RAPPORT SAE 2.04

Partie 3 - Analyse statistique des données

- IUT Lannion - 2024/2025
- Département Informatique

- LE CHEVERE Yannis
- LE SECH Marceau

SOMMAIRE (lien cliquable)

I - Présentation des données

1) Présentation des données

Population étudiée : Étudiant ayant fourni leur prénom, leur moyenne générale, leur code postal, leur niveau d'études et leur mention au bac.
Description des variables :

- *prenom* : Prénom de l'étudiant (ex : Maurice)
- *moyenne* : Moyenne générale obtenue de l'étudiant (ex : 12.75)
- *code_postal* : Code postal de résidence de l'étudiant (ex : 22300)
- *niveau_etude* : Niveau d'étude de l'étudiant (ex : Terminale)
- *mention_bac* : Mention au bac de l'étudiant (ex : TB)

2) Problématique

En utilisant ces données, on va essayer de répondre à la problématique suivante :

Existe-t-il une influence de la première lettre du prénom sur le niveau d'études, la mention obtenue au bac, le code postal et la moyenne générale des étudiants ?

II - Import des données, mise en forme

1) Importer les données en Python

Les données sont importées en Python sous forme de DataFrame à l'aide de la commande suivante :

```
# Lecture du fichier CSV avec un séparateur ';' en DataFrame pandas
cheminFichier = "./vue.csv"
VueDf = pd.read_csv(cheminFichier, sep=";")
```

2) Mise en forme

On a besoin de supprimer les cases vides (qui contiennent nan en Pythons), puis on transforme notre DataFrame en Array :

```
# Suppression des lignes contenant des valeurs manquantes pour éviter
erreurs
Voyelle_df = Voyelle_df.dropna()
# Conversion optionnelle du DataFrame en tableau NumPy (pour certaines
opérations numériques)
Voyelle_ar = Voyelle_df.to_numpy()
```

3) Normalisation

Pour cela on prend toutes les variables qui étaient des chaînes de caractère pour les transformer en variable quantitatif.

- *prenom* devient *initiale*
- *mention_bac* devient *mention_bac_num*
- *niveau_etude* devient *niveau_etude_num*

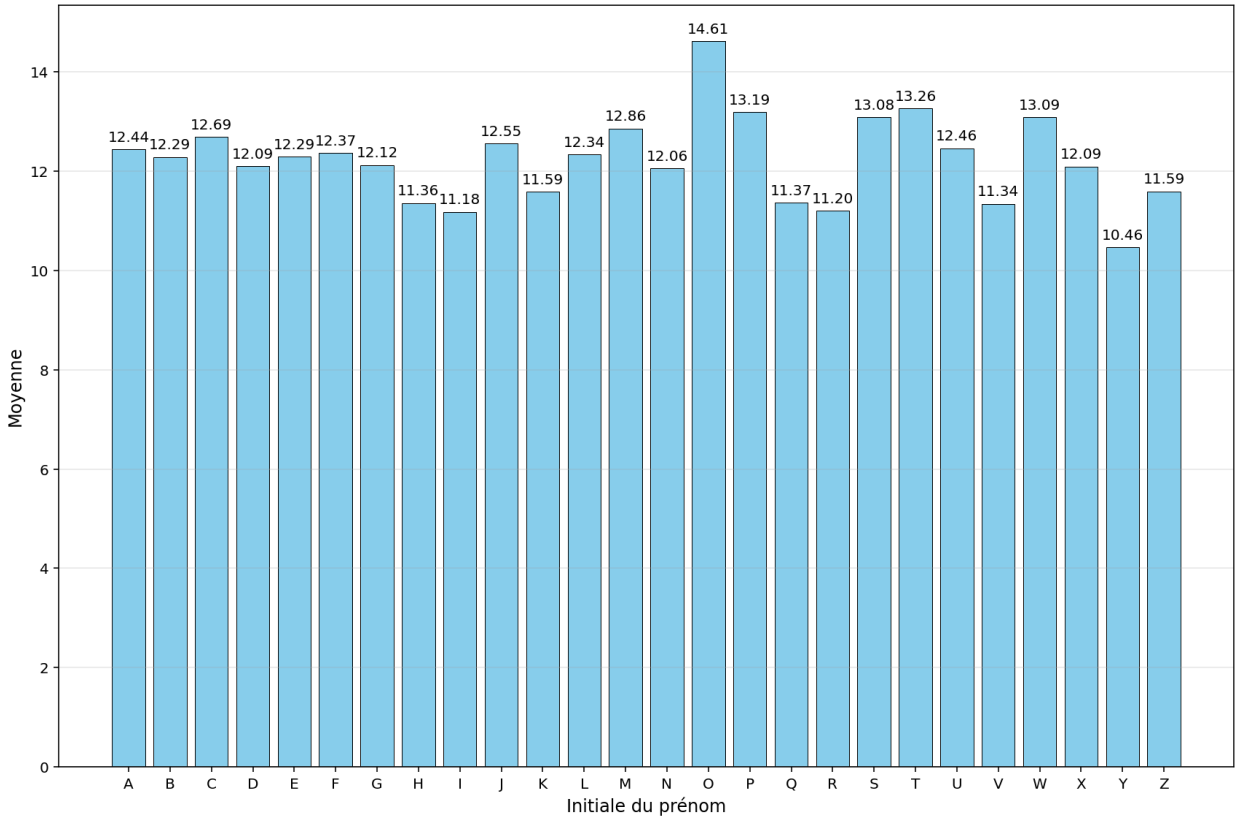
```
# Création d'une colonne contenant la première lettre du prénom
VueDf['initiale'] = VueDf['prenom'].str[0]
# Conversion des lettres en chiffres (A=1, B=2, ..., Z=26)


alphabet = list("ABCDEFGHIJKLMNOPQRSTUVWXYZ")
mapping_initiale = {lettre: i+1 for i, lettre in enumerate(alphabet)}
# Application de la conversion des initiales en valeurs numériques dans une
nouvelle colonne
VueDf['initiale_num'] = VueDf['initiale'].map(mapping_initiale)
# Encodage numérique des mentions au bac
VueDf['mention_bac_num'] = VueDf['mention_bac'].map({
    'P': 1, 'AB': 2, 'B': 3, 'TB': 4
})
```

```
# Encodage numérique des niveaux d'étude
VueDf['niveau_etude_num'] = VueDf['niveau_etude'].map({
    "Terminale": 1, "Année préparatoire aux études supérieures": 2, "1ère année d'études supérieures": 3, "2nd année d'études supérieures": 4
})
```

III - Représentations graphiques

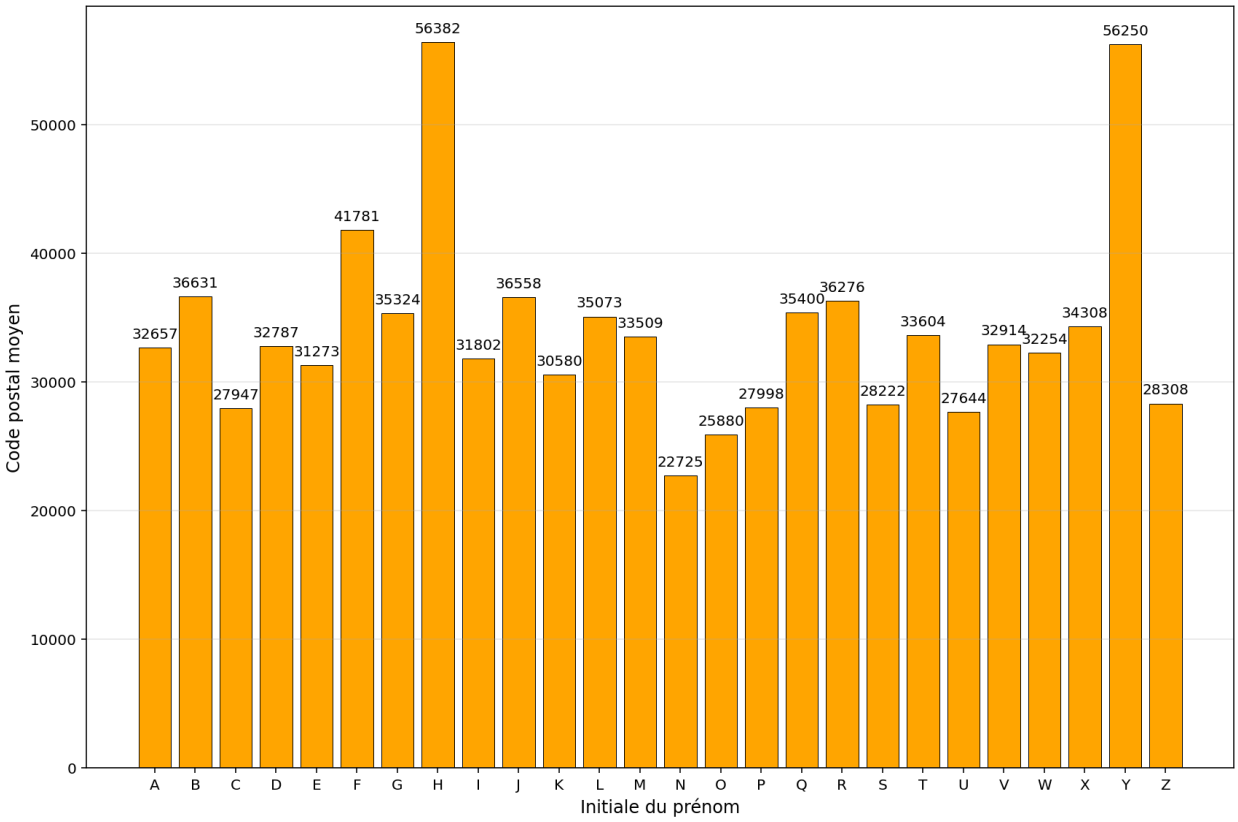
Pour les diagrammes suivants, on a choisi d'utiliser des diagrammes en bâtons car on avait seulement le choix entre diagramme en bâtons et diagramme en moustaches, et donc on a fait la moyenne de chaque variable en fonction de l'initiale du prénom. Seulement, dans la plupart des cas, ce n'est absolument pas adapté de faire un diagramme en bâtons. Il aurait été plus efficace de faire un diagramme de points (avec scatterplot) ou autre, mais ce n'est pas autorisé dans les consignes.

- Diagramme en bâtons : Moyenne générale en fonction de l'initiale du prénom



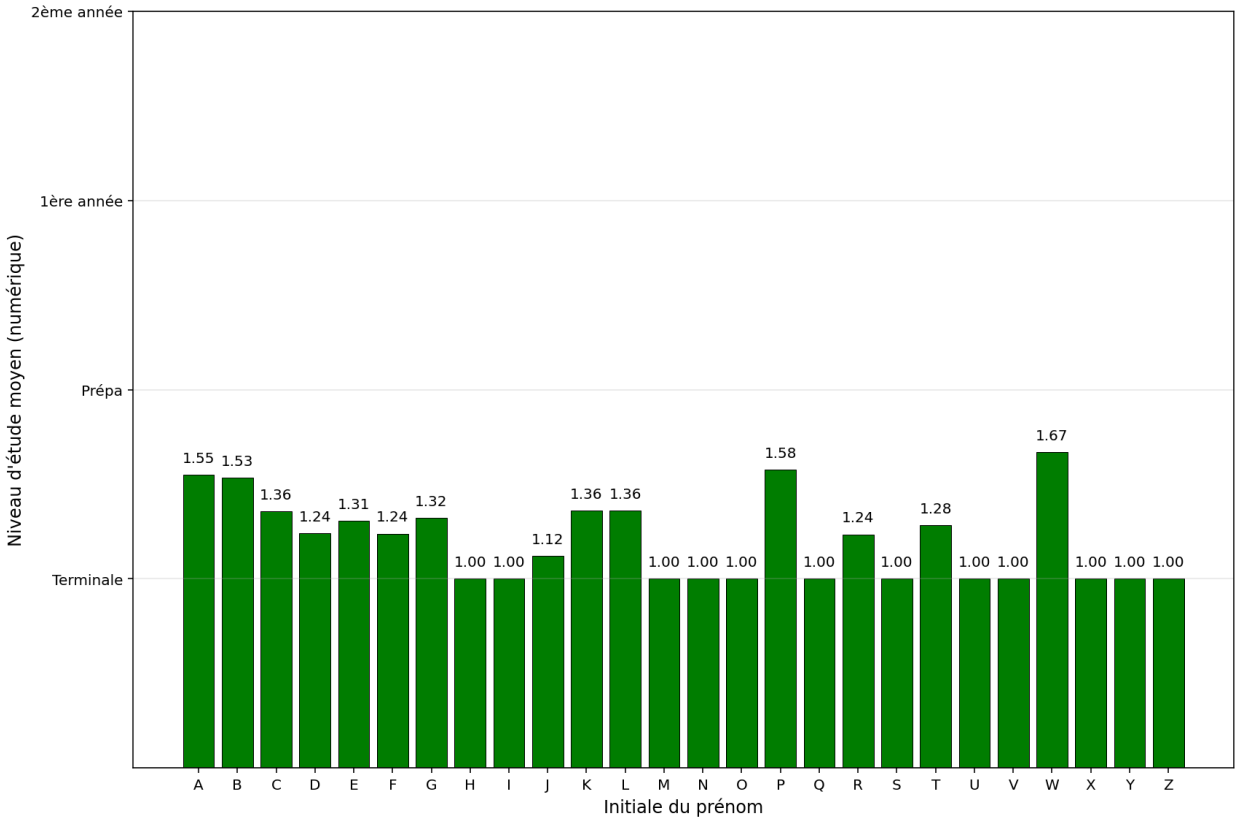
On remarque que les notes moyennes sont très homogènes. Seuls les prénoms avec O comme initiale sont bien au-dessus du reste. Les prénoms commencent par un Y sont aussi en dessous.

- Diagramme en bâtons : Code postal moyen en fonction de l'initiale du prénom



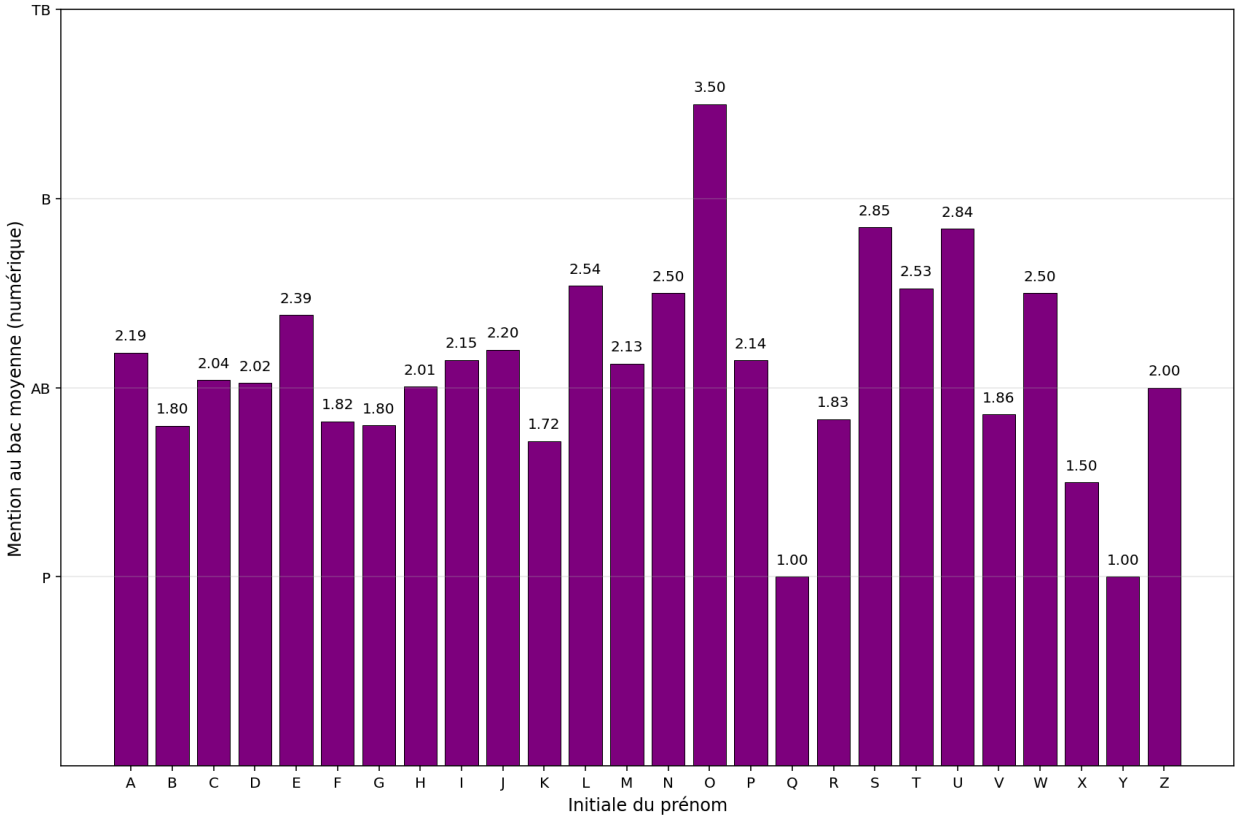
On remarque qu'il y a deux valeurs bien au-dessus des autres. La représentation en diagramme en bâtons n'est pas adaptée car on est obligé de faire la moyenne de codes postaux, ce qui est peu utile pour se donner une représentation des codes postaux en fonction des initiales du prénom. Le diagramme en point aurait été beaucoup plus efficace pour ce cas.

- Diagramme en bâtons : Niveau d'étude moyen en fonction de l'initiale du prénom



Les valeurs cette fois-ci sont très homogènes. Certains niveaux d'étude ne sont pas atteints alors que certaines personnes ont ce niveau d'étude. C'est tout simplement parce qu'on fait une moyenne du niveau d'étude, ce qui nous permet de représenter les données en diagramme en bâtons. Mais le diagramme en point aurait été beaucoup plus efficace pour ce cas.

- Diagramme en bâtons : Mention au bac en fonction de l'initiale du prénom



On remarque qu'il y a une valeur bien au-dessus (lettre O) et deux valeurs en dessous du reste (lettres Q et Y). Encore une fois, le diagramme en point aurait été beaucoup plus efficace pour ce cas.

IV - Matrice des coefficients de corrélation

1) Démarche

Dans cette partie, on calcule la matrice des corrélations afin de mesurer les relations linéaires entre nos différentes variables numériques.

```
# Sélection des variables numériques pertinentes pour le calcul de la corrélation
df_corr = VueDf[['initiale_num', 'code_postal', 'moyenne', 'mention_bac_num', 'niveau_etude_num']]
# Calcul de la matrice de corrélation de Pearson entre ces variables
corr_matrix = df_corr.corr()
```

2) Matrice des corrélations

On obtient la matrice suivante :

Indice	initiale_num	code_postal	moyenne	ıention_bac_nuı	veau_etude_nu
initiale_num	1	-0.0418204	0.0154403	0.0978615	-0.0997443
code_postal	-0.0418204	1	-0.0611063	-0.130334	0.00292531
moyenne	0.0154403	-0.0611063	1	0.182407	0.05776
mention_bac_num	0.0978615	-0.130334	0.182407	1	-0.00256795
niveau_etude_num	-0.0997443	0.00292531	0.05776	-0.00256795	1

Les corrélations qui nous intéressent sont celles de *initiale_num* avec le reste de variables. On remarque donc que pour :

- *code_postal* : Corrélation très faible et négative (pratiquement aucun lien)
- *moyenne* : Corrélation très proche de 0 donc très faible positive (encore moin de lien)
- *mention_bac_num* : Corrélation faible et positive (un petit lien)
- *niveau_etude_num* : Corrélation faible et négative (un petit lien)

V - Régression linéaire multiple

1) Utilisation de la Régression linéaire multiple : comment ?

Dans cette analyse, nous choisissons comme variable endogène la variable *initiale_num*, c'est-à-dire la première lettre du prénom des étudiants, convertie en nombre selon l'ordre alphabétique (A = 1, ..., Z = 26).
Ce choix est atypique car l'initiale d'un prénom n'est généralement pas influencée par des caractéristiques telles que la moyenne ou le niveau d'étude.
Les variables explicatives choisies sont :

- *moyenne* (la moyenne générale de l'étudiant),
- *mention_bac_num* (mention obtenue au bac, de 1 à 4),
- *niveau_etude_num* (niveau d'étude, de 1 à 4),
- *code_postal* (code postal de provenance).

2) Variables explicatives les plus pertinentes

Les variables explicatives sont toutes numériques car précédemment normalisées. Elles couvrent à la fois des aspects académiques (*moyenne*, *mention_bac_num*, *niveau_etude_num*) et géographiques (*code_postal*). Ces données sont disponibles pour tous les étudiant de notre vue, ce qui rend leur utilisation pratique pour modéliser une relation, même hypothétique.

3) Lien avec la problématique

La problématique étant :

Existe-t-il une influence de la première lettre du prénom sur le niveau d'études, la mention obtenue au bac, le code postal et la moyenne générale des étudiants ?

Cette problématique est donc discutable car il n'y a pas de corrélation entre les variables comme vue avant. Cette démarche est volontairement absurde.

4) Régression Linéaire Multiple en Python

On fait maintenant la régression linéaire multiple avec Python. Dans un premier temps il nous faut les fonctions "*coefficients_regression_lineaire(X, y)*", qui nous ont été fournis en TP.

```
def coefficients_regression_lineaire(X, y):
    """
    Calcule les coefficients de l'hyperplan pour une régression linéaire multiple.
    X : ndarray de shape (n, m)
    y : ndarray de shape (n, 1) ou (n,)
    Retourne : theta (ndarray de shape (m+1,) avec b à l'indice 0)
    """
```

```
    n_samples = X.shape[0]
    X_aug = np.hstack((np.ones((n_samples, 1)), X))
    theta = np.linalg.inv(X_aug.T @ X_aug) @ X_aug.T @ y
    return theta.flatten()
```

Par la suit, on peut faire la Régression Linéaire Multiple :

```
# Variables explicatives : moyenne, mention au bac, niveau d'étude, code postal
X_Ar = VueDf[['moyenne', 'mention_bac_num', 'niveau_etude_num',
'code_postal']].to_numpy()
# Variable cible : initiale du prénom
y_Ar = VueDf['initiale_num'].to_numpy()

# Calcul des coefficients
theta = coefficients_regression_lineaire(X_Ar, y_Ar)

print("Coefficients :", theta)
```

5) Paramètres obtenus et interprétation détaillée

L'exécution du code de régression nous donne les coefficients suivants :

- a_0 (ordonnée à l'origine) : 10.661
- a_1 (coefficient de la moyenne) : 0.0048
- a_2 (coefficient de mention_bac_num) : 0.739
- a_3 (coefficient de niveau_etude_num) : -0.936
- a_4 (coefficient de code_postal) : -0.0000144

Modèle mathématique obtenu :
Le modèle s'écrit sous la forme : *initiale_num* = a_0 + a_1 × *moyenne* + a_2 × *mention_bac_num* + a_3 × *niveau_etude_num* + a_4 × *code_postal*

Interprétation des coefficients :

- a_0 (ordonnée à l'origine) : La valeur 10.661 représente la valeur prédite de l'initiale numérique lorsque toutes les variables explicatives sont nulles. Cette valeur correspond approximativement à la lettre "K" (11ème lettre), ce qui n'a aucun sens pratique dans notre contexte.
- a_1 (coefficient de *moyenne*) : Le coefficient 0.0048 est extrêmement faible, indiquant qu'une augmentation de 1 point de moyenne "prédirait" une augmentation de seulement 0.0048 dans l'ordre alphabétique de l'initiale. Cette valeur proche de zéro confirme l'absence de relation entre moyenne et initiale du prénom.
- a_2 (coefficient de *mention_bac_num*) : Le coefficient 0.739 est le plus élevé en valeur absolue. Il suggère que passer d'une mention à la suivante (ex: AB → B) "prédit" une augmentation de 0.739 dans l'initiale numérique. Cette relation est causalement impossible.

- a_3 (coefficient de *niveau_etude_num*) : Le coefficient -0.936 suggérerait qu'avancer d'un niveau d'étude "prédit" une diminution de l'initiale numérique. Cette relation négative n'a aucun fondement logique.
- a_4 (coefficient de *code_postal*) : Le coefficient -0.0000144 est négligeable, confirmant l'absence totale de relation entre code postal et initiale du prénom.

Analyse critique des résultats :
Tous ces coefficients, bien qu'ils puissent être calculés mathématiquement, n'ont aucun sens causal réel. Ils représentent des corrélations purement fortuites dans notre échantillon, confirmant ainsi que :

- La régression peut toujours produire des coefficients, même en l'absence de relation logique
- L'existence de coefficients non nuls ≠ existence d'une relation causale
- Les très faibles corrélations observées dans la matrice (partie IV) se retrouvent dans les coefficients obtenus (a_1 et a_4 quasi-nuls)

6) Coefficient de corrélation multiple, interprétation

Avec la fonction de sklearn (importation de LinearRegression du module sklearn) :

```
modele_sk = LinearRegression()
modele_sk.fit(X_Ar, y_Ar)
r2_sklearn = modele_sk.score(X_Ar, y_Ar)

print("R2 avec sklearn :", r2_sklearn)
```

Le coefficient de corrélation multiple avec la formule vue dans le Cours 1 des Statistiques pour la SAE

```
def coefficient_correlation_multiple_cours(X, y, theta):
    N = X.shape[0]  # Nombre d'individus
    # Matrice augmentée avec colonne de 1
    X_aug = np.hstack((np.ones((N, 1)), X))
    y_pred = X_aug @ theta
    # Calcul selon la formule du cours
    numerateur = np.sum((y_pred - y) ** 2)
    var_y = np.var(y, ddof=1)
    denominateur = N * var_y
    resu = 1 - (numerateur / denominateur)
    return resu
# Calcul et affichage du résultat
resu = coefficient_correlation_multiple_cours(X_Ar, y_Ar, theta)

print("Coefficient de corrélation multiple (R2) :", resu)
```

VI - Conclusion

1) Réponse à la problématique

La problématique que nous avons étudiée était la suivante :

Existe-t-il une influence de la première lettre du prénom sur le niveau d'études, la mention obtenue au bac, le code postal et la moyenne générale des étudiants ?

D'après ce que nous avons vu dans le rapport, il n'y a aucune influence entre la première lettre du prénom et les variables. Cela est prévisible car on a fait exprès de prendre une problématique absurde.

2) Argumentation à partir des résultats de la régression linéaire

D'après nos résultat on peut voir qu'il n'y a aucun liens :

- **Coefficient de la moyenne (a_1 = 0,0048)** : Valeur très faible donc même si on augmente de 1 point de moyenne aucune conséquence dans l'ordre alphabétique de l'initiale.
- **Coefficient de mention_bac_num (a_2 = 0,739)** : Notre coefficient le plus élevé, mais reste faible et donc n'a aucune conséquence.
- **Coefficient de niveau_etude_num (a_3 = -0,936)** : Vu que c'est une valeur négative on y prête pas attention.
- **Coefficient de code_postal (a_4 = -0,0000144)** : Valeur très proche de 0 donc aucun lien géographique avec l'initiale du prénom.

Le coefficient de corrélation multiple (R^2) est égale à 0,0203, même pas 2,03%. C'est très faible, cela s'explique par la différence entre les initiales des prénoms et nos variables.97.97% de la variation de l'initiale du prénom ne peut pas être expliquée par nos variables.

Maintenant quand on regarde les liens entre la première lettre du prénom et les autres variables, on remarque que les liens sont très faibles voir même inexistants. Donc, rien n'est lié à la première lettre du prénom.

3) Interprétations personnelles

Interprétation sérieuse :

Ce projet, nous a permis de bien savoir définir, encadrés et produire une bonne problématique, car au début nous étions partis sur autre chose mais après quelque essai on s'est rendu compte que c'était pas optimal. Même si les calculs donnent des résultats, ça ne veut pas dire qu'ils ont du sens. On peut avoir un modèle qui

“fonctionne” en apparence, mais qui ne renvoie rien dans la réalité.
Notre travail montre que :

- Avoir des chiffres ≠ avoir une vraie relation entre les choses
- Un calcul correct ≠ une idée correcte
- Faut réfléchir à “Est ce que c’est bon/bien ?”

Maintenant on a appris à faire attention à ne pas croire n’importe quoi juste parce qu’il y a des nombres ou des graphes.

Interprétation absurde :

fait le fichier python car jsp si on a la normal et la matriciel
Si on voulait s’amuser à inventer des conclusions farfelues, on pourrait dire que :

- Les élèves dont le prénom commence par un **"O"** sont naturellement plus forts, sûrement parce que la lettre est bien ronde, ce qui aide à penser en rond, donc plus efficacement
- Les personnes qui vivent dans le code postal 22300 sont influencées par une force mystérieuse qui pousse leurs parents à choisir des prénoms qui commencent par le début de l’alphabet.
- Avoir une mention Très Bien au bac donnerait à vos parents, dans le passé, l’envie de vous appeler avec un prénom du milieu de l’alphabet. Oui, on parle bien d’un voyage dans le temps alphabétique.
- Les prénoms qui commencent par un "Y" sont peut-être maudits parce qu’ils arrivent à la fin de l’alphabet. Et hop, une malédiction de plus !

Bien sûr, toutes ces idées sont complètement absurdes. Elles montrent qu’on peut faire dire n’importe quoi aux chiffres, si on ne réfléchit pas un peu à ce qu’ils veulent vraiment dire.