# 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) : 0.002624242424242423

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)

    # Prédictions 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\_\_":

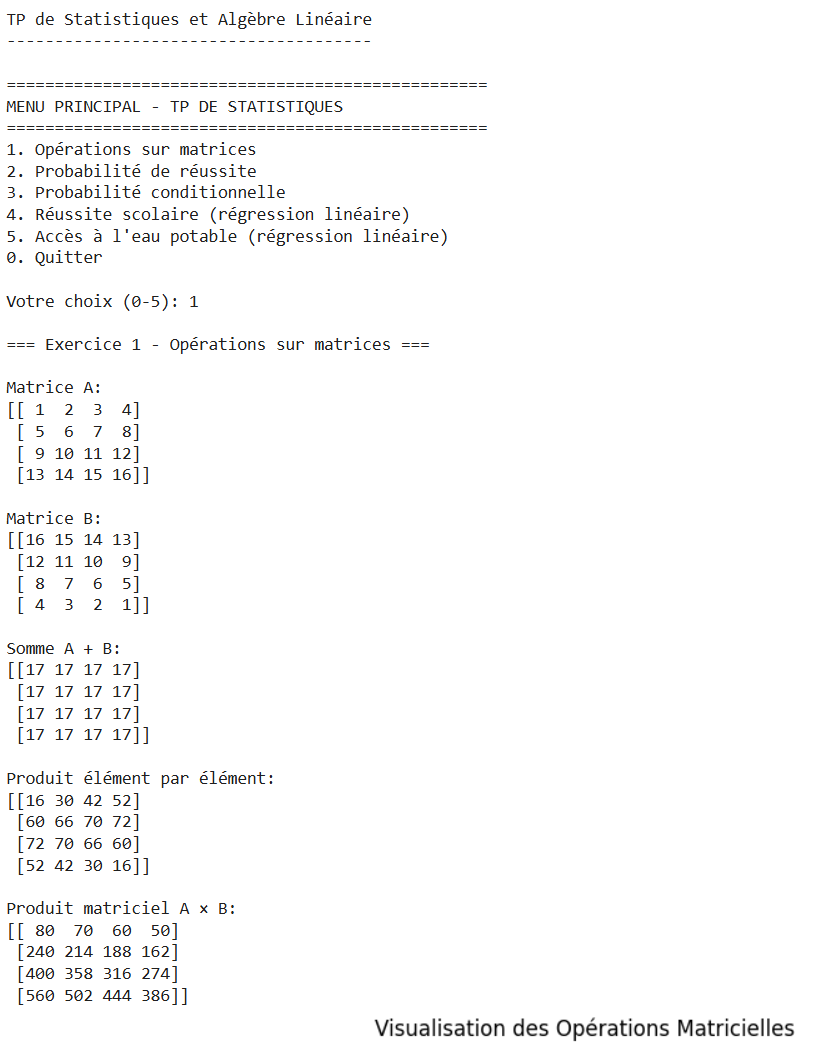
    print("TP de Statistiques et Algèbre Linéaire")

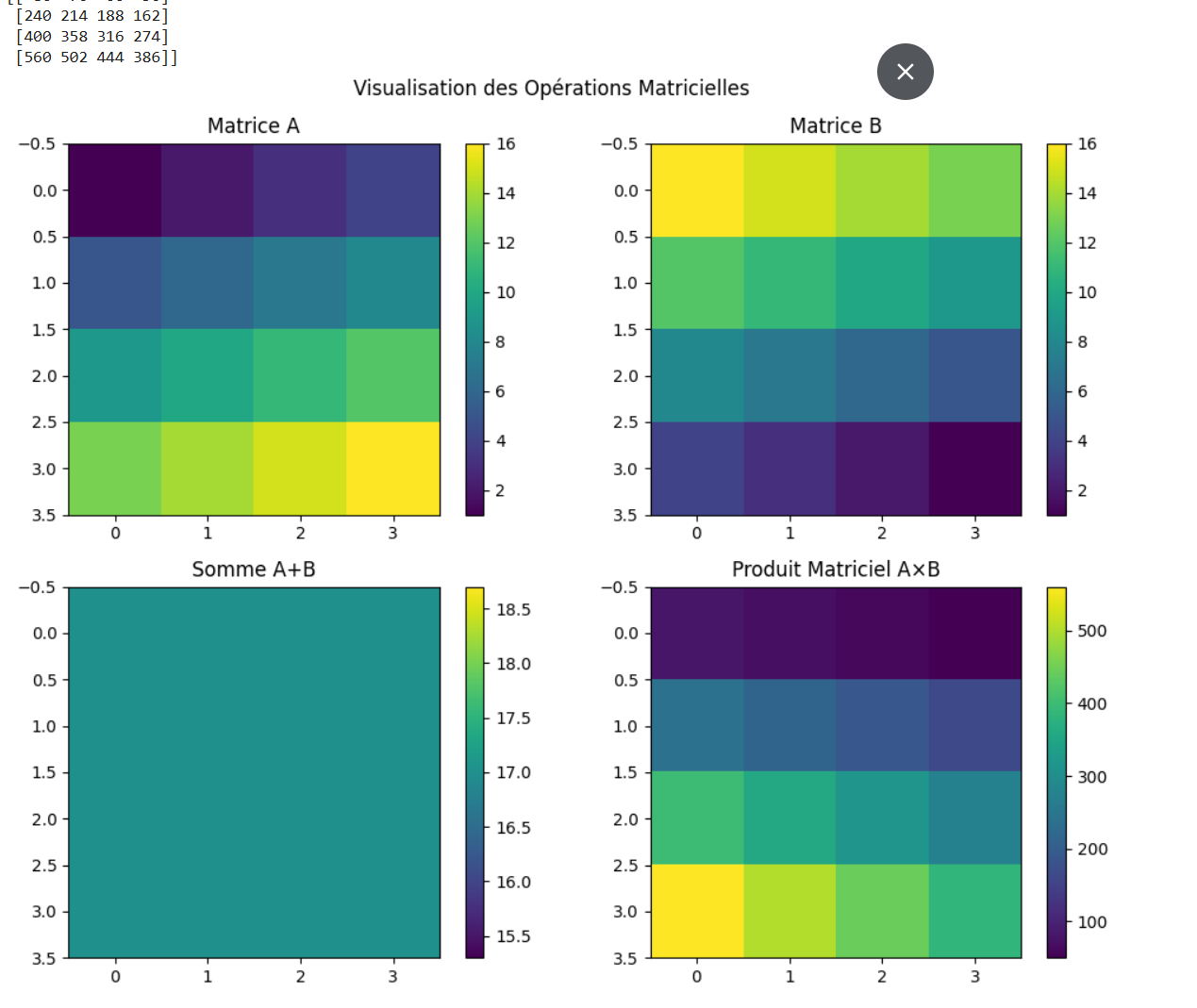
    print("--------------------------------------")

    main()

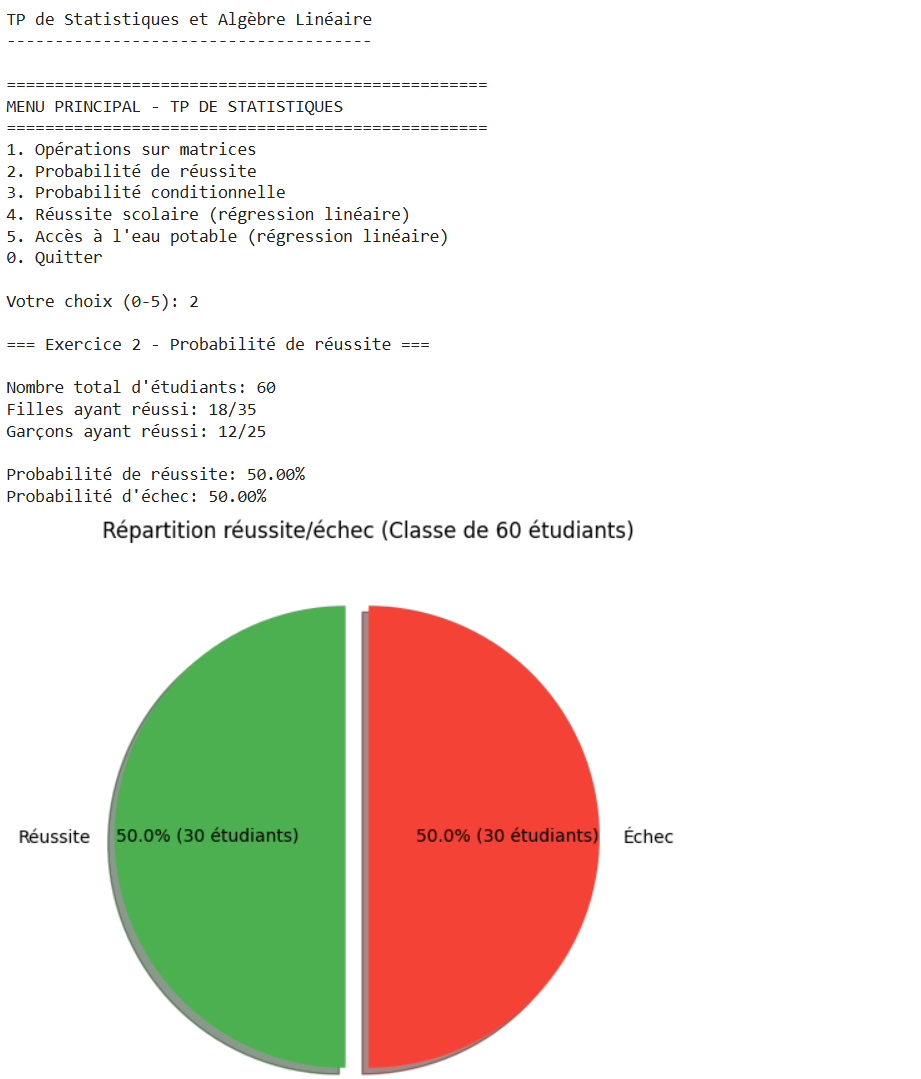
**Les tests de 1 à 5**

Execrcice 1

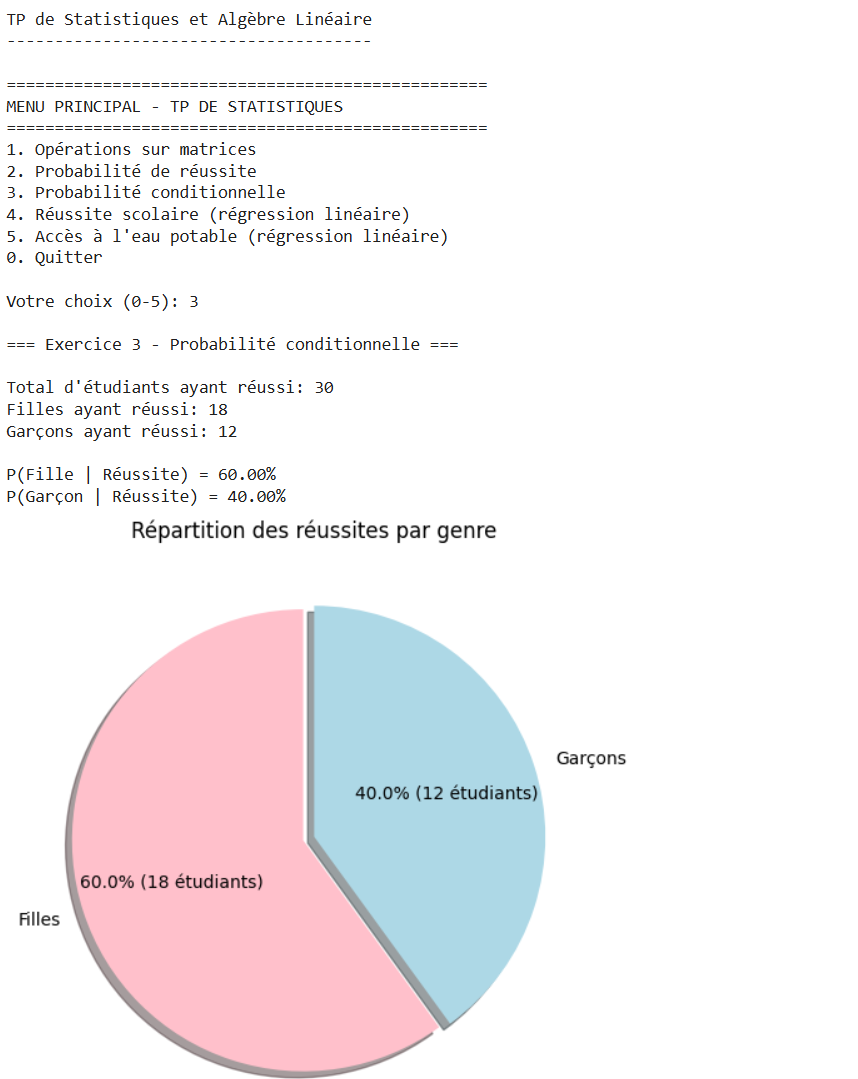




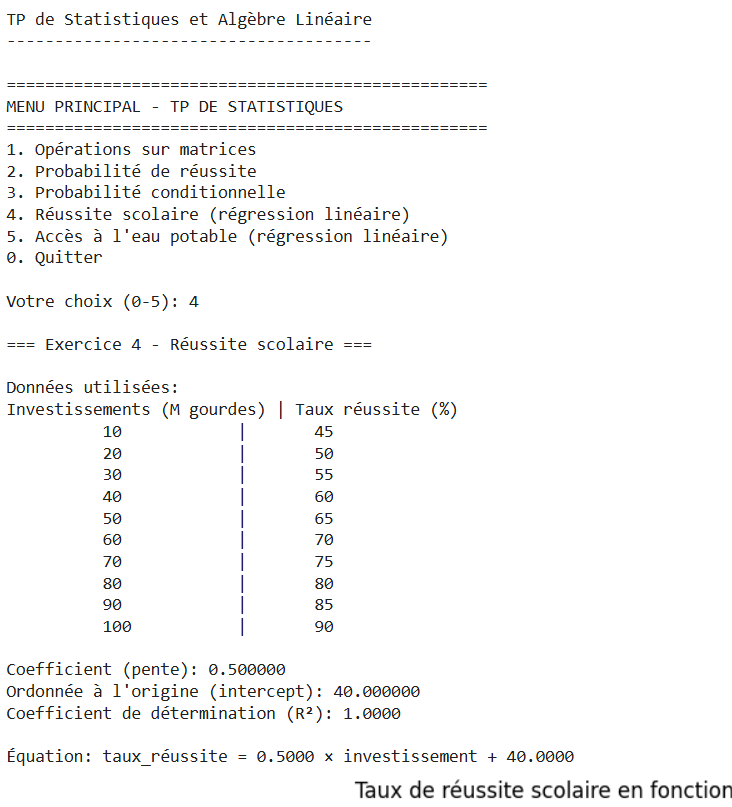
Execrcice 2

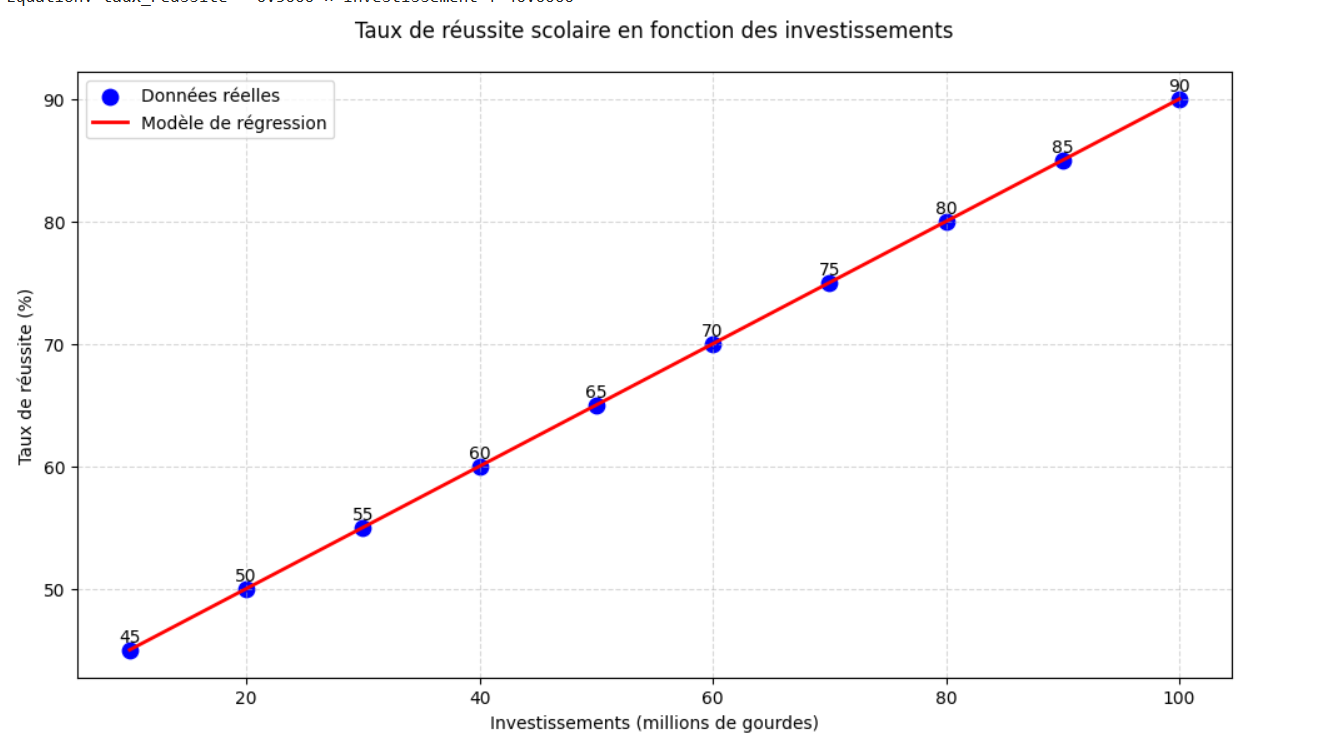


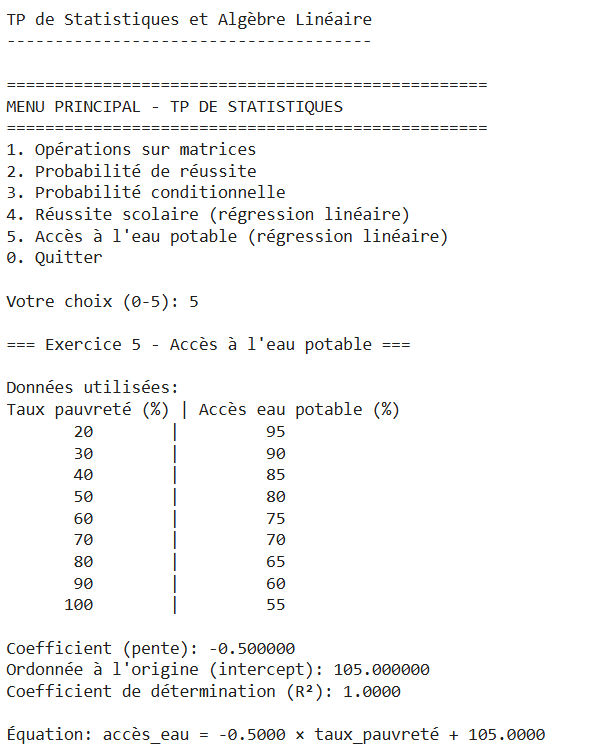
Execrcice 3

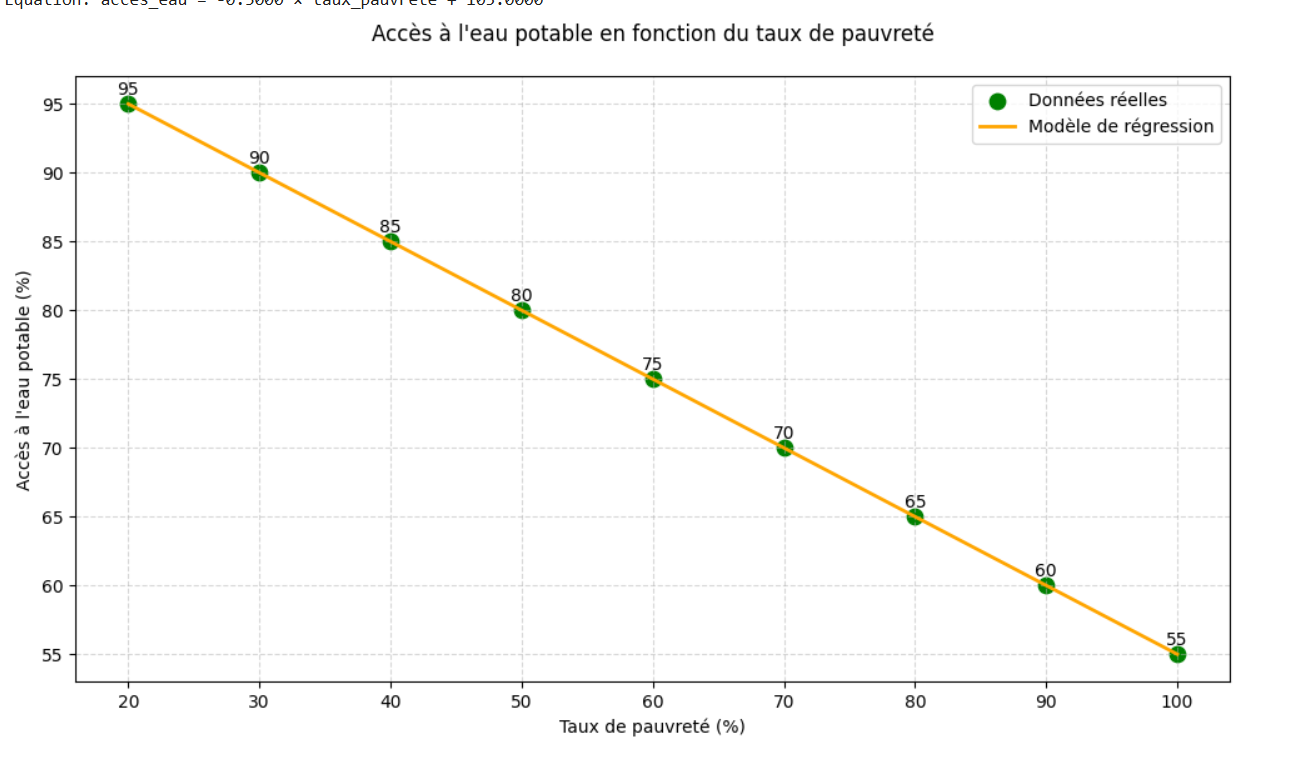


Execrcice 4





Execrcice 5



**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.