

```

#R5.A.L1 - Compléments d'Intelligence Artificielle - Hospital Jules
import pandas as pd
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import StandardScaler
import statsmodels.api as sm

df =
pd.read_excel('Multivariate_Linear_Regression_dataset.xlsx',
engine='openpyxl')
print(df.head())
print("Colonnes du DataFrame:",
df.columns)

y = df['prix']
X = df[['taille_en_pieds_carre',
'nb_chambres']]

scaler = StandardScaler()
X_scaled =
scaler.fit_transform(X)

print("Données normalisées
:")
print(X_scaled)

X_scaled_with_const = sm.add_constant(X_scaled)
model_with_const
= sm.OLS(y, X_scaled_with_const)
results_with_const =
model_with_const.fit()
print(results_with_const.summary())

a0 =
results_with_const.params[0]
a1 = results_with_const.params[1]
a2 =
results_with_const.params[2]

def predict_price_of_house(taille_maison, nb_chambres):
    taille_maison_scaled = (taille_maison - X['taille_en_pieds_carre'].mean()) /
X['taille_en_pieds_carre'].std()
    nb_chambres_scaled = (nb_chambres -
X['nb_chambres'].mean()) / X['nb_chambres'].std()

    return a0 + a1
* taille_maison_scaled + a2 * nb_chambres_scaled

def predict_all(lst_sizes,
lst_nb_chambres):
    predicted_prices = []
    for n in range(len(lst_sizes)):

predicted_prices.append(predict_price_of_house(lst_sizes[n], lst_nb_chambres[n]))
    return
predicted_prices

fig = plt.figure()
ax = fig.add_subplot(111,
projection='3d')

ax.scatter(X['taille_en_pieds_carre'],
X['nb_chambres'], y, c='r',
marker='o')
ax.plot_trisurf(X['taille_en_pieds_carre'],
X['nb_chambres'], predict_all(X['taille_en_pieds_carre'],
X['nb_chambres']), color='blue', alpha=0.5)

ax.set_xlabel('Taille
en pieds carrés')
ax.set_ylabel('Nombre de
chambres')

```

```
chambres')
ax.set_zlabel('Prix')

plt.show()

print(predicted_price =
predict_price_of_house(4500, 5))
```

