

Chapter 9 - Ex3: Lung Function in 1 to 10 Year Old Children

- Cho dữ liệu children_lung.txt. Bộ dữ liệu có 345 trẻ em từ 1 đến 10 tuổi. Với output $y = \text{forced exhalation volume (FEV)}$, thước đo lượng không khí mà ai đó có thể buộc phải thở ra từ phổi của họ và $x = \text{age in years}$. (Nguồn dữ liệu: Dữ liệu ở đây là một phần của bộ dữ liệu được đưa ra trong Kahn, Michael (2005). "An Exhalent Problem for Teaching Statistics", The Journal of Statistical Education, 13".

Yêu cầu:

- Đọc dữ liệu, tiền xử lý dữ liệu, tổng quan ban đầu về dữ liệu.
- Trực quan hóa dữ liệu. Quan sát và nhận xét. Có thấy vấn đề gì đặc biệt từ dữ liệu không? Nếu có thì đó là gì?
- Từ câu 2., xem xét việc tách bài toán này thành 2 phần có được không? Nếu được thì triển khai.

Gợi ý, dữ liệu phần 1

- Thực hiện Simple Linear Regression để dự đoán FEV từ age. Nhận xét kết quả. Trực quan hóa kết quả.
- Cho age lần lượt là: [2, 3, 4, 5]. Hãy cho biết FEV lần lượt là bao nhiêu?
- Đánh giá mô hình vừa xây dựng: có cần phải cải tiến gì không? Nếu cần thì thay đổi mô hình.
- Nhận xét kết quả. Trực quan hóa kết quả với mô hình mới thay đổi. So sánh với mô hình ban đầu. Mô hình sau có tốt hơn không? Quyết định chọn mô hình nào? Lý do?

Gợi ý, dữ liệu phần 2

- Thực hiện Multiple Linear Regression để dự đoán FEV từ age và ht. Nhận xét kết quả. Trực quan hóa kết quả.
- Cho age và ht in year lần lượt là: age = [5, 6, 7, 8, 9], ht = [49.5, 55, 57, 60, 62] . Hãy cho biết FEV lần lượt là bao nhiêu?
- Đánh giá mô hình vừa xây dựng: có cần phải cải tiến gì không? Nếu cần thì thay đổi mô hình.
- Nhận xét kết quả. Trực quan hóa kết quả với mô hình mới thay đổi. So sánh với mô hình ban đầu. Mô hình sau có tốt hơn không? Quyết định chọn mô hình nào? Lý do?

- download dữ liệu: [link](https://newonlinecourses.science.psu.edu/stat462/node/101/) (<https://newonlinecourses.science.psu.edu/stat462/node/101/>)

```
In [1]: # from google.colab import drive
# drive.mount("/content/gdrive", force_remount=True)
# %cd '/content/gdrive/My Drive/MDS5_2022/Practice_2022/Chapter9/'
```

```
In [2]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [3]: ### Đọc dữ liệu  
data = pd.read_csv("children_lung.txt", sep=" ")  
data.head()
```

Out[3]:

	Unnamed: 0	Unnamed: 1	age	FEV	ht	Unnamed: 5	Unnamed: 6	sex	Unnamed: 8	Unnamed: 9	Unn
0	NaN	NaN	9.0	1.708	57.0	NaN	NaN	0.0	NaN	NaN	NaN
1	NaN	NaN	8.0	1.724	67.5	NaN	NaN	0.0	NaN	NaN	NaN
2	NaN	NaN	7.0	1.720	54.5	NaN	NaN	0.0	NaN	NaN	NaN
3	NaN	NaN	9.0	1.558	53.0	NaN	NaN	1.0	NaN	NaN	NaN
4	NaN	NaN	9.0	1.895	57.0	NaN	NaN	1.0	NaN	NaN	NaN

```
In [4]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 654 entries, 0 to 653  
Data columns (total 15 columns):  
 #   Column      Non-Null Count  Dtype     
---  --          --          --  
 0   Unnamed: 0    0 non-null     float64  
 1   Unnamed: 1    345 non-null   float64  
 2   age          654 non-null   float64  
 3   FEV          654 non-null   float64  
 4   ht           309 non-null   float64  
 5   Unnamed: 5    0 non-null     float64  
 6   Unnamed: 6    345 non-null   float64  
 7   sex          309 non-null   float64  
 8   Unnamed: 8    0 non-null     float64  
 9   Unnamed: 9    0 non-null     float64  
 10  Unnamed: 10   0 non-null     float64  
 11  Unnamed: 11   345 non-null   float64  
 12  smoke         309 non-null   float64  
 13  Unnamed: 13   0 non-null     float64  
 14  Unnamed: 14   0 non-null     float64  
 dtypes: float64(15)  
 memory usage: 76.8 KB
```

```
In [5]: data = data[["age", "FEV", "ht", "sex", "smoke"]]  
data.head()
```

Out[5]:

	age	FEV	ht	sex	smoke
0	9.0	1.708	57.0	0.0	0.0
1	8.0	1.724	67.5	0.0	0.0
2	7.0	1.720	54.5	0.0	0.0
3	9.0	1.558	53.0	1.0	0.0
4	9.0	1.895	57.0	1.0	0.0

```
In [6]: data.info()
```

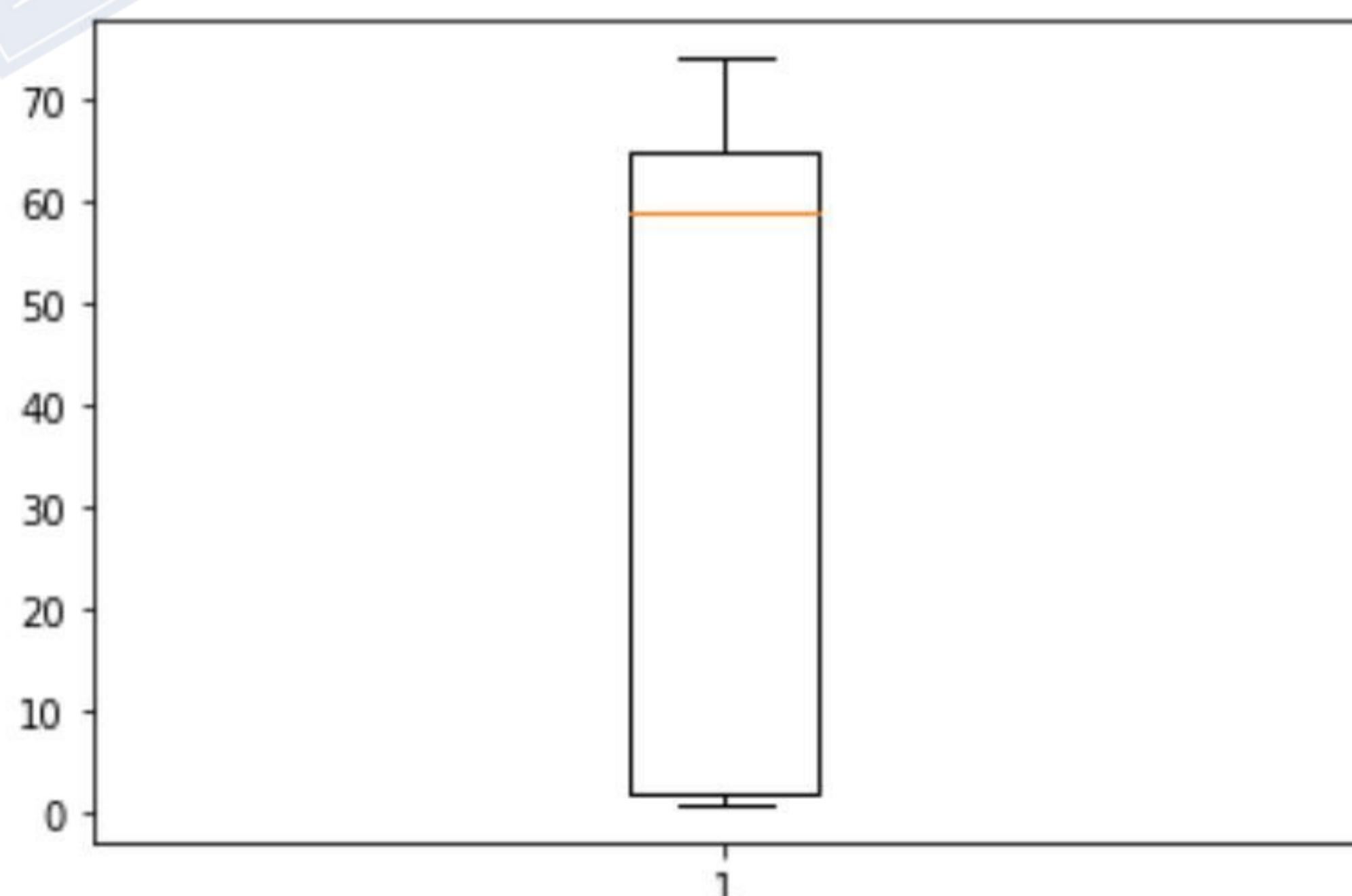
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 654 entries, 0 to 653
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype  
--- 
 0   age      654 non-null    float64
 1   FEV      654 non-null    float64
 2   ht       309 non-null    float64
 3   sex      309 non-null    float64
 4   smoke    309 non-null    float64
dtypes: float64(5)
memory usage: 25.7 KB
```

```
In [7]: data.describe()
```

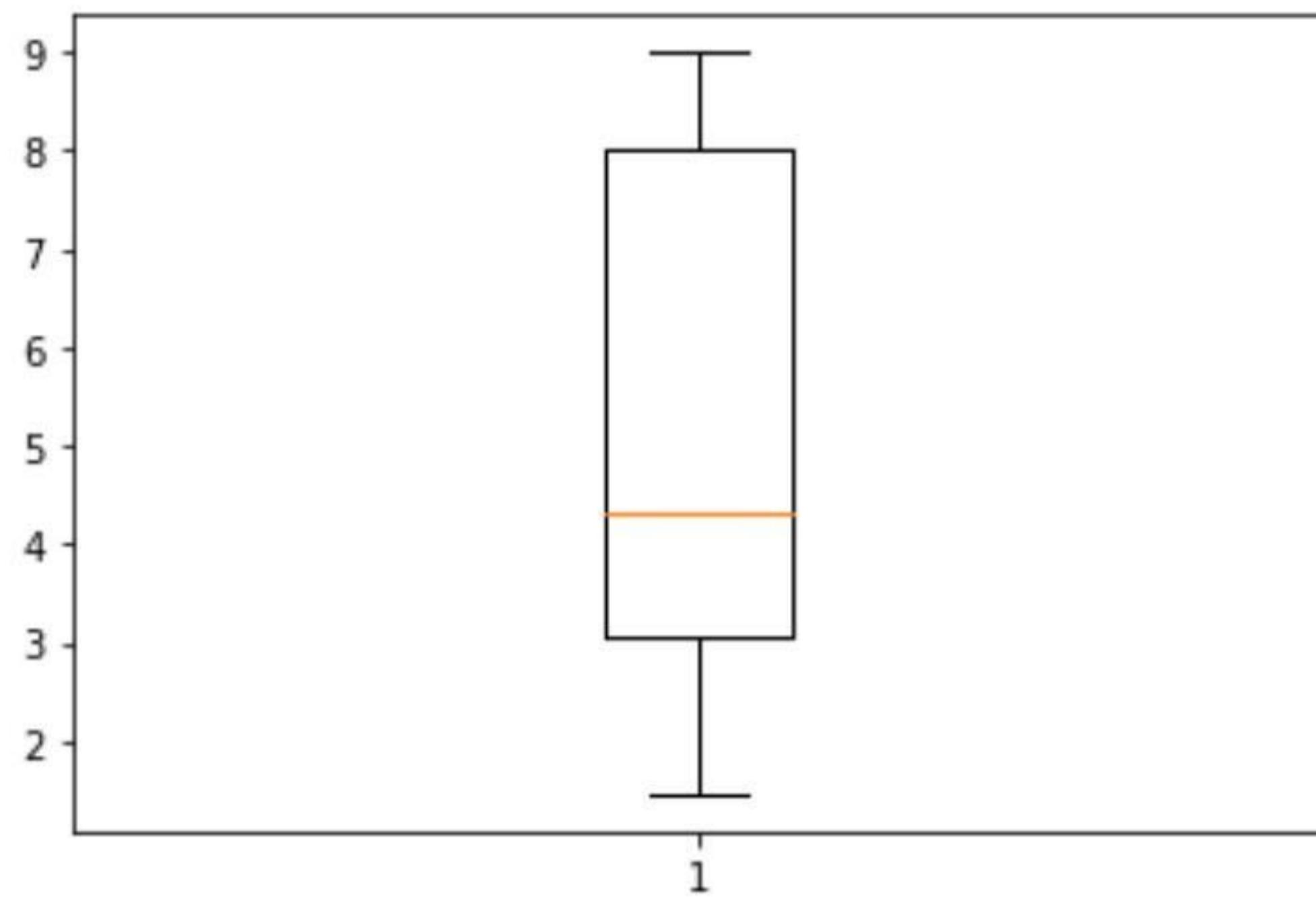
Out[7]:

	age	FEV	ht	sex	smoke
count	654.000000	654.000000	309.000000	309.000000	309.000000
mean	5.205369	35.209546	56.922330	0.498382	0.003236
std	2.429315	31.541283	4.408388	0.500808	0.056888
min	1.458000	0.791000	46.000000	0.000000	0.000000
25%	3.048500	2.040500	53.500000	0.000000	0.000000
50%	4.330000	59.000000	57.000000	0.000000	0.000000
75%	8.000000	65.000000	60.000000	1.000000	0.000000
max	9.000000	74.000000	69.000000	1.000000	1.000000

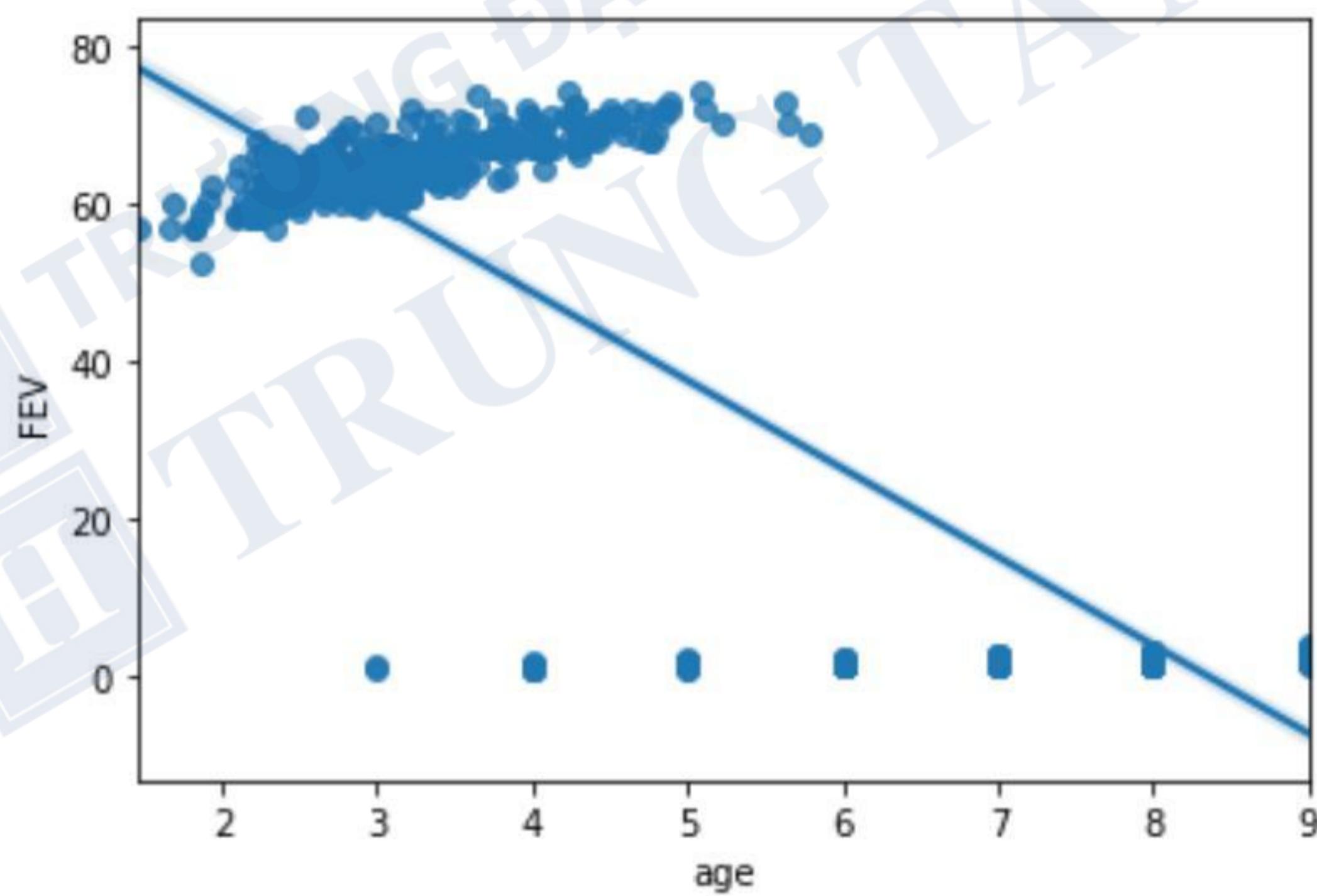
```
In [8]: plt.boxplot(data.FEV)
plt.show()
```



```
In [9]: plt.boxplot(data.age)
plt.show()
```

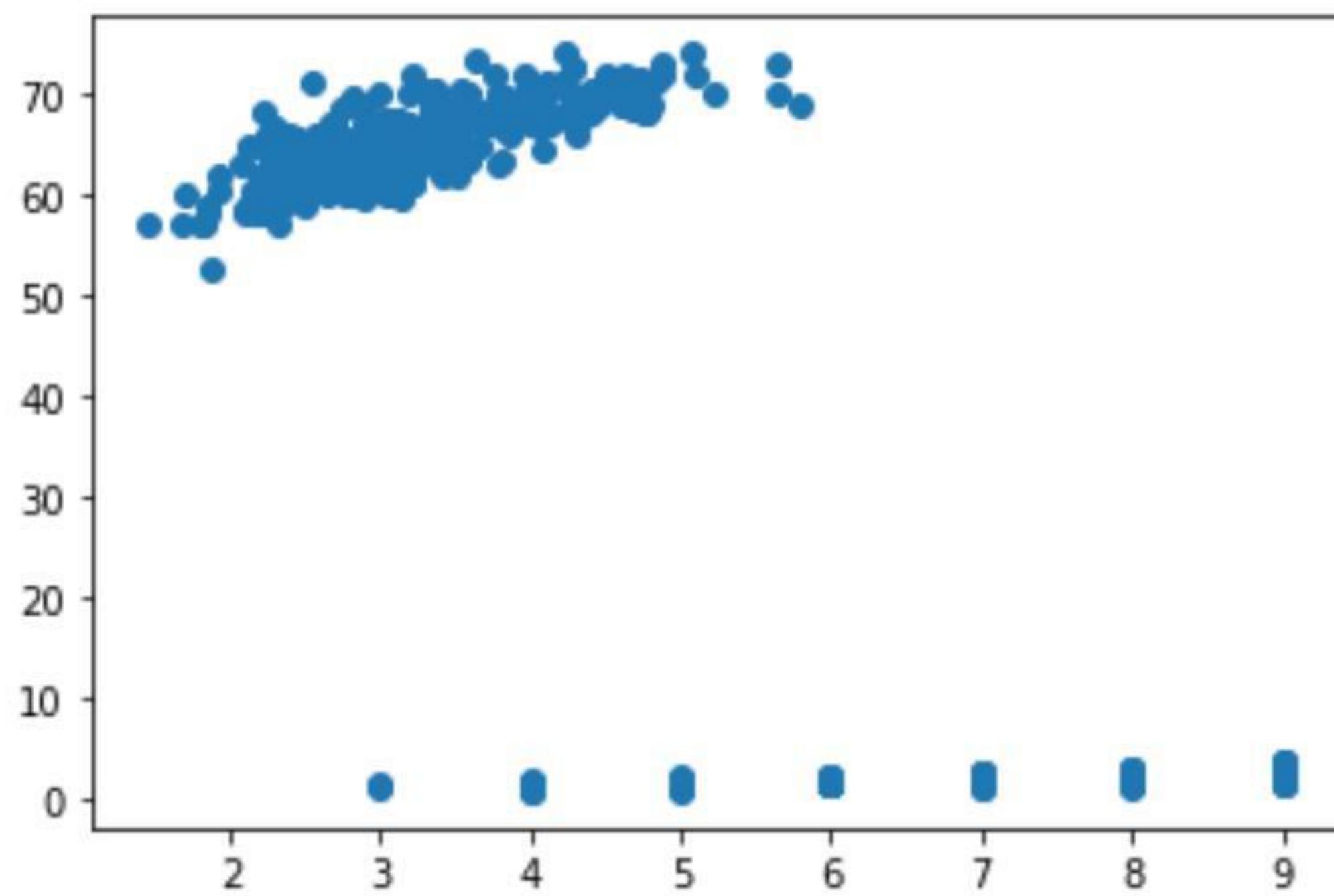


```
In [10]: sns.regplot(data=data, x='age', y='FEV')
plt.show()
```



```
In [11]: plt.scatter(data.age, data.FEV)
```

```
Out[11]: <matplotlib.collections.PathCollection at 0x1d7f22e3eb8>
```



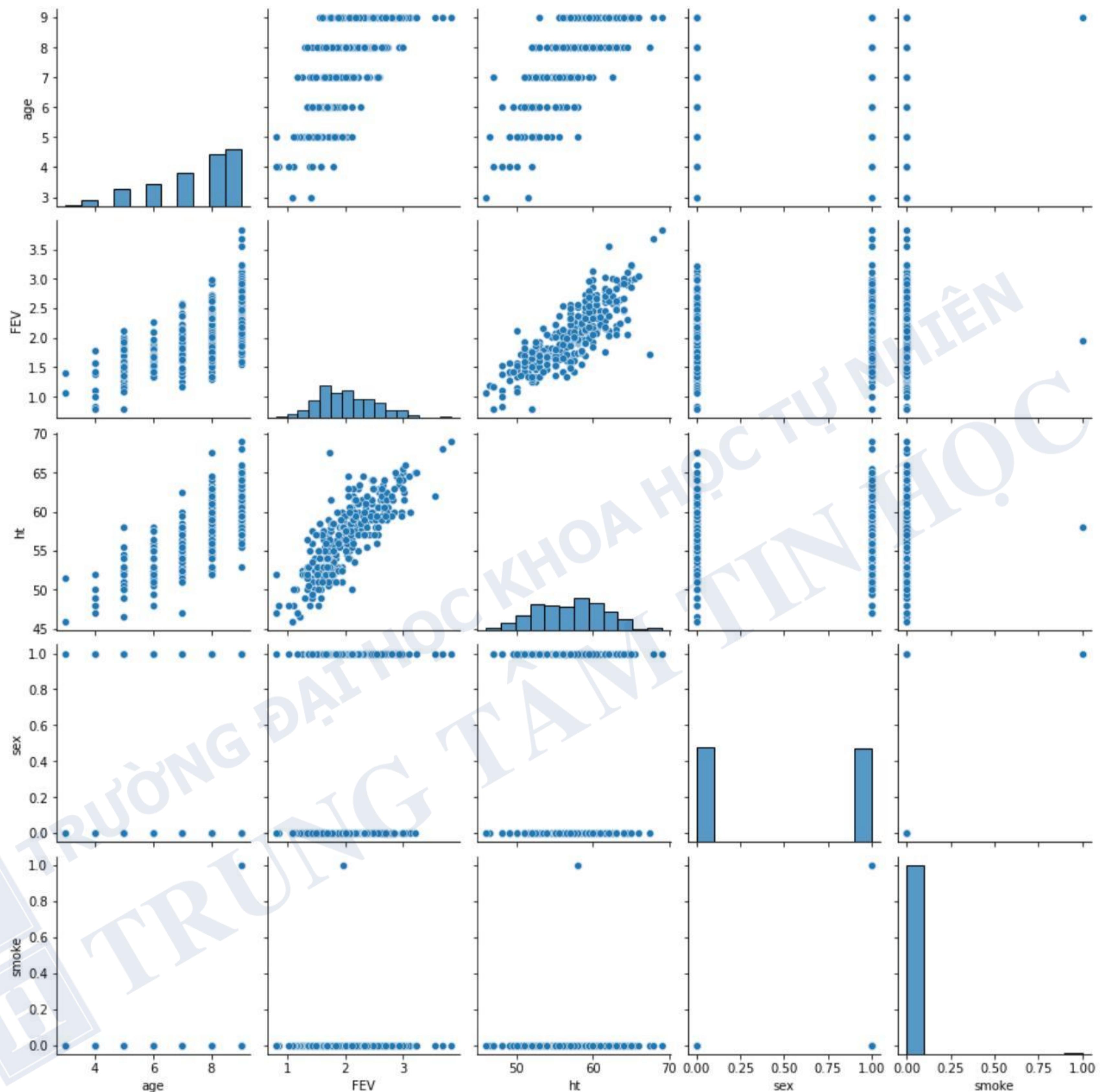
```
In [12]: FEV_lower_10 = data[data.FEV < 10]
FEV_lower_10.shape
```

```
Out[12]: (309, 5)
```

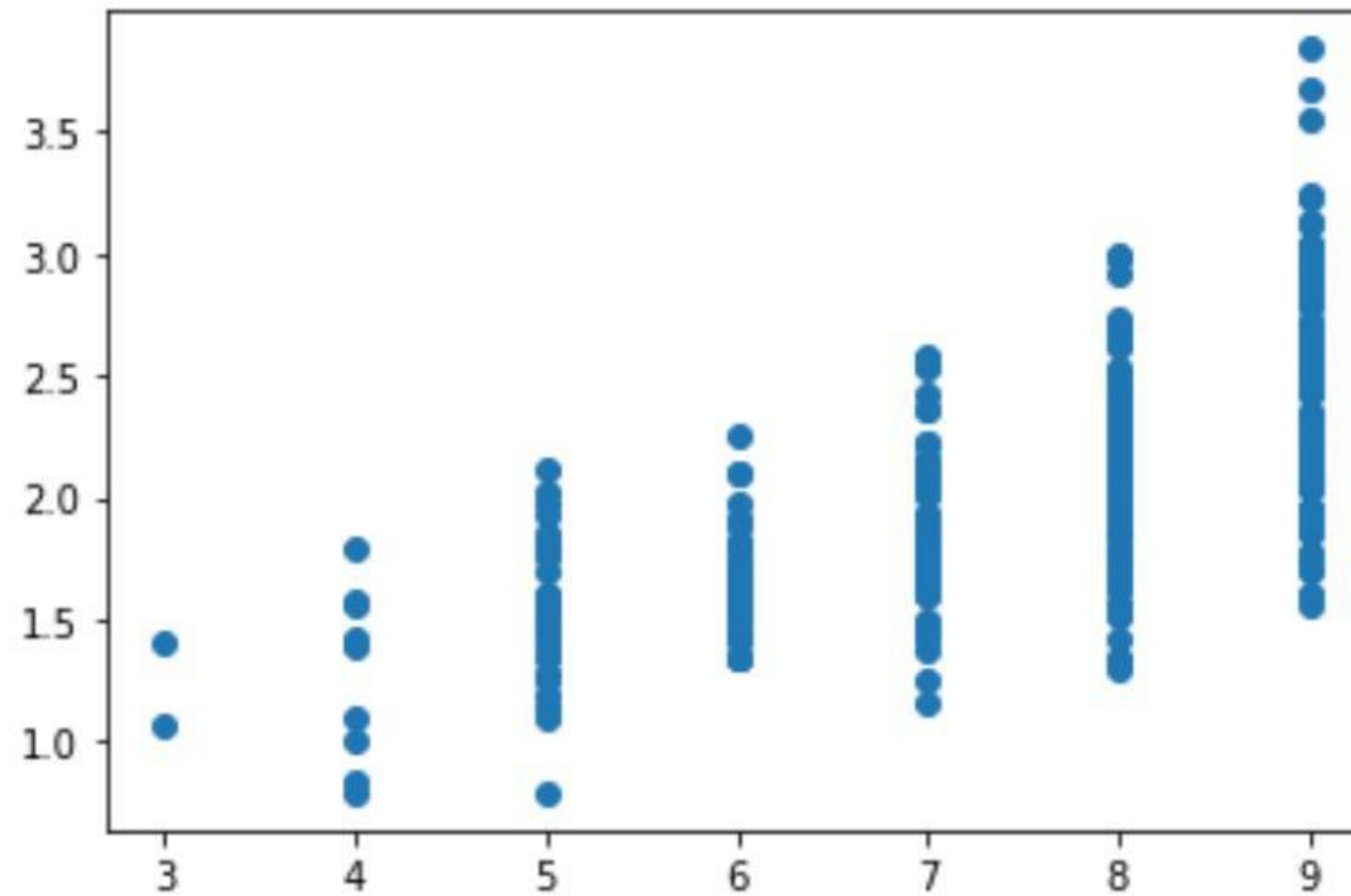
```
In [13]: FEV_lower_10.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 309 entries, 0 to 308
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype  
--- 
 0   age     309 non-null    float64
 1   FEV    309 non-null    float64
 2   ht     309 non-null    float64
 3   sex    309 non-null    float64
 4   smoke   309 non-null    float64
dtypes: float64(5)
memory usage: 14.5 KB
```

```
In [14]: sns.pairplot(FEV_lower_10)
plt.show()
```

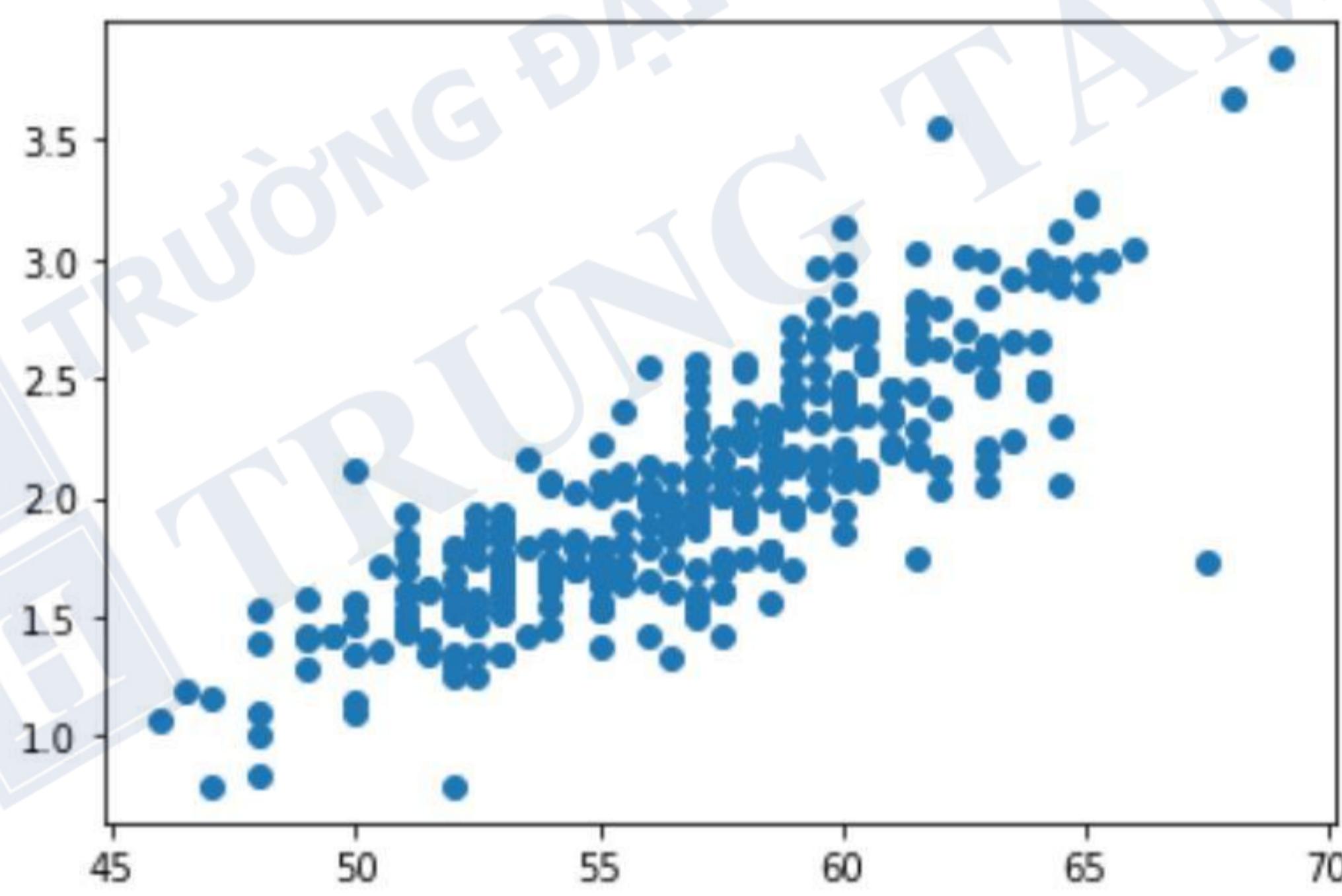


```
In [15]: plt.scatter(FEV_lower_10.age, FEV_lower_10.FEV)  
plt.show()
```



```
In [16]: # Nhóm này có tuổi từ 3 đến 9, chủ yếu tập trung ở Lứa tuổi từ 5 trở lên.  
# Có thể tuyến tính.
```

```
In [17]: plt.scatter(FEV_lower_10.ht, FEV_lower_10.FEV)  
plt.show()
```



```
In [18]: # ht tương đối tuyến tính so với FEV
```

```
In [19]: # Smoke: trẻ em không hút thuốc
```

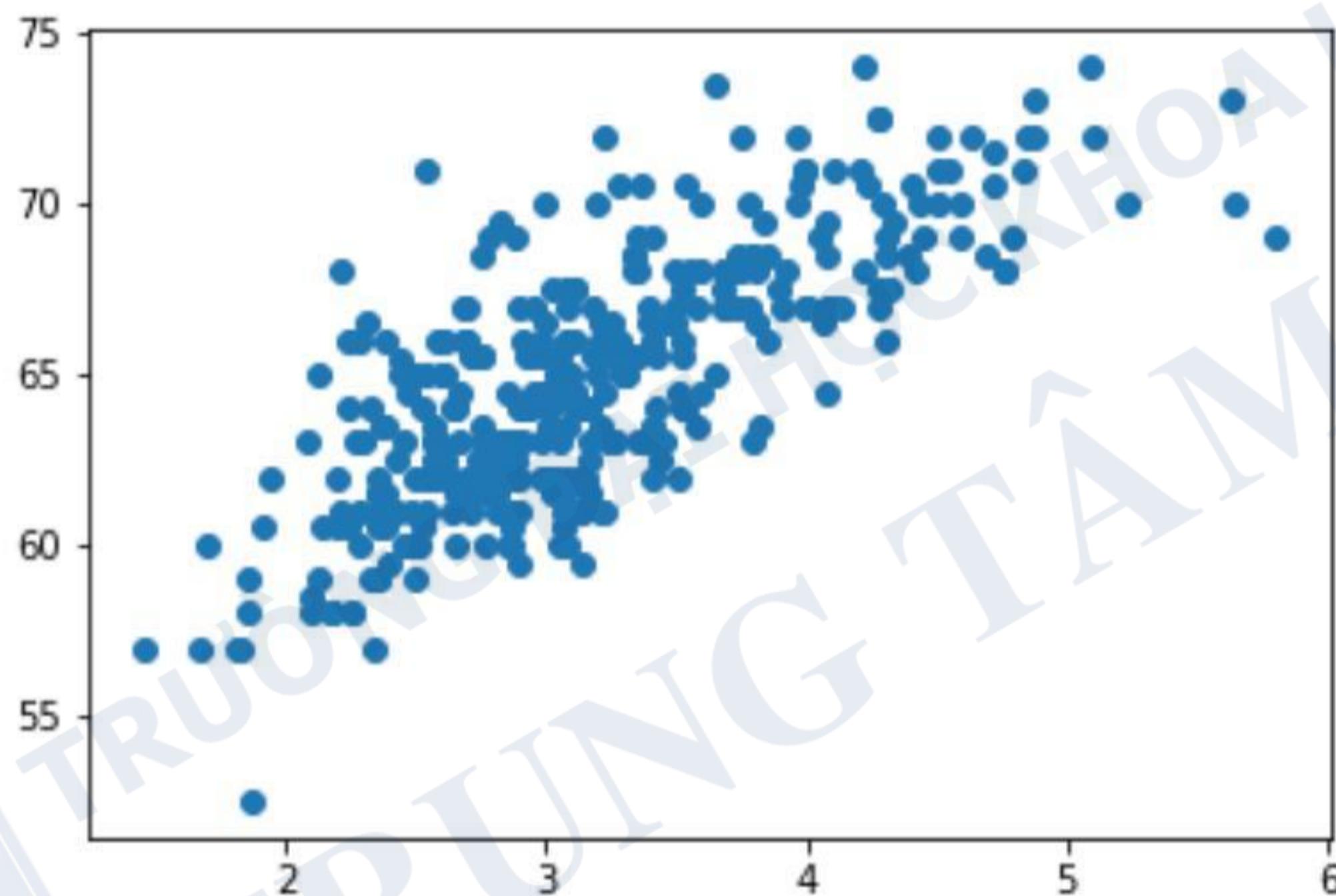
```
In [20]: FEV_upper_10 = data[data.FEV >= 10]  
FEV_upper_10.shape
```

```
Out[20]: (345, 5)
```

```
In [21]: FEV_upper_10.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 345 entries, 309 to 653
Data columns (total 5 columns):
 #   Column   Non-Null Count   Dtype  
--- 
 0   age      345 non-null    float64
 1   FEV      345 non-null    float64
 2   ht       0 non-null     float64
 3   sex      0 non-null     float64
 4   smoke    0 non-null     float64
dtypes: float64(5)
memory usage: 16.2 KB
```

```
In [22]: plt.scatter(FEV_upper_10.age, FEV_upper_10.FEV)
plt.show()
```



```
In [23]: # Nhóm này có tuổi từ 0 đến 6, tập trung nhiều từ 2 đến 5, tương đối tuyến tính
```

```
In [24]: # Chia làm 2 nhóm tương đương là FEV nhỏ hơn 10 và FEV Lớn hơn 10.
# Quan sát thấy với nhóm có FEV_Lower_10 có 345 mẫu với đầy đủ dữ liệu,
# Trong khi đó nhóm FEV_upper_10 chỉ có dữ liệu của age và FEV.
# Liệu có nên chia bài toán thành 2 phần, dự đoán theo từng phần?
# Tham số của từng phần như thế nào?
```

Phần 1: FEV_upper_10, dành cho dữ liệu chỉ có age

Dự đoán FEV từ age

Simple Linear Regression

```
In [25]: from sklearn.linear_model import LinearRegression
```

```
In [26]: lm = LinearRegression()  
lm
```

```
Out[26]: LinearRegression()
```

```
In [27]: X_u = FEV_upper_10[['age']]  
Y_u = FEV_upper_10['FEV']
```

```
In [28]: # Chia dữ liệu thành 2 phần, train và test  
from sklearn.model_selection import train_test_split
```

```
In [29]: X_train_u, X_test_u, y_train_u, y_test_u = train_test_split(X_u,  
Y_u,  
test_size=0.2)
```

```
In [30]: # Train model  
lm.fit(X_train_u,y_train_u)
```

```
Out[30]: LinearRegression()
```

```
In [31]: b = lm.intercept_  
b
```

```
Out[31]: 52.976291690247514
```

```
In [32]: m = lm.coef_[0]  
m
```

```
Out[32]: 3.7626078222145445
```

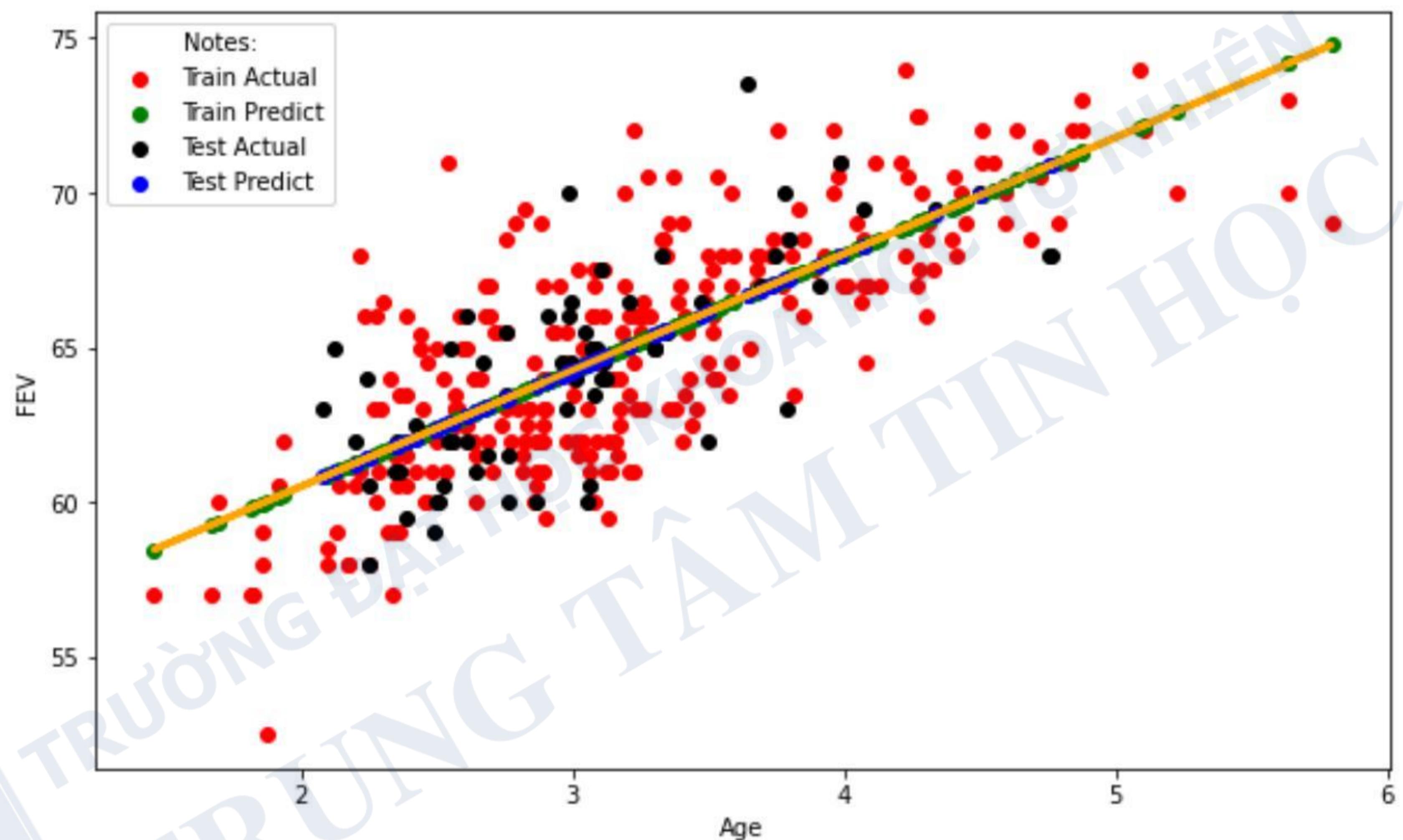
"y = mx + b "

```
In [33]: # Predict trên X_train_u  
yHat_train = lm.predict(X_train_u)
```

```
In [34]: # Predict trên X_test_u  
yHat_test = lm.predict(X_test_u)
```

```
In [35]: reg_line = [(m* float(x)) + b for x in np.array(X_u)]
```

```
In [36]: plt.figure(figsize=(10,6))
plt.plot(X_u,reg_line, color="orange", linewidth=3)
plt.scatter(X_train_u, y_train_u, color='red', label='Train Actual')
plt.scatter(X_train_u, yHat_train, color='green', label='Train Predict')
plt.scatter(X_test_u, y_test_u, color='black', label='Test Actual')
plt.scatter(X_test_u, yHat_test, color='blue', label='Test Predict')
plt.xlabel("Age")
plt.ylabel("FEV")
plt.legend(title="Notes:")
plt.show()
```



```
In [37]: # Find the MSE
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(Y_u, lm.predict(X_u))
print('The MSE of FEV and predicted value is: ', mse)
```

The MSE of FEV and predicted value is: 5.883580542664589

```
In [38]: # Find the R^2
print('The R-square is: ', lm.score(X_u, Y_u))
```

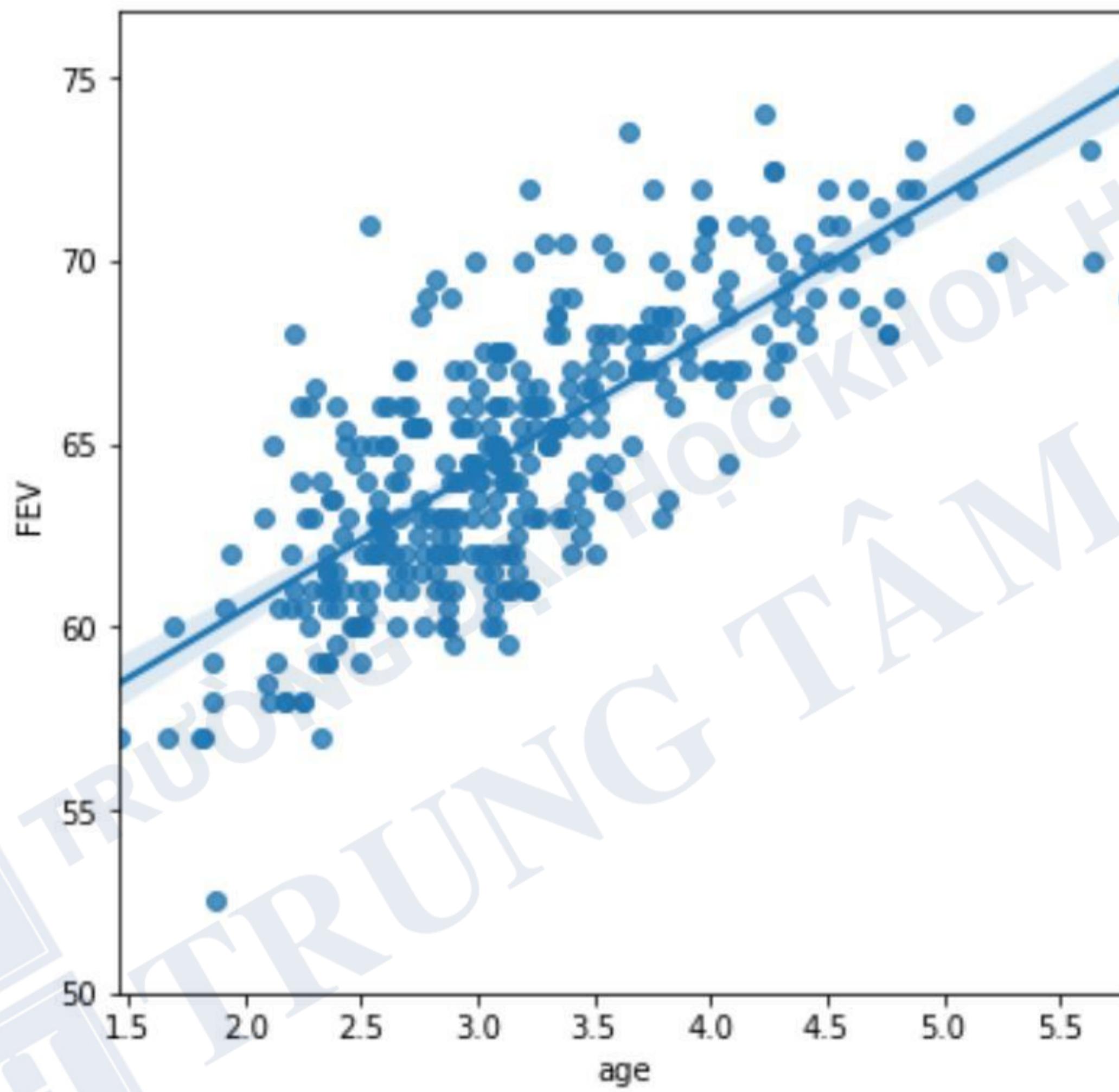
The R-square is: 0.5787793183004307

```
In [39]: # predict new data  
X_new_u = pd.DataFrame(np.array([2, 3, 4, 5]))  
y_new_u = lm.predict(X_new_u)  
y_new_u
```

```
Out[39]: array([60.50150733, 64.26411516, 68.02672298, 71.7893308 ])
```

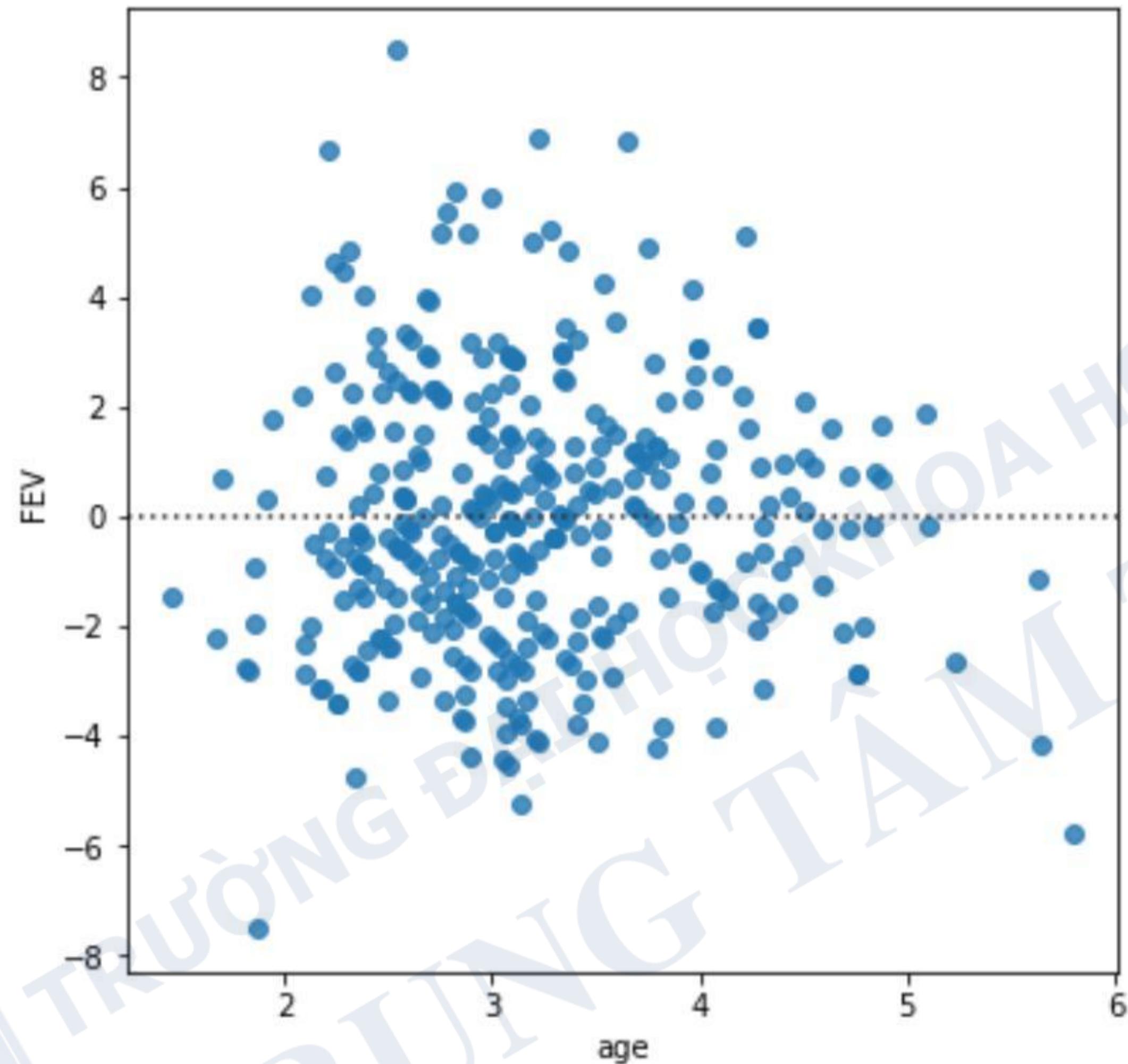
```
In [40]: # regression plot  
plt.figure(figsize=(6,6))  
sns.regplot(x="age", y="FEV", data=FEV_upper_10)  
plt.ylim(50,)
```

```
Out[40]: (50.0, 76.85186879656412)
```



```
In [41]: # Residual plot  
plt.figure(figsize=(6,6))  
sns.residplot(FEV_upper_10['age'], FEV_upper_10['FEV'])  
plt.show()
```

c:\program files\python36\lib\site-packages\seaborn_decorators.py:43: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
FutureWarning



```
In [42]: Y_u_pred = lm.predict(X_u)
```

```
In [43]: # Distribution plot
plt.figure(figsize=(8,8))
ax1 = sns.distplot(Y_u, hist=False, color="r",
                    label="Actual Value")
sns.distplot(Y_u_pred, hist=False, color="b",
              label="Fitted Values" ,
              ax=ax1)

plt.title('Actual vs Fitted Values for FEV')
plt.xlabel('FEV')
plt.ylabel('freq')

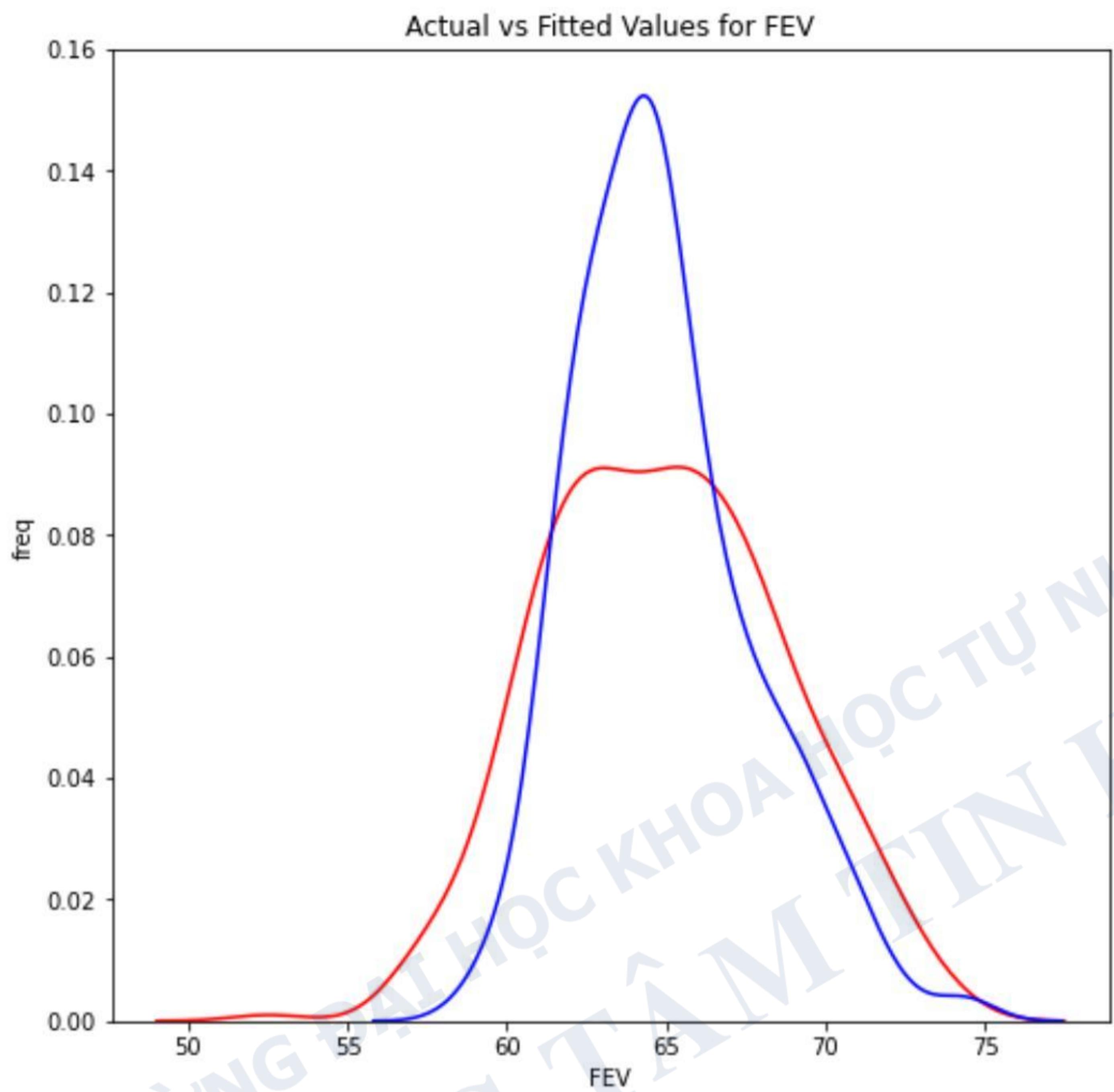
plt.show()
plt.close()
```

c:\program files\python36\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).

```
warnings.warn(msg, FutureWarning)
```

c:\program files\python36\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).

```
warnings.warn(msg, FutureWarning)
```



Polinorminal Regression

```
In [44]: import numpy as np
```

```
In [45]: X_u_p = FEV_upper_10[['age']]
Y_u_p = FEV_upper_10['FEV']
```

```
In [46]: from sklearn.preprocessing import PolynomialFeatures
```

```
In [47]: pr=PolynomialFeatures(degree=2)
pr
```

```
Out[47]: PolynomialFeatures()
```

```
In [48]: pr.fit(X_u_p)
```

```
Out[48]: PolynomialFeatures()
```

```
In [49]: X_u_p_pr = pr.transform(X_u_p)
```

```
In [50]: X_u_p.shape, X_u_p_pr.shape
```

```
Out[50]: ((345, 1), (345, 3))
```

```
In [51]: lm1_pr = LinearRegression()  
lm1_pr.fit(X_u_p_pr, Y_u_p)
```

```
Out[51]: LinearRegression()
```

```
In [52]: Y_u_p_hat_pr = lm1_pr.predict(X_u_p_pr)
```

```
In [53]: lm1_pr.intercept_
```

```
Out[53]: 47.71519711405193
```

```
In [54]: lm1_pr.coef_
```

```
Out[54]: array([ 0.           ,  6.98881744, -0.46863146])
```

```
In [55]: # Find R^2  
from sklearn.metrics import r2_score  
r_squared = r2_score(Y_u_p, Y_u_p_hat_pr)  
print('The R-square of Polinomial Regression model is: ', r_squared)
```

```
The R-square of Polinomial Regression model is:  0.5885078184205568
```

```
In [56]: # Find the MSE  
from sklearn.metrics import mean_squared_error  
mse = mean_squared_error(Y_u_p, Y_u_p_hat_pr)  
print('The MSE of FEV and predicted value is: ', mse)
```

```
The MSE of FEV and predicted value is:  5.747693544464183
```

```
In [58]: # Distribution plot
plt.figure(figsize=(8,8))
ax1 = sns.distplot(Y_u_p, hist=False, color="r",
                    label="Actual Value")
sns.distplot(Y_u_p_hat_pr , hist=False, color="b",
              label="Fitted Values PR" ,
              ax=ax1)

plt.title('Actual vs Fitted Values for FEV')
plt.xlabel('FEV')
plt.ylabel('freq')

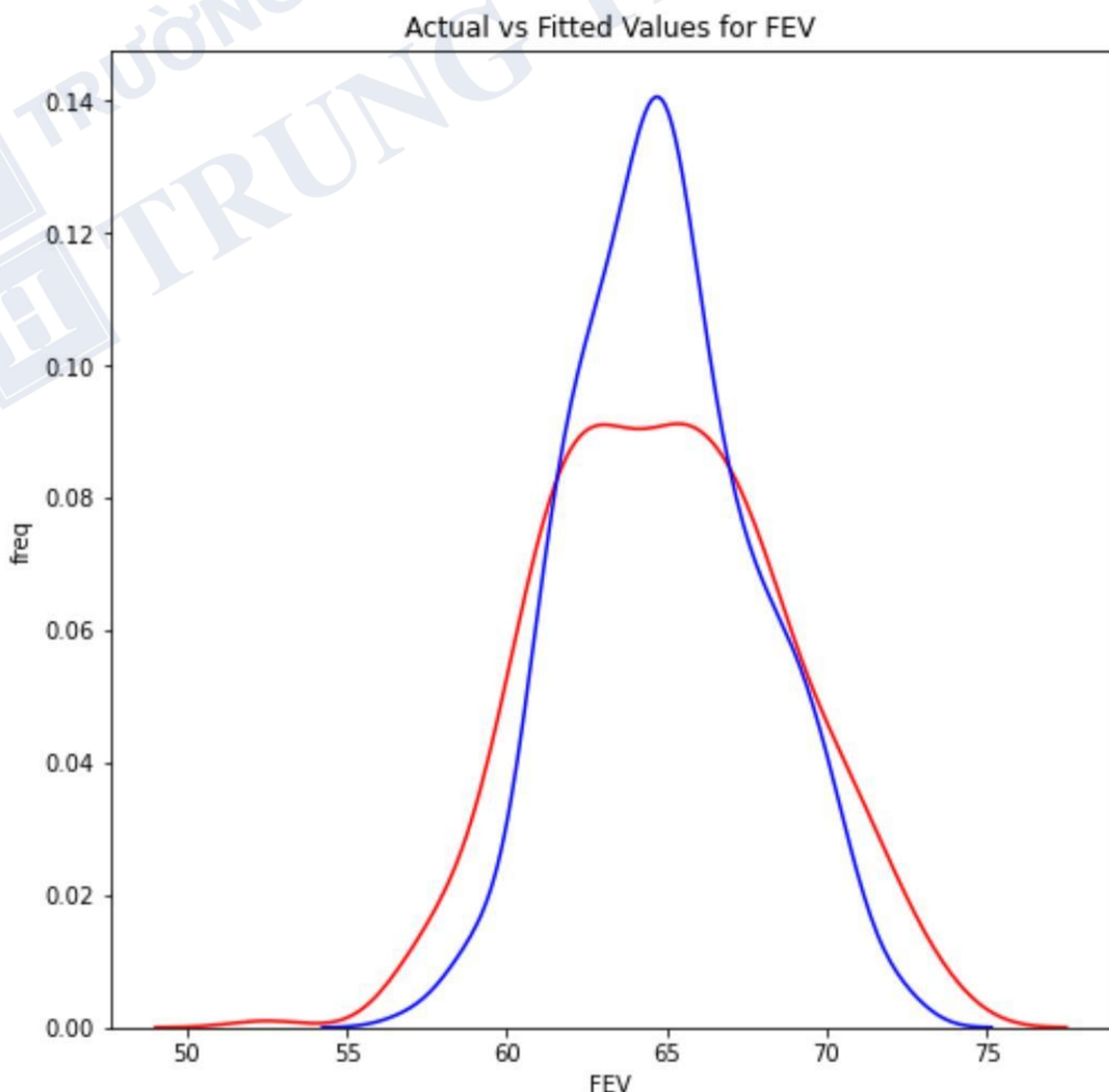
plt.show()
plt.close()
```

c:\program files\python36\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).

warnings.warn(msg, FutureWarning)

c:\program files\python36\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).

warnings.warn(msg, FutureWarning)



```
In [59]: # PR tốt hơn so với SR => có thể chọn PR
```

```
In [60]: # predict new data  
X_new_p = pd.DataFrame({'age': [2, 3, 4, 5]})  
X_new_p_pr = pr.transform(X_new_p)  
Y_new_p = lm1_pr.predict(X_new_p_pr)  
Y_new_p
```

```
Out[60]: array([59.81830614, 64.46396626, 68.17236347, 70.94349774])
```

Phần 2: FEV_lower_10, dành cho dữ liệu có age, ht

Dự đoán FEV từ age, ht

Multiple Linear Regression

```
In [61]: # Với X L có age, ht => y L
```

```
In [62]: X_1 = FEV_lower_10[['age', 'ht']]  
Y_1 = FEV_lower_10['FEV']
```

```
In [64]: lm1 = LinearRegression()
lm1
```

```
Out[64]: LinearRegression()
```

```
In [65]: # Train model
lm1.fit(X_train_l,y_train_l)
```

```
Out[65]: LinearRegression()
```

```
In [66]: b = lm1.intercept_
b
```

```
Out[66]: -3.213843531097137
```

```
In [67]: m1, m2 = lm1.coef_[0], lm1.coef_[1]
m1, m2
```

```
Out[67]: (0.025304714634211935, 0.08906782190648137)
```

```
In [68]: # Predict trên X_train_l
yHat_train_l = lm1.predict(X_train_l)
```

```
In [69]: # Predict trên X_test_u
yHat_test_l = lm1.predict(X_test_l)
```

```
In [70]: # Find the MSE
mse1 = mean_squared_error(Y_l, lm1.predict(X_l))
print('The MSE of price and predicted value is: ', mse1)
```

The MSE of price and predicted value is: 0.08555676121376274

```
In [71]: # Find the R^2
print('The R-square is: ', lm1.score(X_l, Y_l))
```

The R-square is: 0.6720000946101135

```
In [72]: # New prediction
age = [5, 6, 7, 8, 9]
ht = [49.5, 55, 57, 60, 62]
X_l_new = pd.DataFrame({'age': age, 'ht': ht})
yHat_l_new = lm1.predict(X_l_new)
yHat_l_new
```

```
Out[72]: array([1.32153723, 1.83671496, 2.04015532, 2.3326635 , 2.53610386])
```

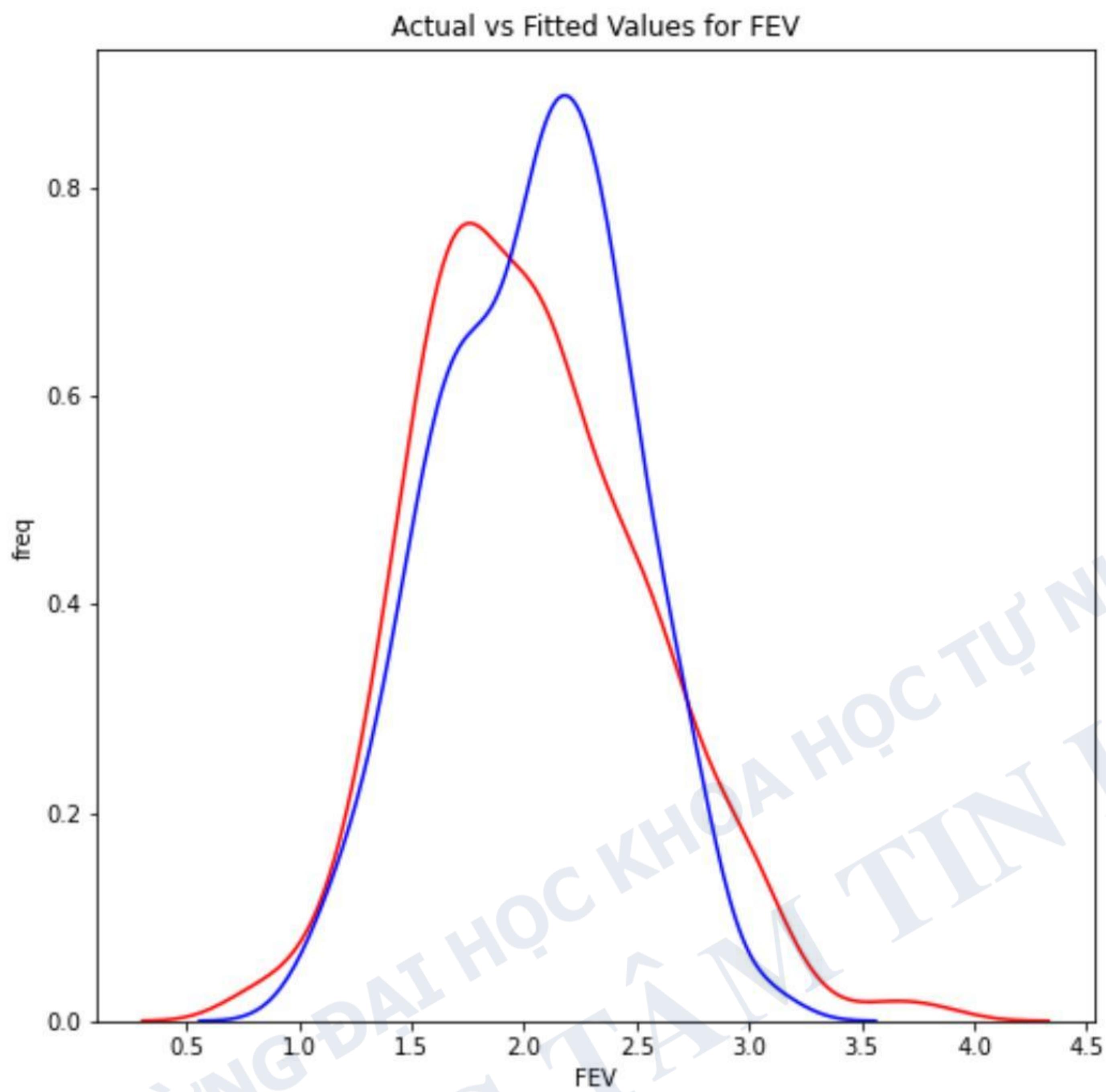
```
In [73]: yHat_l = lm1.predict(X_l)
```

```
In [74]: # Đánh giá mô hình  
# Distribution plot  
plt.figure(figsize=(8,8))  
ax1 = sns.distplot(Y_1, hist=False, color="r",  
                    label="Actual Value")  
sns.distplot(yHat_1, hist=False, color="b",  
             label="Fitted Values" ,  
             ax=ax1)  
  
plt.title('Actual vs Fitted Values for FEV')  
plt.xlabel('FEV')  
plt.ylabel('freq')  
  
plt.show()  
plt.close()
```

c:\program files\python36\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).

```
warnings.warn(msg, FutureWarning)  
c:\program files\python36\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).
```

```
warnings.warn(msg, FutureWarning)
```



Multiple Polynominal

```
In [75]: from sklearn.preprocessing import PolynomialFeatures
```

```
In [76]: pr=PolynomialFeatures(degree=2)  
pr
```

```
Out[76]: PolynomialFeatures()
```

```
In [77]: X_1_pr=pr.fit_transform(X_1)
```

```
In [78]: X_1.shape, X_1_pr.shape
```

```
Out[78]: ((309, 2), (309, 6))
```

```
In [79]: lm1_t = LinearRegression()
```

```
In [80]: lm1_t.fit(X_1_pr, Y_1)
```

```
Out[80]: LinearRegression()
```

```
In [81]: Yhat_t = lm1_t.predict(X_1_pr)
```

```
In [82]: b1_t = lm1_t.intercept_
```

```
In [83]: m1_t = lm1_t.coef_
```

```
In [84]: b1_t, m1_t
```

```
Out[84]: (-0.42182399116147584,  
 array([ 0. , -0.58556875,  0.06971195, -0.01718083,  0.01546802,  
 -0.00085841]))
```

```
In [85]: X_1_new = pd.DataFrame({'age': age, 'ht': ht})  
X_1_pr_new=pr.fit_transform(X_1_new)  
yHat_l_t_new = lm1_t.predict(X_1_pr_new)  
yHat_l_t_new
```

```
Out[85]: array([1.39657488, 1.78817454, 1.99368881, 2.31115186, 2.56998682])
```

```
In [86]: # Find R^2  
from sklearn.metrics import r2_score  
r_squared = r2_score(Y_1, Yhat_t)  
print('The R-square value is: ', r_squared)
```

```
The R-square value is:  0.6823609472684569
```

```
In [87]: # Find MSE  
print('The MSE of price and predicted value using Polinormial Fit: ', \  
     mean_squared_error(Y_1, Yhat_t))
```

```
The MSE of price and predicted value using Polinormial Fit:  0.0828541964194004  
4
```

```
In [88]: # Đánh giá mô hình
# Distribution plot
plt.figure(figsize=(8,8))
ax1 = sns.distplot(Y_1, hist=False, color="r",
                    label="Actual Value")
sns.distplot(Yhat_t, hist=False, color="b",
              label="Fitted Values MPR" ,
              ax=ax1)
sns.distplot(yHat_1, hist=False, color="g",
              label="Fitted Values MR" ,
              ax=ax1)

plt.title('Actual vs Fitted Values for FEV')
plt.xlabel('FEV')
plt.ylabel('freq')

plt.show()
plt.close()
```

```
c:\program files\python36\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).
```

```
    warnings.warn(msg, FutureWarning)
```

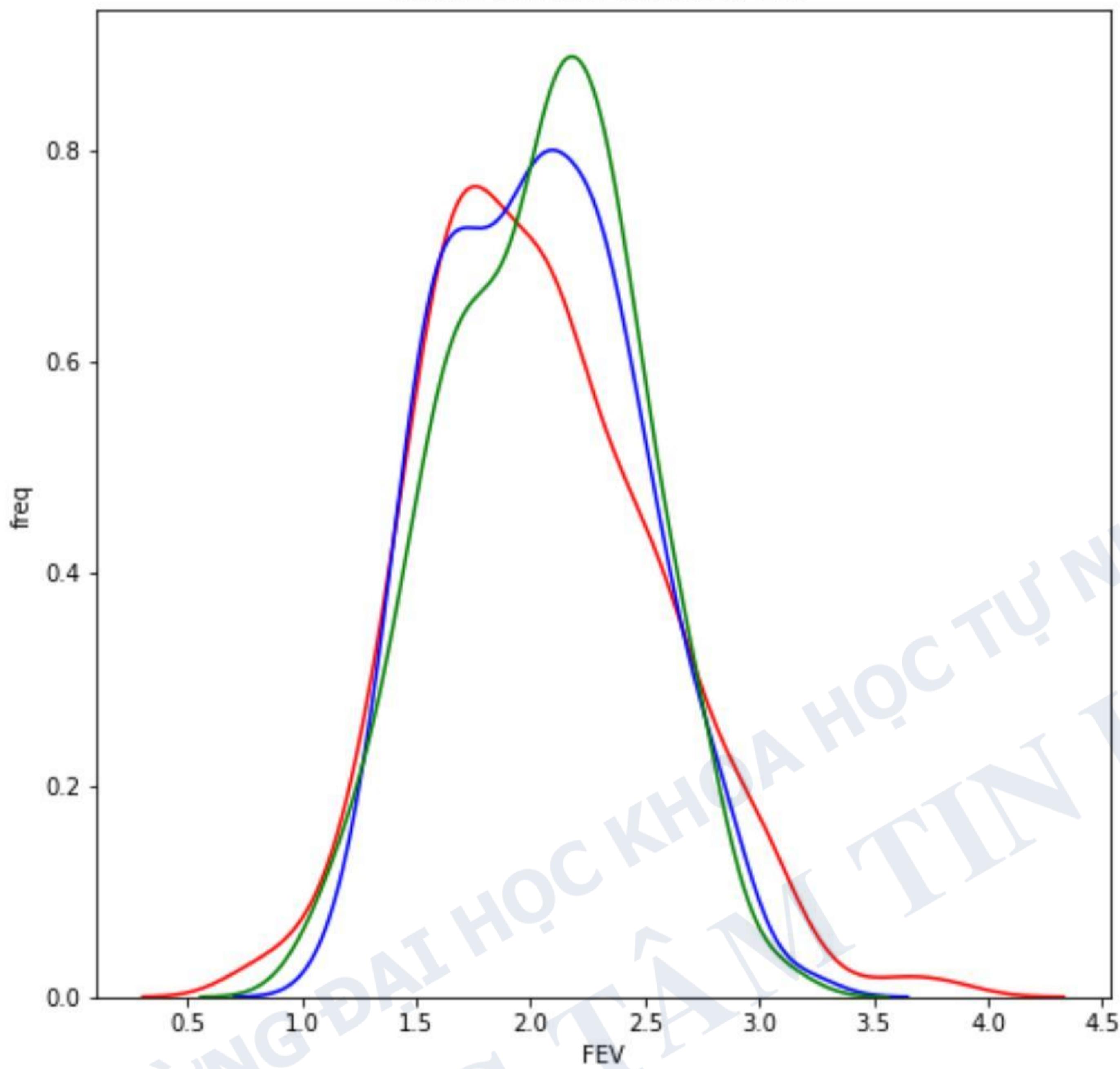
```
c:\program files\python36\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).
```

```
    warnings.warn(msg, FutureWarning)
```

```
c:\program files\python36\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).
```

```
    warnings.warn(msg, FutureWarning)
```

Actual vs Fitted Values for FEV



In [89]: # MPL cho kết quả tốt hơn so với MR => Có thể chọn MPL