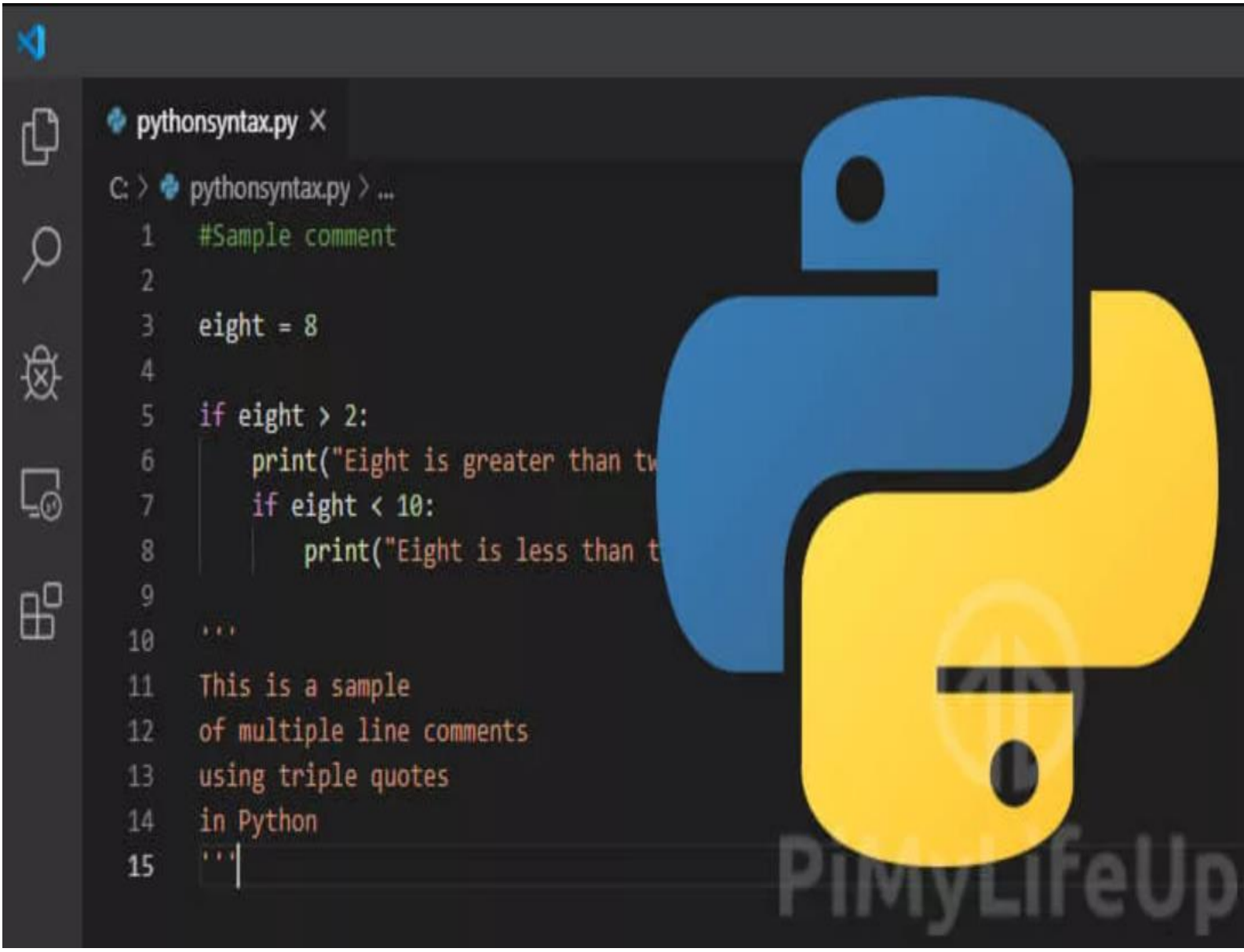
Compte rendu d’exercice 5 TP 5



Réalisé par :

Hassan NACHIT

Spécialité : Impression 3D et Intelligence Artificiel

Introduction

Ce rapport présente les solutions détaillées pour les trois parties de l'Exercice 5, qui couvrent différentes techniques de manipulation et d'analyse de données en Python. Nous utiliserons principalement les bibliothèques NumPy et Matplotlib pour effectuer ces opérations.

1. Matrice de covariance d'un tableau de variables aléatoires

Objectif

Créer un tableau de données de dimensions (100, 3) où chaque colonne représente une variable aléatoire, puis calculer sa matrice de covariance.

Méthodologie

1. Génération d'un tableau de 100 observations pour 3 variables aléatoires en utilisant np.random.rand().
2. Calcul de la matrice de covariance avec np.cov().

Code python :

```
import numpy as np
data = np.random.rand(100, 3) # Génération des données (100 observations, 3 variables)
print("Matrice de covariance (3x3):") # afficher la matrice
cov_matrix = np.cov(data, rowvar=False) # Calcul de la matrice de covariance (3x3)
```

Résultats :

La matrice de covariance est une matrice symétrique 3x3 où :

```
Matrice de covariance (3x3):
[[ 1.16901489  0.22976433 -0.008734  ]
 [ 0.22976433  1.20966487 -0.11917286]
 [-0.008734   -0.11917286  1.35210141]]
```

- Les éléments diagonaux représentent les variances de chaque variable.
- Les éléments hors diagonale représentent les covariances entre les variables.

Interprétation

Une covariance positive entre deux variables indique qu'elles évoluent dans le même sens, tandis qu'une covariance négative suggère une relation inverse.

2. Transformation de Fourier d'un signal sinusoïdal

Objectif

Appliquer la transformation de Fourier sur un tableau de données sinusoïdales et afficher le spectre de fréquences.

Méthodologie

1. Création d'un signal sinusoïdal avec une fréquence donnée.
2. Application de la FFT (Fast Fourier Transform) avec np.fft.fft().
3. Calcul des fréquences correspondantes et affichage du spectre.

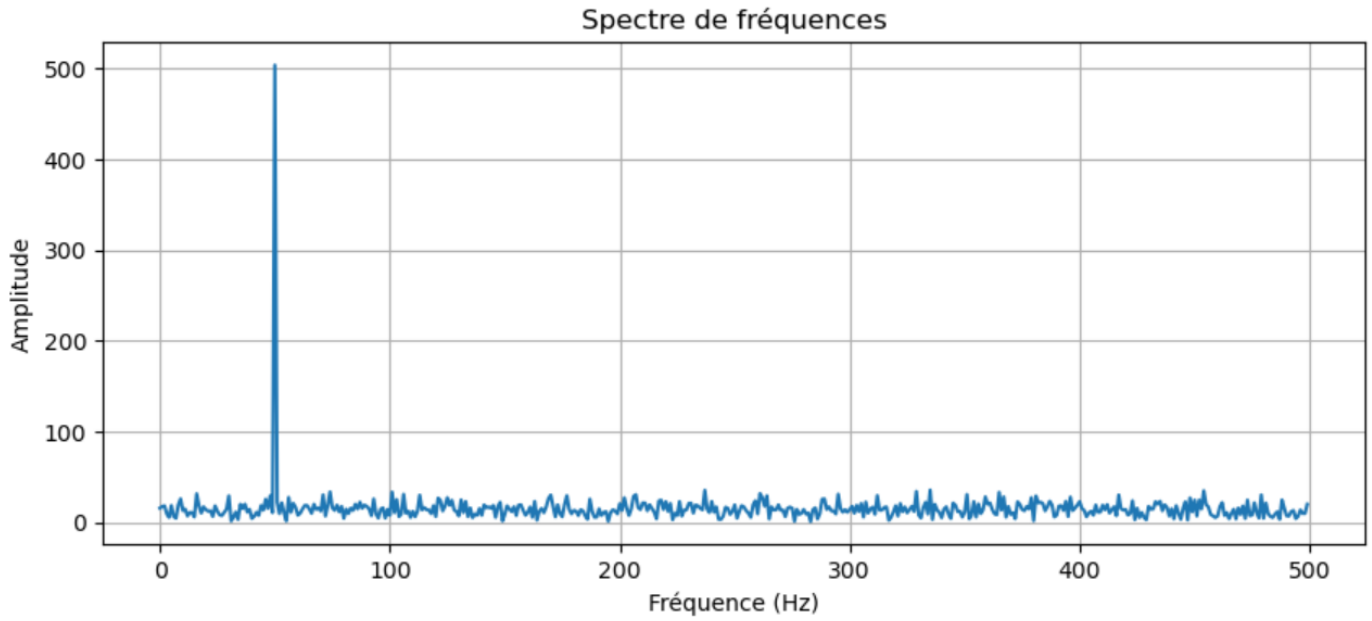
Code python :

```
import matplotlib.pyplot as plt # Paramètres du signal
fs = 100  # Fréquence du signal (Hz)
sampling_rate = 100  # Taux d'échantillonnage (Hz)
duration = 1  # Durée du signal (s)
t = np.arange(0, 1, 1/fs)
 Transformation de Fourier :
fft_result = np.fft.fft(signal)
frequencies = np.fft.fftfreq(len(signal), 1/sampling_rate)

# Affichage du spectre
plt.figure()
plt.plot(frequencies[:len(frequencies)//2], np.abs(fft_result[:len(fft_result)//2]))
plt.xlabel('Fréquence (Hz)')
plt.ylabel('Amplitude')
plt.title('Spectre de fréquences')
plt.grid()
plt.show()
```

Résultats :

Le spectre affiche un pic à la fréquence du signal sinusoïdal (50 Hz), confirmant la présence de cette composante fréquentielle.



Interprétation

La FFT permet d'identifier les fréquences dominantes dans un signal. Ici, le pic isolé correspond à la fréquence pure du signal généré.

3. Simulation de lancers de dés et histogramme des sommes

Objectif

Simuler 1000 lancers de deux dés et afficher l'histogramme des sommes obtenues.

Méthodologie

1. Simulation de 1000 lancers pour deux dés en utilisant np.random.randint().
2. Calcul des sommes pour chaque lancer.
3. Génération d'un histogramme avec plt.hist().

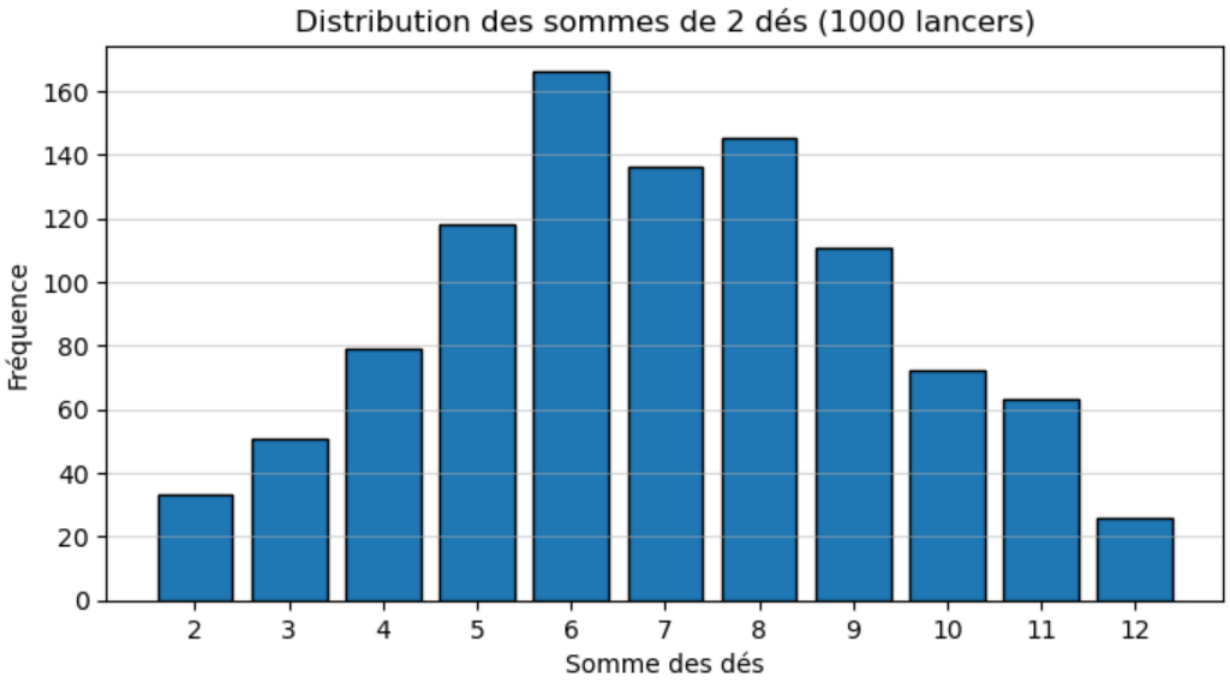
Code python :

```
# Simulation des lancers
np.random.seed(42)
lancers = np.random.randint(1, 7, (1000, 2))
sommes = lancers.sum(axis=1)

# Affichage de l'histogramme
plt.figure(figsize=(8, 4))
plt.hist(sommes, bins=range(2, 14), edgecolor='black', align='left', rwidth=0.8)
plt.title("Distribution des sommes de 2 dés (1000 lancers)")
plt.xlabel("Somme des dés")
plt.ylabel("Fréquence")
plt.xticks(range(2, 13))
plt.grid(axis='y', alpha=0.5)
plt.show()
```

Résultats :

L'histogramme montre une distribution triangulaire des sommes, avec un maximum autour de 7 (la somme la plus probable).



Interprétation

La somme de deux dés suit une distribution connue où les valeurs centrales (6-8) sont plus probables que les extrêmes (2 ou 12), ce qui est confirmé par l'histogramme.

Conclusion

Cet exercice a permis d'explorer trois techniques fondamentales en analyse de données :

1. La **matrice de covariance** pour étudier les relations entre variables.
2. La **transformation de Fourier** pour analyser les composantes fréquentielles d'un signal.
3. La **simulation et visualisation** de distributions probabilistes.

Ces méthodes sont largement applicables en science des données, en traitement du signal et en modélisation statistique.