

# QCM Data Science Python

Voici un autre exemple d'examen QCM sur les modules **Matplotlib**, **NumPy**, **Pandas**, et **Seaborn** avec des questions de type "compléter le code", un peu plus avancé :

## QCM Avancé - Examen Data Science avec Python

### Partie 1 : Matplotlib

1. Compléter le code pour tracer un graphique simple avec Matplotlib :

python

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [1, 4, 9, 16, 25]

_____(x, y)
_____("x values", "y values", "Square Numbers")
_____()
```

- A) `plt.plot()`, `plt.xlabel()`, `plt.title()`
- B) `plt.show()`, `plt.grid()`, `plt.legend()`
- C) `plt.bar()`, `plt.annotate()`, `plt.legend()`
- D) `plt.scatter()`, `plt.set_xlim()`, `plt.set_ylim()`

Réponse correcte : A

### Partie 2 : NumPy

2. Compléter le code pour créer un tableau NumPy et effectuer une addition élément par élément avec un autre tableau :

python

```
import numpy as np

array1 = np.array([1, 2, 3, 4])
array2 = np.array([5, 6, 7, 8])

_____
_____
```

- A) `array1.sum()` , `array2.sum()`
- B) `array1 + array2` , `array1 - array2`
- C) `np.add(array1, array2)` , `np.subtract(array1, array2)`
- D) `array1.append(array2)` , `array1.extend(array2)`

Réponse correcte : B

---

### Partie 3 : Pandas

3. Compléter le code pour charger un fichier CSV et afficher les 5 premières lignes du DataFrame :

python

```
import pandas as pd

df = pd._____('data.csv')
_____()
```

- A) `read_csv` , `head()`
- B) `load_csv` , `show()`
- C) `from_csv` , `top()`
- D) `open_csv` , `first()`

Réponse correcte : A

## Partie 4 : Seaborn

### 4. Compléter le code pour tracer une carte de chaleur (heatmap) avec Seaborn :

```
python

import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt

data = np.random.rand(5, 5)

sns._____(data, annot=True, cmap='coolwarm')
plt.show()
```

- A) `heatmap()`
- B) `barplot()`
- C) `lineplot()`
- D) `scatterplot()`

Réponse correcte : A

## Partie 5 : Applications avancées - Code à compléter

### 5. Compléter le code pour créer un graphique combiné avec Matplotlib et Seaborn montrant à la fois une distribution et un nuage de points :

```
python

import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

# Générer des données aléatoires
data = np.random.normal(size=100)
```

```
plt.figure(figsize=(10, 6))

# Tracer le nuage de points avec Seaborn
sns._____(data)

# Tracer l'histogramme avec Matplotlib
plt._____(data, bins=15, alpha=0.5)

_____( )
```

- A) `scatterplot()`, `hist()`, `plt.show()`
- B) `distplot()`, `histplot()`, `plt.show()`
- C) `boxplot()`, `plot()`, `plt.display()`
- D) `lineplot()`, `bar()`, `plt.display()`

Réponse correcte : B

---

## Partie 6 : Analyse de données avec Pandas et Seaborn

6. Compléter le code pour charger un fichier CSV avec Pandas, filtrer les données et créer un graphique en boîte (boxplot) avec Seaborn :

```
python

import pandas as pd
import seaborn as sns

# Charger les données
df = pd.read_csv('data.csv')

# Filtrer les données où la colonne 'Age' est supérieure à 30
filtered_df = df[_____]

# Tracer un boxplot de la colonne 'Income'
sns._____(x='Age', y='Income', data=filtered_df)
```

- A) `df['Age'] > 30`, `boxplot()`

- B) `df['Age'] < 30, scatterplot()`
- C) `df['Age'] > 30, histplot()`
- D) `df['Age'] == 30, distplot()`

Réponse correcte : A

---

## Partie 7 : NumPy Avancé

7. Compléter le code pour effectuer une opération matricielle avec NumPy :

python

```
import numpy as np

# Créer deux matrices
matrix1 = np.array([[1, 2], [3, 4]])
matrix2 = np.array([[5, 6], [7, 8]])

# Effectuer une multiplication matricielle
result = np._____(matrix1, matrix2)
```

- A) `np.dot()`
- B) `np.multiply()`
- C) `np.add()`
- D) `np.subtract()`

Réponse correcte : A

---

## Partie 8 : Visualisation avancée avec Seaborn

8. Compléter le code pour tracer un graphique de régression avec Seaborn :

python

```
import seaborn as sns
import matplotlib.pyplot as plt

# Charger un jeu de données intégré
tips = sns.load_dataset('tips')

# Tracer un graphique de régression
sns._____(x='total_bill', y='tip', data=tips)
plt.show()
```

- A) `regplot()`
- B) `boxplot()`
- C) `lineplot()`
- D) `scatterplot()`

Réponse correcte : A

---

Do you like this personality