Pollution atmosphérique à l'ozone
- L2 Math, IMIA -

L'ozone est un polluant atmosphérique bien connu pour ses effets délétères sur la santé humaine et sur la végétation. Il se forme dans les basses couches de l'atmosphère en période estivale, sous l'effet du rayonnement solaire et lorsque les températures sont élevées. Ainsi, lors de la canicule d'août 2003 et ses dramatiques conséquences sanitaires, largement imputables aux températures particulièrement élevées, mais aussi aggravées par les niveaux d'ozone, inédits par leur ampleur géographique et leur persistance. Aujourd'hui, le principal problème lié à la pollution à l'ozone est l'absence de tendances à la baisse des niveaux de fond depuis près de deux décennies en Europe, en dépit des mesures de réduction des émissions des polluants précurseurs à sa formation.

Figure 1 – Station de Mordelles-Bellais, capteurs de pollution notamment pour l'ozone O_3

Pour compléter le dispositif de surveillance de la qualité de l'air sur le territoire de Rennes Métropole, une nouvelle station de mesures a été mise en service à Mordelles (fig. 1) au début du mois de décembre 2018. La création de cette nouvelle station répond à un objectif de développement du réseau de mesures d'Air Breizh. Il s'agit d'une station dite "péri-urbaine de fond" qui permet de surveiller l'exposition maximale de la population à l'ozone.

On dispose par ailleurs de données météorologiques de la station METAR positionnée à l'aéroport de Rennes St Jacques.

Dans ce TP, nous allons aborder deux questions

— Le maximum journalier en ozone est il linéairement lié à la température en milieu de journée ?
— Peut-on prédire le maximum journalier en ozone à partir de le température et si oui avec quelle précision ?

Régression linéaire simple

Le premier objectif ici est d'écrire une fonction qui prend en entrée deux séries de données de longueur n (le prédicteur x_i et la réponse y_i) et renvoie les paramètres inconnus de la droite de régression.

On rappelle qu'on a obtenu les formules des estimateurs en TD.

$$\begin{aligned}\widehat{\beta}_1 &= \frac{\frac{1}{n}\sum_{i=1}^n y_i x_i - \frac{1}{n}\sum_{i=1}^n y_i \frac{1}{n}\sum_{i=1}^n x_i}{\frac{1}{n}\sum_{i=1}^n x_i^2 - (\frac{1}{n}\sum_i x_i)^2} \\ \widehat{\beta}_0 &= \frac{1}{n}\sum_{i=1}^n y_i - \widehat{\beta}_1 \frac{1}{n}\sum_{i=1}^n x_i\end{aligned}$$

Interprétation statistique des éléments de la formule

On rappelle que la moyenne empirique d'une série $x_i, i = 1 \cdots, n$ est définie par

$$\bar{x} = \frac{1}{n}\sum_{i=1}^n x_i$$

et que la variance empirique est définie par

$$var(x) = \frac{1}{n}\sum_{i=1}^n x_i^2 - \bar{x}^2$$

En outre, on définit la covariance entre deux séries $x_i, i = 1 \cdots, n$ et d'une série $y_i, i = 1 \cdots, n$ par

$$cov(x, y) = \frac{1}{n}\sum_{i=1}^n x_i y_i - \bar{x}\bar{y}$$

1. Quelles fonctions du module `numpy` de python permettent de calculer la moyenne et la variance d'une série $x_i, i = 1 \cdots, n$ réprésentée par un vecteur `x` ?
2. Quelle fonction du module `numpy` permet de calculer la covariance deux séries représentées par des vecteurs `x` et `y` de même longueur ?
3. Comment peut-on reformuler l'estimateur de $\widehat{\beta}_1$ de β_1 en utilisant les statistiques empiriques ?
4. Comment peut-on reformuler l'estimateur de $\widehat{\beta}_0$ de β_0 en utilisant les statistiques empiriques ? On ne remplacera pas $\widehat{\beta}_1$ par son expression.

Fonction python

5. Ecrire une fonction python nommée `SimpleRegLin`. La fonction ne doit pas faire appel à des boucles `for` mais utiliser les fonctions moyenne, variance et covariance du module `numpy`.
   Entête de la fonction

```
def SimpleRegLin(x,y):
   # Estimation des paramètres inconnus de la régresison linéaire simple
   # Entrées
   # x: prédicteurs (tableau numpy nx1)
   # y: réponses (tableau numpy nx1)
   # Sorties
   # par : vecteur de paramètres contenant l'ordonnée à l'origine beta0
   #       et la pente beta1
```

Mise en oeuvre

On dispose des mesures du maximum journalier de la concentration en ozone (`maxO3`) et de la température à 12h00 (`T12`) pour les mois de juin, juillet et août 2023. La concentration d'ozone est mesurée à Mordelles, en $\mu g/m^3$, et la température à St Jacques, en degrés celsius.

6. Utiliser les commandes suivantes pour charger les données.

```
import pandas as pd
dirname = "https://perso.univ-rennes1.fr/valerie.monbet/IMIA/"
filename = "Ozone_meteo_ete2023.csv"
data = pd.read_table(dirname+filename,sep=",")
print(data) # pour voir ce que contient la table de donnees
maxO3 = data["maxO3"]
T12 = data["T12"]
```

7. Tracer un nuage de points du maximum journalier d'ozone en fonction de la température à 12h00. Le titre et les légendes d'axes seront choisis avec attention.
   Commenter la figure.
8. Estimer les paramètres inconnus de la droite de régression. Quelles valeurs obtient-on ?
9. Tracer de nouveau le nuage de points et y ajouter la droite de régression. Le titre et les légendes d'axes seront choisis avec attention.
10. Peut-on déduire des résultats précédents que la concentration maximale en ozone augmente linéairement avec la température ? Pourquoi ?

Prédiction

11. Quelle formule permet de prédire la concentration maximale en ozone associée à une température donnée, par la régression linéaire ?
12. Ecrire une fonction `predict_SimpleRegLin` qui prend en entrée le résultat de la régression linéaire simple et une (ou plusieurs) valeur(s) du prédicteur et qui renvoie la prédiction de la régression linéaire.
    Entête de la fonction

```
def predict_SimpleRegLin(par,x):
   # prédiction pour régresison linéaire simple
```

```
# Entrées
# par: paramètres estimés par la fonction SimpleRegLin
# x: prédicteurs (tableau numpy mx1)
# Sorties
# y : prédiction
```

13. Estimer la concentration maximale en ozone pour une journée pendant laquelle la température à 12h est de 18.5 degrés celsius, puis 28.5 degrés celsius.
14. Retracer le nuage de points de données observées, ajouter la droite de régression et les deux points prédits à la question précédente.

Qualité du modèle : validation croisée

En machine learning, on utilise une partie des données pour mesurer la qualité du modèle (ici la régression linéaire simple). En pratique on garde de côté un sous ensemble des données tiré au hasard. Ce sous ensemble, appelé *ensemble de test* n'est pas utilisé dans la phase d'entrainement (ie la phase d'estimation des paramètres inconnus). Il est en revanche exploité pour comparer les prédictions à des valeurs réellement observées. Cette méthode s'appelle la **validation croisée**.

15. Exécuter et commenter les commandes suivantes ainsi que les résultats obtenus. En particulier, vous justifierez que les commandes codent la validation croisée décrite ci-dessus.

```
from sklearn.model_selection import train_test_split
[x_train,x_test,y_train,y_test] = train_test_split(T12,max03,test_size=0.2)
par = SimpleRegLin(x_train,y_train)
y_pred = predict_SimpleRegLin(par,x_test)

mse = np.mean((y_test-y_pred)**2)
rmse = np.sqrt(np.mean((y_test-y_pred)**2))
mse_rel = np.mean(((y_test-y_pred)/y_test)**2)

print("Erreur aux moindres carres : ", np.round(mse,2))
print("Racine de l'rreur aux moindres carres : ", np.round(rmse,2))
print("Erreur relative aux moindres carres : ", np.round(mse_rel,2))
```

16. Répéter les commandes précédentes à l'aide d'une boucle `for` de façon à calculer des erreurs moyennes sur 50 tirages différents de l'ensemble d'apprentissage et de l'ensemble de test. Commenter les résultats.
17. Tracer le nuage de points des prédictions en fonction des observations. Critiquer le modèle ajusté.