# TD 1 : Data mining et graphiques

L'objectif de ce TD est d'appliquer sur un set de données inconnu quelques principes de Data Mining et de tracer différents graphiques pour comprendre l'évolution de ces données. Ce travail permettra de se familiariser avec l'utilisation des fonctions de base de R, de manipuler les boucles et d'approfondir des fonctions plus complexes et leurs nombreux arguments.

Nous allons travailler avec les données de qualité de l'air à New-York (entre mai et septembre 1973). Ce set est directement accessible dans R grâce à la commande suivante : ***data("airquality")***

## 1. Exploration des données : Data mining

1) Afin de garder un échantillon brut des données, créer une variable *data* identique à *airquality*.
2) Déterminer à l'aide de la fonction ***class()*** la classe de la variable *data*.
3) Déterminer également à l'aide des fonctions ***dim()*** et ***colnames()*** les dimensions et le nom des colonnes de *data*. Il est également possible d'utiliser les fonctions ***nrow()*** et ***ncol()*** pour connaître le nombre de lignes et le nombre de colonnes ; la fonction ***head()*** pour voir les noms des colonnes ainsi que les premières lignes du tableau. Faire la même c
4) Pour mieux comprendre les données récupérées, retrouver sur internet la documentation correspondante. Il est également possible d'utiliser la commande ***?airquality*** dans R.
5) Déduire de la documentation les unités des grandeurs physiques mesurées dans la donnée.
6) En utilisant la fonction ***summary()***, déterminer le nombre de *NA's* pour chaque grandeur.
7) Utiliser la fonction ***na.approx()*** du package ***zoo*** pour remplacer les NA's de chaque série de données par des valeurs réelles (reconstruction des données manquantes). En comparant l'une des valeurs remplacées avec les valeurs de la veille et du lendemain, évaluer comment fonctionne concrètement cette méthode. Est-ce une bonne méthode ?
8) (Faire cette question si vous êtes en avance) Essayer de reproduire le résultat de la fonction ***na.approx()*** *à la main* pour l'ozone, seulement pour les cas où la valeur manquante est entourée de deux valeurs réelles. Vous pouvez utiliser pour cela la fonction ***is.na()*** ainsi que les boucles ***for*** et ***if***.

## 2. Mise en forme des données : boucles, formules, dates

1) Convertir ces grandeurs en unités plus parlantes. Aussi, à l'aide la fonction ***round()***, arrondir ces valeurs au dixième.
   - 1 langleys = 0.4842 $W/m^2$
   - 1 mph = 0.447 m/s
   - T(°C) = (T(°F) - 32) * 5/9
2) Attribuer des noms de lignes à data avec des dates au format suivant : AAAA-MM-JJ. Pour cela, se servir des boucles ***for*** et ***if*** et des fonctions ***paste()*** et ***rownames()***. Indice : les numéros de jours (seulement certains) et de mois n'ont qu'un digit (« 1 » au lieu de « 01 »). Il faut corriger cette anomalie avant d'aller plus loin.
3) Retirer la colonne des jours à l'aide de la fonction ***which()***. Indice : on peut se servir du signe « - » pour enlever une colonne.

4) (Faire cette question si vous êtes en avance) Refaire la question 1) en utilisant 3 fonctions que vous aurez développées vous-même (rappel : https://abcdr.thinkr.fr/comment-creer-une-fonction-dans-r-function/).

**3. Graphiques : fonction plot et ggplot du package ggplot**

1) Avant de commencer la partie sur les graphiques, assurez-vous que les rownames soient bien au format *« Date »*. La fonction ***as.Date()*** permet de convertir un *« character »* en *« Date »* si le format de la date convient.
2) A l'aide de la fonction ***plot() ou ggplot()***, tracer la série temporelle de la température. Pour cela, précisez que l'argument *« y »* est la colonne des températures et que *« x »* est les rownames que l'on vient de créer.
3) En trouvant l'argument adéquat, afficher un titre.
4) Renommer l'axe des ordonnées.
5) Remplacer les points par des carrés.
6) Tracer une ligne continue à la place du nuage de points.
7) Afficher cette ligne en rouge.
8) Renforcer l'épaisseur de la ligne.
9) Ajuster l'échelle de l'axe des ordonnées pour afficher des valeurs entre 0 et 200.
10) A l'aide de la fonction ***lines() ou geom_line***, superposer la série numérique de l'ozone en pointillés bleus. Quel lien voyez-vous entre la température et l'ozone ?
11) A l'aide de la fonction ***par() ou facet_wrap()*** et de l'argument ***mfrow***, afficher deux graphes sur la même fenêtre, l'un en dessous de l'autre. Indice : ***mfrow*** doit être un vecteur de deux éléments : le premier correspond au nombre de lignes, le deuxième au nombre de colonnes.
12) Tracer sur deux graphiques distincts les séries temporelles de la température et de l'ozone.
13) Sur le graphique de l'ozone, ajouter à l'aide de la fonction ***abline() ou geom_vline() ou geom_hline*** une droite constante à 80 ppb.
14) Tracer sur quatre graphes distincts (sur deux lignes et deux colonnes) les séries temporelles de température, de vent, de radiation solaire et d'ozone de la couleur que vous souhaitez.
15) (Faire cette question si vous êtes en avance) Essayer de faire de même en utilisant une boucle ***for***.

**4. Autre graphique : barplot**

1) En fixant un seuil à 80 ppb, créer un vecteur qualifiant les jours avec une haute concentration et une basse concentration d'ozone. Il faudra pour cela se servir de la notion de test de condition vue dans le cours d'introduction, et voir quelles valeurs sont au-dessus de 80 ppb.
2) A l'aide de la fonction ***table()***, créer une variable qui contient le nombre de jours dans l'échantillon où la concentration est supérieure au seuil et le nombre de jours où elle est inférieure.
3) Tracer un diagramme à bâtons de cette nouvelle variable à l'aide de la fonction ***barplot() ou geom_boxplot()*** .
4) En créant une table à double entrée, déterminer le nombre de jours de haute concentration par mois. Se servir de la fonction ***table()*** et de son argument ***deparse.level***.

5) Tracer sur un diagramme à bâtons cette nouvelle variable.
6) En trouvant l'argument adéquat, juxtaposer (et non empiler) les 2 bâtons d'une même classe.
7) Donner des titres au graphique et aux axes.
8) Attribuer des couleurs aux deux classes (basse et haute concentration).
9) Afficher une légende.