IUT Robert Schuman
Institut universitaire de technologie
Département informatique
Université de Strasbourg

M22 : TP

Exercices

Introduction

Python est un langage de programmation interprété, multiparadigme et multiplateformes. C'est l'un des langages les plus utilisés dans le domaine de l'intelligence artificielle, de l'apprentissage automatique ainsi que pour l'analyse de données scientifiques.
Dans notre cas d'utilisation, c'est un langage pratique pour l'analyse de données.
En 2018, Python est le 2e langage le plus utilisé sur GitHub avec 14.75% des utilisateurs actifs [source]. Au dernier trimestre 2023, c'est le langage avec la communauté la plus active sur la plateforme [source].

Installation Python et les dépendances

1. Installer Python
    a. Windows https://www.python.org/
    b. Linux : `apt install python3`
2. Installer NumPy
   `pip install numpy`
3. Installer Pandas
   `pip install pandas`
4. Installer Matplotlib
   `pip install matplotlib`
5. Optionnel
    a. Installer PyCharm
       https://www.jetbrains.com/fr-fr/pycharm/
    b. Installer Jupyter notebook ou Jupyter lab
       `pip install jupyterlab`

Pour importer les modules installés dans votre script python :

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

Les modules suivants seront également utiles :

```
import math
import random
```

Les différentes documentations :

- https://numpy.org/doc/
- https://pandas.pydata.org/docs/
- https://matplotlib.org/stable/index.html

TP 1 : Statistiques et Python

Exercice 1 : Tableaux NumPy

NumPy est une bibliothèque Python, destinée à manipuler des matrices ou **tableaux** multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux.

1. Créer les listes suivantes

   a : `[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]`

   b : `[1, 4, 7, 10, 13, 16, 19, 22, 25, 28, 31, 34, 37, 40, 43]`

   d : `['quatre', 'chaînes', 'de', 'caractères']`
2. Transformer **a** et **b** en tableau NumPy.
3. Ajouter 2 aux valeurs de **a**.

   Multiplier par 2 les valeurs de **a**.
4. Additionner et multiplier les tableaux **a** et **b** entre eux.
5. Calculer (avec NumPy) pour **b**
   a. moyenne
   b. variance
   c. écart-type
   d. médiane
   e. minimum
   f. maximum
   g. quartiles

Exercice 2 : Un peu de Python

1. Écrire une fonction **`moyenne`** donnant la moyenne d'une série numérique.
2. Écrire une fonction **`variance`** donnant la variance d'une série numérique.
3. Écrire une fonction **`ecart_type`** donnant l'écart-type d'une série numérique.
4. Créer un tableau **`vals_norm`** de 15 valeurs générées aléatoirement via une loi normale de moyenne 10 et d'écart-type 2.
5. Créer un tableau **`vals_unif`** de 15 valeurs générées aléatoirement via une loi uniforme sur l'intervalle [0 ; 20].

Exercice 3 : Matplotlib

Matplotlib est une bibliothèque Python destinée à tracer et à visualiser des données sous forme de graphiques.

1. Visualiser **a** et **b** sur un graphique.
2. Créer les tableaux **`vals_norm_sort`** et **`vals_unif_sort`** contenant les valeurs des tableaux **`vals_norm`** et **`vals_unif`** triées.
3. Visualiser **`vals_norm_sort`** et **`vals_unif_sort`** sur un graphique.

Exercice 4 : DataFrame Pandas

Pandas est une bibliothèque Python permettant la manipulation et l'analyse de données. Elle propose en particulier des structures de données et des opérations de manipulation de tableaux numériques et de séries temporelles comme les DataFrame.
Un **DataFrame** est une structure de données bidimensionnelle, c'est-à-dire que les données sont alignées de façon tabulaire en lignes et en colonnes (comme dans un tableur).

1. Créer un dataframe nommée **df** à partir de **vals_norm** et **vals_unif** avec les noms des colonnes suivants : **norm** et **unif**.
2. Calculer l'écart type de la colonne **norm** (avec pandas) et l'écart type du tableau **vals_norm** (avec votre fonction **ecart_type** ou avec NumPy).
    a. Renseignez-vous sur la correction de Bessel [en.wikipedia.org].
    b. Modifier la valeur renvoyée par pandas pour qu'elle corresponde à la valeur renvoyée par votre fonction.
3. Calculer la covariance, le coefficient de corrélation, ainsi que les constantes pour l'équation de régression linéaire pour les séries **`vals_norm`** et **`vals_unif`**.
4. Créer une deuxième dataframe qu'on nommera **`df2`** avec les mêmes noms de colonnes et 5 valeurs supplémentaires.
5. Ajouter les lignes de **`df2`** à **`df`**.
6. Ajouter une colonne à **`df`** que l'on nommera **`somme`** qui est la somme des deux premières colonnes.
7. Sélectionner le sous dataframe de **`df`** qui ne contient que les valeurs supérieures à 5 dans la colonne **`unif`**.
8. Sélectionner le sous dataframe de **`df`** qui ne contient que les colonnes **`norm`** et **`somme`**.

TP 2 : Analyse de données

Exercice 5 : Parcoursup

Objectif : Visualiser les taux d'admissions par type de bac des différentes composantes de l'IUT Robert Schuman.

Récupérer les fichiers de données concernant les vœux de poursuite d'études Parcoursup pour l'année 2021 sur la plateforme ouverte des données publiques françaises : data.gouv.fr (données relatives à l'éducation).

1. Importer les données dans un dataframe pandas appelé **`df`**.
2. Modifier **`df`** afin de garder les colonnes suivantes :
    a. le code UAI de l'établissement
    b. l'intitulé de la formation
    c. les taux d'admissions
3. Modifier **`df`** afin de garder les lignes correspondant à l'IUT Robert Schuman.
4. Créer une nouvelle colonne **formation** qui contiendra les 32 premiers caractères de l'intitulé de la formation.
5. Afficher un graphique à barres représentant les différents taux d'admission pour chaque formation.

Exercice 6 : Température et changement climatique

Objectif : Visualiser et évaluer statistiquement les changements de température de 1950 à nos jours à Strasbourg.

Récupérer les fichiers de données climatologiques de base pour la période 1950 → 2022 pour le département du Bas-Rhin sur la plateforme ouverte des données publiques françaises : data.gouv.fr (Données climatologiques de base - décadaires agro).

1. Importer les données dans un dataframe pandas appelé **`df`** et le modifier afin de garder les lignes correspondantes à la station de Strasbourg Entzheim.
2. Créer un tableau NumPy **`y_max`** contenant les valeurs de la colonne **`TX`**.
3. Créer un tableau NumPy **`y_min`** contenant les valeurs de la colonne **`TN`**.
4. Créer un tableau Numpy **`x`** contenant les valeurs de 0 à N avec N la longueur de **`TX`** et **`TN`**.
5. Visualiser graphiquement les valeurs de **`y_max`** et **`y_min`** en fonction de **`x`**.
6. Utiliser NumPy (**`polyfit`** et **`polyval`**) pour calculer les tendances des séries **`y_max`** et **`y_min`**.
7. Visualiser graphiquement les valeurs et tendances des séries

Bonus

Exercice 7 : Parcoursup++

Objectif : Visualiser le taux d'admission d'étudiant provenant de bacs technologiques dans trois composantes de l'IUT sur la période 2018 → 2021

Récupérer les fichiers de données concernant les vœux de poursuite d'études Parcoursup pour la période 2018 → 2021 sur la plateforme ouverte des données publiques françaises : data.gouv.fr (données relatives à l'éducation).

1. Créer un dataframe **`df_201x`** pour chaque année concernés par l'étude à partir des données téléchargées.
2. Modifier **`df`** afin de garder les lignes correspondant à l'IUT Robert Schuman.
3. Modifier **`df`** afin de garder les lignes correspondant aux formations suivantes :
    a. Chimie
    b. Informatique
    c. Techniques de commercialisation
4. Modifier **`df`** pour garder les colonnes suivantes :
    a. l'année de la session
    b. l'intitulé de la formation
    c. le pourcentage de bac techno
5. Afficher sous forme de graphique l'évolution du taux d'admission des étudiants provenant de bacs technologiques.

Annexes

Données utilisées

Parcoursup	fr-esr-parcoursup_2021.csv
Température	DECADAGRO_67_previous-1950-2022.csv.gz
Parcoursup++	fr-esr-parcoursup_2021.csv
	fr-esr-parcoursup_2020.csv
	fr-esr-parcoursup_2019.csv
	fr-esr-parcoursup_2018.csv