63]
24.23	[23.74:24.73]
15.91	[15.61:16.21]
16.60	[16.34:16.86]
17.14	[16.86:17.43]
18.97	[18.68:19.26]

```
write.csv(amp.temp4c2, "amp.temp4c2.csv", row.names=FALSE, sep="t ",dec=".")
```

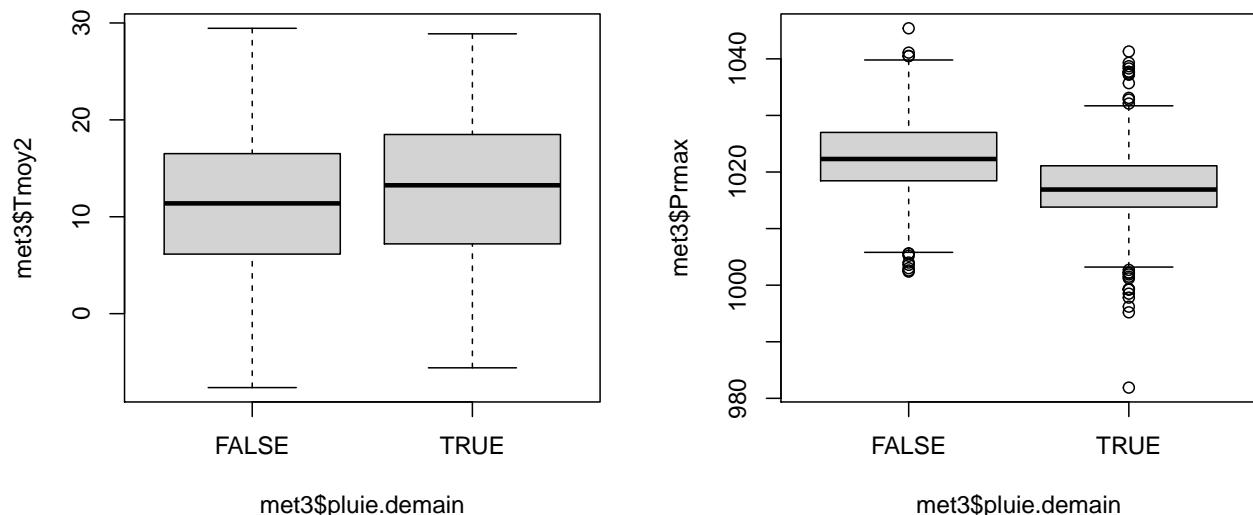
III. La pluie

Dans cette partie, je vais tenter d'expliquer la variable *pluie.demain* en fonction des données récoltées. C'est une variable binaire (avec la convention 1 = TRUE et 0 = FALSE). Je vais donc effectuer une GLM pour proposer un modèle permettant à la fois d'expliquer les données, mais également de prédire s'il pleuvra ou non demain.

A. Premières visualisation de quelques dépendances potentielles

Les couple à étudier étant du type quantitatif/qualitatif, j'utilise un “boxplot”.

```
par(mfrow=c(1,2))
boxplot(met3$Tmoy2~met3$pluie.demain)
boxplot(met3$Prmax~met3$pluie.demain)
```



Les différences sont minimes dans le 1er cas, on peut toutefois noter qu'il a plu lorsque la température de la veille était plus élevée en moyenne.

En revanche, il y a bien un effet Pression maximale sur la pluie le lendemain. J'affinerai cette étude lors de l'étude entre les covariables sélectionnées.

Voici le modèle complet :

```
pluie0=glm(pluie.demain~,family = binomial, data=met3)
summary(pluie0)
```

```
##
## Call:
## glm(formula = pluie.demain ~ ., family = binomial, data = met3)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.9310  -0.8246   0.2799   0.8202   2.9609
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.292e+01  7.115e+01 -1.306 0.191582
## Year         7.507e-02  3.512e-02  2.138 0.032534 *
## Month        -8.515e-03  2.504e-02 -0.340 0.733836
## Day          1.004e-02  8.223e-03  1.221 0.222022
## Tmoy2        3.126e-01  1.696e-01  1.843 0.065293 .
## Tmin2        -1.641e-01  8.754e-02 -1.875 0.060851 .
## Tmax2        5.585e-03  9.694e-02  0.058 0.954059
## Hmoy2        2.321e-02  3.268e-02  0.710 0.477615
## Hmin2        -4.766e-03  1.867e-02 -0.255 0.798515
## Hmax2        -3.267e-03  2.082e-02 -0.157 0.875328
## Prmoy        5.527e-01  1.408e-01  3.926 8.63e-05 ***
## Prmin        -3.341e-01  7.625e-02 -4.382 1.18e-05 ***
## Prmax        -2.811e-01  7.572e-02 -3.713 0.000205 ***
## Plmoy        3.223e-02  2.834e-02  1.137 0.255433
## Snow          -3.122e-01  2.391e-01 -1.306 0.191549
## TNebmoy      1.052e-02  1.212e-02  0.868 0.385204
## TNebmin      7.645e-03  6.273e-03  1.219 0.222913
## TNebmax      2.633e-03  4.897e-03  0.538 0.590738
## HNebmoy      -1.976e-03  6.899e-03 -0.286 0.774553
## HNebmin      4.436e-03  2.129e-02  0.208 0.834955
## HNebmax      3.063e-03  2.906e-03  1.054 0.291881
## MNebmoy      6.129e-03  6.730e-03  0.911 0.362428
## MNebmin      -5.030e-03  9.579e-03 -0.525 0.599541
## MNebmax      6.320e-03  3.178e-03  1.989 0.046741 *
## LNebmoy      -2.401e-03  8.209e-03 -0.293 0.769858
## LNebmin      1.738e-05  7.044e-03  0.002 0.998032
## LNebmax      2.922e-03  3.434e-03  0.851 0.394889
## Sun          2.898e-04  8.930e-04  0.324 0.745581
## Ray          1.058e-04  1.018e-04  1.039 0.298813
## Wsmoy10     -5.000e-02  9.679e-02 -0.517 0.605462
## Wsmin10      1.692e-01  6.426e-02  2.633 0.008464 **
## Wsmax10      4.751e-02  3.466e-02  1.371 0.170477
## Wdmoy10      5.335e-03  5.793e-03  0.921 0.357091
## Wsmoy80      -8.198e-02  6.962e-02 -1.178 0.238973
## Wsmin80      -5.351e-02  4.233e-02 -1.264 0.206206
```

```

## Wsmax80 -2.632e-04 2.862e-02 -0.009 0.992663
## Wdmoy80 -9.491e-03 5.979e-03 -1.587 0.112427
## Wsmoy900 2.290e-02 2.608e-02 0.878 0.379734
## Wsmin900 -7.843e-03 1.917e-02 -0.409 0.682429
## Wsmax900 -1.414e-02 1.218e-02 -1.161 0.245740
## Wdmoy900 5.824e-03 1.467e-03 3.971 7.15e-05 ***
## Wgmoy 1.564e-02 3.691e-02 0.424 0.671780
## Wgmin 7.825e-03 2.796e-02 0.280 0.779553
## Wgmax 2.322e-02 1.736e-02 1.337 0.181137
## temp.demain -1.362e-01 3.910e-02 -3.485 0.000493 ***
## ---
## 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: 1220.4 on 1135 degrees of freedom
## AIC: 1310.4
##
## Number of Fisher Scoring iterations: 5

```

B. Interprétation des coefficients

C'est désormais la “log côte anglaise des p_i ” qui est directement interprétable (lorsque le coefficient positif, la probabilité qu'il pleuve le lendemain augmente et réciproquement).

- à titre d'interprétation, le coefficient de $Prmax$ est négatif (“-2.81e-01”) et est fortement significatif. On retrouve bien le fait que $temp.demain$ et $Prmax$ sont “fortement liées”. Plus précisemment, quand la pression maximale augmente d'une unité, la probabilité qu'il pleuve le lendemain diminue (car le coefficient est négatif) et est multipliée par $\frac{e^{-0.281}}{1+e^{-0.281}}$.
- c'est en revanche le phénomène inverse pour la température moyenne, puisque, cette fois-ci, le coefficient est positif (“3.13e-01”).

C. Sélection d'un premier modèle

Beaucoup de covariables sont peu influentes. Il conviendra de les enlever du modèle.

on procède comme dans la partie I., on supprime progressivement les covariables de **p-valeur supérieure à 0.2**, puis on regarde l'évolution des coefficients restants.

```

pluie1=glm(pluie.demain~Year + Month + Day+ Tmoy2 + Prmoy + Plmoy + Snow +Ray +Wsmoy80 +Wdmoy80
            +Wdmoy900 +Tmin2 +Prmax +Prmin +TNebmin +MNebmax +Wsmax10 +Wsmin10 +Wsmin80 +Wsmax900
            +Wgmax+temp.demain,
            ,family = binomial, data=met3)

pluie2=glm(pluie.demain~Year + Day+ Tmoy2 + Prmoy + Plmoy + Snow +Wsmoy80 +Wdmoy80 +Wdmoy900 +Tmin2
            +Prmax +Prmin +TNebmin +MNebmax +Wsmax10 +Wsmin10 +Wgmax +temp.demain,
            ,family = binomial, data=met3)

pluie3=glm(pluie.demain~Year+ Tmoy2 + Prmoy + Plmoy + Snow +Wsmoy80 +Wdmoy80 +Wdmoy900 +Tmin2
            +Prmax +Prmin +TNebmin +MNebmax +Wsmax10 +Wsmin10 +Wgmax+ +temp.demain,
            ,family = binomial, data=met3)

```

L'intercept est toujours non significative, j'essaie un modèle alternatif en l'ôtant ici.

```

pluie3b=glm(pluie.demain~-1+Year+ Tmoy2 + Prmoy + Plmoy + Snow +Wsmoy80 +Wdmoy80 +Wdmoy900 +Tmin2
            +Prmax +Prmin +TNebmin +M Nebmax +Wsmax10 +Wsmin10 +Wgmax +temp.demain,
            ,family = binomial, data=met3)

```

l'AIC a peu évolué. Je reprends mon procédé précédent, quitte à ré-enlever l'intercept plus tard.

```

pluie4=glm(pluie.demain~Year+ Tmoy2 + Prmoy + Plmoy + Snow +Wsmoy80 +Wdmoy80 +Wdmoy900 +Prmax
            +Prmin +TNebmin +M Nebmax +Wsmax10 +Wsmin10 +Wgmax+temp.demain,
            ,family = binomial, data=met3)

```

Je considère ce modèle *pluie4* comme “modèle de travail”. Il contient les variables : *Year*, *Tmoy2*, *Prmoy*, *Plmoy*, *Snow*, *Wsmoy80*, *Wdmoy80*, *Wdmoy900*, *Prmax*, *Prmin*, *TNebmin*, *MNebmax*, *Wsmax10*, *Wsmin10*, *Wgmax*, *temp.demain*.

Ce modèle contient 16 covariables. J'enlève *Plmoy* (comme variable non significative) et ajoute *Tmin* (comme intuition).

```

pluie4.1=glm(pluie.demain~Year+ Tmoy2 + Prmoy + Snow +Wsmoy80 +Wdmoy80 +Wdmoy900 +Tmin2 +Prmax
            +Prmin +TNebmin +M Nebmax +Wsmax10 +Wsmin10 +Wgmax+temp.demain,
            ,family = binomial, data=met3)

```

Cet essai s'avère infructueux. Je repars du modèle *pluie4* et supprime simplement la covariable *Plmoy*. Ce modèle contient 15 covariables.

```

pluie5=glm(pluie.demain~Year+ Tmoy2 + Prmoy + Snow +Wsmoy80 +Wdmoy80 +Wdmoy900 +Prmax +Prmin
            +TNebmin +M Nebmax +Wsmax10 +Wsmin10 +Wgmax+temp.demain,
            ,family = binomial, data=met3)

```

J'enlève la covariable "Snow". Ce modèle contient 14 covariables

```

pluie6=glm(pluie.demain~Year+ Tmoy2 + Prmoy +Wsmoy80 +Wdmoy80 +Wdmoy900 +Prmax +Prmin +TNebmin
            +M Nebmax +Wsmax10 +Wsmin10 +Wgmax+temp.demain,
            ,family = binomial, data=met3)

```

L'intercept n'est toujours pas significative. Je l'ôte du modèle.

```

pluie7=glm(pluie.demain~-1+Year+ Tmoy2 + Prmoy +Wsmoy80 +Wdmoy80 +Wdmoy900 +Prmax +Prmin +TNebmin +M Nebmax +Wsmax10 +Wsmin10 +Wgmax+temp.demain,
            ,family = binomial, data=met3)
summary(pluie7)

```

```

##
## Call:
## glm(formula = pluie.demain ~ -1 + Year + Tmoy2 + Prmoy + Wsmoy80 +
##       Wdmoy80 + Wdmoy900 + Prmax + Prmin + TNebmin + MNebmax +
##       Wsmax10 + Wsmin10 + Wgmax + temp.demain, family = binomial,
##       data = met3)
##
## Deviance Residuals:
##      Min        1Q     Median        3Q       Max 
## -2.5178   -0.8659    0.2654    0.8627   2.7274 
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## Year          0.035888  0.005541  6.477 9.38e-11 ***
## Tmoy2         0.148376  0.036043  4.117 3.85e-05 ***
## Prmoy         0.488330  0.131464  3.715 0.000204 ***
## Wsmoy80      -0.107667  0.029660 -3.630 0.000283 ***
## Wdmoy80      -0.003254  0.001438 -2.263 0.023626 *  
## 
```

```

## Wdmoy900    0.004523   0.001266   3.572 0.000355 ***
## Prmax     -0.255635   0.069711  -3.667 0.000245 ***
## Prmin     -0.306134   0.072355  -4.231 2.33e-05 ***
## TNebmin    0.009443   0.003398   2.779 0.005446 **
## MNebmax    0.012109   0.001919   6.310 2.78e-10 ***
## Wsmax10    0.052959   0.022284   2.377 0.017474 *
## Wsmin10    0.100284   0.035413   2.832 0.004628 **
## Wgmax      0.024997   0.010233   2.443 0.014580 *
## temp.demain -0.104273  0.035644  -2.925 0.003440 **
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 1635.8 on 1180 degrees of freedom
## Residual deviance: 1255.9 on 1166 degrees of freedom
## AIC: 1283.9
##
## Number of Fisher Scoring iterations: 4

```

Voici un tableau contenant les covariables intervenant dans chacun des modèles ainsi que leurs p-valeurs respectives intervenant dans les modèles étudiés :

covariables	<i>pluie4</i>	<i>pluie4.1</i>	<i>pluie5</i>	<i>pluie6</i>	<i>pluie7</i>
(Intercept)	2.8e-01	3.1e-01	2.9e-01	2.3e-01	
Year	2.7e-02	3.2e-02	2.7e-02	1.8e-02	9.4e-11
Tmoy2	4.4e-05	1.3e-04	4.0e-05	3.2e-05	3.8e-05
Prmoy	1.9e-04	2.9e-04	3.2e-04	2.4e-04	2.0e-04
Plmoy	1.7e-01				
Tmin2		5.3e-02			
Snow	8.4e-02	1.2e-01	1.2e-01		
Wsmoy80	2.0e-04	2.3e-04	2.4e-04	2.6e-04	2.8e-04
Wdmoy80	2.1e-02	9.5e-02	2.8e-02	2.2e-02	2.4e-02
Wdmoy900	8.9e-04	1.9e-04	4.9e-04	3.5e-04	3.5e-04
Prmax	4.1e-04	4.4e-04	6.6e-04	2.9e-04	2.5e-04
Prmin	1.4e-05	3.3e-05	2.2e-05	3.1e-05	2.3e-05
TNebmin	1.1e-02	9.6e-04	2.7e-03	4.2e-03	5.4e-03
MNebmax	1.0e-09	8.3e-11	2.8e-10	2.8e-10	2.8e-10
Wsmax10	1.7e-02	1.9e-02	1.8e-02	2.9e-02	1.7e-02
Wsmin10	5.4e-03	3.8e-03	5.9e-03	5.3e-03	4.6e-03
Wgmax	2.4e-02	1.5e-02	1.5e-02	7.6e-03	1.5e-02
temp.demain	2.7e-03	8.0e-04	2.9e-03	3.0e-03	3.4e-03

Je choisis donc le modèle *pluie7* comme modèle de référence. Je les chercherai à l'améliorer ultérieurement (par le biais des interactions par exemple).

Je décide de vérifier les qualités explicatives du modèle *pluie7* en le comparant au modèle M_0 et M_{sat} .

D. Tests du rapport de vraisemblance

Il y a deux modèles extrêmes auxquels comparer *pluie7* (en l'occurrence M_0 et M_{sat}).

Pour rappel, on appelle :

- M_0 le modèle nul dans lequel les (Y_i) sont indépendantes identiquement distribuées, ce qui équivaut à $\beta_1 = \beta_2 = \dots = \beta_k = 0$.
- M_{sat} le modèle saturé dans lequel il n'y a aucune structure aux (p_i) .

Les trois modèles sont imbriqués ($M_0 \subset pluie7 \subset M_{sat}$). On peut donc effectuer des **tests du χ^2 de rapport de vraisemblance**.

```
# modèle sans covariable
pchisq(summary(pluie7)$null.deviance - summary(pluie7)$deviance,
summary(pluie7)$df[1], lower = F)
```

```
## [1] 2.134851e-72
```

```
# modèle saturé
```

```
pchisq(summary(pluie7)$deviance, summary(pluie7)$df[2], lower = F)
```

```
## [1] 0.03365124
```

Dans le premier cas, on obtient une p-valeur très faible : je rejette le modèle sans covariable. Notre modèle est donc utile.

Dans le second cas, la p-valeur est également faible. Notre modèle ne capture pas toute la variabilité des données. Je devrai normalement rejeter ce modèle au profit du modèle saturé, néanmoins je décide de le conserver pour l'instant.

E. Qualité des modèles proposés en termes de déviance

Je décide de comparer l'ensemble de mes modèles aux deux modèles extrêmes (M_0 et M_{sat}). Cela est rendu possible par le fait qu'ils sont tous emboités avec M_0 et M_{sat} .

```
mod2=c(paste('pluie',0:4,sep=''), "pluie4.1", paste('pluie',5:7,sep=''))

test_devi=function(model){

  # modèle sans covariable
  A=pchisq(summary(model)$null.deviance - summary(pluie7)$deviance,   summary(model)$df[1],
lower = F)

  # modèle saturé
  B=pchisq(summary(model)$deviance, summary(model)$df[2], lower = F)
  return(c(A,B))
}

df7=NULL ; df7a=NULL ; df7b=NULL
for (i in (1:length(mod2))){
  df7b=t(c(df7b,test_devi(get(mod2[i]))))
  df7a=(c(df7a,AIC(get(mod2[i]))))
}
df7b=matrix(df7b,ncol=2,nrow=9, byrow = T)
rownames(df7b)=mod2[1:length(mod2)]
df7=cbind(df7b,df7a)
colnames(df7)=c("vs mod nul","vs modèle sat.", "AIC")
kable(df7)
```

	vs mod nul	vs modèle sat.	AIC
pluie0	0	0.0389923	1310.362
pluie1	0	0.0428788	1286.938

	vs mod nul	vs modèle sat.	AIC
pluie2	0	0.0471332	1280.816
pluie3	0	0.0470828	1279.876
pluie4	0	0.0390224	1283.373
pluie4.1	0	0.0420100	1281.638
pluie5	0	0.0374083	1283.394
pluie6	0	0.0342947	1284.434
pluie7	0	0.0336512	1283.905

Aucun des modèles proposés captent l'ensemble de la variabilité des données. En effet, le test de déviance montre quelque soit le modèle que M_{sat} apporte de l'information supplémentaire (la p-valeur n'est significative).

Je décide de regarder à tout hasard les modèles proposés par les méthodes automatiques.

F. Recherche automatique du meilleur modèle (sans interaction)

Je vérifie si les méthodes automatiques implémentées dans R ne fournissent pas un “meilleur concurrent” en termes d'AIC.

Voici les codes respectifs pour les méthodes ascendantes, descendantes et Stepwise.

```
library(MASS)
modselect_f=stepAIC(pluie0,pluie.demain~,data=
met3,trace=TRUE,direction=c("forward"))

## Start: AIC=1310.36
## pluie.demain ~ Year + Month + Day + Tmoy2 + Tmin2 + Tmax2 + Hmoy2 +
##     Hmin2 + Hmax2 + Prmoy + Prmin + Prmax + Plmoy + Snow + TNebmoy +
##     TNebmin + TNebmax + HNebmoy + HNebmin + HNebmax + MNebmoy +
##     MNebmin + MNebmax + LNebmoy + LNebmin + LNebmax + Sun + Ray +
##     Wsmoy10 + Wsmin10 + Wsmax10 + Wdmoy10 + Wsmoy80 + Wsmin80 +
##     Wsmax80 + Wdmoy80 + Wsmoy900 + Wsmin900 + Wsmax900 + Wdmoy900 +
##     Wgmoy + Wgmin + Wgmax + temp.demain

#modèle sélectionné
reg_for=glm(formula = pluie.demain ~ Year + Tmoy2 + Tmin2 + Prmoy + Prmin +
Prmax + Plmoy + Snow + TNebmoy + TNebmin + MNebmax + LNebmax +
Wsmin10 + Wsmax10 + Wsmoy80 + Wdmoy80 + Wdmoy900 + Wgmax +
temp.demain, family = binomial, data = met3)

test_devi(reg_for)

## [1] 3.58843e-68 6.62677e-02
```

Les 3 méthodes fournissent le même modèle. Ce modèle semble meilleur que *pluie7* au sens du test de la déviance (comparé au modèle M_{sat}), puisque la *p – valeur* est supérieure à 0.05. Néanmoins ce modèle utilise 20 covariables, dont un certain nombre n'est pas significatif !

Au vu de ces résultats, je décide de procéder autrement. J'estime en effet que le volet prédictif de la variable à expliquer est plus important ici que dans le chapitre précédent.

IV. La pluie (prédition et amélioration du modèle)

Je décide donc de modifier mon plan d'étude (comparativement au chapitre précédent) en étudiant dès à présent les qualités prédictives du dernier modèle retenu (en l'occurrence *pluie7*) sur le jeu de données “test”.

Une fois cela accompli, je comparerai les qualités prédictives du modèle de référence à celui obtenu par les méthodes de recherche automatisées.

Je chercherai enfin à améliorer le modèle retenu en étudiant les interactions entre les covariables.

Mon modèle ultime sera celui qui optimisera simultanément le couple *prédictivité / explication* tout en minimisant le nombre de covariables.

A. Prédiction pour le modèle *pluie7*

```
# chargement du jeu de données "test" / suppression des variables inutiles / renommage des variables
metpr0=read.csv("/Users/anthonylezin/Desktop/Projets Stats/Projet GLM/meteo.test.csv",
               header=TRUE,sep=",")
metpr1=metpr0[,-c(1,5,6)]
metpr2=metpr1

names(metpr2)=as.vector(nom1)

#je réarrange les variables pour regrouper celles concernant la même catégorie
metpr2 <- metpr2[,c(1:4,23,22,5,25,24,6,27,26,7:9,29,28,10,31,30,11,33,32,12,35,34,13:15,37,36,16,17,
,39,38,18,19,41,40,20,21,43,42,44,45)]
```

Je calcule les probabilités des valeurs prédites avec notre modèle retenu *pluie7* (“type=response” permettant de passer de l'échelle des “log côtes anglaises” à celle d'une probabilité). J'ai masqué les écarts-type pour plus de lisibilité.

```
pred=function(model){
  A=predict(model, metpr2, type="response", se.fit=F)
  return(A)
}
head(pred(pluie7))
```

```
##          1         2         3         4         5         6
## 0.3441318 0.9102785 0.6369568 0.7010803 0.5914914 0.2256047
```

Je détermine la matrice de confusion associée à un seuil s fixé (par ex. $s = 0.6$) permettant de classer les prédictions selon leurs type d'erreur (ou non) pour le modèle “*pluie7*”. Je choisis de fixer un poids identique à l'erreur de prédiction qu'elle soit de sensibilité ou de spécificité.

```
s=0.6
# matrice de confusion
print(table(pred(pluie7)>=s, metpr2$pluie.demain))

##
##          FALSE  TRUE
##    FALSE    100    59
##    TRUE     33    98

# prédiction en %
print(round(mean((pred(pluie7)>=s) == (metpr2$pluie.demain)),digits=4))

## [1] 0.6828
```

Ainsi pour $s = 0.6$, on a environ 68,3% de bonnes prédictions.

Que se passe-t-il si je fais varier le seuil? Pour cela, je construis une table des bonnes prédictions.

```

B=0
pred.tab = function(model,n,a,b){
  for (i in (0:n)) {
    A=round(seq(a,b,by=(b-a)/n),digits=9)
    length(B)=n+1
    B[i+1]=round(100*mean((pred(model)>=(a+(i*(b-a))/n)) == (metpr2$pluie.demain)),digits = 10)
  }
  C=data.frame(t(data.frame(A,B)))
  rownames(C)=c("seuil s","prédition (en %)")
  colnames(C)=c()
  return(C)
}

```

Ainsi, avec un pas de 0,1 (soit $n = 10$), j'obtiens le tableau suivant :

```
pred.tab(pluie7,10,0,1)
```

```

##
## seuil s      0.00000  0.10000  0.20000  0.30000  0.40000  0.50000
## prédition (en %) 54.13793 58.96552 65.86207 64.82759 70.68966 71.03448
##
## seuil s      0.60000  0.70000  0.8  0.90000  1.00000
## prédition (en %) 68.27586 62.75862 60.0 52.06897 45.86207

```

Avec ce pas, un balayage du tableau indique que les meilleures prédictions sont fournies pour $s = 0.5$ et les la fiabilité associée est de 71%.

Je cherche à améliorer la précision du couple “sensibilité/spécificité” en affinant le pas (toujours par la méthode de balayage).

```
system.time(pred.tab(pluie7,20000,0,1))
```

```

##   user  system elapsed
## 70.339  5.958  76.944

```

La fonction “system.time” montre qu'une précision à 10^{-4} du maximum de prédition prend plus d'une minute.

Le temps de calcul dans le tableau est un peu long et la recherche du seuil adéquat dans le tableau s'avère peu aisée.

Pour y remédier, je crée une fonction qui après avoir “balayé le tableau” :

- sélectionne la prédition maximale p et le seuil s dans le tableau correspondant (d'amplitude $[a, b]$ et de pas $(b - a)/n$)
- affiche la matrice de confusion associée.

```

pred_mat=function(model,n,a,b){

  # selection dans le tableau

  p=max(pred.tab(model,n,a,b)[2,])
  s=a+(which.max(pred.tab(model,n,a,b)[2,]))*(b-a)/n

  # matrice de confusion
  print(table(pred(model)>=s, metpr2$pluie.demain))

  return(c("s"=s,"prédic. (en %)"=p))
}

```

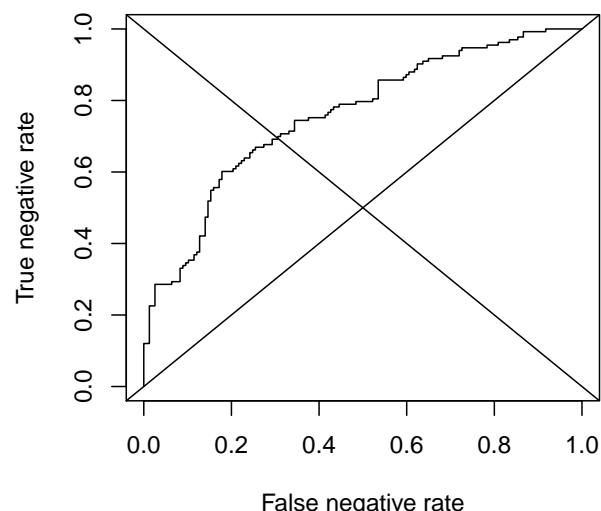
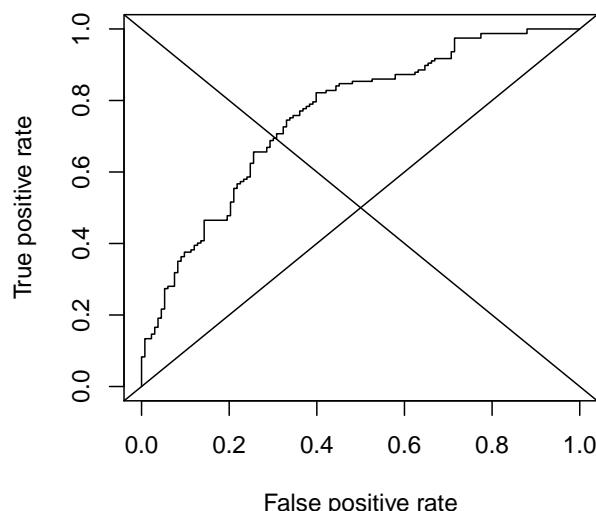
```
pred_mat(pluie7, 20, 0.41, 0.43)
```

```
##  
##          FALSE  TRUE  
##  FALSE     80    28  
##  TRUE      53   129  
  
##          s prédic. (en %)  
## 0.42000    72.06897
```

Le seuil $s = 0,42$ permet d'améliorer les qualités prédictives du modèle, puisque la fiabilité s'élève désormais à 72.07%.

Effectuons des courbes ROC afin de visualiser les pouvoirs prédictifs selon les faux positifs (ou faux négatifs)

```
library(ROCR)  
par(mfrow=c(1,2))  
p=prediction(pred(pluie7),metpr2$pluie.demain)  
plot(performance(p,"tpr","fpr"))  
abline(0,1)  
abline(1,-1)  
  
plot(performance(p,"tnr","fnr"))  
abline(0,1)  
abline(1,-1)
```



```
performance(p, "auc")@y.values[[1]]
```

```
## [1] 0.7579618
```

l'AUC est de 0.758. Etant supérieure à 0.7, on peut la considérer comme bonne.

B. Qualité prédictive des modèles automatiques (sans interaction)

Les méthodes descendantes et Stepwise fournissent un modèle utilisant les 20 covariables suivantes (dont un certain nombre ne sont pas significatives) :

Year, Tmoy2, Tmin2, Prmoy, Prmin, Prmax, Plmoy, Snow, TNebmoy, TNebmin, MNebmax, LNebmax, Wsmin10, Wsm

Je teste les qualités prédictives du modèle sur le jeu test.

```

pred_mat(reg_both, 100, 0, 1)

##
##          FALSE TRUE
##  FALSE    78   31
##  TRUE     55  126
##          s prédic. (en %)
##          0.45000      70.68966

pred_mat(reg_both, 100, 0.4, 0.5)

##
##          FALSE TRUE
##  FALSE    78   30
##  TRUE     55  127
##          s prédic. (en %)
##          0.43900      70.68966

```

Le meilleur modèle proposé par les méthodes de sélection automatique est à mon sens moins bon que notre modèle *pluie7*. En effet, bien que l'AIC soit un peu inférieure,

- Le nombre de covariables présente est trop important.
- la qualité de la prédiction est moins bonne (70.690 %)

-> Je conserve donc le modèle *pluie7* comme base de travail.

C. Amélioration du modèle de référence

Je pars du modèle *pluie7*. Il possède l'avantage d'avoir toutes ses covariables significatives, mais ne capte pas toute la variabilité des données. Pour rappel, le modèle c'est une régression logistique contenant les covariables suivantes :

Year, Tmoy2, Prmoy, Wsmoy80, Wdmoy80, Wdmoy900, Prmax, Prmin, TNebmin, MNebmax, Wsmax10, Wsmin10, Wgm

```

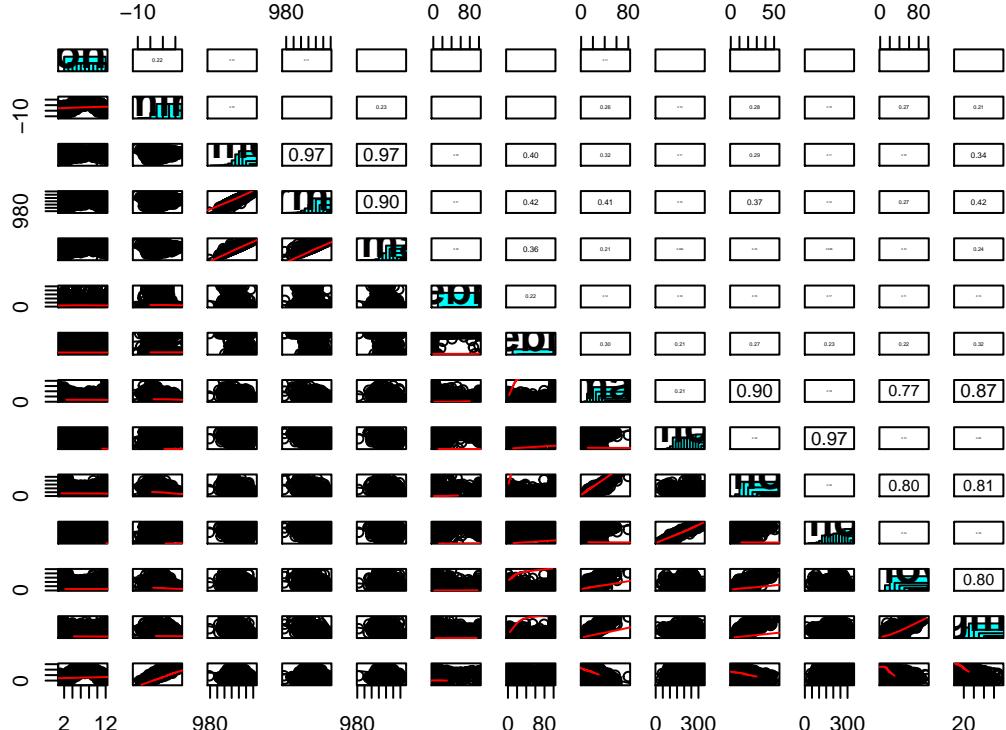
# Scatterplot

# préparation des données

met5 <- met3[,c(2,5,10,11,12,16,23,31,32,33,36,37,43,45)]

pairs(met5,diag.panel=panel.hist,cex.labels=2,font.labels = 2,
      upper.panel=panel.cor, lower.panel=panel.lm)

```



1. Vue d'ensemble - Scattreplot

Le nombre important de covariables entraîne une difficulté de lecture des résultats :

- bien que certaines covariables soient fortement corrélées, elles ne le sont pas avec toutes les autres (il est donc délicat de les enlever du modèle avec ce simple critère)
- Certaines régressions semblent indiquer des essais possibles d'interactions exotiques du type " $y : \sqrt{x}$ " (mais l'interprétation s'avèrerait assez délicate)

Je décide donc de procéder autrement.

2. Ajout des intercations Il paraît possible que la température soit liée à la pression atmosphérique. En effet, les fortes pressions sont généralement signes de temps non pluvieux et à contrario...

```
cor.test(met3$Prmax,met3$Tmoy2)
```

```
##
## Pearson's product-moment correlation
##
## data: met3$Prmax and met3$Tmoy2
## t = -7.5625, df = 1178, p-value = 7.929e-14
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.2689435 -0.1600770
## sample estimates:
## cor
## -0.2151787
```

La corrélation est forte (p-valeur très faible), il peut être intéressant d'en étudier l'interaction dans le choix d'un modèle idéal. Par ailleurs, je remarque que, dans ce modèle, la corrélation est négative (information confirmée par le fait que l'intervalle de confiance ne contienne pas 0). Cela laisse penser que la température et la pression maximale tendent à évoluer en "variations opposées" (ce qui tendrait à contredire mon intuition précédente)

```
pluie7b=glm(pluie.demain~ -1 + Year+ Tmoy2*Prmax + Prmoy +Wsmoy80 +Wdmoy80 +Wdmoy900 +Prmin +TNebmin +M
,family = binomial, data=met3)
```

L'AIC est meilleure ! (1259 au lieu de 1284), mais la covariable "Year" n'est désormais plus du tout significative (la p-valeur très élevée). Je l'enlève du modèle.

```
pluie7c=glm(pluie.demain~ -1 + Tmoy2*Prmax + Prmoy +Wsmoy80 +Wdmoy80 +Wdmoy900 +Prmin +TNebmin +MNebmax
,family = binomial, data=met3)
```

Deux covariables sont à limite du seuil de significativité. Je décide de les conserver (temporairement) en attendant de voir si d'autres interactions pourraient "pencher les seuils du bon côté".

J'essaie d'augmenter la précision de la qualité de l'approximation de la prédiction en augmentant le pas du balayage.

```
pred_mat(pluie7c,10,0,1)
```

```
##
##          FALSE  TRUE
##  FALSE     89    46
##  TRUE      44   111
##
##          s prédic. (en %)
##          0.50000    72.41379
```

```
pred_mat(pluie7c,100,0,1)
```

```
##
##          FALSE  TRUE
##  FALSE     77    24
##  TRUE      56   133
##
##          s prédic. (en %)
##          0.38000    72.75862
```

```
pred_mat(pluie7c,20,0.365,0.367)
```

```
##
##          FALSE  TRUE
##  FALSE     76    21
##  TRUE      57   136
##
##          s prédic. (en %)
##          0.36560    73.10345
```

La prédiction est meilleure ! (p=73,1 % pour une seuil s=0,365), le tout pour un AIC à 1257.

Par ailleurs, le temps de calcul est désormais quasi immédiat.

```
system.time(pred_mat(pluie7c,20,0.365,0.367))
```

```
##
##          FALSE  TRUE
##  FALSE     76    21
##  TRUE      57   136
##
##    user  system elapsed
##    0.072  0.000  0.074
```

J'ôte les covariables "un peu limite" en terme de significativité (Vdmoy80, Wsmin10, Wgmax) et en ne conservant que l'interaction entre les 2 dernières.

```
pluie7d=glm(pluie.demain~ -1 + Tmoy2*Prmax + Prmoy +Wsmoy80 +Wdmoy900 +Prmin +Tmoy2:Wdmoy80 +TNebmin +M
```

J'obtiens un AIC de 1253 et toutes les variables sont désormais significatives.

```
# matrice de confusion (selon précision) et prédiction
pred_mat(pluie7d,100,0,1)
```

```
##
```

	FALSE	TRUE
## FALSE	83	31
## TRUE	50	126

```
##           s prédic. (en %)
##          0.41000      72.75862
```

```
pred_mat(pluie7d,200,0.40,0.42)
```

```
##
```

	FALSE	TRUE
## FALSE	81	26
## TRUE	52	131

```
##           s prédic. (en %)
##          0.40110      73.10345
```

```
# calcul de l'AUC
```

```
print("AUC=")
```

```
## [1] "AUC="
```

```
performance(prediction(pred(pluie7d),metpr2$pluie.demain),"auc")@y.values[1]
```

```
## [1] 0.7664384
```

Je ne progresse pas sur la qualité de la prédiction par rapport au modèle précédent, mais j'ai gagné un peu en AIC. De plus, l'AUC peut être considérée comme bonne (supérieure à 0.7).

J'étudie maintenant une interaction potentielle entre *temp.demain* et *Prmax*.

```
pluie7e=glm(pluie.demain~ -1 + (Tmoy2 + temp.demain)*Prmax + Prmoy +Wsmoy80 +Wdmoy900 +Prmin +Tmoy2:Wdm
```

Je gagne un peu en AIC, mais ..

```
pred_mat(pluie7e,100,0,1)
```

```
##
```

	FALSE	TRUE
## FALSE	83	32
## TRUE	50	125

```
##           s prédic. (en %)
##          0.43000      72.75862
```

```
pred_mat(pluie7e,10,0.42,0.44)
```

```
##
```

	FALSE	TRUE
## FALSE	83	30
## TRUE	50	127

```

##           s prédic. (en %)
##      0.42200      72.75862

```

j'ai ajouté une covariable au modèle et je perds en qualité prédictive.

Je décide de tester diverses interactions en observant la significativité de ces dernières tout en cherchant à minimiser l'AIC des modèles.

Je récapitulerai l'ensemble de ces résultats dans un tableau.

```

pluie7f=glm(pluie.demain~ -1 + (Tmoy2 + temp.demain)*Prmax +temp.demain*Wdmoy900+ Prmoy +Wsmoy80 +Wdmoy
             ,family = binomial, data=met3)

pred_mat(pluie7f,20,0.41,0.43)

##
##          FALSE TRUE
##    FALSE     82    27
##    TRUE      51   130

##           s prédic. (en %)
##      0.41200      73.10345
#


pluie7g=glm(pluie.demain~ -1 + (Tmoy2 + temp.demain)*Prmax +temp.demain*(Wdmoy900+ 0)+ Prmoy +Wsmoy80 +Wsmoy
             ,family = binomial, data=met3)

pred_mat(pluie7g,100,0.38,0.4)

##
##          FALSE TRUE
##    FALSE     79    26
##    TRUE      54   131

##           s prédic. (en %)
##      0.38220      72.41379
#


pluie7h=glm(pluie.demain~ -1 + (Tmoy2 + temp.demain)*Prmax +temp.demain*(Wdmoy900 + Wsmax10)+ Prmoy +Wsmoy
             ,family = binomial, data=met3)

pred_mat(pluie7h,100,0.33,0.36)

##
##          FALSE TRUE
##    FALSE     72    20
##    TRUE      61   137

##           s prédic. (en %)
##      0.33840      72.06897
#


pluie7i=glm(pluie.demain~ -1 + (Tmoy2 + temp.demain)*Prmax +Prmax:sqrt(Wgmax) +temp.demain*(Wdmoy900 + Wsmoy
             ,family = binomial, data=met3)
# summary(pluie7i)

pred_mat(pluie7i,100,0.3,0.5)

```

```

##          FALSE TRUE
## FALSE      76   25
## TRUE       57 132
##           s prédic. (en %)
##        0.34800     71.72414

```

Chaque amélioration supplémentaire fait baisser un peu l'AIC (je n'ai pas indiqué les tests non concluants), mais la qualité prédictive ne s'améliore pas et le nombre de covariables intégrées s'incrémentent de 1 à chaque nouvel essai, compliquant ainsi le modèle et son interprétation.

Aiguillé par le scatterplot, j'effectue un dernier essai en ajoutant l'interaction $Prmax : \sqrt{Wgmax}$.

L'AIC baisse encore d'un point (1231), mais le pouvoir prédictif n'évolue pas contrairement au nombre de covariables et à la complexité d'interprétation.

Je décide d'arrêter mes essais ici.

D. Récapitulatif

J'ai choisi *pluie7* comme modèle de base, puis j'ai cherché à étudier diverses interactions entre les covariables explicatives afin d'en minimiser l'AIC tout en maximisant le pouvoir prédictif.

Pour rappel, le modèle *pluie7* est une régression logistique contenant les covariables suivantes :

Year, Tmoy2, Prmoy, Wsmoy80, Wdmoy80, Wdmoy900, Prmax, Prmin, TNebmin, MNebmax, Wsmax10, Wsmin10, Wgm

Voici un tableau résumant les divers résultats obtenus.

modèles	caractéristiques du modèle	nbre de cov.	AIC	s	qualité de la prédiction / (sur 290)
pluie7	aucune interaction	14	1284	0.422	72.069% (209)
pluie7b	(pluie7) + $Tmoy2 * Prmax$	15	1259	0.367	73.103% (212)
pluie7c	(pluie7b) - <i>Year</i>	14	1257	0.366	73.103% (212)
pluie7d	(pluie7c) - ($Vdmoy80 + Wsmin10 + Wgmax$) + $Wsmin10:Wgmax$	12	1253	0.365	73.103% (212)
pluie7e	(pluie7d) + $temp.demain * Prmax$	13	1248	0.413	72.759% (211)
pluie7f	(pluie7e) + $temp.demain * Wdmoy900$	14	1245	0.413	73.103% (212)
pluie7g	(pluie7f) + $temp.demain:MNebmax$ - <i>MNebmax</i>	15	1233	0.382	72.414% (210)
pluie7h	(pluie7g) + $temp.demain * WSmax10$ - $Tmoy2:Wdmoy80$	15	1232	0.339	71.724% (208)
pluie7i	(pluie7h) + $Prmax:\sqrt{Wgmax}$	16	1231	0.348	71.724% (208)

Le meilleur compromis entre :

- nombre minimal de variables explicatives
- AIC la plus basse
- pouvoir prédictif le plus élevé

m'enclint à choisir le modèle *pluie7d* (bien que l'AIC ne soit pas la plus faible). En effet, ce modèle possède un pouvoir prédictif maximal et un nombre de covariable restreint (12 covariables). Mon 2nd choix se serait porté sur le *pluie7f* (même pouvoir prédictif, AIC meilleure de 2% mais 2 covariables supplémentaires).

Voici l'export des prédictions associées au modèle *pluie7d*.

```

# Export des prédictions associées au modèle "temp4c2"

pluie.demain.pred=cbind(pred(pluie7d)>=0.40110,metpr2[44])
colnames(pluie.demain.pred)=c("pred.", "observ.")
head(pluie.demain.pred)

##   pred. observ.
## 1 FALSE  FALSE
## 2 TRUE  FALSE
## 3 TRUE  FALSE
## 4 TRUE   TRUE
## 5 TRUE   TRUE
## 6 FALSE  TRUE

write.table(pluie.demain.pred, "pluie.demain.pred.csv", row.names=FALSE, sep=",")

```

E. Modèles obtenus avec d'autres fonctions de lien

```

pluie7db=glm(pluie.demain~ -1 + Tmoy2*Prmax + Prmoy +Wsmoy80 +Wdmoy900 +Prmin +Tmoy2:Wdmoy80 +TNebmin +
               ,family = binomial(link=probit), data=met3)
AIC(pluie7db)

## [1] 1253.881

pluie7dc=glm(pluie.demain~ -1 + Tmoy2*Prmax + Prmoy +Wsmoy80 +Wdmoy900 +Prmin +Tmoy2:Wdmoy80 +TNebmin +
               ,family = binomial(link=cauchit), data=met3)
AIC(pluie7dc)

## [1] 1257.923

pluie7dd=glm(pluie.demain~ -1 + Tmoy2*Prmax + Prmoy +Wsmoy80 +Wdmoy900 +Prmin +Tmoy2:Wdmoy80 +TNebmin +
               ,family = binomial(link=cloglog), data=met3)
AIC(pluie7dd)

## [1] 1257.392

```

Les fonctions de lien utilisant “probit”, “cauchit” et “cloglog” donne des AIC supérieures à celles de “logit” (1254, 1258 et 1257 respectivement), je conserve donc la fonction de lien “logit”.

V. Mesures prédictives des qualités prédictives (par validation croisée)

Je propose d'évaluer les qualités prédictives du modèle *pluie7d* en effectuant une validation croisée de type “l-folds”.

Pour cela, je divise les données *met3* en l parties. Chacune (tour à tour) des l parties servant de jeu “test” alors que le reste du jeu est dévolue au jeu “train”. Chaque modèle testé sur l'ensemble des l parties fournit l qualités de prévision (construites avec des jeux de données restreints utilisant une fois et une seule l'ensemble des données initiales).

Je modélise le taux de prédition théorique du modèle en répétant la procédure précédente k fois. Ce taux est tout simplement la moyenne des kl résultats précédents.

J'effectue ensuite la même procédure en substituant le jeu “test” au profit de “metpr2” (le jeu de données servant initialement à tester le modèle prédictif).

Mon objectif est de vérifier que le taux de bonnes prédictions de 73.1% est assez proche “du mieux que le modèle retenu peut proposer”.

Plusieurs remarques :

- Le choix de l peut s'avérer délicat (suffisamment grand pour “apprendre” et pas trop grand pour éviter le surapprentissage). L’usage veut que l’on choisisse une valeur l vérifiant $5 \leq l \leq 15$. Je choisis $l = 10$
- Le fait d’apprendre sur des modèles restreints devrait en théorie minimiser la qualité de prédiction
- je pourrais faire cette validation sur l’ensemble des modèles proposés précédemment. La principale contrainte serait le temps de calcul..

Je reprends une partie des fonctions précédentes. Je les modifie légèrement pour les besoins de mes tests.

```
# fonction de prédiction
pred2=function(model,test){
  A=predict(model,test , type="response", se.fit=F)
  return(A)
}

# table de prédiction
B=0
pred.tab2 = function(model,n,a,b,data_test,test2){
  for (i in (0:n)) {
    A=round(seq(a,b,by=(b-a)/n),digits=9)
    length(B)=n+1
    B[i+1]=round(100*mean((pred2(model,data_test)>=(a+(i*(b-a))/n)) == (test2)),digits = 10)
  }
  C=data.frame(t(data.frame(A,B)))
  rownames(C)=c("seuil s","prédiction (en %)")
  colnames(C)=c()
  return(C)
}

# prédiction maximale et seuil associé
pred_mat2=function(model,n,a,b,data_test,test2){

  # sélection dans le tableau

  p=max(pred.tab2(model,n,a,b,data_test,test2)[2,])
  s=a+(which.max(pred.tab2(model,n,a,b,data_test,test2)[2,]))*(b-a)/n

  return(c("s"=s,"prédic. (en %)"=p))
}
```

Voici ma fonction effectuant ma validation croisée en découplant mon jeu de données en l parties.

```
val_crois =function(l,n,a,b,jeu_pred){
  B=NULL
  C=NULL
  D=NULL
  m=nrow(jeu_pred)
  for (i in 1:l){
    k = sample(m,(l-1)/l*m)
    data_train = jeu_pred[-k,]
    data_test = jeu_pred[k,]

    #utilisation des covariables intervenant dans le modèle "temp4d"

    reg=glm(pluie.demain~ -1 + (Tmoy2 + temp.demain)*Prmax +temp.demain*Wdmoy900+ Prmoy +Wsmoy80 +Wdmoy
            family = binomial, data=data_train)
```

```

C[i]=pred_mat2(reg,n,a,b,jeu_pred,jeu_pred$pluie.demain)
D[i]=pred_mat2(reg,n,a,b,jeu_pred,jeu_pred$pluie.demain)[2]

}
df9=t(data.frame(C,D))
rownames(df9)=c("s","p")
return(df9)
}

kable(val_crois(3,400,0,1,metpr2))

```

s	0.52000	0.43500	0.35250
p	71.72414	71.72414	72.06897

Cette fonction détermine l'erreur moyenne de prévision après simulation de k répétitions.

```

prev_moy=function(l,n,a,b,jeu_pred,k){
  E=NULL
  F=NULL

  for (i in 1:k){
    E=val_crois(l,n,a,b,jeu_pred)[2,]
    F=c(E,F)
  }
  F=F[-1*k-1]
  F=mean(F)
  return(F)
}

```

voici un récapitulatif après 2, 3 et 5 essais.

```

df10=data.frame(c(prev_moy(10,100,0,1,met3,2),prev_moy(10,100,0,1,met3,3),prev_moy(10,100,0,1,met3,5)),
c(prev_moy(10,100,0,1,metpr2,2),prev_moy(10,100,0,1,metpr2,3),prev_moy(10,100,0,1,metpr2,5)))
rownames(df10)=c("2 répétitions","3 répétitions","5 répétitions")
colnames(df10)=c("met3","metpr2")
kable(df10)

```

	met3	metpr2
2 répétitions	74.92373	72.18966
3 répétitions	74.79661	72.16092
5 répétitions	74.83220	72.10345

Je décide d'effectuer des simulations de plus grand effectif

```

prev_moy(10,100,0,1,met3,10)
prev_moy(10,100,0,1,metpr2,10)
prev_moy(10,100,0,1,met3,100)
prev_moy(10,100,0,1,metpr2,100)
prev_moy(10,100,0,1,met3,500)
prev_moy(10,100,0,1,metpr2,500)

```

Voici les taux de prédiction en % (obtenus auprès d'un certain temps pour les simulations comprenant un grand nombre d'essais)

nombre d'essais	<i>met3</i>	<i>metpr2</i>
10	74.8	72.1
100	74.8	72.1
500	74.8	72.1

Je constate que le jeu de test *met3* minimise l'erreur moyenne de prédiction. Cela ne semble pas illogique puisque le jeu “test” à servi à construire le modèle. L'adéquation aux données est donc meilleure qu'avec le jeu de données *metpr2*.

Au vu des résultats obtenus sur le jeu d'entraînement, je pense être assez proche de la vérité.

En effet, le jeu “test” *met3* fournit une qualité prédictive meilleure d'environ 3.7%, mais il sous-évalue l'erreur moyenne de prédiction pour les raisons énoncées ci-dessus.

VII. Conclusion

Pour chacune de ces deux variables d'intérêt, j'ai proposé et validé un modèle (*temp4c2* pour la température et *pluie7d* pour la pluie). J'ai tenté de **sélectionner un modèle optimal** au sens de la dualité *explicatif / prédictif*.

J'ai toutefois privilégié le cadre prédictif pour la 2nde variable d'intérêt, car il m'est apparu moins important de ne pas me tromper de plus de 1°C sur la température du lendemain que de savoir si je devais ou non “prendre mon parapluie demain...”