Projet de régression linéaire

Application de la régression linéaire simple et multiple sur une base de données ozone et détection des valeurs aberrantes

Nom complet : Soufiane Oukessou

Filière : Data Science DS

Objectif du projet

L’objectif de ce projet est de modéliser les pics de concentration d’ozone en fonction de variables météorologiques telles que la température et la nébulosité. Pour cela, nous nous appuyons sur un jeu de données comportant 112 observations.

Ce travail s'inscrit dans une démarche d’analyse statistique à travers la régression linéaire, en explorant successivement la régression linéaire simple puis multiple. L’objectif final est d’identifier, à l’aide d’une mise en œuvre pratique en Python, le meilleur modèle de régression linéaire multiple (avec constante) permettant d’expliquer efficacement la variable cible **Y** (le pic d’ozone), à partir des variables explicatives disponibles.

Importation des bibliothèques et de la data

###Importation des bibliothèques

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

1] Python

###Importation des données

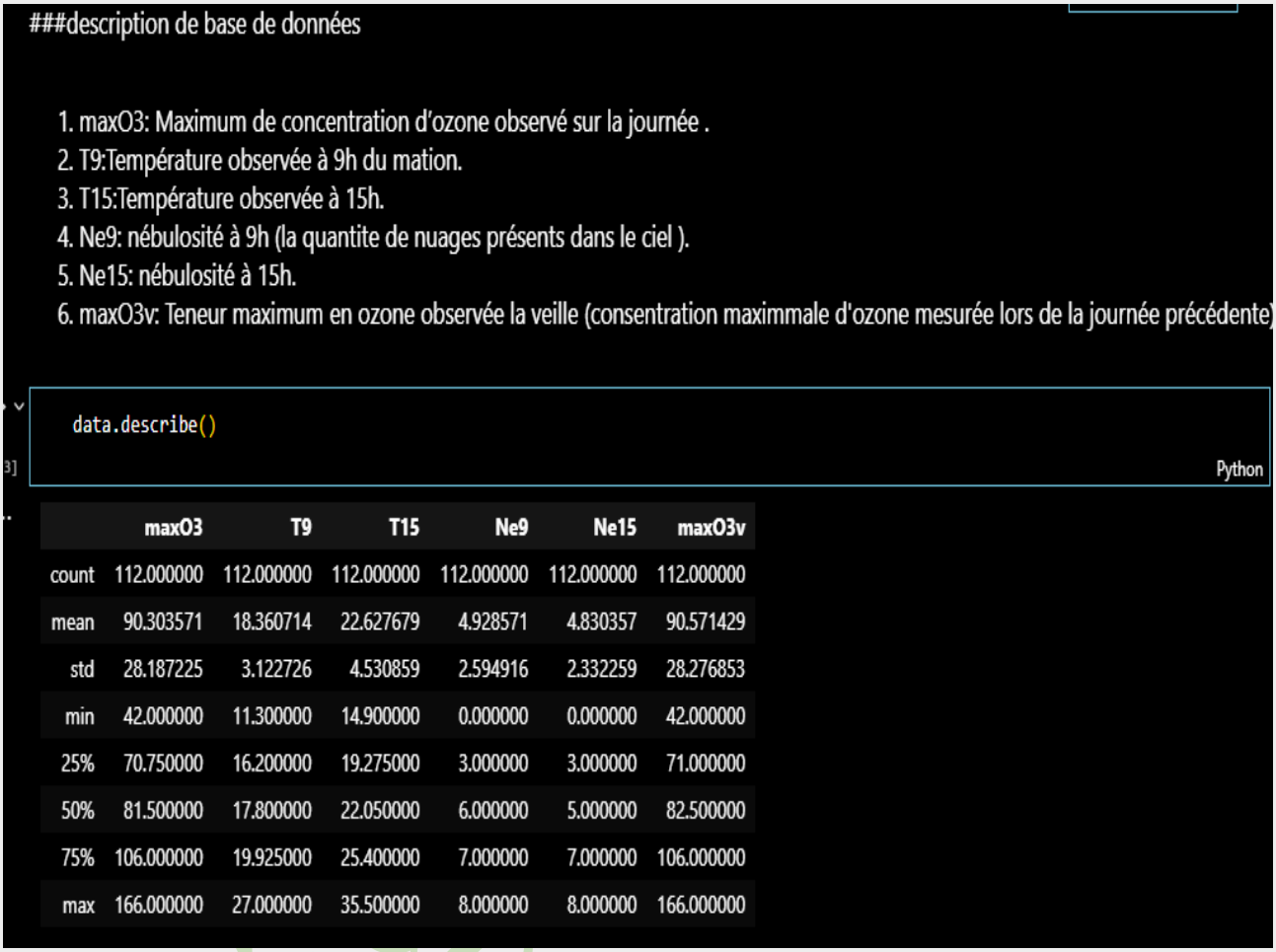
```
data=pd.read_csv('ozone.txt',delimiter='\s+')
data.to_csv('ozone.csv')
data.head()
```

2] Python

	maxO3	T9	T15	Ne9	Ne15	maxO3v
1	87	15.6	18.4	4	8	84
2	82	17.0	17.7	5	7	87
3	92	15.3	19.5	2	4	82
4	114	16.2	22.5	1	0	92
5	94	17.4	20.4	8	7	114

On trouve **pandas**, qui fournit des fonctions pour préparer, analyser et traiter les données ; **numpy**, qui offre des fonctionnalités mathématiques ; et **matplotlib**, qui permet la visualisation des données à l'aide de graphiques. J'importe notre base de données à l'aide de la fonction ***'read_csv'***.

Description de la base de données



###description de base de données

1. maxO3: Maximum de concentration d'ozone observé sur la journée .
2. T9:Température observée à 9h du mation.
3. T15:Température observée à 15h.
4. Ne9: nébulosité à 9h (la quantite de nuages présents dans le ciel).
5. Ne15: nébulosité à 15h.
6. maxO3v: Teneur maximum en ozone observée la veille (consentration maximmale d'ozone mesurée lors de la journée précédente)

```
data.describe()
```

Python

	maxO3	T9	T15	Ne9	Ne15	maxO3v
count	112.000000	112.000000	112.000000	112.000000	112.000000	112.000000
mean	90.303571	18.360714	22.627679	4.928571	4.830357	90.571429
std	28.187225	3.122726	4.530859	2.594916	2.332259	28.276853
min	42.000000	11.300000	14.900000	0.000000	0.000000	42.000000
25%	70.750000	16.200000	19.275000	3.000000	3.000000	71.000000
50%	81.500000	17.800000	22.050000	6.000000	5.000000	82.500000
75%	106.000000	19.925000	25.400000	7.000000	7.000000	106.000000
max	166.000000	27.000000	35.500000	8.000000	8.000000	166.000000

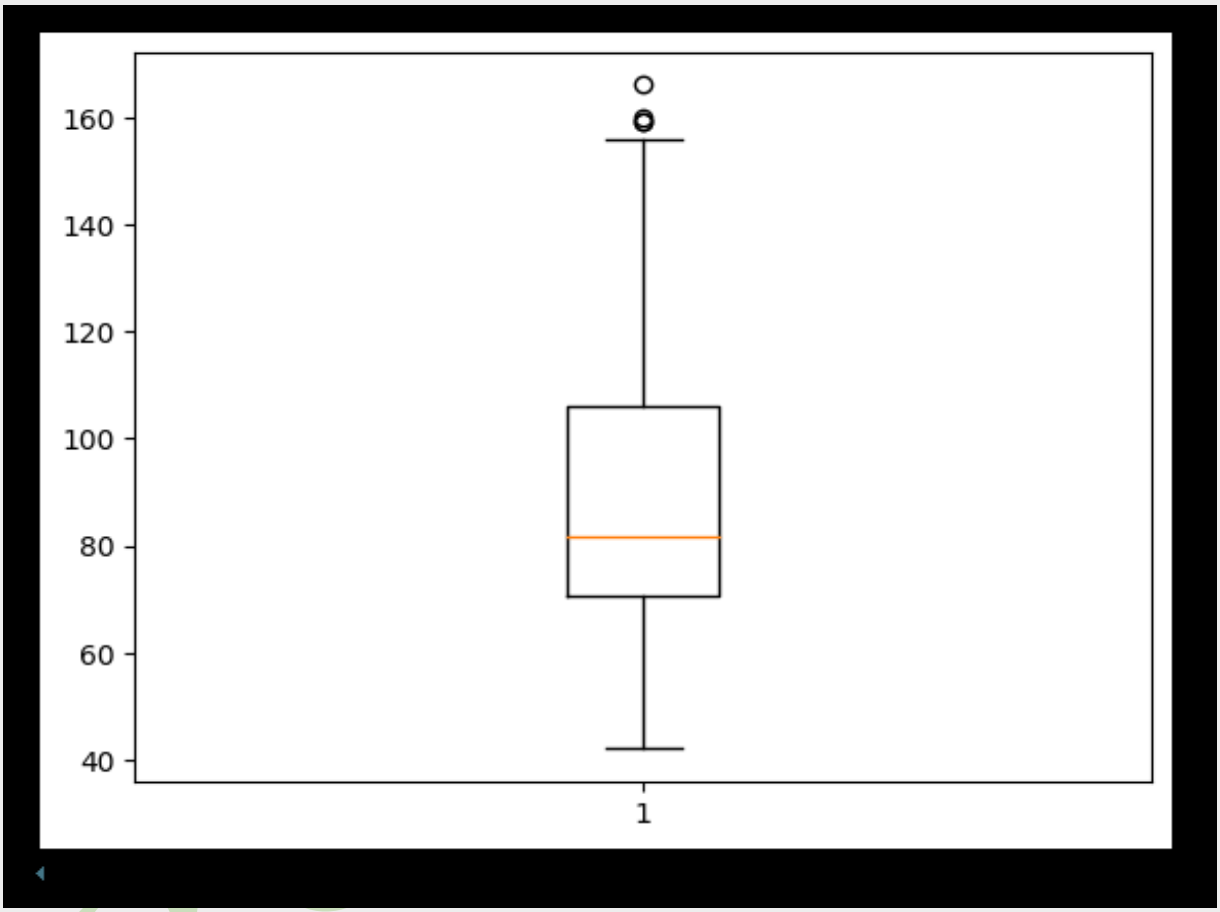
Afin d'avoir une première vue d'ensemble sur le jeu de données, nous avons utilisé la méthode **describe ( )** de pandas. Celle-ci nous fournit des statistiques descriptives pour chaque variable quantitative, telles que la moyenne, l'écart-type, les valeurs extrêmes (minimum et maximum) ainsi que les quartiles. Cette étape permet d'identifier les ordres de grandeur des variables, et de mieux comprendre la distribution générale des données avant d'appliquer des techniques de modélisation.

```
data.isnull().sum()
```

```
maxO3     0
T9        0
T15       0
Ne9       0
Ne15      0
maxO3v    0
dtype: int64
```

La base de données ne contient aucune valeur manquante.

Préparation de la base de données et détection des valeurs aberrantes



Afin d'examiner la distribution des valeurs maximales d'ozone (maxO3), nous avons utilisé un diagramme en boîte. Ce type de visualisation permet de repérer rapidement la dispersion des données, la médiane, ainsi que la présence de valeurs aberrantes. Le boxplot montre que la majorité des observations de maxO3 sont concentrées autour de la médiane, mais on observe également quelques valeurs extrêmes au-delà de la borne supérieure. Ces outliers indiquent la présence de jours avec des pics d'ozone exceptionnellement élevés, ce qui pourrait avoir des implications importantes pour la santé publique et justifie une attention particulière dans la modélisation.

```python
#calculer les quartiles (Q1,Q3)
Q1=data['maxO3'].quantile(0.25)
Q3=data['maxO3'].quantile(0.75)
#calculer interquartile
IQR=Q3-Q1
#determiner la borne sup et la borne inf
born_sup=Q3+1.5*IQR
born_inf=Q1-1.5*IQR
valeurs_aberrantes=data[(data['maxO3']<born_inf)|(data['maxO3']>born_sup)]
valeurs_aberrantes.shape[0]
```

```
4
```

```python
valeurs_aberrantes
```

	maxO3	T9	T15	Ne9	Ne15	maxO3v
54	159	24.0	26.5	2	7	153
56	160	25.0	31.1	0	5	149
79	166	19.8	30.8	4	1	131
80	159	25.0	35.5	1	1	166

Autre méthode pour identifier les valeurs aberrantes dans les pics d'ozone (maxO3), nous avons utilisé la méthode de l'intervalle interquartile (IQR). Après avoir calculé les 1er et 3e quartiles (Q1 et Q3), nous avons déterminé les bornes inférieure et supérieure en appliquant la règle de 1,5 × IQR. Toute valeur située en dehors de cet intervalle a été considérée comme aberrante.

```
data_cleaned=data.drop(valeurs_aberrantes.index)
```

```
data_cleaned.describe()
```

	maxO3	T9	T15	Ne9	Ne15	maxO3v
count	108.000000	108.000000	108.000000	108.000000	108.000000	108.000000
mean	87.685185	18.172222	22.318519	5.046296	4.879630	88.379630
std	25.102648	2.989916	4.268011	2.551834	2.306908	26.226803
min	42.000000	11.300000	14.900000	0.000000	0.000000	42.000000
25%	70.000000	16.175000	19.175000	3.000000	3.000000	70.750000
50%	81.000000	17.700000	21.850000	6.000000	5.000000	81.000000
75%	101.000000	19.750000	24.925000	7.000000	7.000000	101.000000
max	156.000000	27.000000	33.700000	8.000000	8.000000	160.000000

Notre nouvelle base de données contient 108 observations.

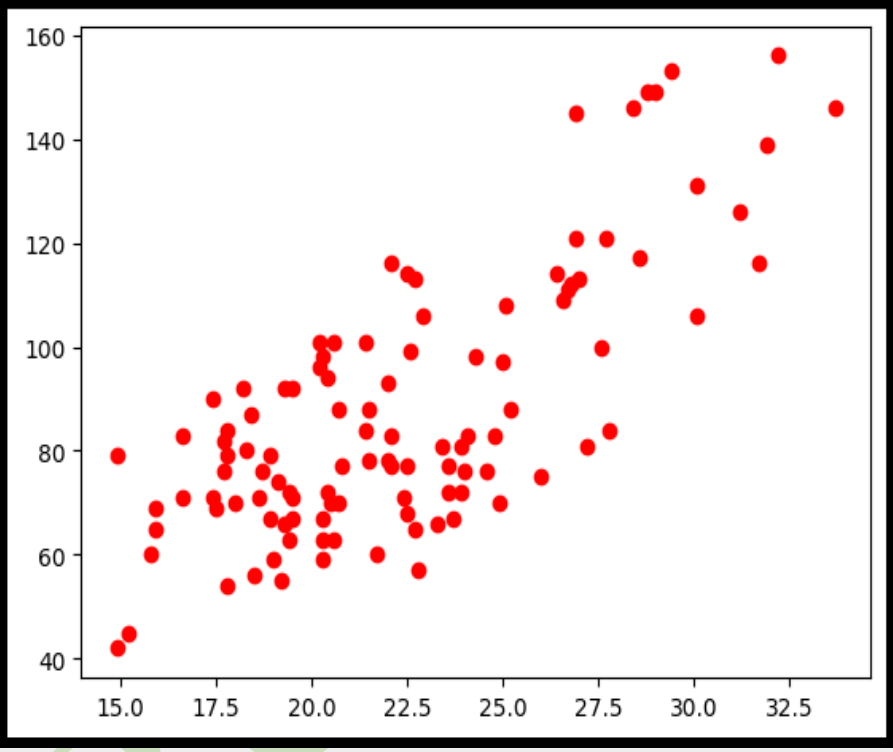
Après avoir identifié les observations aberrantes dans la variable maxO3, nous avons procédé à leur suppression afin d'obtenir un jeu de données plus représentatif. Cette étape de nettoyage est cruciale pour améliorer la qualité des modèles statistiques. Le nouveau jeu de données, nommé data_cleaned, ne contient plus que les observations jugées normales selon la méthode de l'IQR.

Il contient 108 observations et non 112 maintenant.

Régression linéaire simple entre maxO3 et T15 et test de significativité

```
X=data_cleaned['T15']
Y=data_cleaned['maxO3']
plt.scatter(X,Y,color='red',label='nuage de points')
```

```
<matplotlib.collections.PathCollection at 0x7b9c8e74cdf0>
```



Afin d'explorer visuellement la relation entre la température (T15) et la concentration maximale d'ozone (maxO3), nous avons tracé un nuage de points. Ce graphe permet d'observer la relation linéaire entre les deux variables.

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
X=data_cleaned['T15'].values.reshape(-1,1)
Y=data_cleaned['maxO3'].values
model=LinearRegression()
model.fit(X,Y)
b1=model.coef_
b0=model.intercept_
Y_pred=model.predict(X)
r2=r2_score(Y,Y_pred)
```

```
b0
```

```
-10.029375146553647
```

```
b1
```

```
array([4.37818309])
```

```
r2
```

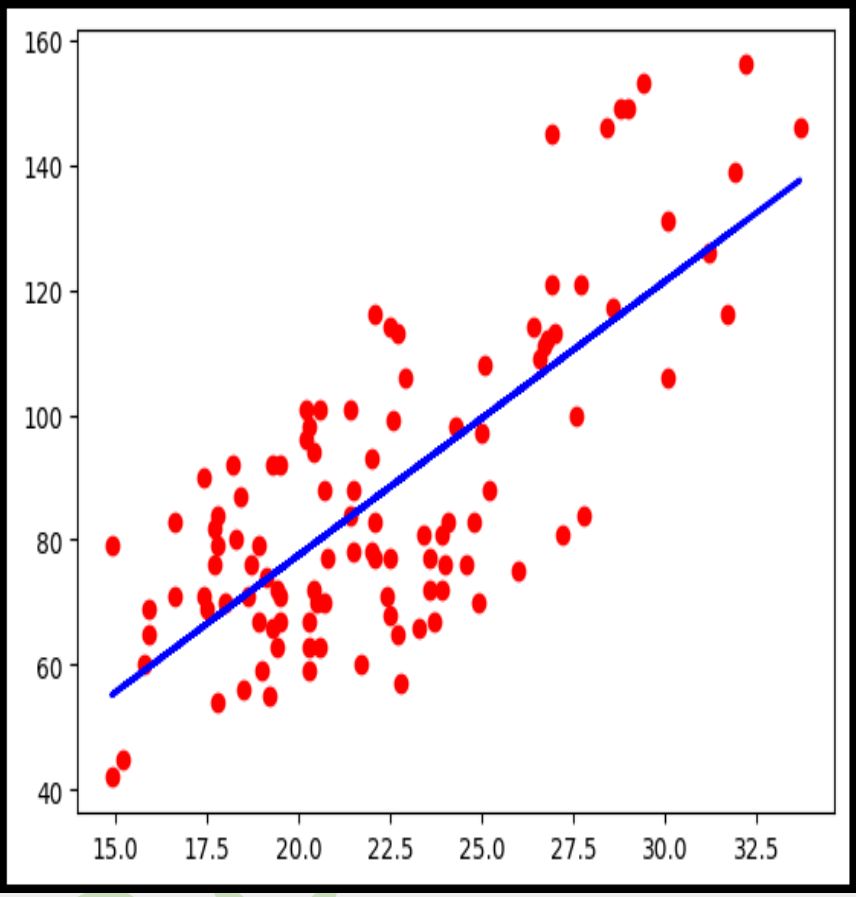
```
0.5541148079353468
```

Une régression linéaire simple a été réalisée afin de modéliser la relation entre la température (T15) et les pics d'ozone (maxO3). Grâce à **Scikit-learn,** on estime **les coefficients** (coefficient directeur et ordonnée à l'origine) et **le coefficient de détermination R au carré**.

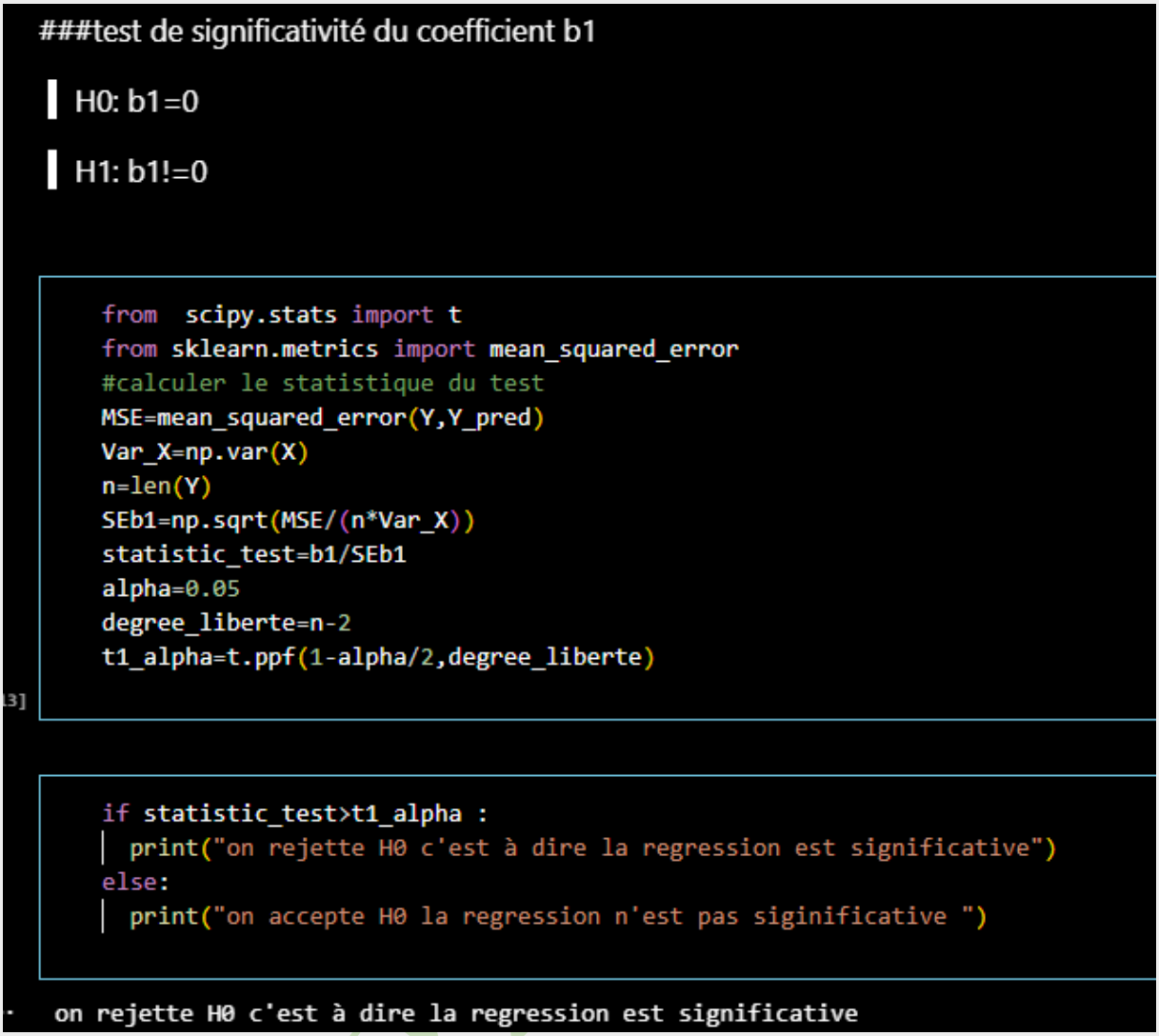
On trouve que notre modèle explique **55%** de la variance totale des données.

```
plt.scatter(X,Y,color='red',label='nuage de points')
plt.plot(X, Y_pred, color='blue', linewidth=2, label='Régression linéaire')
```

```
[<matplotlib.lines.Line2D at 0x7aebfd6b8790>]
```



Le graphe ci-dessus présente à la fois le nuage de points représentant les observations (T15, maxO3) et la droite de régression ajustée.



On réalise **un test de significativité de la régression (test de Fisher)** en comparant la statistique de test avec le quantile d'ordre alpha=5%. On conclut que **la régression est significative**.

Régression linéaire multiple et test de significativité

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
XM=data_cleaned[['T9','T15','Ne9','Ne15','maxO3v']]
YM=data_cleaned['maxO3']
model=LinearRegression()
model.fit(XM,YM)
b=model.coef_
b0=model.intercept_
YM_pred=model.predict(XM)
r2=r2_score(YM,YM_pred)
```

```
b1=b[0]
b2=b[1]
b3=b[2]
b4=b[3]
b5=b[4]
```

```
b0,b1,b2,b3,b4,b5
```

```
(19.09284821458138,
 0.45917281870942833,
 2.0587169708977267,
 -3.1206418711598487,
 0.06644725562102693,
 0.3363227394703032)
```

Une régression linéaire multiple a été réalisée, Ce modèle intègre **cinq variables explicatives**. Le modèle a été entraîné à l'aide de la classe **LinearRegression** de la bibliothèque **scikit-learn**. Une fois le modèle ajusté, nous avons extrait **les coefficients de régression** ainsi que **l'ordonnée à l'origine.**

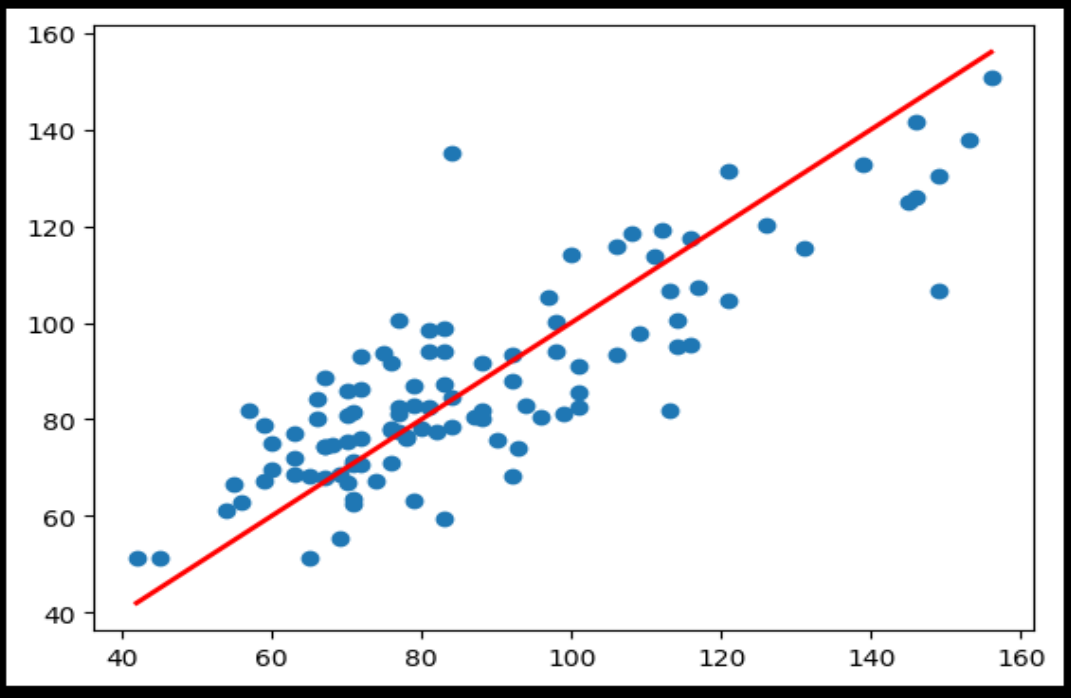
```
r2
```

```
0.7013469498763538
```

Le modèle explique 70% de la variance totale des données.

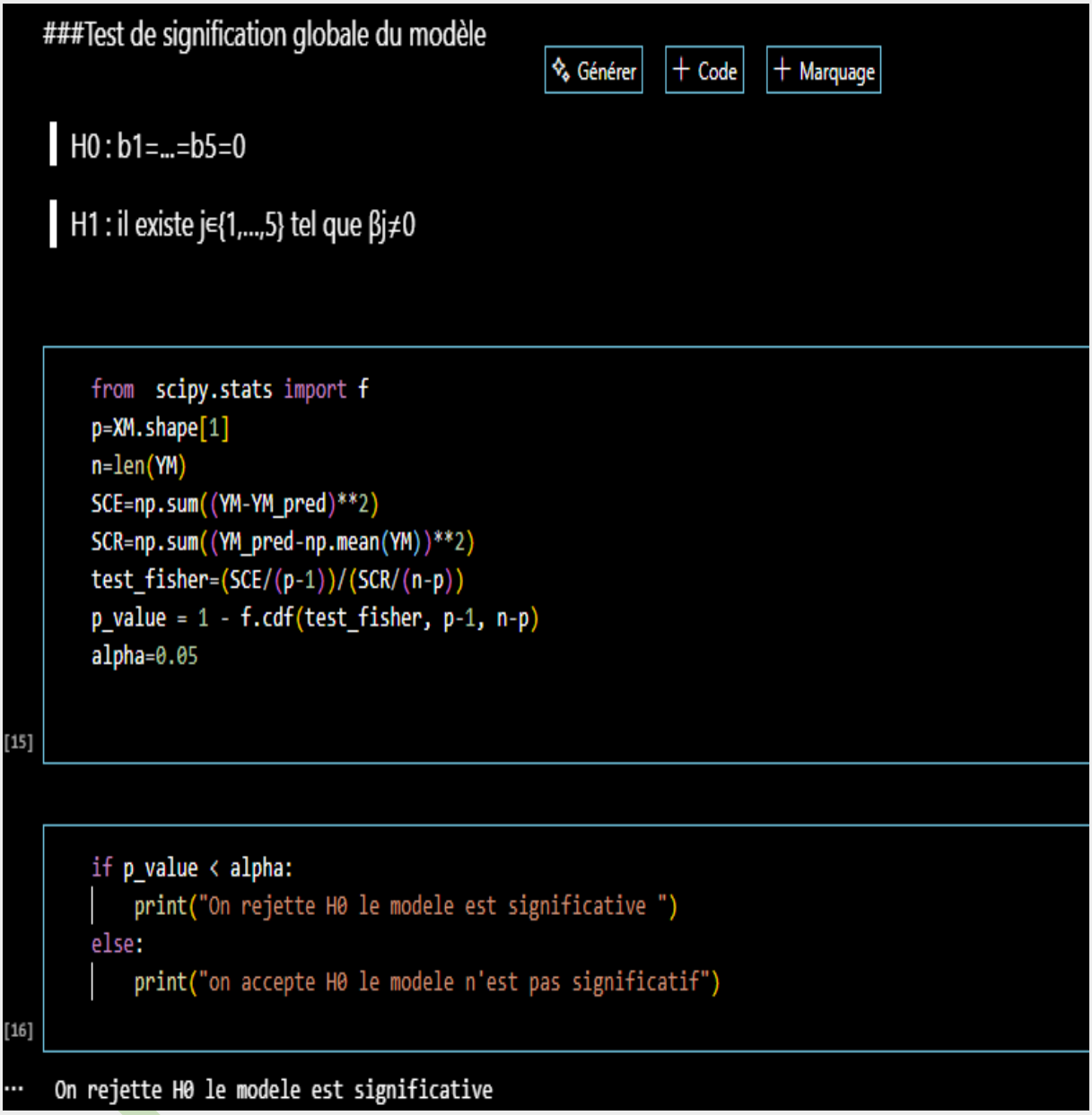
```
plt.scatter(YM,YM_pred)
plt.plot([Y.min(), Y.max()], [Y.min(), Y.max()], color='red', lw=2)
```

```
[<matplotlib.lines.Line2D at 0x7b9c8c5c6110>]
```



Le graphique suivant présente la comparaison entre les valeurs réelles de maxO3 et celles prédites par le modèle de régression multiple. Chaque point correspond à une observation du jeu de données. La ligne rouge représente

la diagonale idéale y=x, où les valeurs prédites seraient égales aux valeurs réelles.



Afin de tester la validité globale du modèle de régression multiple, nous avons réalisé un **test de Fisher**. Ce test évalue si au moins une des variables explicatives a un effet significatif sur la variable à expliquer. On conclut que le modèle est globalement **significatif** et donc pertinent pour expliquer la concentration d'ozone.

Méthode de la Sélection des variables

```
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_classif
X=data_cleaned[['T9','T15','Ne9','Ne15','maxO3v']]
Y=data_cleaned['maxO3']
selector=SelectKBest(score_func=f_classif,k=2)
selector.fit(X,Y)
selected_mask = selector.get_support()
selected_features = X.columns[selected_mask]
print(selected_features)
```

```
Index(['T9', 'T15'], dtype='object')
```

Une étape de sélection de variables a été entreprise à l'aide de la méthode **SelectKBest**, afin d'identifier les deux variables explicatives les plus pertinentes pour prédire **maxO3**. Après application de la méthode, les variables **T9** et **T15** ont été retenues. Un modèle de régression linéaire multiple a ensuite été construit avec ces deux variables et c'est le suivant : **maxO3=b0+b1T9+b2T15**

```python
from sklearn.linear_model import LinearRegression
from scipy.stats import f
X=data_cleaned[['T9','T15']]
Y=data_cleaned['maxO3']
model=LinearRegression()
model.fit(X,Y)
Y_pred=model.predict(X)
p=X.shape[1]
n=len(Y)
SCE=np.sum((Y-Y_pred)**2)
SCR=np.sum((Y_pred-np.mean(Y))**2)
test_fisher=(SCE/(p-1))/(SCR/(n-p))
p_value = 1 - f.cdf(test_fisher, p-1, n-p)
alpha=0.05
if p_value < alpha:
    print("On rejette H0 le modele est significative ")
else:
    print("on accepte H0 le modele n'est pas significatif")
```

```
On rejette H0 le modele est significative
```

Le modèle est évalué via un test de Fisher. Ce test permet de vérifier si le modèle réduit reste globalement significatif.

Donc les variables *'T15'*, *'T9'* expliquent bien la variable cible *'maxO3'*.