Chapitre n°15

Statistiques bivariées

I. Statistiques descriptives

I. 1 Rappels de statistiques univariées

Pour décrire un échantillon de données $x = [x_1, x_2, \ldots, x_n]$, on introduit quelques mesures :

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

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

- **La variance empirique** $\boxed{\hat{\sigma}_n^2(x) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x}_n)^2}$ 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) $\boxed{\hat{\sigma}_n(x) = \sqrt{\hat{\sigma}_n^2(x)}}$
  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.
- **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).
- **Les 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.

Exercice 1

Écrire des fonctions `esperance(X)`, `variance(X)` et une fonction `ecarttype(X)` qui calculent la moyenne, la variance et l'écart-type empirique d'une liste de réels `X`.

I. 2 Statistiques bivariées

Définition 1.1

Une **série statistique double** est la donnée d'un certain nombre $n \in \mathbb{N}$ de mesures simultanées $(x_i, y_i)_{i\in[\![1,n]\!]}$.

On cherche à savoir s'il est possible d'expliquer une série de données à partir d'une autre. Par exemple, la consommation moyenne d'éléctricité par habitant (en KWh) peut-elle expliquer le taux d'émission de C02 (en kT) dans l'atmosphère ? Plus généralement, on considère deux séries statistiques $x = [x_1, \ldots, x_n]$ et $y = [y_1, .., y_n]$ que l'on observe simultanément. On étudie alors les couples $[(x_1, y_1), \ldots, (x_n, y_n)]$ que l'on appelle observations dans le cas de statistiques bivariées.

Nuages de points.

Définition 1.2

- 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 ?

Nous allons, dans un premier temps, utiliser les générateurs aléatoires

Exemple :

- Recopier l'exemple suivant

```
import numpy.random as rd
N=1000
Xu=rd.random(N)
Yu=rd.random(N)
plt.plot(Xu,Yu,"+", color="cyan")
plt.show()
```

- Remplacer les lois uniformes par des loi binomiales $\mathcal{B}(15; 0, 4)$. On nommera Xb et Yb les listes.
- Recopier la fonction suivante, elle renvoie deux séries en fonction d'un paramètre α. Afficher les nuages de points obtenus pour différentes valeur de α dans $[-1, 1]$

```
def series(alpha):
    ''' alpha doit etre compris entre -1 et 1'''
    x,z=rd.normal(0,1,1000),rd.normal(0,1,1000)
    y=alpha*x+np.sqrt(1-alpha**2)*z
    return x,y
```

Exercice 2

Écrire une fonction `pm(X,Y)` qui renvoie le point moyen de la série double.

Covariance empirique

Définition 1.3 — Covariance empirique

Soit $(x, y) = (x_i, y_i)_{i \in [\![1, n]\!]}$ une série double. La *covariance empirique* est

$$\hat{\sigma}_{x,y} = \frac{1}{n} \sum_{i=1}^{n} \left((x_i - \overline{x_n}) (y_i - \overline{y_n}) \right)$$

Attention:

Comme nous travaillons avec des séries finies, les sommes sont finies et il n'y a pas de problèmes de convergence à soulever.

Proposition 1.4 — Propriétés

La covariance empirique partage les même propriétés que la covariance vue dans le cours sur les couples de variables aléatoires. Soit x, y, z trois séries de même taille.

- $\hat{\sigma}_{x,x} = \hat{\sigma}_n(x)^2$.
- $\hat{\sigma}_{x,y+\lambda z} = \hat{\sigma}_{x,y} + \lambda\hat{\sigma}_{x,z}$ et linéarité à gauche.
- Formule de Koenig Huygens :

$$\hat{\sigma}_{x,y} = (\overline{xy})_n - \bar{x}_n\bar{y}_n$$

Exercice 3

1. Démontrer mathématiquement les propriétés de la proposition 1.4.
2. Écrire une fonction Python `cov(x,y)` qui calcule la covariance de x et y.
3. Soit `X` et `Y` deux vecteurs contenant 100 réalisations de la loi exponentielle de paramètre 3. Qu'obtient-on avec la commance `np.cov(X,Y)` ?

I. 3 Droite de régression linéaire

On se place dans la situation où l'on souhaite savoir 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 dispose donc d'une séries double de données.

Définition 1.5 — coefficient de corrélation linéaire

$$\rho_{xy} = \frac{\hat{\sigma}_{xy}}{\hat{\sigma}_n(x)\hat{\sigma}_n(y)} \qquad \text{où} \qquad \hat{\sigma}_n(x) = \sqrt{\hat{\sigma}_n(x)^2} \text{ est l'écart-type empirique de la série.}$$

On rappelle alors le résultat suivant. Soit $\rho_{(X,Y)}$ le coefficient de corrélation linéaire du couple (X, Y). Alors

- $\rho_{(X,Y)} \in [-1; 1]$;
- $\rho_{(X,Y)} = \pm 1$ si et seulement si la régression $Y = aX + b$ est exacte.

Remarques :

R1 – Il parait alors assez naturel de penser que si $\rho_{(X,Y)}$ est "assez proche" de 1 (en valeur absolue), l'approximation affine pourrait être pertinente.
Si $|\rho_{(X,Y)}|$ 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.

R2 – En sciences humaines et en sciences économiques, une valeur de $|\rho_{(X,Y)}|$ de l'ordre de 0,85 est souvent considérée comme bonne.

Attention:

Trouver une corrélation ne permet pas de prouver un lien de causalité !

http ://www.tylervigen.com/spurious-correlations

Exercice 4

Écrire une fonction `rho(X,Y)` qui calcule le coefficient de corrélation linéaire

On cherche la meilleure approximation de ces données par une droite. On utilise alors la méthode des moindres carrés (cf. Annexe) 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.

Définition 1.6

Soit x, y une série double. La droite de régression linéaire a pour équation :

$$Y = \frac{\hat{\sigma}_{xy}}{\hat{\sigma}_x^2}(X - \bar{x}) + \bar{y}$$

Remarques :

R1 – On a donc ici $a =$ et $b =$

R2 – Cette droite passe par le point moyen $(\overline{x}_n, \overline{y}_n)$.

Exercice 5 — calcul de la droite

Écrire une fonction `regression(x,y)` qui retourne a, b tels que $Y = aX + b$ est la droite de regression linéaire de la série double (x, y).

Exercice 6 — illustration

Sur un même graphique faire afficher une série obtenue avec `série(0.7)` et sa droite de régression linéaire. Faire varier le coefficient α.

II. Application à l'étude de données sur Python

Le regroupement des données concernant une étude sont regroupées dans ce qu'on appelle des *bases de données*. Une introduction à la manipulation de ces bases de données sera faite dans un chapitre lié au langage SQL. Nous allons des fichiers déjà existant et utiliser dans Python la bibliothèque `pandas`, qui permet la lecture de fichiers .csv (Comma Separated Values) et la création/manipulation de tables.
Tout au long de ce qui suit, nous allons nous appuyer sur certaines données qui illustrerons l'application de ces définitions et de ces méthodes.
Pour ce faire nous aurons besoin d'importer certains modules, dont en particulier `numpy`, `matplotlib.pyplot` et `pandas`.

Avec Python

Pour demander à Python de lire un fichier csv, il suffit de taper la commande

`pd.read_csv("Nom du fichier")` ou `pd.read_csv("url")`

où `url` est un lien internet vers un fichier csv.

Exemple :

Taper la commande suivante :

```
données=pd.read_csv("https://cahier-de-prepa.fr/ece2-civ/download?id=1517")
print(données)
```

Avec Python

Quelques commandes à connaître :

1. La variable `données` est alors une table de données (ou DataFrame). On rappelle que
   - `données.head` permet de n'afficher que les 5 premiers rangs du tableau ;
   - `données.shape` renvoie une couple (n, p) où n est le nombre de lignes et p le nombre de colonnes du tableau ;
   - `données.columns` permet d'afficher l'ensemble des colonnes du tableau.
   - `données.loc[...]` permet d'afficher des lignes ou des coefficients du tableau.

1. Lien entre croissance et chômage

1. Ouvrir avec Python et pandas le fichier csv qui se trouve sur le lien `"https://cahier-de-prepa.fr/ece2-civ/download?id=1519"`

```
url='https://cahier-de-prepa.fr/ece2-civ/download?id=1519'
etude=pd.read_csv(url,sep=';')
print(etude)
```

   Que contient ce fichier ?
2. On s'intéresse au lien qu'il peut y avoir entre l'évolution d'une série statistique par rapport à une autre. Il convient de choisir une variable *explicative* `X` et une variable à expliquer `Y`.
3. On souhaite tracer l'évolution du taux de chômage et celle du PIB au cours du temps. Quel programme peut-on proposer ?
4. On souhaite à présent tracer le nuage de point de la série `(X,Y)` :
5. On souhaite décrire ces données à l'aide des statistiques descriptives définies dans la première partie.

Avec Python

- `etude.mean()` : Créé un tableau regroupant les différentes moyennes des quantités du tableau `etude`.
- `etude.var()` : Créé un tableau regroupant les différentes variances des quantités du tableau `etude`.
- `etude.cov()` : Créé un tableau regroupant les différentes variances des quantités du tableau `etude`.
-
- `etude.std()` : Créé un tableau regroupant les différents écart-types des quantités du tableau `etude`. (`std` est pour *Standard Deviation*).
- `etude.median()` : Créé un tableau regroupant les différentes médianes des quantités du tableau `etude`.
- `etude.min()` ou `etude.max()` : Créé un tableau regroupant les différentes valeurs minimales ou maximales des quantités du tableau `etude`.

Attention:

Les données étudiées sont sous forme de **DataFrame**. La bibliothèque `numpy` dont on se sert ne peut pas être utilisée directement. (*Python peut tout de même afficher quelque chose mais renverra un message d'erreur*).

6. Écrire un programme donnant les moyennes, écart-types, covariances des colonnes réservées au chômage et croissance.
7. Que faut-il taper pour obtenir le coefficient de corrélation linéaire ? Que peut on dire de la valeur obtenue ?
8. Écrire un programme permettant d'obtenir le nuage de point de la série `(X,Y)` avec la droite de régression linéaire.
9. Reprendre ces questions avec la bibliothèque `numpy`.

2. Une étude de marché

Vous êtes le chef de direction d'une franchise de camions à pizza. 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 donne l'affichage ci-après. Que contient le fichier data.csv importé ?

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

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

> > donnees.head ( )

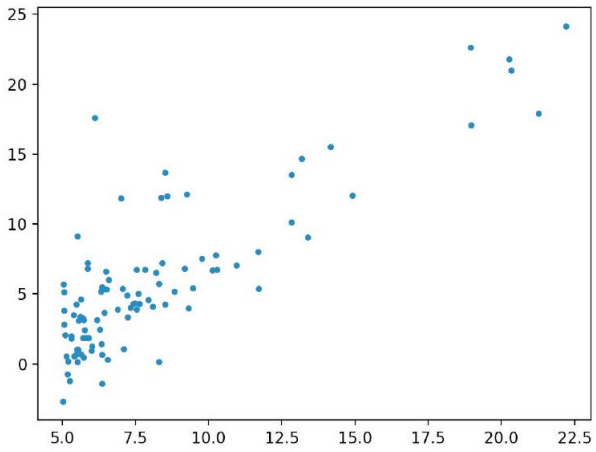
	Population (en 10k)	Profit (en 10k EUR)
0	6.1101	17.5920
1	5.5277	9.1302
2	8.5186	13.6620
3	7.0032	11.8540
4	5.8598	6.8233

2. On ajoute les commandes suivantes

```
table=données.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, b tels que $ax+b$, où x est le nombre d'habitants de la ville (en dizaines de milliers d'habitants), est une approximation raisonnable du profit (en dizaines de milliers d'euros) d'un camion à pizza installé dans cette même ville.

(c) Quelle quantité pourrait-on calculer pour conforter cette approximation ? Donner une suite d'instruction 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 0.8738733891854535 . Est-ce cohérent ?

(e) Il y a 182354 habitants à Legumeville et pas encore de camion à pizza. Quelle(s) commande(s) Python permettraient 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 Etats-Unis, et on décide d'exprimer le profit en dollars. Sachant qu'un euro vaut au moment de faire le calcul 1.045 dollar, que devient la covariance des séries statistiques habitants/profit ? Même question avec le coefficient de corrélation linéaire.

3. Régression linéaire avec transformation

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

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

Prenons à présent le fichier

```
PIBbonheur:"https://cahier-de-prepa.fr/ece2-civ/download?id=1518"
données=pd.read_csv(PIBbonheur)
print(données)
```

1. Créer des variables X et Y qui contiennent respectivement le PIB et l'indice de bonheur.
2. Tracer le nuage de point des données PIB/ indice de bonheur. On constate que ce nuage à une forme qui ne se prête pas à une régression linéaire.
3. Transformation logarithmique :
   (a) Représenter le nuage de points $(\ln(X), Y)$.
   (b) Calculer le coefficient de corrélation linéaire de Y en $\ln(X)$.
   (c) Déterminer l'équation de la droite de régression de Y en $\ln(X)$.
   (d) Représenter cela avec le nuage de points précédent.
4. Représenter le nuage de points (X, Y) avec la courbe de la fonction $y = a\ln(t) + b$.