



**IUS**  
INSTITUT  
UNIVERSITAIRE  
DES SCIENCES

**Faculté des Sciences et Technologie**  
(FST)

**Niveau : L3-FST**

**Cours : Mathématiques pour l'Informatique**

**Soumis au chargé de cours : Ismaël SAINT AMOUR**

**Préparé par : Jameson DOMINIQUE**

**Date : 07 Juin 2025**

# TD3- Programmation Mathématiques pour la science des données en python.

## ◆ Exercice 1 — Matrices de base

Objectif : Créer deux matrices 4x4, puis :

Afficher leur somme

Calculer leur produit élément par élément

Calculer leur produit matriciel

Afficher les 3 résultats et Visualisation Graphique des Matrices

## ◆ Exercice 2 — Probabilité de réussite

Objectif : Calculer la probabilité qu'un étudiant ait réussi un test dans une classe de 60 étudiants.

Données :

35 filles, dont 18 ont réussi.

25 garçons, dont 12 ont réussi.

Instructions :

Pente (coef) : 0.0026242424242423

Intercept : -0.23636363636363455

Calculer la probabilité qu'un étudiant pris au hasard ait réussi.

Afficher le résultat sous forme de graphe circulaire (camembert) en utilisant matplotlib.

Interprétez les résultats.

## ◆ Exercice 3 — Probabilité conditionnelle

Objectif : Calculer la probabilité qu'un étudiant soit une fille sachant qu'il a réussi.

Données :

35 filles, dont 18 ont réussi.

25 garçons, dont 12 ont réussi.

Instructions :

Calculer la probabilité qu'un étudiant qui a réussi soit une fille.

Afficher les résultats sous forme de graphe circulaire.

Interprétez la probabilité obtenue.

## ◆ Exercice 4 : Prédiction du taux de réussite scolaire en fonction des investissements en éducation

Objectif :

Utiliser la régression linéaire pour prédire le taux de réussite scolaire (en %) en fonction des investissements dans l'éducation (en millions de gourdes).

Instructions :

Crée un jeu de données simulées représentant les investissements dans l'éducation et les taux de réussite scolaire.

Applique une régression linéaire pour prédire le taux de réussite en fonction des investissements.

Visualise les résultats sous forme de graphique.

Interprète la pente et l'intercepte du modèle.

Données simulées :

Investissements (en millions de gourdes) : [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

Taux de réussite scolaire (%) : [45, 50, 55, 60, 65, 70, 75, 80, 85, 90]

## ◆ Exercice 5 : Prédiction de l'accès à l'eau potable en fonction du taux de pauvreté en Haïti

Objectif :

Utiliser la régression linéaire pour prédire l'accès à l'eau potable (en pourcentage de la population ayant accès) en fonction du taux de pauvreté dans différentes régions d'Haïti.

Instructions :

Crée un jeu de données simulées représentant le taux de pauvreté (%) et l'accès à l'eau potable (%) pour plusieurs régions d'Haïti.

Applique une régression linéaire pour prédire l'accès à l'eau potable en fonction du taux de pauvreté.

Affiche les résultats sous forme de graphique.

Calcule et interprète la pente et l'intercepte du modèle.

Données simulées :

Taux de pauvreté (%) : [20, 30, 40, 50, 60, 70, 80, 90, 100]

Accès à l'eau potable (%) : [95, 90, 85, 80, 75, 70, 65, 60, 55]

```

# -*- coding: utf-8 -*-
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

def afficher_menu():
    """Affiche le menu principal"""
    print("\n" + "="*50)
    print("MENU PRINCIPAL - TP DE STATISTIQUES")
    print("="*50)
    print("1. Opérations sur matrices")
    print("2. Probabilité de réussite")
    print("3. Probabilité conditionnelle")
    print("4. Réussite scolaire (régression linéaire)")
    print("5. Accès à l'eau potable (régression linéaire)")
    print("0. Quitter")

def exercice1():
    """Exercice sur les opérations matricielles"""
    print("\n=== Exercice 1 - Opérations sur matrices ===")

    # Création de matrices fixes pour reproductibilité
    A = np.array([[1, 2, 3, 4],
                  [5, 6, 7, 8],
                  [9, 10, 11, 12],
                  [13, 14, 15, 16]])

    B = np.array([[16, 15, 14, 13],
                  [12, 11, 10, 9],
                  [8, 7, 6, 5],
                  [4, 3, 2, 1]])

    # Calculs matriciels
    somme = A + B
    produit_element = A * B
    produit_matriciel = A @ B # Alternative à np.dot

    # Affichage des résultats
    print("\nMatrice A:")
    print(A)
    print("\nMatrice B:")
    print(B)
    print("\nSomme A + B:")
    print(somme)
    print("\nProduit élément par élément:")
    print(produit_element)
    print("\nProduit matriciel A × B:")
    print(produit_matriciel)

```

```

# Visualisation
fig, axs = plt.subplots(2, 2, figsize=(10, 8))
fig.suptitle('Visualisation des Opérations Matricielles')

im = axs[0,0].imshow(A, cmap='viridis')
axs[0,0].set_title('Matrice A')
fig.colorbar(im, ax=axs[0,0])

im = axs[0,1].imshow(B, cmap='viridis')
axs[0,1].set_title('Matrice B')
fig.colorbar(im, ax=axs[0,1])

im = axs[1,0].imshow(somme, cmap='viridis')
axs[1,0].set_title('Somme A+B')
fig.colorbar(im, ax=axs[1,0])

im = axs[1,1].imshow(produit_matriciel, cmap='viridis')
axs[1,1].set_title('Produit Matriciel A×B')
fig.colorbar(im, ax=axs[1,1])

plt.tight_layout()
plt.show()

def exercice2():
    """Calcul de probabilité de réussite"""
    print("\n=== Exercice 2 - Probabilité de réussite ===")

    # Données initiales
    filles_reussies = 18
    filles_total = 35
    garcons_reussis = 12
    garcons_total = 25
    total_etudiants = 60

    # Calculs des probabilités
    prob_reussite = (filles_reussies + garcons_reussis) / total_etudiants
    prob_echec = 1 - prob_reussite

    # Affichage des résultats
    print(f"\nNombre total d'étudiants: {total_etudiants}")
    print(f"Filles ayant réussi: {filles_reussies}/{filles_total}")
    print(f"Garçons ayant réussi: {garcons_reussis}/{garcons_total}")
    print(f"\nProbabilité de réussite: {prob_reussite:.2%}")
    print(f"Probabilité d'échec: {prob_echec:.2%}")

    # Visualisation
    plt.figure(figsize=(8, 6))
    plt.pie([prob_reussite, prob_echec],
            labels=['Réussite', 'Échec'],

```

```

        autopct=lambda p: f'{p:.1f}% ({p*total_etudiants/100:.0f}
étudiants)',
        colors=['#4CAF50', '#F44336'],
        explode=(0.1, 0),
        shadow=True,
        startangle=90)
    plt.title(f'Répartition réussite/échec (Classe de {total_etudiants}
étudiants)')
    plt.show()

def exercice3():
    """Calcul de probabilité conditionnelle"""
    print("\n=== Exercice 3 - Probabilité conditionnelle ===")

    # Données reprises de l'exercice 2
    filles_reussies = 18
    garcons_reussis = 12
    total_reussis = filles_reussies + garcons_reussis

    # Calcul de la probabilité conditionnelle
    prob_fille_reussite = filles_reussies / total_reussis
    # Corrected indentation for the next line
    prob_garcon_reussite = garcons_reussis / total_reussis

    # Affichage des résultats
    print(f"\nTotal d'étudiants ayant réussi: {total_reussis}")
    print(f"Filles ayant réussi: {filles_reussies}")
    print(f"Garçons ayant réussi: {garcons_reussis}")
    print(f"\nP(Fille | Réussite) = {prob_fille_reussite:.2%}")
    print(f"P(Garçon | Réussite) = {prob_garcon_reussite:.2%}")

    # Visualisation
    plt.figure(figsize=(8, 6))
    plt.pie([filles_reussies, garcons_reussis],
            labels=['Filles', 'Garçons'],
            autopct=lambda p: f'{p:.1f}% ({p*total_reussis/100:.0f}
étudiants)',
            colors=['#FFC0CB', '#ADD8E6'],
            explode=(0.05, 0),
            shadow=True,
            startangle=90)
    plt.title('Répartition des réussites par genre')
    plt.show()

def exercice4():
    """Régression linéaire - Réussite scolaire"""
    print("\n=== Exercice 4 - Réussite scolaire ===")

    # Données simulées

```

```

investissements = np.array([10, 20, 30, 40, 50, 60, 70, 80, 90, 100])
taux_reussite = np.array([45, 50, 55, 60, 65, 70, 75, 80, 85, 90])

# Modèle de régression
X = investissements.reshape(-1, 1)
y = taux_reussite

model = LinearRegression()
model.fit(X, y)

# Prédiction et métriques
predictions = model.predict(X)
r_squared = model.score(X, y)

# Affichage des résultats
print("\nDonnées utilisées:")
print("Investissements (M gourdes) | Taux réussite (%)")
for inv, taux in zip(investissements, taux_reussite):
    print(f"{inv:^23} | {taux:^15}")

print(f"\nCoefficient (pente): {model.coef_[0]:.6f}")
print(f"Ordonnée à l'origine (intercept): {model.intercept_:.6f}")
print(f"Coefficient de détermination (R²): {r_squared:.4f}")
print(f"\nÉquation: taux_réussite = {model.coef_[0]:.4f} × investissement
+ {model.intercept_:.4f}")

# Visualisation
plt.figure(figsize=(10, 6))
plt.scatter(X, y, color='blue', s=80, label='Données réelles')
plt.plot(X, predictions, color='red', linewidth=2, label='Modèle de
régression')

# Ajout des annotations
for i, txt in enumerate(taux_reussite):
    plt.annotate(txt, (investissements[i], taux_reussite[i]),
                 textcoords="offset points", xytext=(0,5), ha='center')

plt.title('Taux de réussite scolaire en fonction des investissements',
pad=20)
plt.xlabel('Investissements (millions de gourdes)')
plt.ylabel('Taux de réussite (%)')
plt.grid(True, linestyle='--', alpha=0.5)
plt.legend()
plt.tight_layout()
plt.show()

def exercice5():
    """Régression linéaire - Accès à l'eau potable"""
    print("\n=== Exercice 5 - Accès à l'eau potable ===")

```

```

# Données simulées
taux_pauvrete = np.array([20, 30, 40, 50, 60, 70, 80, 90, 100])
acces_eau = np.array([95, 90, 85, 80, 75, 70, 65, 60, 55])

# Modèle de régression
X = taux_pauvrete.reshape(-1, 1)
y = acces_eau

model = LinearRegression()
model.fit(X, y)

# Prédictions et métriques
predictions = model.predict(X)
r_squared = model.score(X, y)

# Affichage des résultats
print("\nDonnées utilisées:")
print("Taux pauvreté (%) | Accès eau potable (%)")
for pauv, eau in zip(taux_pauvrete, acces_eau):
    print(f"{pauv:^16} | {eau:^18}")

print(f"\nCoefficient (pente): {model.coef_[0]:.6f}")
print(f"Ordonnée à l'origine (intercept): {model.intercept_:.6f}")
print(f"Coefficient de détermination (R²): {r_squared:.4f}")
print(f"\nÉquation: accès_eau = {model.coef_[0]:.4f} × taux_pauvreté + {model.intercept_:.4f}")

# Visualisation
plt.figure(figsize=(10, 6))
plt.scatter(X, y, color='green', s=80, label='Données réelles')
plt.plot(X, predictions, color='orange', linewidth=2, label='Modèle de régression')

# Ajout des annotations
for i, txt in enumerate(acces_eau):
    plt.annotate(txt, (taux_pauvrete[i], acces_eau[i]),
                  textcoords="offset points", xytext=(0,5), ha='center')

plt.title("Accès à l'eau potable en fonction du taux de pauvreté",
pad=20)
plt.xlabel('Taux de pauvreté (%)')
plt.ylabel('Accès à l\'eau potable (%)')
plt.grid(True, linestyle='--', alpha=0.5)
plt.legend()
plt.tight_layout()
plt.show()

def main():

```

```

"""Fonction principale"""
while True:
    afficher_menu()
    choix = input("\nVotre choix (0-5): ").strip()

    if choix == "1":
        exercice1()
    elif choix == "2":
        exercice2()
    elif choix == "3":
        exercice3()
    elif choix == "4":
        exercice4()
    elif choix == "5":
        exercice5()
    elif choix == "0":
        print("\nMerci d'avoir utilisé ce programme. Au revoir!")
        break
    else:
        print("\nErreur: Veuillez entrer un nombre entre 0 et 5.")

    input("\nAppuyez sur Entrée pour continuer...")

# Corrected the if name == "main": block indentation
if __name__ == "__main__":
    print("TP de Statistiques et Algèbre Linéaire")
    print("-----")
    main()

```



## Les tests de 1 à 5

### Exercice 1

TP de Statistiques et Algèbre Linéaire

-----

=====

MENU PRINCIPAL - TP DE STATISTIQUES

=====

1. Opérations sur matrices
2. Probabilité de réussite
3. Probabilité conditionnelle
4. Réussite scolaire (régression linéaire)
5. Accès à l'eau potable (régression linéaire)
0. Quitter

Votre choix (0-5): 1

=== Exercice 1 - Opérations sur matrices ===

Matrice A:

```
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]
 [13 14 15 16]]
```

Matrice B:

```
[[16 15 14 13]
 [12 11 10  9]
 [ 8  7  6  5]
 [ 4  3  2  1]]
```

Somme A + B:

```
[[17 17 17 17]
 [17 17 17 17]
 [17 17 17 17]
 [17 17 17 17]]
```

Produit élément par élément:

```
[[16 30 42 52]
 [60 66 70 72]
 [72 70 66 60]
 [52 42 30 16]]
```

Produit matriciel A x B:

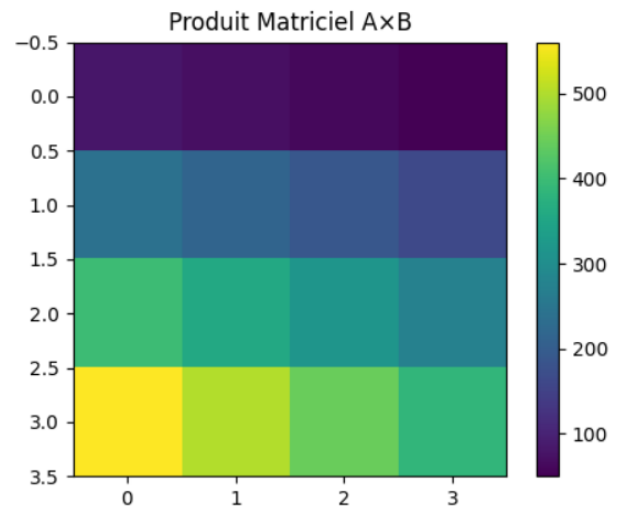
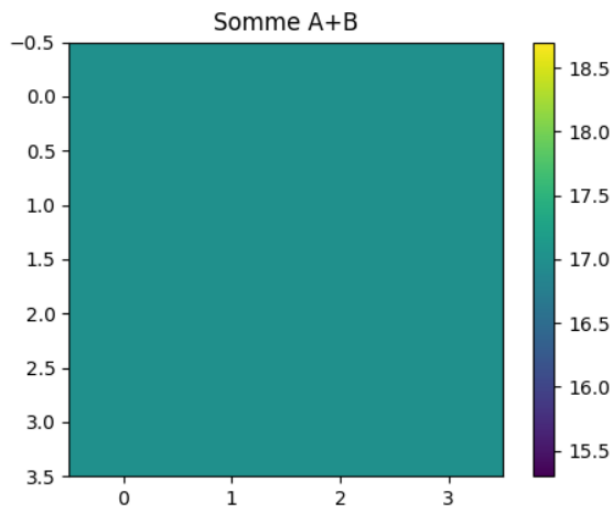
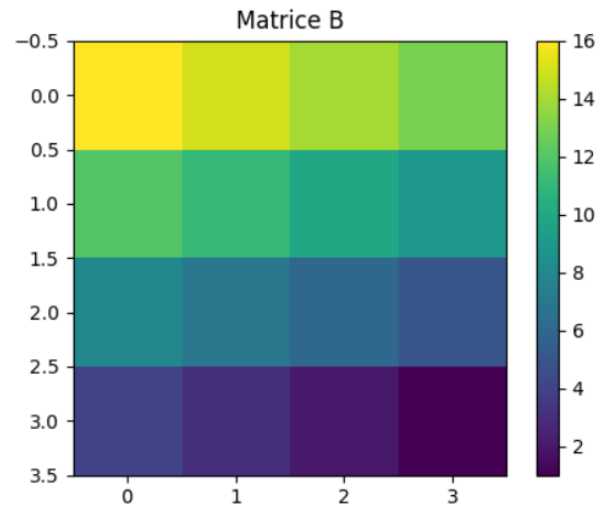
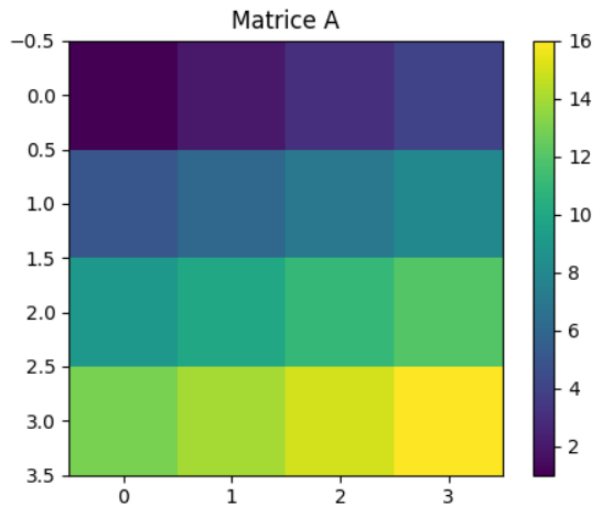
```
[[ 80  70  60  50]
 [240 214 188 162]
 [400 358 316 274]
 [560 502 444 386]]
```

Visualisation des Opérations Matricielles

---

```
[[240 214 188 162]
 [400 358 316 274]
 [560 502 444 386]]
```

## Visualisation des Opérations Matricielles



## Exercice 2

TP de Statistiques et Algèbre Linéaire

=====

MENU PRINCIPAL - TP DE STATISTIQUES

=====

1. Opérations sur matrices
2. Probabilité de réussite
3. Probabilité conditionnelle
4. Réussite scolaire (régression linéaire)
5. Accès à l'eau potable (régression linéaire)
0. Quitter

Votre choix (0-5): 2

=== Exercice 2 - Probabilité de réussite ===

Nombre total d'étudiants: 60

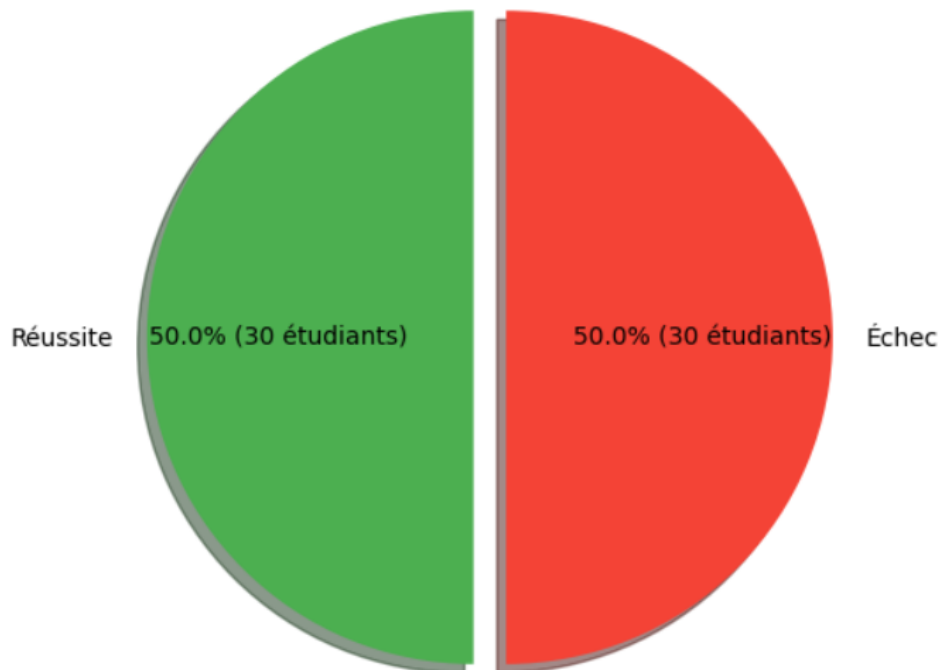
Filles ayant réussi: 18/35

Garçons ayant réussi: 12/25

Probabilité de réussite: 50.00%

Probabilité d'échec: 50.00%

Répartition réussite/échec (Classe de 60 étudiants)



## Exercice 3

TP de Statistiques et Algèbre Linéaire

=====

MENU PRINCIPAL - TP DE STATISTIQUES

=====

1. Opérations sur matrices
2. Probabilité de réussite
3. Probabilité conditionnelle
4. Réussite scolaire (régression linéaire)
5. Accès à l'eau potable (régression linéaire)
0. Quitter

Votre choix (0-5): 3

=== Exercice 3 - Probabilité conditionnelle ===

Total d'étudiants ayant réussi: 30

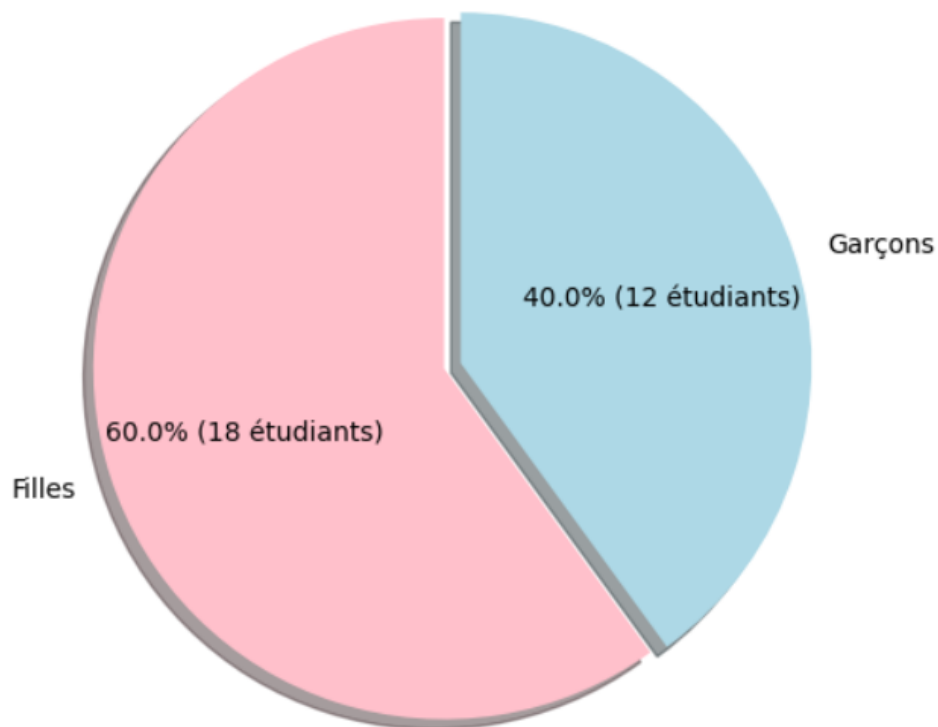
Filles ayant réussi: 18

Garçons ayant réussi: 12

$P(\text{Fille} \mid \text{Réussite}) = 60.00\%$

$P(\text{Garçon} \mid \text{Réussite}) = 40.00\%$

Répartition des réussites par genre



## Exercice 4

TP de Statistiques et Algèbre Linéaire

-----

=====

MENU PRINCIPAL - TP DE STATISTIQUES

=====

1. Opérations sur matrices
2. Probabilité de réussite
3. Probabilité conditionnelle
4. Réussite scolaire (régression linéaire)
5. Accès à l'eau potable (régression linéaire)
0. Quitter

Votre choix (0-5): 4

=== Exercice 4 - Réussite scolaire ===

Données utilisées:

Investissements (M gourdes) | Taux réussite (%)

10		45
20		50
30		55
40		60
50		65
60		70
70		75
80		80
90		85
100		90

Coefficient (pente): 0.500000

Ordonnée à l'origine (intercept): 40.000000

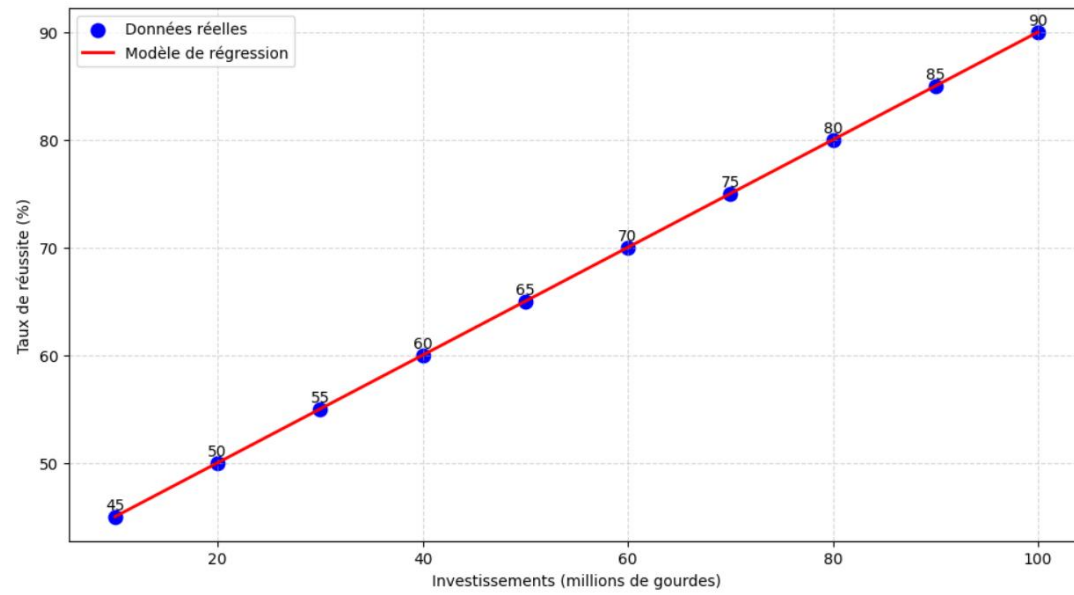
Coefficient de détermination ( $R^2$ ): 1.0000

Équation:  $\text{taux\_réussite} = 0.5000 \times \text{investissement} + 40.0000$

**Taux de réussite scolaire en fonction**

---

### Taux de réussite scolaire en fonction des investissements



## Exercice 5

TP de Statistiques et Algèbre Linéaire

-----

=====

MENU PRINCIPAL - TP DE STATISTIQUES

=====

1. Opérations sur matrices
2. Probabilité de réussite
3. Probabilité conditionnelle
4. Réussite scolaire (régression linéaire)
5. Accès à l'eau potable (régression linéaire)
0. Quitter

Votre choix (0-5): 5

=== Exercice 5 - Accès à l'eau potable ===

Données utilisées:

Taux pauvreté (%)		Accès eau potable (%)
20		95
30		90
40		85
50		80
60		75
70		70
80		65
90		60
100		55

Coefficient (pente): -0.500000

Ordonnée à l'origine (intercept): 105.000000

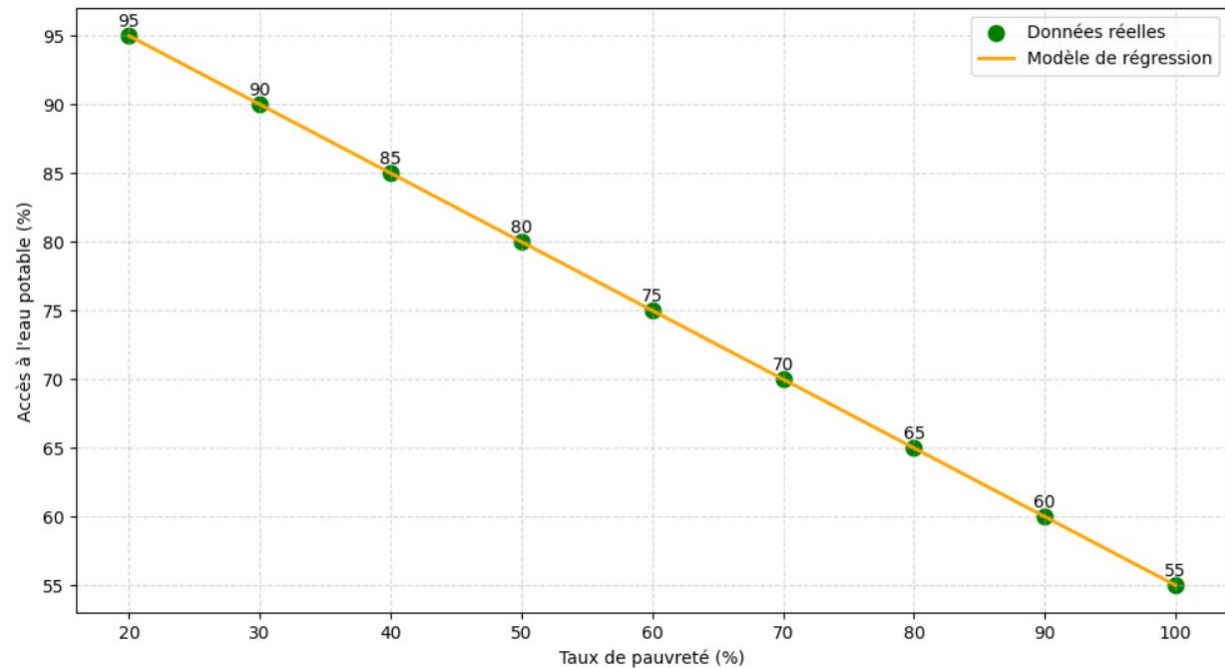
Coefficient de détermination ( $R^2$ ): 1.0000

Équation: accès\_eau = -0.5000 × taux\_pauvreté + 105.0000

---

$$\text{Equation: } \text{accès\_eau} = -0.0001 \times \text{taux\_pauvreté} + 10.0000$$

Accès à l'eau potable en fonction du taux de pauvreté



## Conclusion

Ce TP m'a permis d'approfondir mes connaissances en algèbre linéaire et en statistiques, en particulier sur les matrices, les probabilités et la régression linéaire.

### **Bilan des apprentissages**

Matrices : Compréhension des opérations de base et de leur visualisation.

Probabilités : Calcul de probabilités simples et conditionnelles, avec interprétation des résultats.

Régression linéaire : Modélisation de relations entre variables et interprétation des coefficients.

Visualisation : Utilisation de Matplotlib pour des graphiques clairs et professionnels.

Ce TP a renforcé mes compétences en programmation Python et en analyse de données, ce qui sera utile pour de futurs projets en science des données.