**Dữ liệu**

Dữ liệu 1:

* File csv : AllstarFull.csv
* Mô tả dữ liệu:
* Cấu trúc dữ liệu:

Dữ liệu bao gồm các cột chính:

* playerID: Mã định danh của cầu thủ.
* yearID: Năm diễn ra trận đấu.
* gameNum: Số thứ tự của trận đấu trong mùa giải.
* gameID: Mã định danh của trận đấu.
* teamID: Mã định danh cho đội mà cầu thủ tham gia.
* lgID: Mã định danh cho giải đấu hoặc liên đoàn.
* GP (Games Played): Số trận đã tham gia.
* startingPos: Vị trí xuất phát của cầu thủ trong trận đấu.
* Ứng dụng trong phân tích dữ liệu:
* Phân tích hiệu suất cầu thủ.
* So sánh giữa các cầu thủ.
* Phân tích theo thời gian.
* Phân tích vị trí.
* Kết luận:
* Dữ liệu này rất phong phú và có thể cung cấp nhiều thông tin quan trọng cho việc phân tích hiệu suất của cầu thủ trong một giải đấu thể thao. Nó giúp xác định các yếu tố ảnh hưởng đến kết quả và có thể hỗ trợ việc ra quyết định cho các đội bóng hoặc các nhà phân tích thể thao.
* Mô tả và diễn giải quá trình HQTT:
* Thư viện và đọc dữ liệu:

import pandas as pd

import seaborn as sns

import numpy as np

import matplotlib.pyplot as plt

import statsmodels.api as sm

from  sklearn.linear\_model import LinearRegression

# Đọc dữ liệu của file csv

data = pd.read\_csv('AllstarFull.csv')

print(data)

* HQTT các giả định :

# HQTT

# Lọc dữ liệu không có giá trị NaN

df = data[['startingPos', 'GP']].dropna()

# GD1

plt.scatter(data['startingPos'], data['GP'], color='b')

plt.title('Quan hệ tuyến tính giữa Starting Position và GP')

plt.xlabel('Starting Position')

plt.ylabel('GP')

plt.show()

# GD2

x = sm.add\_constant(df[['startingPos']])

y = df['GP']

model = sm.OLS(y, x).fit()

# Phần dư

residuals = model.resid

# Vẽ biểu đồ phân phối phần dư

sns.histplot(residuals, kde=True, color='b')

plt.title('Phân phối của phần dư (Residuals)')

plt.xlabel('Residuals')

plt.ylabel('Frequency')

plt.show()

# GD3

x1 = df['startingPos']

x2 = df['GP']

y = df['GP']

sns.scatterplot(x = x1, y = y, color = 'b', label = 'X1')

sns.scatterplot(x = x2, y = y, color = 'g', label = 'X2')

plt.xlabel('X')

plt.ylabel('Y')

plt.title('Biểu đồ phân tán của X1 (Starting Position) và X2 (GP) với Y (GP)')

plt.legend()

plt.show()

# GD4

x2 = df['startingPos']

y2 = df['GP']

x2 = sm.add\_constant(x2)

model = sm.OLS(y2, x2).fit()

residuals = model.resid

residuals\_variance = np.var(residuals)

sns.scatterplot(x=df['startingPos'], y=residuals, color='b')

plt.xlabel('Starting Position')

plt.ylabel('Residuals')

plt.title('Phân tán của phần dư (Residuals) theo Vị trí xuất phát (Starting Position)')

plt.axhline(0, color='red', linestyle='--')

plt.show()

print('Phương sai của phần dư:', residuals\_variance)

#hdtt

X = df['startingPos'].values.reshape(-1, 1)

y = df['GP'].values

model = LinearRegression()

model.fit(X, y)

GP\_pred = model.predict(X)

coef = model.coef\_[0]

intercept = model.intercept\_

plt.scatter(X, y, color='blue', label='Dữ liệu thực tế')

plt.plot(X, GP\_pred, color='red', label='Đường hồi quy')

plt.xlabel('Starting Position')

plt.ylabel('GP')

plt.title('Mối quan hệ giữa Vị trí xuất phát và Số trận đấu (GP)')

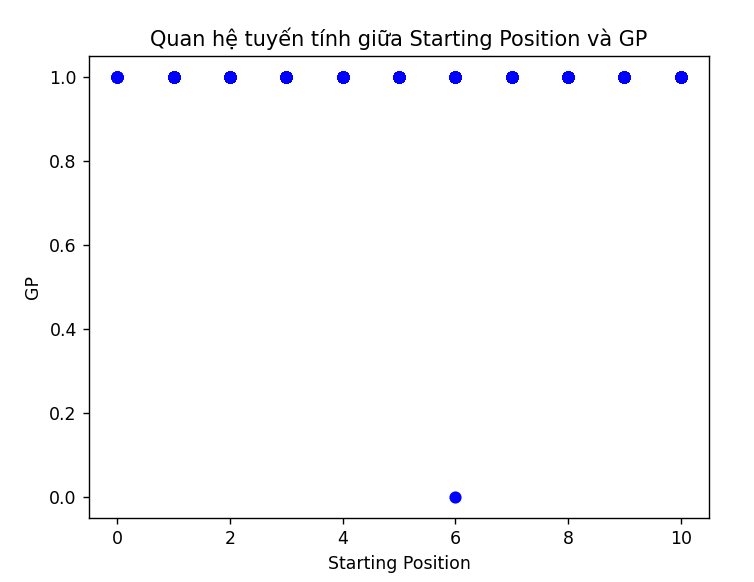
plt.legend()

plt.show()

print('Hệ số:', coef)

print('Hằng số:', intercept)

* Hình ảnh trực quan:



A graph with lines and numbers

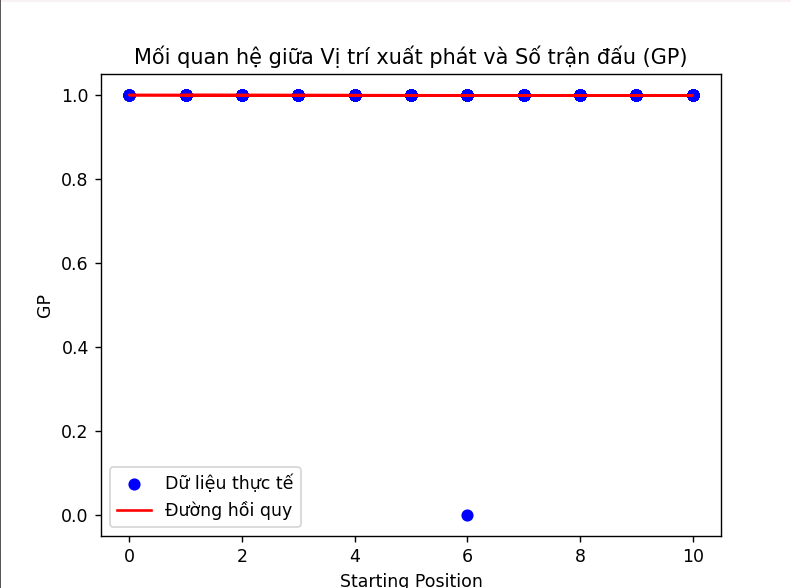
Description automatically generated with medium confidence

A graph with numbers and points

Description automatically generated

A graph with a line

Description automatically generated



A black background with white numbers

Description automatically generated

Dữ liệu 2:

* File csv: basketball\_abbrev.csv
* Mô tả dữ liệu:
* Cấu trúc dữ liệu:

Dữ liệu bao gồm 3 cột chính:

* abbrev\_type: Loại sự kiện hoặc kết quả.
* code : Mã của các loại sự kiện hoặc kết quả.
* full\_name : Tên đầy đủ của các dữ liệu trong cột code.
* Ứng dụng trong phân tích dữ liệu:
  + Phân loại sự kiện.
  + Phân tích kết quả.
  + So sánh giữa các nhóm.
  + Theo dõi lịch sử.
  + Trực quan hóa dữ liệu.
* Mô tả và diễn giải quá trình tiền xử lý dữ liệu:
  + Thư viện và đọc dữ liệu

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import numpy as np

import statsmodels.api as sm

from sklearn.linear\_model import LinearRegression

# Đọc dữ liệu của file csv

data = pd.read\_csv('basketball\_abbrev.csv')

print(data)

* + HQTT các giả định:

#HQTT

df = data[['abbrev\_type', 'code']].dropna()

# Mã hóa các loại vòng đấu và mã code

df['abbrev\_type\_code'] = df['abbrev\_type'].astype('category').cat.codes

df['code\_code'] = df['code'].astype('category').cat.codes

# GD1: Kiểm tra tính tuyến tính

plt.scatter(df['abbrev\_type\_code'], df['code\_code'], color='b')

plt.title('Quan hệ tuyến tính giữa Mã loại vòng đấu và Mã code')

plt.xlabel('Mã loại vòng đấu')

plt.ylabel('Mã code')

plt.show()

# GD2: Mô hình hồi quy tuyến tính

x = sm.add\_constant(df[['abbrev\_type\_code']])

y = df['code\_code']

model = sm.OLS(y, x).fit()

# Phần dư

residuals = model.resid

# Vẽ biểu đồ phân phối phần dư

sns.histplot(residuals, kde=True, color='b')

plt.title('Phân phối của phần dư (Residuals)')

plt.xlabel('Residuals')

plt.ylabel('Frequency')

plt.axvline(0, color='red', linestyle='--')  # Đường tại 0

plt.show()

# GD3: Vẽ biểu đồ phân tán

x1 = df['abbrev\_type\_code']

x2 = df['code\_code']

sns.scatterplot(x=x1, y=y, color='b', label='X1: Mã loại vòng đấu')

sns.scatterplot(x=x2, y=y, color='g', label='X2: Mã code')

plt.xlabel('X')

plt.ylabel('Y')

plt.title('Biểu đồ phân tán của X1 và X2 với Y')

plt.legend()

plt.show()

# GD4: Kiểm tra phương sai đồng nhất

x2 = sm.add\_constant(df['abbrev\_type\_code'])

model = sm.OLS(y, x2).fit()

residuals = model.resid

residuals\_variance = np.var(residuals)

sns.scatterplot(x=df['abbrev\_type\_code'], y=residuals, color='b')

plt.xlabel('Mã loại vòng đấu')

plt.ylabel('Residuals')

plt.title('Phân tán của phần dư theo Mã loại vòng đấu')

plt.axhline(0, color='red', linestyle='--')  # Đường ngang tại 0

plt.show()

print('Phương sai của phần dư:', residuals\_variance)

# Hồi quy tuyến tính với scikit-learn

X = df['abbrev\_type\_code'].values.reshape(-1, 1)  # Đổi thành định dạng 2D

y = df['code\_code'].values

model = LinearRegression()

model.fit(X, y)

code\_pred = model.predict(X)

coef = model.coef\_[0]

intercept = model.intercept\_

plt.scatter(X, y, color='blue', label='Dữ liệu thực tế')

plt.plot(X, code\_pred, color='red', label='Đường hồi quy')

plt.xlabel('Mã loại vòng đấu')

plt.ylabel('Mã code')

plt.title('Mối quan hệ giữa Mã loại vòng đấu và Mã code')

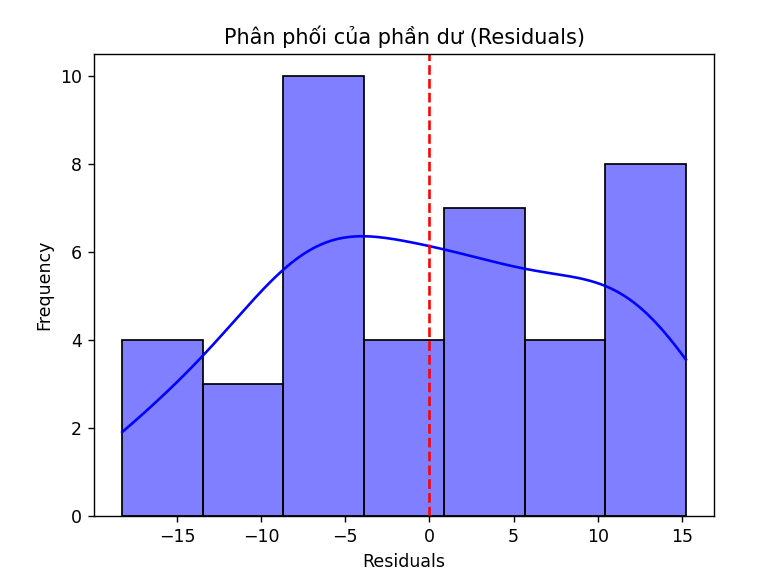
plt.legend()

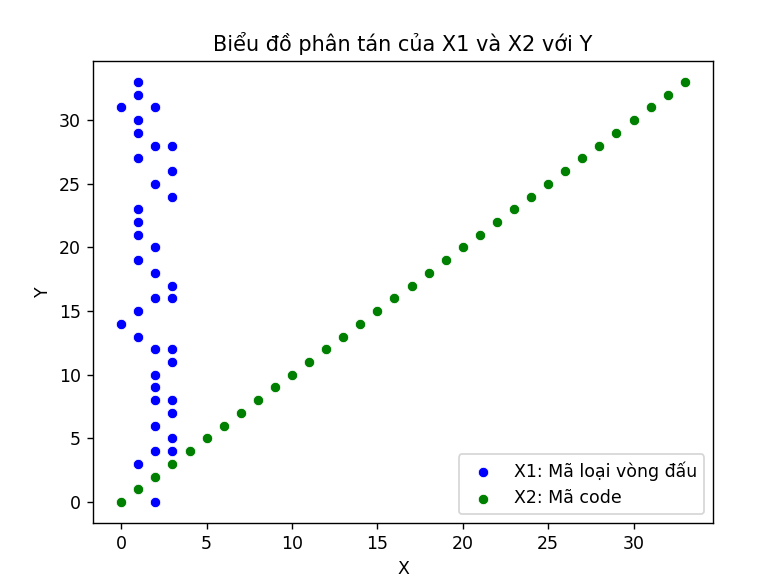
plt.show()

* Hình ảnh trực quan:

A graph with blue dots

Description automatically generated





A graph with blue dots

Description automatically generated

A graph with blue dots and a red line

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A graph of blue rectangular bars

Description automatically generated with medium confidence