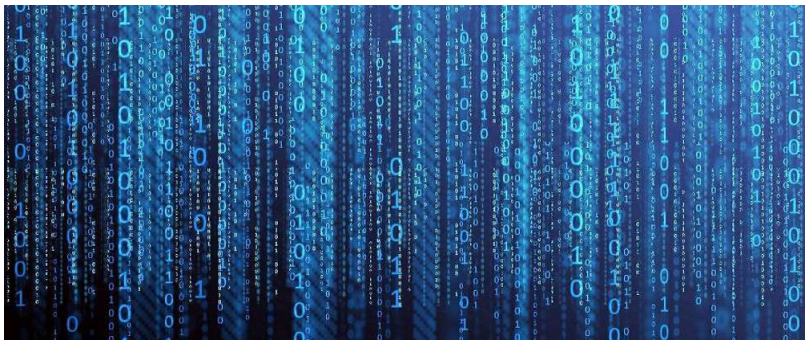


TP2

STATISTIQUES, RÉGRESSION LINÉAIRE, MANIPULATION DE FICHIERS .CSV.

1. Pré-requis

Dans tout le T.P, on utilise la bibliothèque pandas qui permet la lecture de fichiers .csv (*Comma Separated Values*) et la création/manipulation de tables. Si certaines commandes et instructions seront rappelées ci-après, on renvoie au cours de première année pour tout le détail. On importe une fois pour toutes

```
import pandas as pd
```

On va utiliser comme document de travail, tout au long de ce TP, le fichier tp2_data.csv, qui regroupe tout un tas de données publiques récupérées sur le site World Bank Data. En particulier, pour la période (1960-2020) et dans le Monde entier

- le taux de fertilité des jeunes femmes (nombre d'enfants pour 1000 jeunes femmes entre 15 et 19 ans),
- le pourcentage (du groupe concerné) de jeunes femmes étant scolarisé dans l'enseignement secondaire,
- l'espérance de vie,
- le pourcentage de population ayant accès à l'électricité,
- les émissions de CO_2 (en kT),
- la consommation électrique moyenne *per capita* (en KWh par habitant),
- la surface de forêt (en km^2).

On commence donc par importer le fichier de données dans Python avec la commande

```
donnees=pd.read_csv('http://louismerlin.fr/FTP/tp2_data.csv', sep=';')
```

Ici, on rajoute l'argument sep=' ;' car les données du fichier sont séparées avec un point virgule.

Remarque 1. La variable donnees est alors une *table de données* (ou *DataFrame*). On rappelle que

- donnees.head() permet de n'afficher que les 5 premiers rangs du tableau ;
- donnees.shape() renvoie une couple (n, p) où n est le nombre de lignes et p le nombre de colonnes du tableau ;
- donnees.columns permet d'afficher l'ensemble des colonnes du tableau.

Une colonne intitulée index est ajoutée par la bibliothèque pandas à la table de données lors de sa lecture afin de donner un numéro à chaque ligne de la table de données (la numérotation commençant comme toujours avec Python à 0).

Notre jeu de données manipulé ici est (relativement) grand. Il contient 8 colonnes et 61 lignes... On va dans un premier temps ne considérer qu'une sous-table.

ECG 2 Maths appliquées, http://louismerlin.fr.

```
table1=donnees[['Annee', 'taux fertilite j. femmes', 'femmes scol. sec.']]
table1=table1.rename(columns={ 'taux fertilite j. femmes': 'TF', 'femmes scol. sec.':'FSS'
    })
```

2. Statistiques descriptives

2.1. **Rappels : statistiques univariées.** Pour *décrire* un jeu de données $x = [x_1, x_2, ..., x_n]$, on introduit quelques mesures :

- La **moyenne (empirique)** (*Mean Value* en anglais), souvent notée $\overline{x}_n$ définie par

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

Remarque 2. Si la commande `mean` de bibliothèque `numpy` permet d'obtenir la moyenne des valeurs d'une liste, il faut ici faire attention ; on travaille avec *DataFrame* et il faut donc utiliser la commande `table1.mean()` qui renvoie la liste des moyennes pour chaque colonne numérique ou plus précisément `table1['nom_de_la_colonne'].mean()` pour obtenir la moyenne des valeurs d'une colonne précise.

- La **variance empirique**

$$\hat{\sigma}_n^2(x) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x}_n)^2$$

Il s'agit de la moyenne des carrées des écarts à la moyenne. Cette valeur n'est pas facile à interpréter car son unité de mesure n'est pas la même que celle des données. C'est pourquoi, pour l'interprétation (et notamment en statistiques descriptives), on lui préfère la mesure suivante.

- L'**écart-type empirique** (*Standard Deviation* en anglais)

$$\hat{\sigma}_n(x) = \sqrt{\hat{\sigma}_n^2(x)}$$

Cette mesure permet de quantifier la dispersion des observations autour de la moyenne et a l'avantage de s'exprimer dans la même unité de grandeur que nos données.

Remarque 3. Avec les *DataFrames*, on utilise la commande `table1.std()` ou `table1['nom_de_la_colonne'].std()` sur le même modèle que précédemment.

- La **médiane** de la série statistique. Il s'agit de la valeur m telle que 50% des données sont inférieures à m et 50% supérieures à m. Intuitivement, la médiane est le point milieu des observations (à ne pas confondre avec le point moyen).

Remarque 4. Avec les *DataFrames*, on utilise la commande `table1.median()` ou `table1['nom_de_la_colonne'].median()` sur le même modèle que précédemment.

- On s'intéresse aussi parfois à d'autres **quantiles**. On note q_α le quantile d'ordre α qui désigne le réel tel qu'une proportion α des observations est inférieure à q_α et une proportion $1-\alpha$ est supérieure à q_α. La médiane est le quantile d'ordre 1/2.
- Le **minimum** ou le **maximum** de la série statistique qui correspond à la plus petite (ou la plus grande valeur) des observations.

Remarque 5. Avec les *DataFrames*, on utilise les commande `table1.min()` ou `table1.max()` sur le même modèle que précédemment.

2.2. **Nuage de points, point moyen.** On cherche maintenant à savoir s'il est possible d'*expliquer* une série de données à partir d'une autre. Par exemple, le pourcentage de jeunes femmes scolarisées peut-il *expliquer* le nombre moyen d'enfant (pour 1000) des jeunes femmes entre 15 et 19 ans ?

Plus généralement, on considère deux séries statistiques $x = [x_1, ..., x_n]$ et $y = [y_1, .., y_n]$ que l'on observe **simultanément**. On étudie alors les couples $[(x_1, y_1), ..., (x_n, y_n)]$ que l'on appelle observations dans le cas de statistiques bivariées.

Définition : Nuage de points

On appelle **nuage de points** associé à la série statistique (x, y) l'ensemble des points M_k de coordonnées (x_k, y_k) (pour $1 \leq k \leq n$) tracés dans un repère orthonormé du plan (où $X = (x_k)$ et $Y = (y_k)$).

Le **point moyen** du nuage est le point de coordonnées $(\overline{x}_n, \overline{y}_n)$, où $\overline{x}_n$ désigne la moyenne empirique des x_k et $\overline{y}_n$ celle des y_k.

L'examen du nuage de points permet de faire des constatations qualitatives :

- est-il concentré ou dispersé ?
- relève-t-on une tendance ?
- y a-t-il des valeurs *a priori* aberrantes ?

On reprend notre jeu de données sur la scolarisation et la fertilité des jeunes filles. Recopier et exécuter les instructions suivantes. Commenter.

```
import matplotlib.pyplot as plt

table1=table1.dropna() # on supprime les rangs avec donnees manquantes
X=table1['FSS']
Y=table1['TF']

plt.grid()
plt.plot(X,Y, 'k+')
plt.show()
```

Quelles commandes peut-on ajouter pour faire apparaître le point moyen du nuage ?

Définition : Covariance et coefficient de corrélation empiriques

La **covariance empirique** d'une série statistique (x, y) est définie par

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

Le **coefficient de corrélation linéaire empirique** est défini par

$$\rho_{x,y} = \frac{\mathrm{cov}(x, y)}{\hat{\sigma}_n(x)\hat{\sigma}_n(y)}$$

Calculer le coefficient de corrélation linéaire des séries X et Y considérées ci-avant. Commenter.

3. Régression

3.1. **Variable explicative et variable à expliquer.** Dans une population donnée, on peut souhaiter étudier simultanément deux caractères X et Y. On peut alors s'intéresser aux propriétés de chacun des deux caractères pris séparément (statistiques univariées), mais aussi au lien entre ces deux caractères (statistiques bivariées) ; on étudie alors le couple de caractère $Z = (X, Y)$. En particulier, on peut penser que l'une des variables, par exemple X, est une cause de l'autre, par exemple Y. On dit alors que X est *la variable explicative* et que Y est *la variable à expliquer*. Dans tous les cas, on tentera d'exprimer Y en fonction de X pour deviner la relation entre ces données (voir la partie suivante). Dans la suite de ce TP, nous allons développer un critère, qui permet d'établir (ou non) s'il y a corrélation linéaire entre deux variables. Il faut cependant garder en tête que

une causalité entraîne une corrélation mais la réciproque est fausse.

Il y a deux écueils à éviter lorsqu'on établit une corrélation linéaire entre deux variables :

- Le premier consiste à conclure qu'il y a un lien de causalité alors qu'il y a corrélation. Comme dit précédemment, ce n'est pas toujours le cas. Par exemple, il pourrait y avoir une variable cachée C qui ne fait pas partie de l'étude statistique qui explique X et Y simultanément.

  Exemple 6. On observe une corrélation positive entre le fait de pratiquer la course à pied et le fait de développer un cancer de la peau. Une conclusion simpliste consiste à affirmer que la course à pied est une cause de cancer de la peau. Une explication plus fine consiste à dire qu'il existe une variable cachée : l'exposition au soleil. Plus les personnes sont exposées au soleil, plus il est probable qu'elles fassent de l'activité physique. De même, plus les personnes sont exposées au soleil, plus il est probable qu'elles développent un cancer de le peau.

- Quand bien même une causalité existerait entre X et Y, le deuxième écueil consiste à se tromper dans le sens de la causalité. En effet, même en cas de corrélation forte entre X et Y, celle-ci est symétrique entre X et Y et ne permet pas de dire laquelle des deux variables explique l'autre.

  Exemple 7. Un lien de corrélation est établit entre l'usage de cannabis et le fait de développer une psychose. Est-ce que fumer participe au développement de la psychose ou est-ce que fumer est une conséquence visant à calmer l'angoisse de la maladie ?

  C'est une partie du travail de la recherche en sciences sociales que de déterminer si une variable peut en expliquer une autre ou non, et la seule analyse mathématique ne peut pas répondre définitivement à une telle question, puisqu'elle ne démontre que des liens de corrélations et jamais de causalité.

3.2. **Droite de régression linéaire. Méthode des moindres carrées.** On se place dans la situation où l'on souhaite savoir si on peut trouver une "formule" permettant de donner une *approximation* de Y en fonction de X. Cette formule pouvant notamment servir à faire de la prévision.

On rappelle alors le résultat suivant.

Proposition : Propriétés du coefficient de corrélation linéaire

Soit ρ le coefficient de corrélation linéaire du couple (X, Y). Alors

1. $\rho \in [-1; 1]$;
2. $\rho = \pm 1$ si et seulement si la régression $Y = aX + b$ est exacte.

Il parait alors assez naturel de penser que si ρ est "assez proche" de 1 (en valeur absolue), l'approximation *affine* pourrait être pertinente.

Remarque 8. Si $|\rho|$ est proche de 1 **et qu'on a visualisé une relation linéaire entre les données**, on peut confirmer qu'il y a bien corrélation linéaire entre X et Y.

En sciences humaines et en sciences économiques, une valeur de $|\rho|$ de l'ordre de $0,85$ est souvent considérée comme bonne.

On cherche donc deux constante a et b telles que

$$Y = aX + b + \varepsilon.$$

On utilise alors la *méthode des moindres carrés* qui nous donne l'équation de la droite la plus proche des points en terme de distance, c'est à dire l'unique droite D d'équation $y = ax + b$ qui rend minimale la somme des carrés des erreurs d'ajustement

$$d^2(a, b) = \frac{1}{n} \sum_{i=1}^{n} (y_i - (ax_i + b))^2 .$$

Le résultat suivant donne la valeur de a et b et est **admis** (voir une preuve dans un fichier TOP).

Proposition : Droite de régression

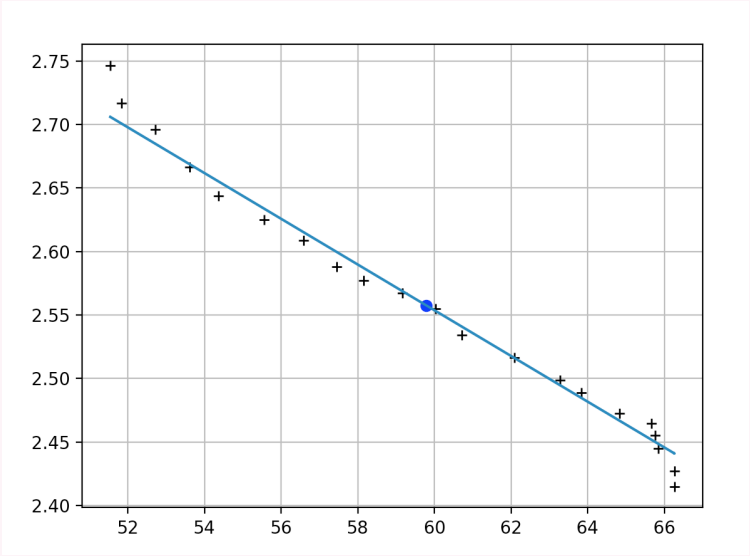
La droite la plus proche du nuage de points associé au couple (x, y) est la droite d'équation $y = ax + b$ avec

$$a = \frac{\mathrm{cov}(x, y)}{\hat{\sigma}_n^2(x)}, \quad \text{et} \quad b = \overline{y}_n - a \times \overline{x}_n.$$

En particulier, cette droite passe par le point moyen $(\overline{x}_n, \overline{y}_n)$. On appelle cette droite la *droite de régression*.

Écrire une suite d'instructions permettant de représenter la droite de régression linéaire de `Y` en fonction de `X`, sur la même figure que le nuage de point (ainsi que le point moyen), comme ci-dessous.

Droite de régression



Exercice 1

Étudier la pertinence d'une régression linéaire pour expliquer l'espérance de vie en fonction de l'accès à l'électricité (pour les années où les données sont fournies).

3.3. **Régression linéaire avec transformations.** Dans certains cas (qui seront pour nous complètement guidés par l'énoncé du sujet), on peut appliquer le principe de régression linéaire à un couple obtenu par transformées de Y (ou aussi de X) et obtenir une relation de la forme

$$Y = a\varphi(X) + b + \varepsilon, \quad \text{ou} \quad \varphi(Y) = a\varphi(X) + b + \varepsilon.$$

Considérons un exemple avec des données correspondant à l'évolution du PIB par habitant (en USD) et du pourcentage de la population en zone urbaine de la Norvège, de 1960 à 2020 (source : World Bank Data).

1. Recopier et exécuter les instructions suivantes. Commenter le nuage de points.

```
data2=pd.read_csv('https://louismerlin.fr/FTP/tp2_nor.csv', sep=';')

X=data2['PIB per capita']
Y=data2['Pop urbaine %']

plt.grid()
plt.plot(X,Y, '.') # nuage de points
plt.show()
```

2. Représenter le nuage de points $(\ln(X), Y)$.

3. Calculer le coefficient de corrélation linéaire de Y en $\ln(X)$.

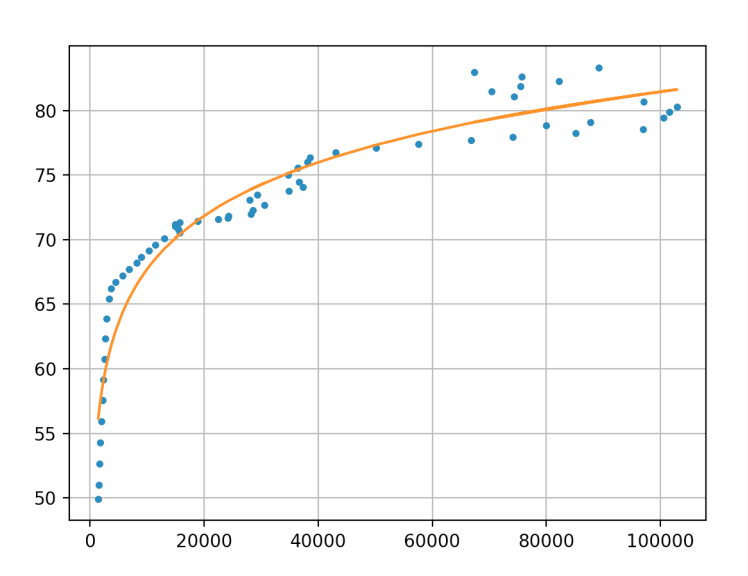
4. Déterminer l'équation de la droite de régression de Y en $\ln(X)$.

5. En déduire qu'on peut supposer que la dépendance entre Y et X est de la forme

$$Y = a\ln(X) + b$$

6. Représenter le nuage de points précédent sur lequel on fera apparaître la courbe d'équation $y = a\ln(t) + b$.

Courbe de régression



3.4. **Un autre exercice.** Supposons que vous soyez le chef de direction d'une franchise de camions ambulants (*Food trucks*). Vous envisagez différentes villes pour ouvrir un nouveau point de vente. La chaîne a déjà des camions dans différentes villes et vous avez des données pour les bénéfices et les populations des villes. Vous souhaitez utiliser ces données pour vous aider à choisir la ville pour y ouvrir un nouveau point de vente.

On dispose d'un fichier data.csv et on utilise la bibliothèque pandas

1. On exécute les instructions suivantes qui donnent l'affichage ci-après. Que contient le fichier *data.csv* ?

```
import pandas as pd
import numpy as np
import numpy.random as rd
import matplotlib.pyplot as plt

donnees=pd.csv_read('data.csv', sep=';')
donnees.head()
```

Retour Python

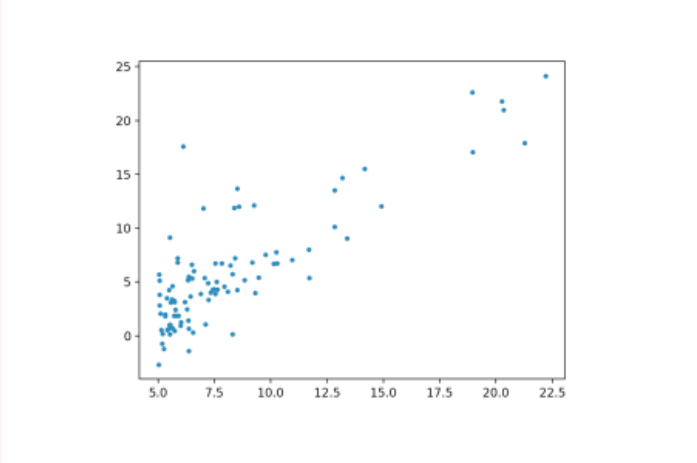
```
>>>> donnees.head()

   Population (en 10k) Profit (en 10k EUR)
0        6.1101         17.5920
1        5.5277          9.1302
2        8.5186         13.6620
3        7.0032         11.8540
4        5.8598          6.8233
```

2. On ajoute les commandes suivantes

```
table = donnees.rename(columns={'Population (en 10k)' : 'pop', 'Profit (en 10k EUR)'
    : 'profit'})
X=table['pop']
Y=table['profit']
plt.grid()
plt.plot(X,Y,'.')
plt.show()
```

Affichage Python



a. Que représente cette figure ?

b. Expliquer pourquoi la figure ci-dessus permet de conjecturer qu'il existe deux réels a et b tels que $ax + b$ où x est le nombre d'habitants de la ville (en dizaine de milliers d'habitants), est une approximation raisonnable du profit (en dizaines de milliers d'euros) d'un *Food Truck* installé dans cette même ville.

c. Quelle quantité peut-on calculer pour conforter cette approximation ? Donner une suite d'instructions en Python permettant de la calculer.

d. On suppose qu'on a été en mesure de répondre à la question précédente correctement. L'exécution des commandes affiche alors une valeur de 0.83792835822348835. Est-ce cohérent ?

e. Il y a 182354 habitants à *Legumeville* et pas encore de *Food Truck*. Quelle commande Python permettrait d'estimer raisonnablement le profit suivant l'installation d'un camion dans cette localité ?

3. Votre société a beau être établie en zone euro, son siège social est dans le Delaware aux États-Unis, et on décide d'exprimer le profit en dollars. Sachant qu'un euro vaut, au moment de faire le calcul, 1.10 dollars, que devient la covariance des séries statistiques habitants/profits ? Même question pour le coefficient de corrélation.