INSTITUT NATIONAL DES SCIENCES APPLIQUÉES ROUEN

INSA ROUEN

Projet de Fin d'Études

Pluie en Australie

Théophile THIERRY

M. PORTIER

2021-2022

Table des matières

Première partie

Description des données

1 Présentation des données

Les données utilisées pour ce projet peuvent être trouvées sur le site Kaggle. Elles ont été récupérées des données du gouvernement australien, dans la partie Daily Weather Observations, et ont été complétées avec les données de la partie : Climate Data Online.

Ces données contiennent 10 ans d'observations de la météo australienne sur 49 lieux différents entre 2007 et 2017. Une observation est constituée de (presque) toutes ces variables :

- Date : date de la mesure.
- Location : localisation de la mesure.
- MinTemp : température minimale dans les 24h jusqu'à 9h du matin (en °C).
- MaxTemp : température maximale dans les 24h jusqu'à 9h du matin (en °C).
- Rainfall : précipitation dans les 24h jusqu'à 9h du matin (en mm).
- Evaporation : bac d'évaporation de classe A dans les 24h jusqu'à 9h du matin (en mm).
- Sunshine : ensoleillement en heure dans les 24h jusqu'à minuit.
- WindGustDir : direction de la plus forte rafale dans les 24 heures jusqu'à minuit (16 points cardinaux/intercardinaux).
- WindGustSpeed : vitesse de la plus forte rafale dans les 24 heures jusqu'à minuit (16 points cardinaux/intercardinaux).
- WindDir9am : direction du vent à 9h du matin.
- WindDir3pm : idem à 15h.
- WindSpeed9am : vitesse du vent à 9h.
- WindSpeed3pm : idem à 15h.
- Humidity9am : taux d'humidité relative à 9h.
- Humidity3pm : idem à 15h.
- Pressure9am : pression atmosphérique réduite au niveau moyen de la mer à 9h.
- Pressure3pm : idem à 15h.
- Cloud9am : fraction du ciel couverte par un nuage à 9h (en huitième).
- Cloud3pm : idem à 15h.
- Temp9am : température à 9h (en °C).
- Temp3pm : idem à 15h.
- RainToday : s'il a plu le jour même.
- RainTomorrow : s'il a plu le lendemain.

Jetons tout d'abord un coup d'œil aux variables numériques de nos données.

1.1 Variables numériques

Comme nous pouvons le constater dans la Table 1, pour certaines d'entre elles, il manque beaucoup d'observations (voir la ligne *NA's*). Nous allons devoir remédier à cela dans les futures parties, et principalement sur les variables *Sunshine*, *Evaporation*, *Cloud9am* et *Cloud3pm*, dont nous avons moins de 60% des observations.

Nous pouvons de plus noter certaines choses : comme on pouvait s'y attendre, il fait en général moins froid à 15h qu'à 9h, et il fait aussi moins humide.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's	Std.
MinTemp	-8.50	7.60	12.00	12.19	16.90	33.90	1485.00	6.40
MaxTemp	-4.80	17.90	22.60	23.22	28.20	48.10	1261.00	7.12
Rainfall	0.00	0.00	0.00	2.36	0.80	371.00	3261.00	8.48
Evaporation	0.00	2.60	4.80	5.47	7.40	145.00	62790.00	4.19
Sunshine	0.00	4.80	8.40	7.61	10.60	14.50	69835.00	3.79
WindGustSpeed	6.00	31.00	39.00	40.04	48.00	135.00	10263.00	13.61
WindSpeed9am	0.00	7.00	13.00	14.04	19.00	130.00	1767.00	8.92
WindSpeed3pm	0.00	13.00	19.00	18.66	24.00	87.00	3062.00	8.81
Humidity9am	0.00	57.00	70.00	68.88	83.00	100.00	2654.00	19.03
Humidity3pm	0.00	37.00	52.00	51.54	66.00	100.00	4507.00	20.80
Pressure9am	980.50	1012.90	1017.60	1017.65	1022.40	1041.00	15065.00	7.11
Pressure3pm	977.10	1010.40	1015.20	1015.26	1020.00	1039.60	15028.00	7.04
Cloud9am	0.00	1.00	5.00	4.45	7.00	9.00	55888.00	2.89
Cloud3pm	0.00	2.00	5.00	4.51	7.00	9.00	59358.00	2.72
Temp9am	-7.20	12.30	16.70	16.99	21.60	40.20	1767.00	6.49
Temp3pm	-5.40	16.60	21.10	21.68	26.40	46.70	3609.00	6.94

TABLE 1 – Résumé des variables numériques

1.2 Variables factorielles

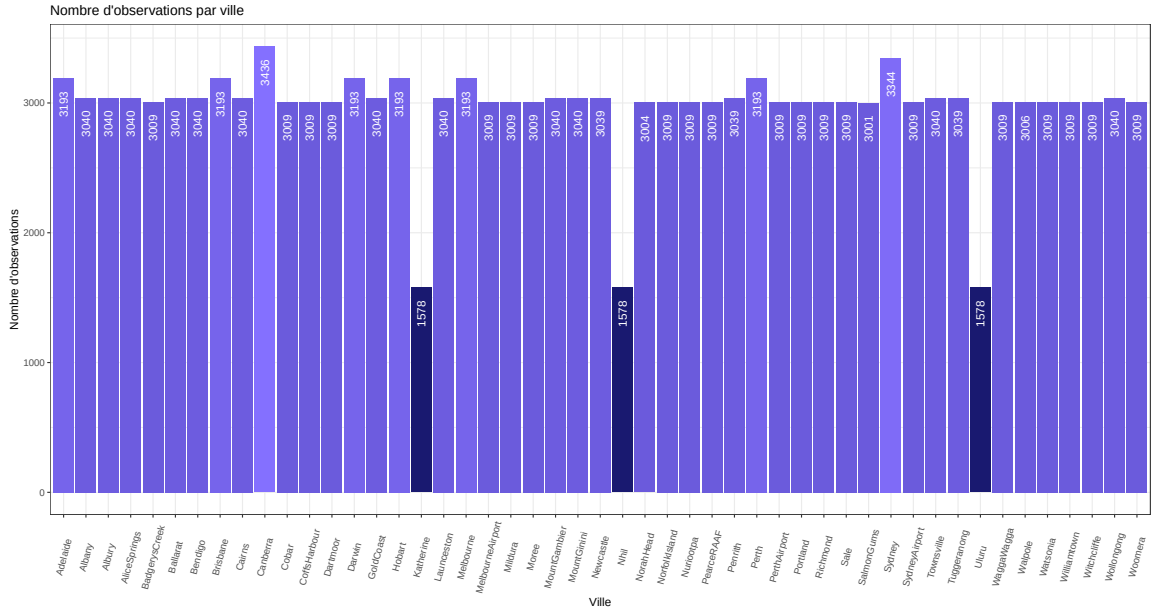

FIGURE 1 – Nombre d'observations pour chaque ville

Regardons désormais les variables avec des facteurs. Les dates, tout d'abord, vont du 2007-11-01 au 2017-06-25, ce qui représente 3524 jours. Si l'on regarde la distribution du nombre d'observations par ville, on

devrait donc voir que chacune d'entre elles en ont 3524 (voir Figure 1). On voit déjà qu'il manque certaines dates d'observations pour la plupart des villes, et pour 3 d'entre elles : Katherine, Nhil et Uluru, nous avons moins de la moitié. Ceci est dû au fait que les observations démarrent à ces endroits en 2013.

Penchons-nous désormais sur les variables de direction du vent en regardant la distribution de leurs facteurs dans les Tables 2, 3 et 4.

Les 16 niveaux utilisés pour ces variables sont tous représentés avec à peu près la même distribution. Nous souhaitons changer ces facteurs en valeurs numériques, pour cela, nous allons remplacer chaque direction par sa valeur en degrés : utiliser l'angle pour la directions nous servira peut-être lorsque nous utiliserons des modèles de prédiction comme des arbres CART. En effet, baisser le nombre de facteurs que nous avons nous permettra sûrement de réduire le nombre de feuilles que nous aurions pu avoir. Nous utiliserons les valeurs indiquées dans la Table 5.

	E	ENE	ESE	N	NE	NNE	NNW	NW	S	SE	SSE	SSW	SW	W	WNW	WSW
Degré	0.0	22.5	45.0	67.5	90.0	112.5	135.0	157.5	180.0	202.5	225.0	247.5	270.0	292.5	315.0	337.5

Table 5 – Les 16 points cardinaux en degrés

On remarque une fois de plus qu'une partie de ses observations sons manquantes (plus de 7% pour *WindGustDir* et *WindDir9am* et un peu moins de 3% pour *WindDir3pm*)

Enfin, nous avons les deux dernières variables booléennes concernant la pluie (Tables 6 et 7). On voit que notre base de données est déséquilibrée : la variable que nous voulons prédire étant *RainTomorrow*, nous voulons avoir un équilibre entre les observations *Yes* et les observations *No*. Pour veiller à ceci, nous utiliserons des méthodes de resampling telle que SMOTE.

On remarque de plus que nous avons 2.25% des variables manquantes pour ces deux variables. Ceci s'explique par le fait qu'il manque 2.25% des observations de la variable *Rainfall*, sur laquelle sont basées ces deux variables.

Enfin, nous avons la variables *Location* qui contient tous les lieux d'observations de la base de données. Nous nous pencherons sur celle-ci dans la section Cartographie.

Avant de nous lancer plus loin, nous allons tout d'abord modifier la base de données pour la rendre utilisable. Pour cela, nous devons nous occuper des variables qui contiennent des facteurs et les changer en numériques, comme nous avons fait pour les variables de direciton du vent. Nous allons ensuite remplacer la variable lieu avec une variable de longitude et de latitude, pour des raisons que nous expliquerons dans la section suivante. Enfin, nous allons rajouter une variable correspondant aux climats de chaque lieu (on prendra pour cela une carte des climats et on pourra rentrer à la main chaque climat de chaque lieu). Pour ce qui est de la variable date, nous la remplacerons par une variable saison à seulement 4 niveaux. Ce choix sera expliqué dans une partie sur le climat et sur les périodicités. Au final, nous pourrons nous occuper du traitement des observations manquantes.

2 Cartographie

Notre base de donnée comprend donc une variable *Location*, qui est une variable qualitative avec le nom du lieu de mesure. Nous en avons 49 différentes, et afin de visualiser un peu mieux ces différents points d'observation, nous voulons les afficher sur une carte.

Pour cela, nous allons utiliser le paquet R "rnaturalearth", qui nous offre un moyen simple de dessiner nos propres cartes en utilisant le standard WGS84 (World Geodetic System).

	E	ENE	ESE	N	NE	NNE	NNW	NW	S	SE	SSE	SSW	SW	W	WNW	WSW	NA's
Compte	9181	8104	7372	9313	7133	6548	6620	8122	9168	9418	9216	8736	8967	9915	8252	9069	10326
%	6.31	5.57	5.07	6.40	4.90	4.50	4.55	5.58	6.30	6.47	6.34	6.01	6.16	6.82	5.67	6.23	7.10

Table 2 – Variable WindGustDir

	E	ENE	ESE	N	NE	NNE	NNW	NW	S	SE	SSE	SSW	SW	W	WNW	WSW	NA's
Compte	9176	7836	7630	11758	7671	8129	7980	8749	8659	9287	9112	7587	8423	8459	7414	7024	10566
%	6.31	5.39	5.25	8.08	5.27	5.59	5.49	6.01	5.95	6.38	6.26	5.22	5.79	5.82	5.10	4.83	7.26

Table 3 – Variable WindDir9am

	E	ENE	ESE	N	NE	NNE	NNW	NW	S	SE	SSE	SSW	SW	W	WNW	WSW	NA's
Compte	8472	7857	8505	8890	8263	6590	7870	8610	9926	10838	9399	8156	9354	10110	8874	9518	4228
%	5.82	5.40	5.85	6.11	5.68	4.53	5.41	5.92	6.82	7.45	6.46	5.61	6.43	6.95	6.10	6.54	2.91

Table 4 – Variable WindDir3pm

	No	Yes	NA's
Compte	110319	31880	3261
%	75.84	21.92	2.24

Table 6 – Variable RainToday

	No	Yes	NA's
Compte	110316	31877	3267
%	75.84	21.91	2.25

Table 7 – Variable RainTomorrow

2.1 Un standard de localisation et une projection

Afin de localiser avec précision un point sur Terre, nous avons besoin d'un standard de localisation. Un standard est basé sur un système de coordonnées géodésique. Il peut utiliser notamment un système de coordonnées en Longitude et Latitude.

2.1.1 Latitude et Longitude

Afin d'avoir une coordonnée pour n'importe quel point sur Terre, nous utilisons des coordonnées de Longitude et de Latitude. Ce sont des valeurs exprimées en degré à partir d'un degré 0 de référence.

La Terre ne peut être représentée comme une sphère car cela rendrait les coordonnées trop imprécises par rapport à la réalité. Elle est de plus arrondie aux pôles et c'est pour ces raisons que nous représentons la Terre par un éllipsoïde.

La Longitude est une coordonnée géographique représentée par une valeur angulaire, expression du positionnement est-ouest d'un point sur Terre [Wik21b]. Tous les points étant situés sur une courbure de l'élipsoïde reliant les pôle Nord et Sud et traversant l'équateur perpendiculairement ont la même longitude. Une courbure de référence, appelé "méridien" est choisi arbitrairement (le méridien de Greenwich) comme degré 0. Les valeurs de Longitude s'étendent de -180° vers l'ouest à 180° à l'est par rapport à ce méridien.

La Latitude est une coordonnée similaire mais qui à pour plan de référence l'équateur. Tous les points sur Terre ayant une même latitude forment un cercle dont le plan est parallèle à celui de l'équateur [Wik21a].

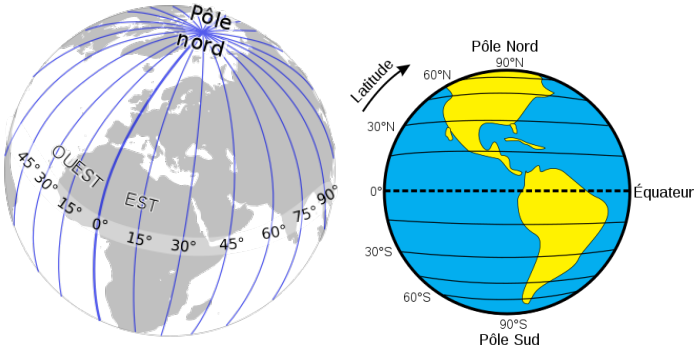

FIGURE 2 – Illustration du système de coordonnées de Longitude et Latitude

Lorsque l'on combine ce système de coordonnées et une représentation de la Terre en ellipsoïde (au travers de mesures précises des dimensions de la planète), on obtient un système géodésique.

2.1.2 Le WSG84

Le World Geodetic System 84 (WGS84) est un système géodésique, et nous pouvons l'utiliser pour nos cartes grâce au paquet "rnaturalearth". Il est notamment utilisé par le système GPS (Global Positioning System). Ce standard à été établi et est maintenu par le National Geospatial Intelligence Agency (NGA) des Etats-Unis [Wik22b] depuis 1984. Il est basé sur un ellipsoïde de référence raffiné avec le temps pour représenter au mieux la Terre, ainsi que le système de coordonnées en Longitude et Latitude.

Nous avons maintenant un moyen de localiser précisément un point sur Terre grâce à deux valeurs numériques. Pour pouvoir les afficher sur une carte, il nous faut cependant une projection.

2.1.3 Projections

La projection cartographique est "un ensemble de techniques permettant de représenter la surface de la Terre dans son ensemble ou en partie sur la surface plane d'une carte" [Wik21c]. La Terre étant sphérique, afin de l'afficher sur une carte plane, il faut la projeter. Il existe différent types de projections, certaines permettent de conserver localement les surfaces, d'autres les angles ou encore les distances sur les méridiens.

Notre paquet utilise de base une projection dite géographique : elle consiste simplement à prendre les valeurs de latitude et de longitude et des les utiliser comme si elles étaient les coordonnées X et Y (respectivement) d'un repère en deux dimensions. Cette "projection" peut avoir des résultats différents en fonction du système géodésique utilisé.

Le plus gros inconvénient de cette pratique est la distorsion des surfaces lorsque l'on s'éloigne de l'équateur. Cependant, cela est suffisant dans notre cas, où nous voulons avoir seulement une idée globale de la position des lieux observés des uns par rapport aux autres. De plus, comme nous ne prévoyons pas de mesurer précisément la distance entre deux points, ce système de "projection" géographique est le plus pratique.

2.2 Affichage sur une carte

Nous avons désormais tous les éléments pour placer les lieux sur une carte. Le paquet "rnaturalearth" nous permet donc d'avoir une liste de polygone de pays. Le paquet "ozmaps" nous permet d'avoir les polygones des états australiens. Pour les lieux, nous récupérons les latitudes et longitudes manuellement grâce à n'importe quelle base que nous pouvons trouver sur internet et nous les rajoutons à chaque observation en ajoutant deux colonnes. Au final nous pouvons afficher notre carte grâce à ggplot2 :

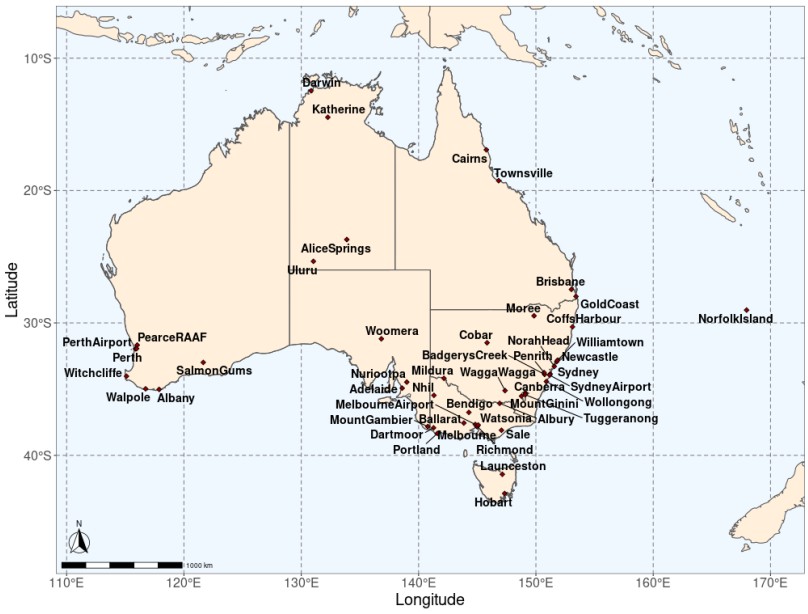

FIGURE 3 – Carte de l'Australie et villes dont la météo est observée

Nous pouvons desormais utiliser les colonnes de longitude et latitude à la place de la colonne localisation.

3 Etude des climats

Maintenant que nous pouvons afficher les lieux sur une carte, nous pouvons déterminer à quelle zone climatique appartient chaque point.

Comme on peut le voir sur la carte précédente, la plupart des observations ont lieu dans le sud-est du pays, où la concentration d'habitants et de ville est la plus grande. Cette zone correspond à un climat tempéré pour les villes les plus au sud et subtropical pour les villes plus au nord comme Brisbane. Au nord du pays nous avons les villes sur les littoraux dans une zone plus tropicale, et enfin au sud-ouest nous avons d'autre villes subtropicales. Plus à l'intérieur des terres, où il le climat est désertique, nous avons les observations de Uluru, Alice Springs et Woomera. Enfin, nous avons aussi les données de villes sur l'île de Tasmanie et l'île Norfolk.

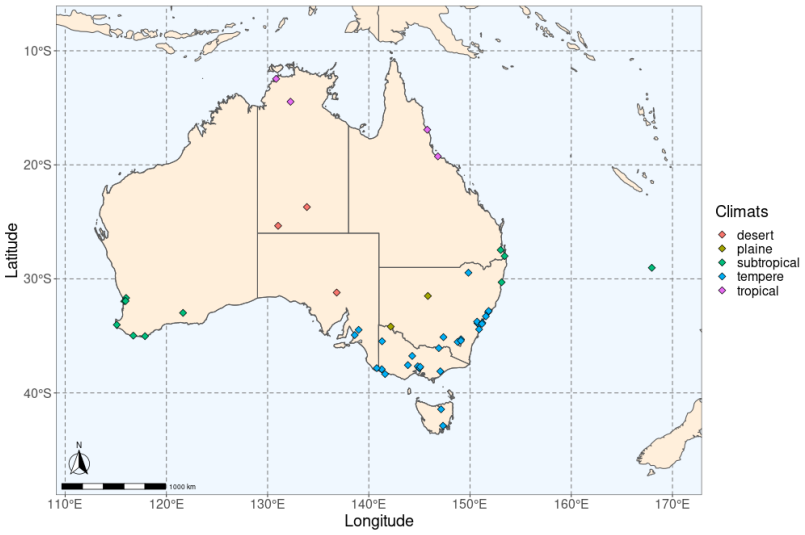

FIGURE 4 – Carte de l'Australie et climat des villes

Nous avons donc les observations de 4 lieux tropicaux, 3 lieux désertiques, 2 lieux dans les plaines (le climat de transition entre désertique et tempéré, en quelques sortes), 11 lieux subtropicaux et enfin 29 lieux tempérés. Certaines données sont liées à des villes mais d'autres à des aéroport ou encore des lieux touristiques.

3.1 Particularités des climats et périodicité

Les données étant étalées sur 10 ans, on peut trouver une périodicité dans les mesures à l'année. Nous pouvons alors, pour chaque ville, faire un graphique comprenant les moyennes des température maximales, minimales et moyennes de chaque jour sur 10 ans. Et faire de même pour les précipitations.

Nous pouvons afficher ces données en fonction des saisons. Nous pourrons ainsi remplacer la variable date par une variable avec uniquement 4 niveaux différents comme expliqué précédemment, à savoir les saisons. Nous prendrons comme saisons :

— L'été, de décembre à février
— L'automne, de mars à mai
— L'hiver, de juin à août
— Le printemps, de septembre à novembre

Nous obtenons les graphiques de la Figure 5

On remarque tout de suite que les températures les plus élevées sont aux alentours de décembre / janvier ; l'Australie étant dans l'hémisphère Sud, il s'agit de l'été.

On remarque ensuite quelques particularité dues aux climats. Dans les régions tempérée et subtropicale tout d'abord, nous avons des températures qui évoluent entre en dessous de 10 degrés et environ 30 degrés, avec en hiver (mai, juin, juillet) plus de pluie que sur le reste de l'année.

Du côté des régions dans les plaines, il pleut moins tout au long de l'année et nous n'observons pas de période de pluie comme pour les deux premières régions. Les températures sont en revanche à peu près les même, voire plus chaudes pendant l'été. Lorsque l'on se penche sur les régions désertiques, les températures sont encore plus hautes et les précipitations sont encore moins importantes, avec seulement quelques millimètres tout au long de l'année.

A l'opposé, dans les régions tropicales, la température tout au long de l'année évolue moins et reste plus proche de 30 degrés tout au long de l'année (avec une légère baisse en hiver). Dans ces régions, il pleut énormément pendant l'hiver et quasiment pas pendant l'été.

Le choix de ne garder que les saisons et pas les dates se justifie par le fait que si nous voulons prédire s'il pleut le lendemain, nous n'avons pas besoin de savoir précisemment quel jour nous sommes, voire quel mois. On remarque sur les graphiques des différences notables entre les saisons, et celle-ci suffiront sûrement pour nous aider à prédire ce que nous voulons. De plus, nous nous débarassons d'une variable avec beaucoup de facteurs différents, ce qui nous sera bénéfique lors de la mise en place de modèle de prédictions.

Nous nous retrouvons au final avec les variables suivantes :
— MinTemp, MaxTemp, Temp9am, Temp3pm
— Rainfall, RainToday, RainTomorrow, Evaporation
— Sunshine, Cloud9am, Cloud3pm
— WindGustDir, WindGustSpeed, WindDir9am, WindDir3pm, WindSpeed9am, WindSpeed3pm
— Humidity9am, Humidity3pm
— Pressure9am, Pressure3pm
— Season, Climate
— Latitude, Longitude

Qui sont toutes des variables numériques (0 ou 1 pour les variables booléennes RainToday et RainTomorrow).

4 Complétion des données

4.1 Pourquoi compléter ?

Afin de se rendre compte de la distribution des valeurs manquantes, on affiche ce qu'on appelle la missingness map de nos données (Figure 6).

Afin de faire des prédictions sur nos données, nous avons besoin de nous débarasser des observations avec des valeurs *NA*. Pour cela, on utilise la commande `na.omit`. On se retrouve avec 56420 observations sur les 145460 de base, ce qui est très peu. De plus, la plupart des lieux ne sont plus représentés comme nous l'indique la Figure 7.

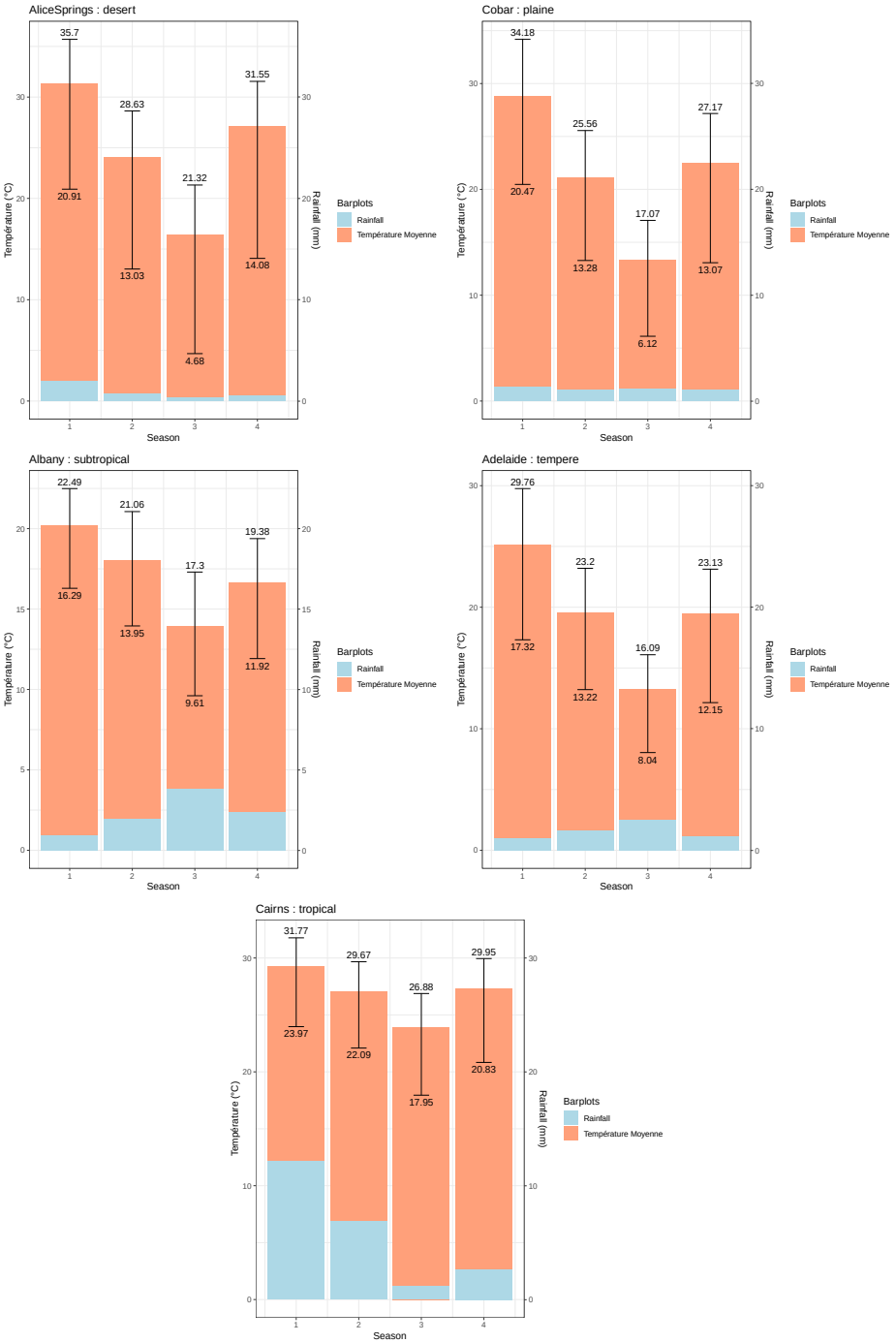

FIGURE 5 – Température Minimale et Maximale ainsi que pluviométrie au cours d'une année pour certaines des villes des données (une par climat)

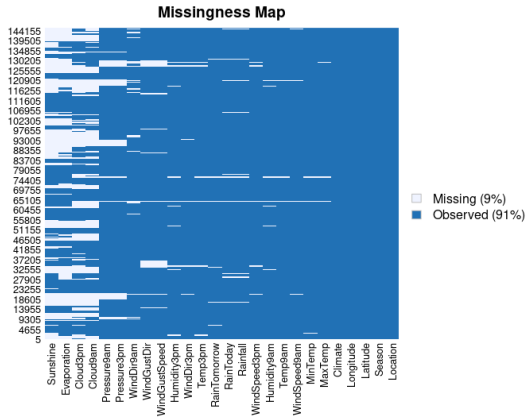

FIGURE 6 – Missingness Map des données avant la complétion.

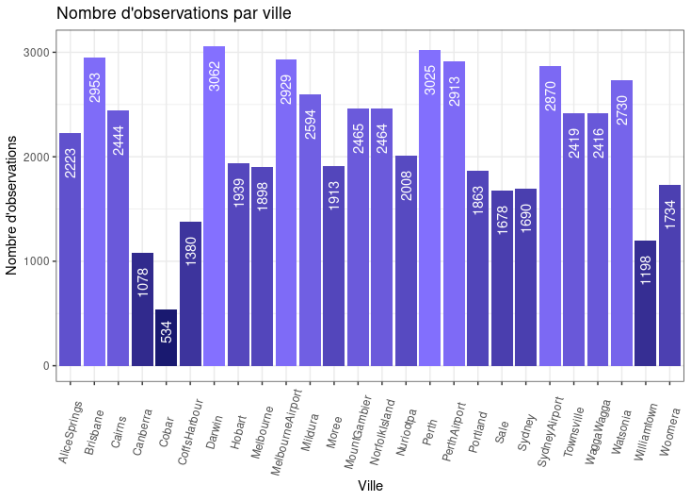

FIGURE 7 – Distribution du nombre d'observations après un *na.omit*.

Nous allons donc chercher un moyen de compléter les valeurs *NA*, et de le faire de façon à ne pas avoir trop de données redondantes et de garder une cohérence vis-à-vis des climats : nous allons copier les données d'un lieu à l'autre d'une certaine façon.

4.2 Comment compléter ?

Pour compléter nos données, nous allons regarder pour chaque variable quels sont les lieux qui ont besoin de complétion, disons ceux qui ont plus de 20% de *NA* pour cette variable, et quels sont ceux avec lesquels nous pouvons compléter : les autres qui ont plus de 2500 observations et qui ont au moins 80% de leurs observations complètes. Nous excluons ainsi les lieux dont nous n'avons pas les observations sur les dix années : lorsque nous complétons nos données, nous voulons que les dates coincident pour ne pas perdre en cohérence, ainsi nous voulons compléter les données avec les lieux pour lesquels nous avons des observation sur la période maximale d'observation de notre base de données, soit plus de 2500 jours. Le

seuil de 80% des données est choisi pour avoir un maximum d'observations sans *NA* et pour être sûr d'avoir une ville depuis laquelle copier les données.

Lorsque nous copions les données d'un lieu à un autre, nous nous soucions donc de la date d'observation. Cependant, cela ne suffira pas. Nous avons vu en effet que les différents lieux observés appartiennet à des zones climatiques très différentes. Pour chaque variable, nous allons donc associer nos deux listes de lieux (ceux à compléter et ceux avec lesquels compléter) en cherchant les lieux les plus proches de la même zone climatique.

Au final, nous nous retrouvons avec une liste de couples pour chaque variable.

Avec cette méthode, nous allons copier jusqu'à 5 fois maximum les données d'un lieu pour un autre lieu, et nous le ferons de manière "intelligente", sans perte de cohérence par rapport aux climats des lieux observés. Nous pouvons afficher quelles données de quelles villes vont compléter quelle autre villes sur des cartes comme celle de la Figure 8. On peut voir que pour les variables où il y avait le plus de *NA*, beaucoup de lieux sont complétés. Dans d'autre cas moins extrêmes, nous n'avons copié les données que d'un lieu à un autre.

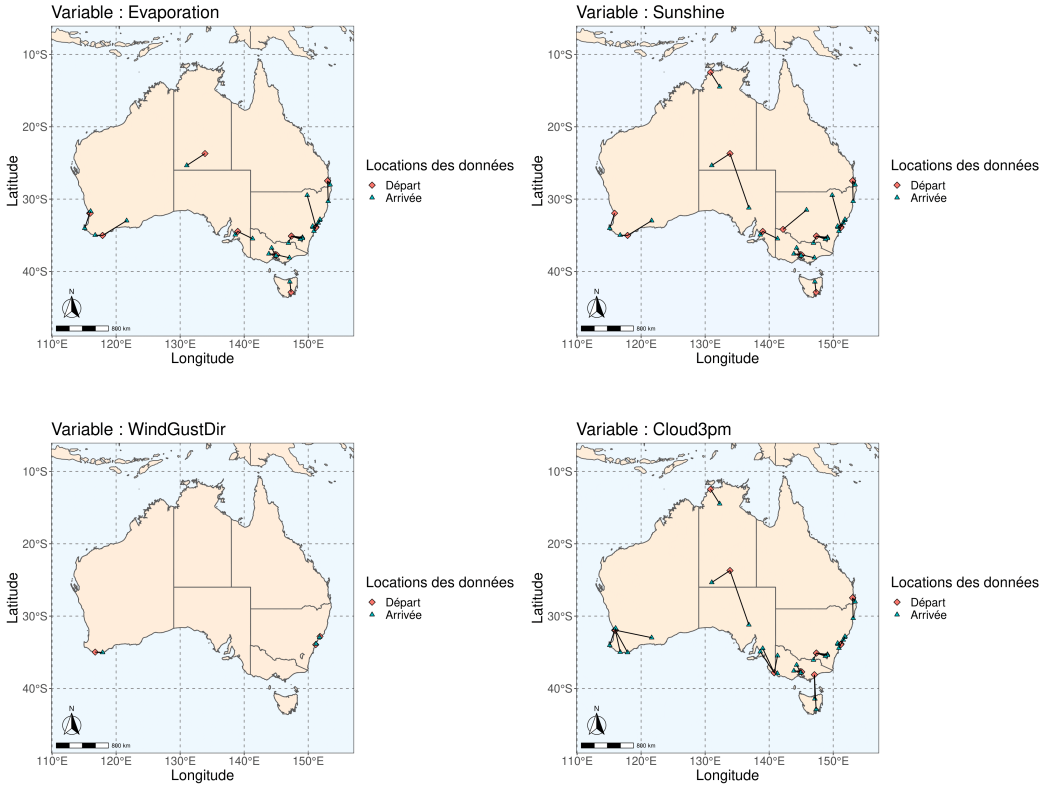

FIGURE 8 – Chemin des observations copiées (ville de départ et ville(s) d'arrivée(s)) pour certaines des variables complétées.

4.3 Après complétion

Au final, le *na.omit* nous donne une base de données avec 105546 observations. On peut réafficher la missingness map et la distribution des observations par lieu (respectivement Figures 9 et 10)

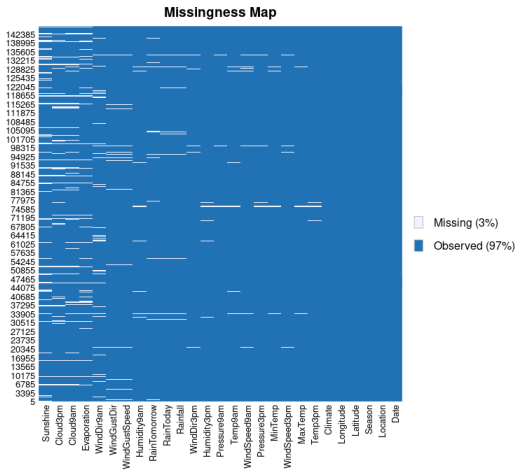

FIGURE 9 – Missingness Map des données complétées.

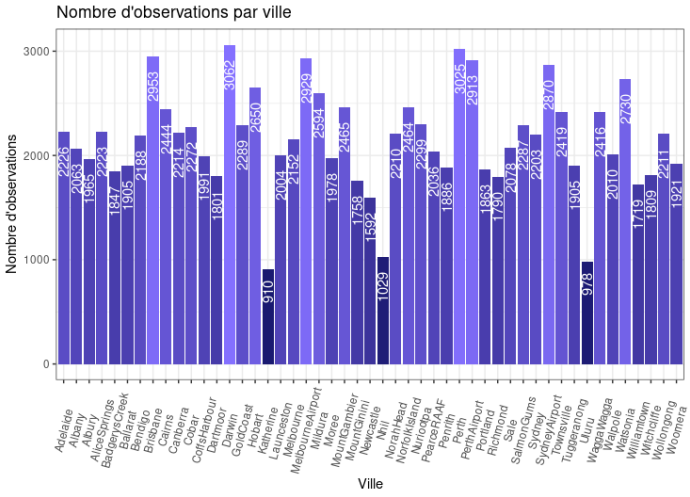

FIGURE 10 – Distribution des observations par villes dans notre base de données finale.

On remarque que certaines plages de dates n'ont pas été complétées pour certaines variables, il peut y avoir deux raisons à cela :

— Le lieu pour cette variable n'a pas été considéré comme à compléter, malgré quelques valeurs *NA* ;

— Le lieu qui a servi pour la complétion certaines dates en moins que celles du lieu à compléter.

Le nombre d'observations de notre base de données finale reste cependant satisfaisante et tous les lieux y sont représentés.

5 Relations entre les variables

Dans cette partie nous allons chercher les relations entre les variables.

5.1 Corrélations entre les variables numériques

Commençons tout d'abord par nous renseigner sur les corrélations entre les variables numériques (Figure 11).

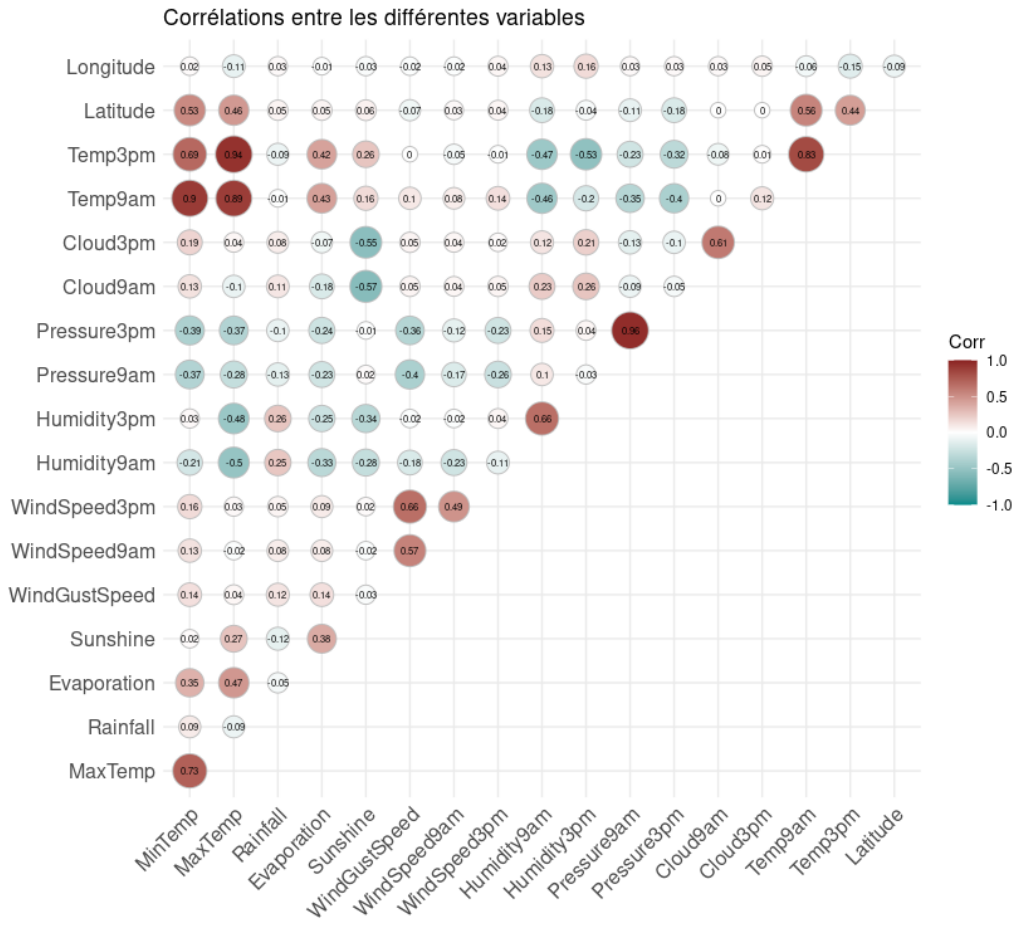

FIGURE 11 – Corrélations des variables deux-à-deux.

Nous pouvons conclure de cette figure les points suivants :

- Concernant les variables de températures :
  - MinTemp et MaxTemp sont deux variables corrélées (coeff. = 0.73). C'est un résultat attendu car ces deux valeurs sont les températures maximales et minimales de la même journée.
  - Temp9am et Temp3pm sont fortement corrélées (coeff. = 0.83). Ce résultat était attendu pour la même raison qui explique la corrélation entre MinTemp et MaxTemp.
  - MinTemp et Temp9am sont très corrélées (coeff. = 0.9). La température minimale d'une journée est atteinte aux alentours de 9h du matin, il est normal d'avoir ce résultat.

  - MaxTemp et Temp3pm sont très corrélées (coeff. = 0.94). La température maximale d'une journée est atteinte aux alentours de 15h, ce résultat était donc attendu.
  - Nous pouvons aussi remarquer que le couple Temp3pm et MinTemp et le couple Temp9am et MaxTemp sont aussi corrélés (coeff. resp. = 0.69 et 0.89). Comme ces variables sont corrélées entre elles deux à deux, il est normal de trouver ces corrélations. La température maximale est cependant plus corrélées à celle à 9h du matin que ne l'est la température minimale et la température à 15h. Cela peut s'expliquer par le fait qu'en fonction des saisons, le soleil se lève plus ou moins tôt, et que donc à 9h déjà, il peut faire très chaud. Dans tous les cas, il est normal que ces températures soient corrélées puisqu'il s'agit des écarts de températures d'une même journée. Ces derniers ne sont pas très importants et dépendent énormément de la saison (donc de la journée) comme nous l'avons vu Figure 5.
- La variable Latitude est corrélées positivement aux variables de températures (la température dépend des climats et ces climats sont très dépendants de la latitude comme on a pu le voir dans la partie cartographie).
- Pression3am et Pression9pm sont très corrélées (coeff. = 0.96) pour les même raisons que les variables de température. Il en va de même pour Humidity3pm et Humidity9am (coeff = 0.66) et WindSpeed3pm et WindSpeed9am (0.49).
- WindSpeed3pm est corrélées positivement avec WindGustDir (coeff. = 0.66).
- Les variables Cloud9am et Cloud3pm sont corrélées négativement avec la variable Sunshine (-0.57 et -0.55 respectivement) ce qui est logique car ils mesurent à peu près la même chose.
- Les variables d'humidité sont corrélées négativement à MaxTemp (-0.48 et -0.5).

5.2 Boxplots pour les variables à facteurs

Affichons maintenant les boxplots des valeurs des différents variables continues en fonction de nos variables à facteurs.

Pour la variable Climate (Figure 12), on voit une fois de plus la différence entre les climats de désert et de plaine et les autres au niveau de leur humidité, cela se voit aussi avec les grandes valeurs de Rainfall. On peut aussi voir que les climats sont arrangés en couche du nord au sud de l'Australie, comme on pouvait voir sur la carte plus haut. En effet, la région tropicale est au nord du pays, comme l'indique la boîte à moustache concernant la Latitude. Tout au sud on trouve la région tempérée. On peut aussi trouver une différence de températures entre ces 5 climats, avec une distribution ressemblant celle de la Latitude : la région tropicale est la plus chaude, et la moins chauds est la région tempérée.

Penchons-nous désormais sur les boîtes à moustache de la variable Season (Figure 13). On trouve la ditribution des températures auxquelles nous nous attendions : il fait plus chaud au été et moins en hiver. Ce même schéma est visible pour le taux d'ensolleillement (et donc un peu pour les variables des nuages). Les variables de pression et d'humidité évolues similairement : elles augmentent en hiver. En effet la pression peut changer en fonction des dépressions (qui apportent souvent le mauvais temps) ou la vitesse du vent [Wik22a].

Jetons maintenant un coup d'oeil aux boîtes des variables RainToday (Figure 14) et RainTomorrow (Figure 15). Comme on peut s'y attendre elles sont très similaires. En effet, d'un jour au lendemain, il n'y a pas de changement énorme. Comme ces deux variables sont liées (Si RainToday alors RainTomorrow pour le jour d'avant), nous avons ces similarités. Une différence notable est celle des boîtes affichées pour la variable Rainfall : par construction de la variable RainToday, il n'y à pas d'observations ou Rainfall a une valeur supérieure à 0 et ou RainToday est à 1. Ça n'est pas le cas pour RainTomorrow évidemment, il y aura toujours un moment ou il pleut aujourd'hui mais il ne pleuvra pas demain (sinon la pluie ne s'arrêterait jamais).

Comme on peut s'y attendre, les variables d'humidité sont plus elevée lorsqu'il pleut (RainToday). On peut voir cependant ces mêmes différences pour RainTomorrow, avec quelques variations : l'humidité à 15h

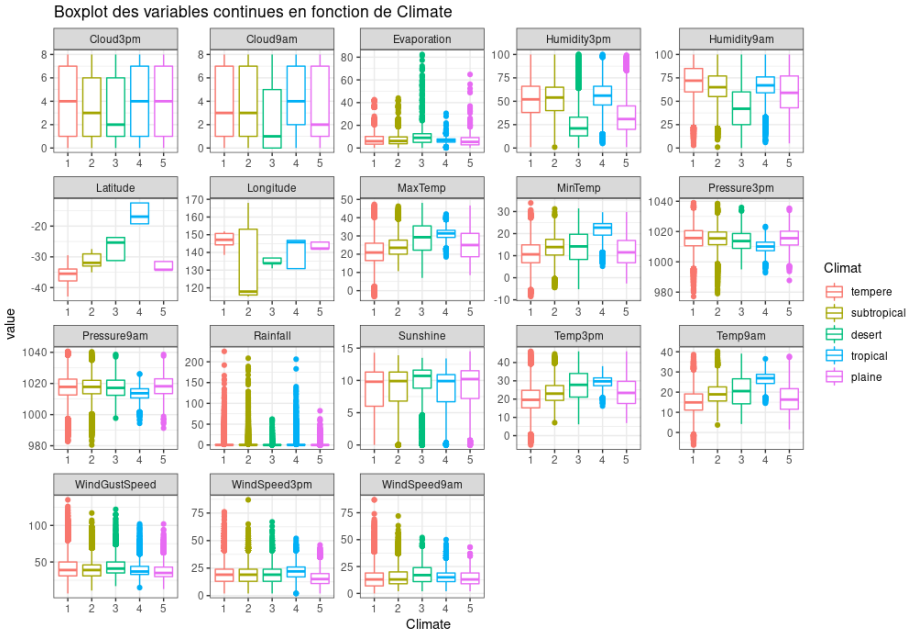

FIGURE 12 – Boxplots pour la variable Climate

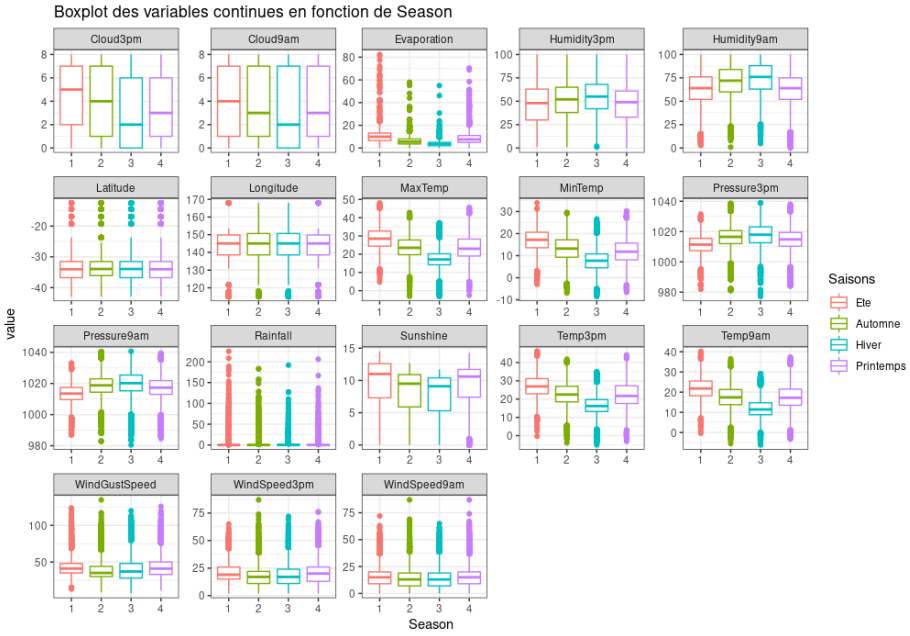

FIGURE 13 – Boxplots pour la variable Season

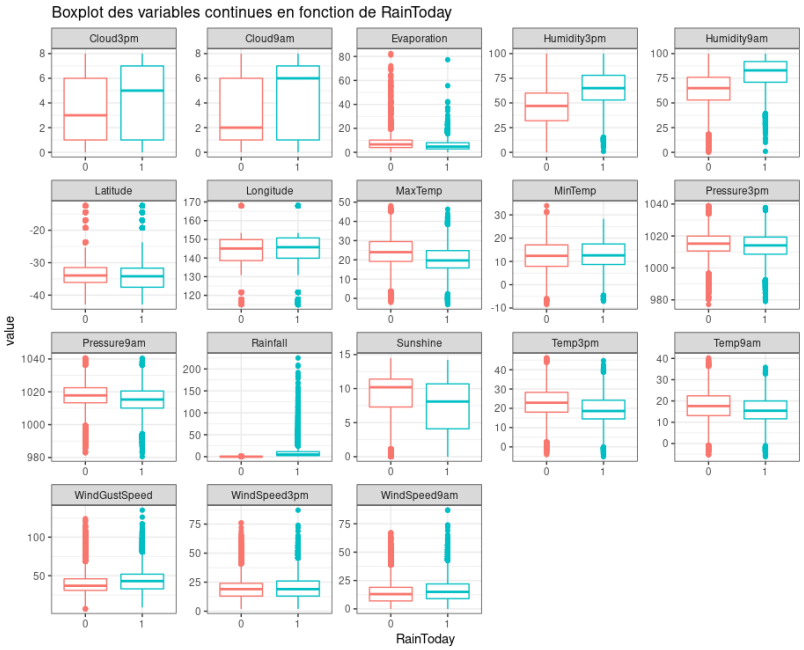

FIGURE 14 – Boxplots pour la variable RainToday

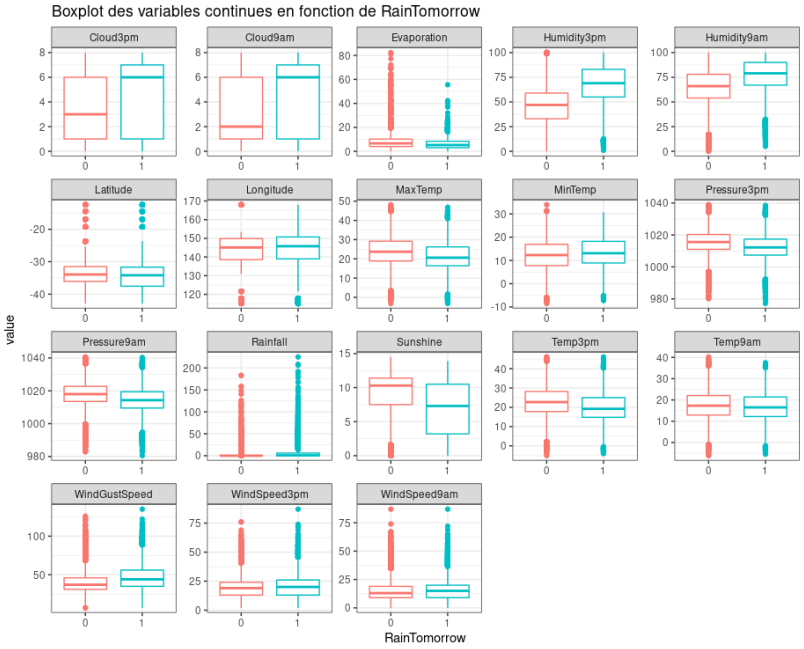

FIGURE 15 – Boxplots pour la variable RainTomorrow

et à 9h est plus elevée lorsqu'il pleut le lendemain (cette différence est surtout notable pour Humidity3pm, les distances inter-quartiles ne se chevauchent presque pas).

Deuxième partie
Prédiction

Maintenant que notre base de données est prête que nous la connaissons plus en détail, nous pouvons commencer à créer nos modèles de prédiction. Commençons tout d'abord par faire une analyse en composantes principales, pour avoir une meilleure idée de la distribution des observations.

6 ACP

Rappelons tout d'abord la distribution des observations où il pleut le lendemain et où il ne pleut pas :

RainTomorrow	Compte	%
0	82367	78.04
1	23179	21.96

TABLE 8 – Distribution des valeurs de la variable RainTomorrow dans nos données finales.

Et lorsque l'on affiche une analyse en composantes principales de ces observations, et en les colorant en fonction de leur valeur de RainTomorrow, on obtient le graphique de la Figure 16.

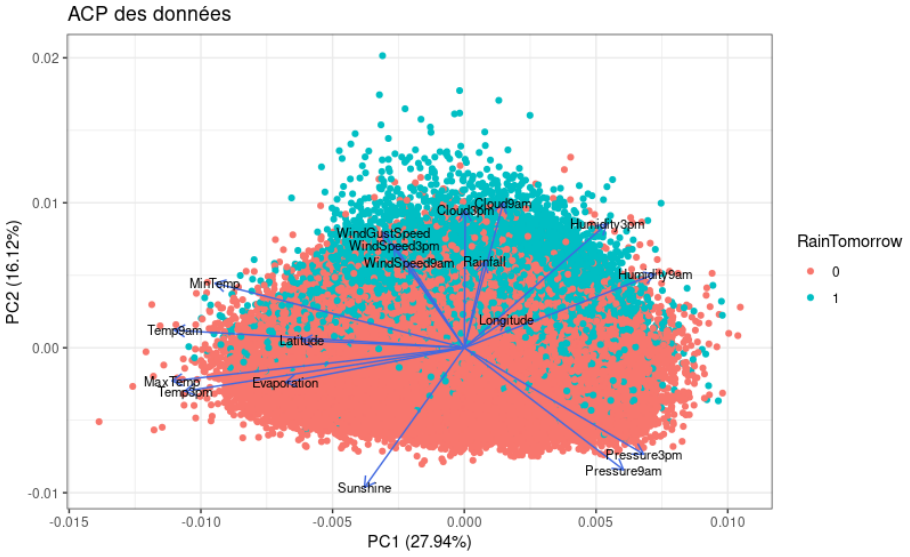

FIGURE 16 – Analyse en Composantes Principales de nos données, points colorés en fonction de RainTomorrow.

Nous allons donc chercher un moyen de séparer ces deux groupes pour faire des prédictions. On voit

déjà (avec 44.06% de l'information, l'axe 1 représentant 27.94% et l'axe 2 16.12%) que les observations pour lesquelles il pleut le lendemain sont un peu séparées de celles ou il ne pleut pas. On voit d'ailleurs bien le déséquilibre de nos données sur ce graphique.

On peut noter par ailleurs que les variables d'humidité et les nuages sont importantes pour différencier ces deux groupes. En effet les valeurs de ces variables sont plus élevées pour RainTomorrow = 1. A l'inverse, lorsque la valeur de Sunshine est élevée on va plutôt vers le groupe RainTomorrow = 0. On peut aussi s'amuser à regarder les 2 composantes principales suivantes pour avoir un autre point de vue (on a cette fois-ci 28.51% de l'information, Figure 17).

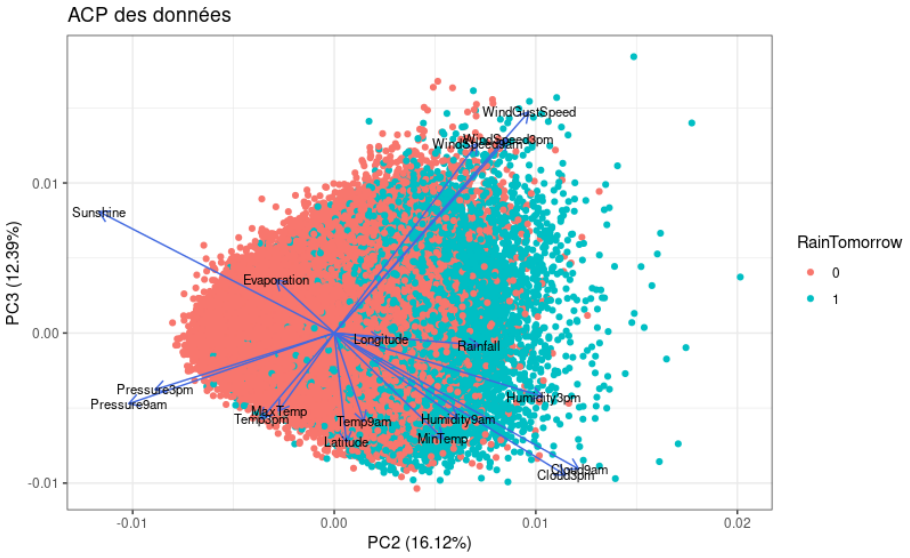

FIGURE 17 – Analyse en Composantes Principales de nos données, axes 2 et 3, points colorés en fonction de RainTomorrow.

Il semble aussi que les variables de pression sont importantes pour différencier ces deux groupes.

7 Premières prédictions

Dans cette partie nous allons prendre telle quelle la base de données, sans nous soucier de son déséquilibre. Nous allons mettre en place plusieurs modèles de prédiction :

— Un modèle de régression linéaire
— Un arbre CART
— Un modèle de Random Forest

Evidemment, nous séparerons nos données en deux échantillon : un échantillon d'apprentissage (dataApp), correspondant à 80% des données choisies de manière aléatoire, et un échantillon pour les tests (dataTest) correspondant aux 20% restants.

Nous allons construire ces modèles sur différentes "versions" de notre base de données : équilibrée ou non, avec du *one-hot encoding* ou non, centrées ou non (à chaque fois séparés en dataApp et dataTest).

7.1 Le *One-hot encoding*

Lorsque nous avons une variable à plusieurs niveaux, dans notre cas les variables Climate et Season, nous pouvons les transformer pour nous servir de l'information qu'elles nous procurent avec des variables numériques.

Pour chaque niveau de la variable, nous allons créer une variable. Ainsi, comme nous avons 4 saisons, nous aurons 4 nouvelles variables : Season.1, Season.2, Season.3 et Season.4. Comme nous avons déjà fait un travail pour réduire le nombre de niveaux pour les variables avec des facteurs, nous n'aurons pas beacoup plus de colonnes dans notre base de données.

Pour remplir ces nouvelles variables, nous allons mettre un "1" dans la colonne qui correspond au niveau de la saison de la mesure et "0" dans les autres. Imaginons en effet qu'une observation ai eu lieu en Autmone (c'est-à-dire que sa valeur pour la variable Season est 2), alors les 4 colonnes seront remplies de la façon, suivante :

Season.1	Season.2	Season.3	Season.4
0	1	0	0

Nous faisons de même avec Climate. Ainsi nous passons de 2 colonnes à 9 (extrait du début des variables) :

Season.1	Season.2	Season.3	Season.4	Climate.1	Climate.2	Climate.3	Climate.4	Climate.5
1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00

Nous voila maintenant avec une base de données ne contenant que des variables numériques et une variable (RainTomorrow) de facteurs. Nous pouvons utiliser des techniques d'upSampling comme le SMOTE (nous rentrerons dans les détails plus tard), et nous pourrons voir si cela change quelque chose aux performances des modèles.

7.2 Une première régression linéaire logistique

Commençons par un modèle de régression logistique. Pour cela, nous allons utiliser la fonction *glm* du paquet *stats* de R. En utilisant le paramètre "family = binomial", nous pouvons faire une régression logistique, qui prend en compte les variables factorielles. Nous ne faisons donc pas encore de *one-hot encoding* et gardons la base de données telle quelle. Notons aussi que nous ne centrons et réduisons pas les données numériques.

Un summary de ce modèle nous montre :

```
Call:
glm(formula = RainTomorrow ~ ., family = binomial, data = dataApp)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.0799  -0.5630  -0.3302  -0.1500   3.2469
```

```
Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)    4.538e+01  1.790e+00  25.349  < 2e-16 ***
MinTemp        1.567e-02  5.504e-03   2.847  0.00442 **
MaxTemp       -6.312e-02  5.926e-03 -10.652  < 2e-16 ***
Rainfall       1.668e-02  1.584e-03  10.526  < 2e-16 ***
Evaporation    2.418e-04  3.100e-03   0.078  0.93783
Sunshine      -6.412e-02  3.735e-03 -17.168  < 2e-16 ***
WindGustDir    1.754e-04  1.318e-04   1.331  0.18333
WindGustSpeed  5.620e-02  1.162e-03  48.359  < 2e-16 ***
WindDir9am    -3.727e-04  1.190e-04  -3.132  0.00174 **
WindDir3pm    -7.989e-05  1.329e-04  -0.601  0.54767
WindSpeed9am  -1.354e-02  1.573e-03  -8.604  < 2e-16 ***
WindSpeed3pm  -3.047e-02  1.639e-03 -18.597  < 2e-16 ***
Humidity9am    9.532e-03  1.077e-03   8.852  < 2e-16 ***
Humidity3pm    5.313e-02  1.023e-03  51.915  < 2e-16 ***
Pressure9am    1.019e-01  5.925e-03  17.192  < 2e-16 ***
Pressure3pm   -1.517e-01  5.860e-03 -25.893  < 2e-16 ***
Cloud9am       2.585e-03  4.836e-03   0.534  0.59301
Cloud3pm       4.043e-02  5.098e-03   7.931 2.17e-15 ***
Temp9am        5.863e-02  8.532e-03   6.872 6.31e-12 ***
Temp3pm        3.384e-02  4.235e-03   7.989 1.36e-15 ***
RainToday1     5.710e-01  2.791e-02  20.457  < 2e-16 ***
Season2        3.692e-01  3.288e-02  11.226  < 2e-16 ***
Season3        6.623e-01  4.272e-02  15.505  < 2e-16 ***
Season4        4.318e-01  3.324e-02  12.992  < 2e-16 ***
Latitude       6.860e-03  4.780e-03   1.435  0.15121
Longitude     -1.338e-02  1.096e-03 -12.205  < 2e-16 ***
Climate2      -1.288e-01  3.931e-02  -3.276  0.00105 **
Climate3      -3.819e-01  8.957e-02  -4.264 2.01e-05 ***
Climate4      -1.001e+00  1.026e-01  -9.757  < 2e-16 ***
Climate5       1.398e-01  6.631e-02   2.108  0.03502 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 88900  on 84437  degrees of freedom
Residual deviance: 61568  on 84408  degrees of freedom
AIC: 61628

Number of Fisher Scoring iterations: 5
```

On remarque que la fonction a fait elle-même le *one-hot encoding*, et on retrouve des variables supplémentaires pour les différents niveaux des variables à niveaux. On remarque cependant qu'il y en a à chaque fois une de moins que le nombre de niveaux : en fait, cela suffit à donner l'information du dernier niveau si dans tous ceux déjà présent nous avons "0". En effet, si pour une observation nous avons les variables Climate "2" à "5" égales à "0", alors nous savons que cette observation est dans une région climatique "1".

Chose intéressante : si d'ailleurs on relançait cette fonction avec nos données après le *one-hot encoding*

que nous avons spécifié plus haut, nous verrons devant chacune des colonnes "redondantes" un coefficient égal à *NA* (nous verrons ceci après le SMOTE).

La première chose que nous pouvons remarquer de cette sortie R est que nous avons des coefficients qui ne peuvent pas être jugés comme significativement non nuls. En effet pour les variables Evaporation, WindGustDir, WindDir3pm, Cloud9am et Latitude, la p-value du test de Student est bien supérieure à 0.05. Nous pouvons donc mettre en oeuvre une sélection pas-à-pas de ces régresseurs.

La fonction *confusionMatrix* du paquet *caret* nous permet de voir rapidement les performances de notre modèle. Sur dataApp (à gauche) et dataTest (à droite) :

```
Confusion Matrix and Statistics

          y.app
y.glm         0     1
    0     62377  9722
    1      3517  8822

               Accuracy : 0.8432
                 95% CI : (0.8407,
                   0.8457)
    No Information Rate : 0.7804
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4801

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9466
            Specificity : 0.4757
         Pos Pred Value : 0.8652
         Neg Pred Value : 0.7150
             Prevalence : 0.7804
         Detection Rate : 0.7387
   Detection Prevalence : 0.8539
      Balanced Accuracy : 0.7112

       'Positive' Class : 0
```

```
Confusion Matrix and Statistics

          y.test
yt.glm        0     1
    0     15552  2454
    1       921  2181

               Accuracy : 0.8401
                 95% CI : (0.8351,
                   0.845)
    No Information Rate : 0.7804
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4706

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9441
            Specificity : 0.4706
         Pos Pred Value : 0.8637
         Neg Pred Value : 0.7031
             Prevalence : 0.7804
         Detection Rate : 0.7368
   Detection Prevalence : 0.8530
      Balanced Accuracy : 0.7073

       'Positive' Class : 0
```

Analysons ligne à ligne ces sorties :

- Nous avons tout en haut la matrice de contingence, qui nous montre la table des valeurs prédites contre les valeurs attendues. On voit que l'on a bien classé 62 377 observations en prédisant qu'il ne pleuvrait pas le lendemain. On retrouve le déséquilibre des données que nous avions déjà remarqué, car nous avons seulement 17000 observations pour lesquelles il pleut le lendemain dans cet échantillon (idem pour l'échantillon dataTest à droite).
- On peut voir que sur les deux échantillons, nous avons une précision de 84%, nous ne notons donc pas beaucoup de sur-apprentissage, donc pas de dégradation d'un échantillon à l'autre.
- Le No Information Rate nous indique la précision que nous aurons eu si nous avions prédit "0" pour toutes les observations. On voit que cette précision est très haute, de 78%, à cause encore une fois du déséquilibre de nos données.

— Cependant, la p-value donnée en dessous nous fait rejeter l'hypothèse au risque de 5% que les performances sont les mêmes entre notre régression et le cas où nous n'aurions pas de régresseurs (si nous prenions "0" pour toutes les prédictions). Notre modèle n'est donc pas complètement inutile.
— Plus bas, nous pouvons voir que nous avons une sensitivité d'environ 94% pour les deux échantillons. Cela correspond au nombre de prédictions de "0" qui sont bien classées sur tous les "0" attendus. En revanche, la spécificité (le taux de bonnes prédictions de "1" sur tous les "1" attendus) est seulement de 47%. Il s'agit encore une fois d'une conséquence du déséquilibre de nos données : le modèle sait très bien reconnaître les jours où il ne pleuvra pas le lendemain, car ce sont les observations que nous lui avons données en majorité pour s'entraîner.
— La valeur prédictive positive est la probabilité que la condition soit présente lorsque le test est positif, elle est ici de 86%. La valeur prédictive négative est la probabilité que la condition ne soit pas présente lorsque le test est négatif, elle est ici de 72% [Wik21d]. "Positif" dans notre cas est la prédiction que RainTomorrow soit à "0", c'est-à-dire qu'on prédise qu'il ne pleut pas. Ainsi lorsque le modèle prédit qu'il va pleuvoir le lendemain d'une observation, il y a 72% de chance qu'il ait raison.
— La prévalence est le score que nous aurions eu si nous avions prédit "0" pour toutes les observations.
— Afin de prendre en compte ce déséquilibre de nos données, nous devrions plutôt regarder la *Balanced Accuracy*, qui est égale à (sensibilité + spécificité)/2, qui est plus parlante pour comparer les performances de ces modèles. Ici, nous sommes autour de 71% dans les deux échantillons.

Nous voyons donc bien le problème que posent des données déséquilibrées. Sans plus attendre, essayons de résoudre ce problème. Nous avons ici plusieurs possibilités. Nous pouvons faire de l'up ou down sampling, ou bien utiliser une méthode un peu plus sophistiquée : SMOTE.

7.3 *up sampling*

La fonction *upSample* du paquet *caret* nous permet de ré-échantillonner de manière aléatoire notre base de données afin d'avoir le même nombre d'observations avec RainTomorrow égal à 1 et égal à 0. Cependant, cela à pour conséquence de réduire le nombre d'observations total. Comme nous avons 23179 observations avec RainTomorrow = 1, nous n'aurons plus alors que 23719 observations avec RainTomorrow = 0, ce qui nous emmènera loin des initiales 140000 observations de la base de données initiale. Nous privilégierons donc l'up sampling, qui fait la même chose, mais copie les données en minorité afin d'avoir 50% de chaque. Cette dernière méthode ajoute néanmoins beaucoup de données redondantes. En effet, à partir des 23179 observations que nous avons avec RainTomorrow égal à 1, nous allons en copier une partie de manière aléatoire. Cela aura pour conséquence d'avoir beaucoup d'informations redondantes. Cependant, nous pouvons utiliser cette méthode sur tout type de variables (quantitatives et qualitatives), et donc n'avons pas besoin d'utiliser le *one-hot encoding* par exemple.

Au final on voit bien qu'on a beaucoup plus de points bleus sur notre distribution (Figure 18).

7.4 Impact sur les performances de la régression logistique

Cette fois-ci, pour la régression linéaire, on trouve une *Balanced Accuracy* de 77% pour les deux échantillons (pas trop de surapprentissag encore une fois), pour une sensibilité de 78% et une spécificité de 76%. Nous avons donc équilibré les bonnes prédictions des deux classes, et d'ailleurs nous pouvons voir que la prévalence est égale à 0.5, car nous avons exactement 50% d'observations pour chaque classes.

```
Confusion Matrix and Statistics

          y.test
```

```
yt.glm     0     1
     0 12874  3919
     1  3599 12554
```

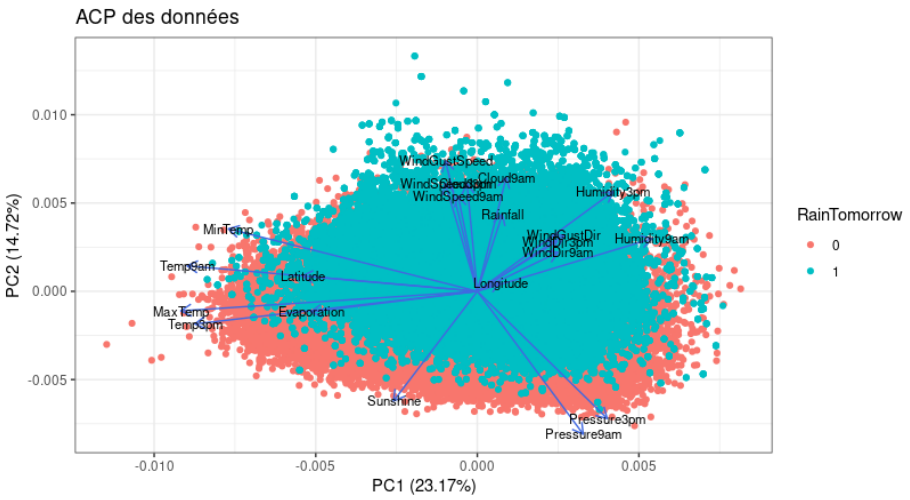

FIGURE 18 – Analyse en Composantes Principales de nos données, après upsampling, points colorés en fonction de RainTomorrow.

```
                Accuracy : 0.7718
                  95% CI : (0.7672,
                    ↪  0.7763)
     No Information Rate : 0.5
     P-Value [Acc > NIR] : < 2.2e-16

                   Kappa : 0.5436

 Mcnemar's Test P-Value : 0.0002341
```

```
            Sensitivity : 0.7815
            Specificity : 0.7621
         Pos Pred Value : 0.7666
         Neg Pred Value : 0.7772
             Prevalence : 0.5000
         Detection Rate : 0.3908
   Detection Prevalence : 0.5097
      Balanced Accuracy : 0.7718

       'Positive' Class : 0
```

7.5 SMOTE

7.5.1 Principe de la méthode

Penchons-nous alors sur la méthode SMOTE (Synthetic Minority Oversampling Technique). Cette méthode a pour but de créer de nouvelles observations dans le groupe de celles en minorité. Elle va faire en cela en traçant des lignes entre des observations proches au niveau de leur modalité et prendre un point sur cette ligne, créant ainsi une nouvelle observations.

On peut imaginer cela en 2D sur le graphe de l'ACP. On trace une droite entre deux points très proches, et on prend le milieu de cette droite. on peut alors lire la valeurs de chaque variable sur les axes. Evidemment, il s'agit de faire cela en dimension n (n étant le nombre de modalités -de variables- que nous avons).

De plus, la méthode fait ça en prenant un échantillon aléatoire dans le groupe minoritaire, et en le faisant plusieurs fois.

Cette méthode est efficace car elle créé des observations qui sont cohérentes avec la réalité. Un problème qu'il peut y avoir cependant, est que la classe majoritaire n'est pas prise en compte, et cela pose un problème pour les classes qui se superposent beaucoup.

Nous utiliserons la méthode *SMOTE* du paquet *smotefamily*.

La question que nous devons nous poser avant d'utiliser cette méthode set : que faire des variables à facteurs ? C'est ici que la technique de *one-hot encoding* entre en jeu. Une fois que nous n'aurons plus que des variables à deux niveaux (0 ou 1), nous pourrons les considérer comme des variables numériques. Nous aurons ainsi des valeurs entre 0 et 1 pour ces variables, ce qui peut représenter la ressemblance de la nouvelle observations avec tel ou tel climat, tel ou tel saison.

Au final nous nous retrouvons avec une base de données avec la distribution suivante :

RainTomorrow	Compte	%
0	82367	54.22
1	69537	45.78

TABLE 9 – Distribution des valeurs de la variable RainTomorrow après SMOTE.

7.5.2 Impact sur la régression logistique

```
    Confusion Matrix and Statistics

          y.test
yt.glm     0     1
     0 13475  3662
     1  2998 10245

               Accuracy : 0.7808
                 95% CI : (0.7761,
                   0.7854)
    No Information Rate : 0.5422
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5568

    Mcnemar's Test P-Value : 4.506e-16

               Sensitivity : 0.8180
               Specificity : 0.7367
            Pos Pred Value : 0.7863
            Neg Pred Value : 0.7736
                Prevalence : 0.5422
            Detection Rate : 0.4435
      Detection Prevalence : 0.5641
         Balanced Accuracy : 0.7773

          'Positive' Class : 0
```

Les performances sont un tout petit peu meilleures (d'un simple %). Nous privilégions cependant cette méthode car elle créée moins de redondances dans nos données que l'upsampling.

7.6 Limites de la régression linéaire

Il semblerait donc que l'on ne puisse pas faire beaucoup mieux avec la régression linéaire tout en gardant une cohérence dans les données par rapport à la réalité. Nous pouvons cependant avoir une idée des variables moins importantes dans la prédiction de la pluie en faisant une backward selection.

Références

[Wik21a] Wikipédia. *Latitude — Wikipédia, l'encyclopédie libre.* [En ligne ; Page disponible le 29-décembre-2021]. 2021. URL : %5Curl%7Bhttp://fr.wikipedia.org/w/index.php?title=Latitude&oldid=189341688%7D.

[Wik21b] Wikipédia. *Longitude — Wikipédia, l'encyclopédie libre.* [En ligne ; Page disponible le 6-décembre-2021]. 2021. URL : %5Curl%7Bhttp://fr.wikipedia.org/w/index.php?title=Longitude&oldid=188614923%7D.

[Wik21c] Wikipédia. *Système de coordonnées (cartographie) — Wikipédia, l'encyclopédie libre.* [En ligne ; Page disponible le 9-avril-2021]. 2021. URL : https://fr.wikipedia.org/w/index.php?title=Syst%C3%A8me_de_coordonn%C3%A9es_(cartographie).

[Wik21d] Wikipédia. *Valeur prédictive — Wikipédia, l'encyclopédie libre.* [En ligne ; Page disponible le 17-juin-2021]. 2021. URL : http://fr.wikipedia.org/w/index.php?title=Valeur_pr%C3%A9dictive&oldid=183891886.

[Wik22a] Wikipédia. *Pression atmosphérique — Wikipédia, l'encyclopédie libre.* [En ligne ; Page disponible le 22-janvier-2022]. 2022. URL : http://fr.wikipedia.org/w/index.php?title=Pression_atmosph%C3%A9rique&oldid=190112465.

[Wik22b] Wikipedia contributors. *World Geodetic System — Wikipedia, The Free Encyclopedia.* https://en.wikipedia.org/w/index.php?title=World_Geodetic_System&oldid=1065796786. [Online ; accessed 15-January-2022]. 2022.