MTH8302 - Devoir 2 : Régression Linéaire et Analyse des Résidus

March 9, 2025

Introduction

Ce devoir a pour objectif de vous familiariser avec les concepts fondamentaux de la régression linéaire, des tests d'hypothèses et de l'analyse des résidus. Vous travaillerez avec le dataset **Wage**, qui contient des informations sur les salaires en fonction de différentes caractéristiques sociodémographiques.
Vous devez répondre aux questions en justifiant vos réponses avec des analyses statistiques et des graphiques lorsque nécessaire.

1 Chargement et Exploration des Données (5 points)

1.1 Exploration du dataset Wage *(3 points)*

Questions & Instructions :

1. Téléchargez le fichier `Wage.csv` depuis Moodle et importez-le dans Python ***(1 point)***.
2. Affichez les 5 premières lignes du dataset. (Reportez le résultat affiché en le collant) ***(1 point)***.
3. Listez les colonnes du dataset ***(1 point)***.

Complétez le code ci-dessous en remplaçant les commentaires par les instructions appropriées afin de charger le dataset et pouvoir l'explorer.

```
import pandas as pd  # Importation de la bibliothèque pandas

# 1. Charger les données depuis un fichier CSV
file_path = "_________"  # Complétez avec l'emplacement du fichier
data = pd._________(file_path)  # Complétez avec la méthode appropriée

# 2. Afficher les premières lignes du dataset
_________  # Complétez avec la commande pour afficher les 5 premières lignes

# 3. Afficher la liste des colonnes
_________  # Complétez avec la commande pour afficher les noms des colonnes
```

1.2 Description des Colonnes *(2 points)*

Question : Donnez le nom en anglais et la description en français des attributs de chaque colonne du dataset. Vous pouvez vous référer à la description complète du dataset disponible dans la librairie ISLP.

2 Régression Linéaire Simple (45 points)

Vous allez ajuster un modèle de régression linéaire simple pour prédire le salaire (`wage`) en fonction de l'âge (`age`).

2.1 Expression du modèle de régression linéaire simple et explication de ses composantes (5 points)

Question : Posez le Le modèle de régression linéaire simple que vous allez utiliser pour ce problème et décrivez ses composantes ***(5 points)***.

2.2 Équations mathématiques des coefficients (5 points)

Question : Exprimez les formules mathématiques pour obtenir les estimateurs des coefficients du modèle ***(5 points)***.

2.3 Implémentation manuelle des coefficients (5 points)

Question : Vous allez maintenant calculer ces coefficients sans utiliser de librairie de régression. Reportez le résultat affiché ***(5 points)***.

Instructions : Complétez le code ci-dessous en remplaçant les commentaires par les instructions appropriées afin d'implémenter manuellement un modèle de régression linéaire simple pour prédire le salaire (`wage`) en fonction de l'âge (`age`).

```
import numpy as np

# 1. Extraire les variables indépendantes et dépendantes
X = data["_____"].values  # Complétez avec la colonne contenant l'âge
Y = data["_____"].values  # Complétez avec la colonne contenant le salaire

# 2. Calcul des moyennes de X et Y avec numpy
X_mean = ________ # Complétez avec l'opération adéquate
Y_mean = ________ # Complétez avec l'opération adéquate

# 3. Calcul du coefficient beta_1 (pente)
Sxy = ________ # Complétez avec l'opération adéquate
Sxx = ________  # Complétez avec l'opération adéquate
beta_1 = ________ # Complétez avec l'opération adéquate

# 4. Calcul du coefficient beta_0 (intercept)
beta_0 = _______  # Complétez avec l'opération adéquate

# 5. Affichage des coefficients
```

```
print(f"beta_0 (intercept) = {beta_0}")
print(f"beta_1 (pente) = {beta_1}")
```

2.4 Comparaison avec les bibliothèques `statsmodels` et `scikit-learn` (5 points)

Question : Utilisez maintenant les librairies `statsmodels` ***(2.5 points)*** et `scikit-learn` ***(2.5 points)*** pour comparer les résultats.

Instructions : Complétez le code ci-dessous en remplaçant les commentaires par les instructions appropriées afin d'ajuster un modèle de régression linéaire simple en utilisant la bibliothèque `statsmodels`. Reportez le résultat affiché.

```
import statsmodels.api as sm

# 1. Ajout de la constante pour inclure l'intercept dans le modèle
X_with_const = ________  # Utilisez la fonction de statsmodels qui ajoute une constante

# 2. Ajustement du modèle de régression
model = ________  # Complétez avec la classe et la méthode d'ajustement de statmodels

# 3. Affichage du résumé des résultats avec statmodels
print(model.________)  # Complétez avec l'instruction pour afficher le résumé du modèle
```

Instructions : Complétez le code ci-dessous en remplaçant les commentaires par les instructions appropriées afin d'ajuster un modèle de régression linéaire simple en utilisant la bibliothèque `scikit-learn`. Reportez le résultat affiché.

```
from sklearn.linear_model import LinearRegression

# 1. Création du modèle de régression linéaire
model_sklearn = ________()  # Complétez avec la classe à utiliser

# 2. Transformation de X en une matrice colonne (obligatoire pour scikit-learn)
X_reshaped = X.________(-1, 1)  # Complétez avec la méthode de transformation

# 3. Entraînement du modèle avec les données
model_sklearn.______(X_reshaped, Y)  # Complétez avec la méthode d'ajustement

# 4. Affichage des coefficients
print(f"Coefficient beta_1 avec scikit-learn : {model_sklearn.coef_[0]:.4f}")
print(f"Intercept beta_0 avec scikit-learn : {model_sklearn.intercept_:.4f}")
```

2.5 Visualisation de la relation entre `wage` et `age` (5 points)

Question : Représentez graphiquement la relation entre `wage` et `age` avec la droite de régression linéaire ajustée ***(5 points)***.

Instructions : Complétez le code ci-dessous en remplaçant les commentaires par les instructions appropriées afin de tracer :

- Un nuage de points représentant les salaires (`wage`) en fonction de l'âge (`age`).
- Une droite de régression obtenue avec le modèle ajusté.

```
import matplotlib.pyplot as plt

# 1. Tracer le nuage de points des données
# Complétez avec la ou les opérations adéquates en utilisant matplotlib
plt.scatter(_____, _____, alpha=0.5, label="Données réelles")
# 2. Générer la droite de régression
age_range = np.linspace(min(X), max(X), 100)  # Générer un intervalle d'âges
predicted_wage = _____ + _____ * age_range  # Complétez avec ce qui est adéquat

# 3. Tracer la droite de régression
# Complétez avec la ou les opérations adéquates en utilisant matplotlib
plt.plot(_____, _____, color='red', label="Régression linéaire")

# 4. Ajouter des labels et un titre
plt.xlabel("Âge")
plt.ylabel("Salaire horaire")
plt.title("Régression linéaire : Salaire en fonction de l'âge")
plt.legend()
plt.show()
```

2.6 Test t pour la significativité du coefficient de l'âge (10 points)

Question : Établissez un test t pour vérifier si l'attribut `age` est significatif dans la régression linéaire simple ***(10 points)***.

1. ***(5 points)*** Décrivez les étapes du test t permettant d'évaluer la significativité du coefficient associé à l'attribut `age`, en définissant l'hypothèse nulle et alternative, ainsi que les étapes du test statistique. Assurer-vous de vérifiez la significativité du coefficient associé à `age` en utilisant deux approches différentes :

   - **Méthode 1 :** Comparaison avec les valeurs critiques. Dans votre description des étapes du test t, expliquez la procédure de calcul des valeurs critiques et l'interprétation du résultat obtenu.
   - **Méthode 2 :** Calcul et interprétation de la p-valeur. Dans votre description des étapes du test t, détaillez le processus de calcul de la p-valeur et discutez de son interprétation.

2. ***(5 points)*** Complétez le code ci-dessous en remplaçant les commentaires par les instructions appropriées afin d'établir le test t pour vérifier si l'attribut `age` est significatif. Exécutez le code, reportez les résultats obtenus, et comparez la statistique t ainsi que la p-valeur aux valeurs affichées dans le résumé du modèle que vous avez déjà généré avec `statsmodels`.

```python
import numpy as np
import scipy.stats as stats

# Prédictions du modèle
Y_pred = ________  # Utilisez beta_0, beta_1, et X pour obtenir Y_pred

# Calcul de l'erreur quadratique résiduelle (SCres)
SCres = ________  # Complétez avec le calcul de SCres

# Nombre d'observations
n = len(X)

# Calcul de l'erreur standard de beta_1
SE_beta1 = ________  # Complétez avec la formule de l'erreur standard

# Calcul de la statistique t
t_statistic = ________

# Définition des degrés de liberté
dl = ________

# Calcul des valeurs critiques
t_critique = ________

# Calcul de la p-valeur
p_value = ________

# Affichage des résultats
print(f"Statistique t = {t_statistic:.4f}")
print(f"Valeur critique t = ±{t_critique:.4f}")
print(f"p-valeur = {p_value:.4f}")

# Vérification de la significativité en comparant la statistique t
# avec les valeurs critiques
--------
# remplir ci-haut le if else statement en qui donne un affichage conditionné sur
# la comparaison de la statistique t et les valeurs critiques.

# Vérification de la significativité en comparant la p-valeur
# avec un niveau de signification de 0.05
--------
# remplir ci-haut le if else statement en qui donne un affichage conditionné sur
# la comparaison de la p-valeur avec un niveau de signification de 0.05.
```

2.7 Calcul du coefficient de détermination R^2 et interprétation *(5 points)*

Questions :

1. ***(1 point)*** Donner la formule du coefficient R^2 qui permet de mesurer la proportion de l'ajustement du modèle linéaire sur les données observées .

2. ***(2 points)*** Complétez le code ci-dessous en remplaçant les commentaires par les instructions appropriées afin de calculer R^2 en fonction de SC_{reg} et SC_{totale}. Exécutez le code et reportez le résultat obtenu. Comparez le résultat obtenu avec le coefficient R^2 déjà reproté par le modèle implémenté avec la librairie `statmodels` et qui est affiché dans son résumé.

```
import numpy as np

# 1. Calcul de la somme des carrés totale (SC_totale)
Y_mean = ________  # Complétez avec la moyenne de Y
SC_totale = ________  # Complétez avec la somme des carrés totale

# 2. Calcul de la somme des carrés expliquée par la régression (SC_reg)
SC_reg = ________  # Complétez avec la somme des carrés expliquée par le modèle

# 3. Calcul du coefficient de détermination R^2
R2 = ________  # Complétez avec la formule de R^2

# Affichage du résultat
print(f"Coefficient de détermination R^2 = {R2:.4f}")
```

3. ***(2 points)*** Interprétez le résultat obtenu pour le coefficient de détermination R^2. Que signifie sa valeur en termes d'ajustement du modèle aux données ?

2.8 Analyse des Résidus et Interprétation des Résultats (10 points)

L'analyse des résidus permet d'évaluer la qualité d'un modèle de régression linéaire et de vérifier si ses hypothèses sont respectées.

Questions:

1. ***(5 points)*** Complétez le code ci-dessous et exécutez-le pour générer les graphiques des résidus :

```
# Fonction pour tracer les graphiques de résidus
def plot_residuals(X, Y, Y_pred, title):
    residuals = ________

    fig, axes = plt.subplots(1, 3, figsize=(18, 5))

    # Histogramme des résidus
    axes[0].hist(________, bins=20, edgecolor='black', alpha=0.7)
    axes[0].set_title("Histogramme des résidus")
```

```
    axes[0].set_xlabel("Résidus")
    axes[0].set_ylabel("Fréquence")

    # QQ-Plot des résidus
    stats.probplot(________, dist="norm", plot=axes[1])
    axes[1].set_title("QQ-Plot des résidus")

    # Graphique des résidus
    axes[2].scatter(________, ________, alpha=0.5)
    axes[2].axhline(0, color='red', linestyle='dashed', linewidth=1)
    axes[2].set_title("Résidus en fonction de X")
    axes[2].set_xlabel("X")
    axes[2].set_ylabel("Résidus")

    plt.suptitle(title)
    plt.show()

plot_residuals(________, ________, ________,
               "Analyse des résidus : prédiction des salaires")
```

2. ***(5 points)*** Affichez les graphiques obtenus et interprétez vos résultats en répondant aux questions suivantes :

   (a) ***(2 points)*** Analyser les résultats déduits de l'histogramme des résidus, du QQ-plot et graphique des résidus en fonction de l'âge.

   (b) ***(1 point)*** En combinant ces observations avec R^2 et le graphique de visualisation de la relation entre `wage` et `age`, que pouvez-vous conclure sur la qualité du modèle de régression et ses limites?

   (c) ***(2 points)*** Que suggérez-vous pour améliorer la qualité de la régression? Proposez des pistes d'amélioration et justifiez votre réponse.

3 Partie 2 : Régression Linéaire Multiple (50 points)

Objectif : Dans cette partie, nous allons améliorer notre modèle en ajoutant des variables explicatives supplémentaires pour mieux prédire le salaire.

3.1 Expression du modèle de régression linéaire multiple (5 points)

Questions :

1. Écrivez le modèle de régression linéaire multiple sous sa forme scalaire ***(2 points)***.

2. Reformulez ce modèle en notation matricielle, où la variable cible, les coefficients et les prédicteurs sont représentés sous forme de vecteurs et matrices ***(2 points)***.

3. Donner la formule de l'estimateur $\hat{\boldsymbol{\beta}}$ regroupant les estimateurs des coefficients du modèle obtenus par la méthode d'estimation des moindres carrés ordinaires (MCO) ***(1 point)***.

3.2 Ajout de variables explicatives (5 points)

Question : Ajustez un modèle de régression linéaire multiple en incluant toutes les variables explicatives disponibles dans le dataset, à l'exception de `wage` et `logwage`. Implémentez ce modèle de manière manuelle, en utilisant la formule matricielle des moindres carrés ordinaires (MCO) ***(5 points)*** .

Instructions : Complétez le code ci-dessous en remplaçant les commentaires par les instructions appropriées afin de calculer les coefficients du modèle de régression linéaire multiple sans utiliser de bibliothèque de régression. Exécutez le code et reportez les coefficients obtenus.

```
import numpy as np
import pandas as pd

# Charger les données
file_path = "_________"
data = pd._________(file_path)

# Sélectionner toutes les variables explicatives sauf 'wage' et 'logwage'
X = data.drop(columns=["_________", "_________"])

# Convertir les variables catégoriques en variables numériques (one-hot encoding)
X = pd.get_dummies(X, drop_first=True)
# Convertir toutes les valeurs en float (nécessaire pour l'inversion de matrice)
X = X.astype(float)

# Extraire la variable cible
Y = data["_________"].values

# Ajouter une colonne de 1 pour le terme d'interception (beta_0)
X_with_1 = _________

# Calcul des coefficients de régression selon la formule des MCO
beta = _________

# Affichage des coefficients
print("Coefficients estimés du modèle :")
print(beta)
```

3.3 Comparaison avec les modèles de `statsmodels` et `scikit-learn` (5 points)

Question : Comparez les coefficients obtenus par votre implémentation manuelle avec ceux des modèles de régression linéaire multiple en utilisant les bibliothèques `statsmodels` ***(2.5 points)*** et `scikit-learn` ***(2.5 points)*** .

Instructions : Complétez les codes ci-dessous en remplaçant les commentaires par les instructions appropriées afin d'ajuster un modèle de régression linéaire multiple avec `statsmodels` et `scikit-learn`. Reportez les coefficients obtenus et comparez-les avec ceux obtenus manuellement.

Régression linéaire multiple avec `statsmodels`

```
import statsmodels.api as sm

# Ajouter une constante pour inclure l'intercept dans le modèle
X_with_const = ________  # Complétez avec la fonction qui ajoute une constante

# Ajustement du modèle de régression linéaire multiple
model = ________  # Complétez avec la classe et la méthode d'ajustement de statsmodels

# Affichage du résumé des résultats
print(model.________)  # Complétez avec l'instruction pour afficher le résumé du modèle
```

Régression linéaire multiple avec `scikit-learn`

```
from sklearn.linear_model import LinearRegression

# 1. Création du modèle de régression linéaire multiple
model_sklearn = ________()  # Complétez avec la classe à utiliser

# 2. Entraînement du modèle avec les données
model_sklearn.______(X, Y)  # Complétez avec la méthode d'ajustement

# 3. Affichage des coefficients
print("Coefficients avec scikit-learn :")
print(f"Intercept : {model_sklearn.intercept_:.4f}")
print(f"Coefficients : {model_sklearn.coef_}")
```

3.4 Analyse de la Variance (ANOVA) pour la Régression Multiple (20 points)

L'ANOVA pour la régression multiple permet d'évaluer si l'ensemble des variables explicatives améliore significativement la prédiction de Y.
Question : Implémentez une fonction qui calcule et affiche le tableau ANOVA pour un modèle de régression multiple. Vous devez compléter les formules manquantes.
Instructions :

1. ***(5 points)*** Donnez les formules mathématiques des quantités à calculer pour le tableau ANOVA dans le contexte de la régression linéaire multiple ainsi que la formule de la p-valeur ajoutée à la dernière colonne du tableau.
2. ***(5 points)*** Complétez le code ci-dessous en remplaçant les commentaires par les formules correctes.
3. ***(5 points)*** Affichez le tableau ANOVA sous forme de DataFrame.
4. ***(5 points)*** Analyser les résultats obtenus en répondant aux questions suivantes :
   (a) ***(1 point)*** Comparez les tableaux obtenus pour pour le modèle utilisant une implémentation manuelle des coefficients et celui obtenu en utilisant la librairie `scikit-learn`

(b) ***(1 point)*** Quelle est la signification de la statistique F_{stat} ?

(c) ***(1 point)*** Que nous indique la p-valeur associée ?

(d) ***(2 points)*** Que concluez-vous sur l'ensemble des variables explicatives utilisées dans la régression ?

Code à Compléter :

```
import numpy as np
import pandas as pd
from scipy.stats import f

def compute_anova(Y, Y_pred, n, p):
    Y_mean = np.mean(Y)

    # 1. Calcul des sommes des carrés
    SC_total = ________  # Complétez avec la formule de SC_total
    SC_reg = ________  # Complétez avec la formule de SC_reg
    SC_res = ________  # Complétez avec la formule de SC_res

    # 2. Calcul des degrés de liberté
    df_reg = ________  # Complétez avec la formule de df_reg
    df_res = ________  # Complétez avec la formule de df_res

    # 3. Calcul des moyennes des carrés (MC)
    MC_reg = ________  # Complétez avec la formule de MC_reg
    MC_res = ________  # Complétez avec la formule de MC_res

    # 4. Calcul du F-statistic
    F_stat = ________  # Complétez avec la formule de F_stat

    # 5. Calcul de la p-valeur associée
    p_value = ________  # Complétez avec la formule de p_value (loi de Fisher)

    # 6. Création du tableau ANOVA sous forme de DataFrame
    anova_table = pd.DataFrame({
        "Source": ["Régression", "Résiduel", "Total"],
        "Somme des Carrés (SC)": [________, ________, ________],
        "Degrés de Liberté (dl)": [________, ________, ________],
        "Moyenne des Carrés (MC)": [________, ________, ________],
        "F_stat": [________, ________, ________],
        "p-value": [________, ________, ________]
    })

    return anova_table

# 7. Obtention des Y_pred selon la librarie utilisée
Y_pred_mco = ________
```

```
Y_pred_sklearn = ________

# Appels de la fonction `compute_anova` pour la modèles MCO et sklearn
anova_mco = compute_anova(Y, Y_pred, n=len(Y), p=X.shape[1])
anova_sklearn = compute_anova(Y, Y_pred_sklearn, X.shape[0], X.shape[1])

# Affichage des tableaux ANOVA en utilisant `tabulate`
print("\nTableau ANOVA avec la formule des Moindres Carrés Ordinaires (MCO)")
print(tabulate(anova_mco, headers="keys", tablefmt="grid", showindex=False,
               floatfmt=".4f"))
print("\nTableau ANOVA avec `LinearRegression` de sklearn")
print(tabulate(anova_sklearn, headers="keys", tablefmt="grid", showindex=False,
               floatfmt=".4f"))
```

3.5 Analyse des coefficients individuels avec `statsmodels` (5 points)

L'analyse des coefficients individuels permet de mieux comprendre l'impact de chaque variable explicative sur la variable cible Y. En examinant les p-valeurs individuelles des coefficients de régression ainsi que les statistiques de test F_{stat}, nous pouvons identifier :

- Les variables ayant un effet significatif sur Y.
- Les variables qui n'ont pas d'influence statistiquement significative.

Exécutez le code fourni ci-dessous et affichez le tableau ANOVA contenant les statistiques individuelles des coefficients.

Questions :

1. Analysez et interprétez les résultats en répondant aux questions suivantes :
   (a) ***(1 point)*** Que pouvez-vous conclure de la valeur du coefficient de détermination R^2?
   (b) ***(2 points)*** Quelles sont les variables ayant un impact significatif sur Y? Justifiez votre réponse.
   (c) ***(2 points)*** Quelles variables semblent ne pas avoir d'influence significative? Expliquez pourquoi.

Code à exécuter :

```
import statsmodels.api as sm
import statsmodels.formula.api as smf
import pandas as pd
import re
from sklearn.metrics import r2_score

# Création d'un DataFrame avec les variables explicatives et la cible
data_df = pd.concat([pd.DataFrame(X, columns=X.columns), pd.Series(Y, name="wage")], axis=1)
```

```python
# Remplacement des caractères spéciaux dans les noms de colonnes
def clean_column_name(col_name):
    return re.sub(r"[^\w]", "_", col_name)  # Remplace tout caractère non-alphanumérique par "_"

data_df.columns = [clean_column_name(col) for col in data_df.columns]

# Construction de la formule en évitant les erreurs liées aux caractères spéciaux
formula = "wage ~ " + " + ".join(data_df.columns.drop("wage"))

# Ajustement du modèle avec la notation par formule
model = smf.ols(formula, data=data_df).fit()

# Calcul du tableau ANOVA avec Statsmodels
anova_sm = sm.stats.anova_lm(model, typ=1)

print("\nTableau ANOVA avec `OLS` de statsmodels")
print(tabulate(anova_sm, headers="keys", tablefmt="grid", floatfmt=".4f"))

# Calcul et affichage des coefficients et p-valeurs
print("\nCoefficients et p-valeurs du modèle :")
print(model.summary())

r2_mco = r2_score(Y, Y_pred_mco)
print("\nR2 obtenu par chaque méthode :")
print("Statsmodels R2 :", model.rsquared)
print("Sklearn R2 :", r2_score(Y, model_sklearn.predict(X)))
print("Moindres Carrés R2 :", r2_mco)
```

3.6 Analyse des Résidus et Interprétation des Résultats (10 points)

L'analyse des résidus permet d'évaluer la qualité d'un modèle de régression linéaire et de vérifier si ses hypothèses sont respectées.

Questions:

1. ***(5 points)*** Complétez le code ci-dessous et exécutez-le pour générer les graphiques des résidus :

```python
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats

# Fonction pour tracer les graphiques de résidus
def plot_residuals(X, Y, Y_pred, feature_name, title):
    residuals = ________

    fig, axes = plt.subplots(1, 3, figsize=(18, 5))
```

```
    # Histogramme des résidus
    axes[0].hist(________, bins=20, edgecolor='black', alpha=0.7)
    axes[0].set_title("Histogramme des résidus")
    axes[0].set_xlabel("Résidus")
    axes[0].set_ylabel("Fréquence")

    # QQ-Plot des résidus
    stats.probplot(________, dist="norm", plot=axes[1])
    axes[1].set_title("QQ-Plot des résidus")

    # Graphique des résidus en fonction de Advanced_Degree
    if feature_name in X.columns:
        feature_values = X[feature_name]
    else:
        raise ValueError(f"La variable '{feature_name}' n'existe pas dans X.")

    axes[2].scatter(________, ________, alpha=0.5)
    axes[2].axhline(0, color='red', linestyle='dashed', linewidth=1)
    axes[2].set_title(f"Résidus en fonction de {feature_name}")
    axes[2].set_xlabel(feature_name)
    axes[2].set_ylabel("Résidus")

    plt.suptitle(title)
    plt.show()

# Appel de la fonction avec Advanced_Degree comme variable explicative
plot_residuals(________, ________, ________, "age",
               "Analyse des résidus : prédiction des salaires")
```

2. ***(5 points)*** Affichez les graphiques obtenus et interprétez vos résultats en répondant aux questions suivantes :

   (a) ***(2 points)*** Analyser les résultats déduits de l'histogramme des résidus, du QQ-plot et graphique des résidus en fonction de l'âge. Y a t-il une différence significative entre les graphiques obtenus ici et ceux déjà obtenus dans le contexte de la régression linéaire simple?

   (b) ***(2 points)*** En combinant ces observations avec R^2, que pouvez-vous conclure sur la qualité du modèle de régression et ses limites?

   (c) ***(1 point)*** Que suggérez-vous pour améliorer la qualité de la régression? Proposez des pistes d'amélioration et justifiez votre réponse.