Chapter 2 - Ex2: Automobile

Cho dữ liệu Automobile_data.csv (link tham khảo và dowload Automobile (https://www.kaggle.com/toramky/automobile-dataset?select=Automobile_data.csv))

Yêu cầu:

1. Đọc dữ liệu. Hiển thị các thông tin chung về dữ liệu
2. Cho biết có bao nhiêu giá trị trong cột 'price' là chuỗi số, bao nhiêu giá trị không là chuỗi số. Cho biết vị trí các dòng chứa 'price' không phải là chuỗi số.
3. Thay thế những 'price' không phải là chuỗi số này bằng giá trị median của 'price'. Đổi cột 'price' sang kiểu số.
4. Thực hiện tương tự 2. và 3. cho các cột 'horsepower', normalized-losses
5. Tìm hiểu xu hướng trung tâm của các cột 'height', 'price'
6. Trực quan hóa phân phối của các cột 'height', 'price'. Nhận xét.
7. Trực quan hóa mối quan hệ giữa 'horsepower' and 'price'

In [ ]:
```
# from google.colab import drive
# drive.mount("/content/gdrive", force_remount=True)
# %cd '/content/gdrive/My Drive/MDS5_2022/Practice_2022/Chapter2/'
```

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

Gợi ý làm bài

In [ ]:
```
#1.
df = pd.read_csv("Automobile_data.csv")
```

```
In [ ]: df.head()
```

Out[3]:

	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	.
0	3	?	alfa-romero	gas	std	two	convertible	rwd	front	88.6	.
1	3	?	alfa-romero	gas	std	two	convertible	rwd	front	88.6	.
2	1	?	alfa-romero	gas	std	two	hatchback	rwd	front	94.5	.
3	2	164	audi	gas	std	four	sedan	fwd	front	99.8	.
4	2	164	audi	gas	std	four	sedan	4wd	front	99.4	.

5 rows × 26 columns

In [ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 26 columns):
symboling            205 non-null int64
normalized-losses    205 non-null object
make                 205 non-null object
fuel-type            205 non-null object
aspiration           205 non-null object
num-of-doors         205 non-null object
body-style           205 non-null object
drive-wheels         205 non-null object
engine-location      205 non-null object
wheel-base           205 non-null float64
length               205 non-null float64
width                205 non-null float64
height               205 non-null float64
curb-weight          205 non-null int64
engine-type          205 non-null object
num-of-cylinders     205 non-null object
engine-size          205 non-null int64
fuel-system          205 non-null object
bore                 205 non-null object
stroke               205 non-null object
compression-ratio    205 non-null float64
horsepower           205 non-null object
peak-rpm             205 non-null object
city-mpg             205 non-null int64
highway-mpg          205 non-null int64
price                205 non-null object
dtypes: float64(5), int64(5), object(16)
memory usage: 41.8+ KB
```

In [ ]:
```
#2.
df['price'].str.isnumeric().value_counts()
```

Out[5]:
```
True     201
False      4
Name: price, dtype: int64
```

In [ ]:
```
# List out the values which are not numeric
df['price'].loc[df['price'].str.isnumeric() == False]
```

Out[6]:
```
9      ?
44     ?
45     ?
129    ?
Name: price, dtype: object
```

In [ ]:

```
#3.
price = df['price'].loc[df['price'] != '?']
pmedian = price.astype(float).median()
df['price'] = df['price'].replace('?',pmedian).astype(float)
df['price'].head()
```

Out[7]:

```
0    13495.0
1    16500.0
2    16500.0
3    13950.0
4    17450.0
Name: price, dtype: float64
```

In [ ]:

```
df.head()
```

Out[8]:

	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	.
0	3	?	alfa-romero	gas	std	two	convertible	rwd	front	88.6	.
1	3	?	alfa-romero	gas	std	two	convertible	rwd	front	88.6	.
2	1	?	alfa-romero	gas	std	two	hatchback	rwd	front	94.5	.
3	2	164	audi	gas	std	four	sedan	fwd	front	99.8	.
4	2	164	audi	gas	std	four	sedan	4wd	front	99.4	.

5 rows × 26 columns

In [ ]:

```
df[['price']].dtypes
```

Out[9]:

```
price    float64
dtype: object
```

In [ ]:

```
#4. Cleaning the horsepower losses field
df['horsepower'].str.isnumeric().value_counts()
horsepower = df['horsepower'].loc[df['horsepower'] != '?']
hpmedian = horsepower.astype(int).mean()
df['horsepower'] = df['horsepower'].replace('?',hpmedian).astype(int)
df['horsepower'].head()
```

Out[10]:

```
0    111
1    111
2    154
3    102
4    115
Name: horsepower, dtype: int32
```

In [ ]:

```
# Cleaning the Normalized losses field
df[df['normalized-losses']=='?'].count()
nl=df['normalized-losses'].loc[df['normalized-losses'] !='?'].count()
nmedian=nl.astype(int).mean()
df['normalized-losses'] = df['normalized-losses'].replace('?',nmedian).astype(in
df['normalized-losses'].head()
```

Out[11]:

```
0    164
1    164
2    164
3    164
4    164
Name: normalized-losses, dtype: int32
```

In [ ]:

```
#5.
#calculate mean, median and mode of height
mean = df["height"].mean()
median =df["height"].median()
mode = df["height"].mode()
print(round(mean,2) , median, mode)
```

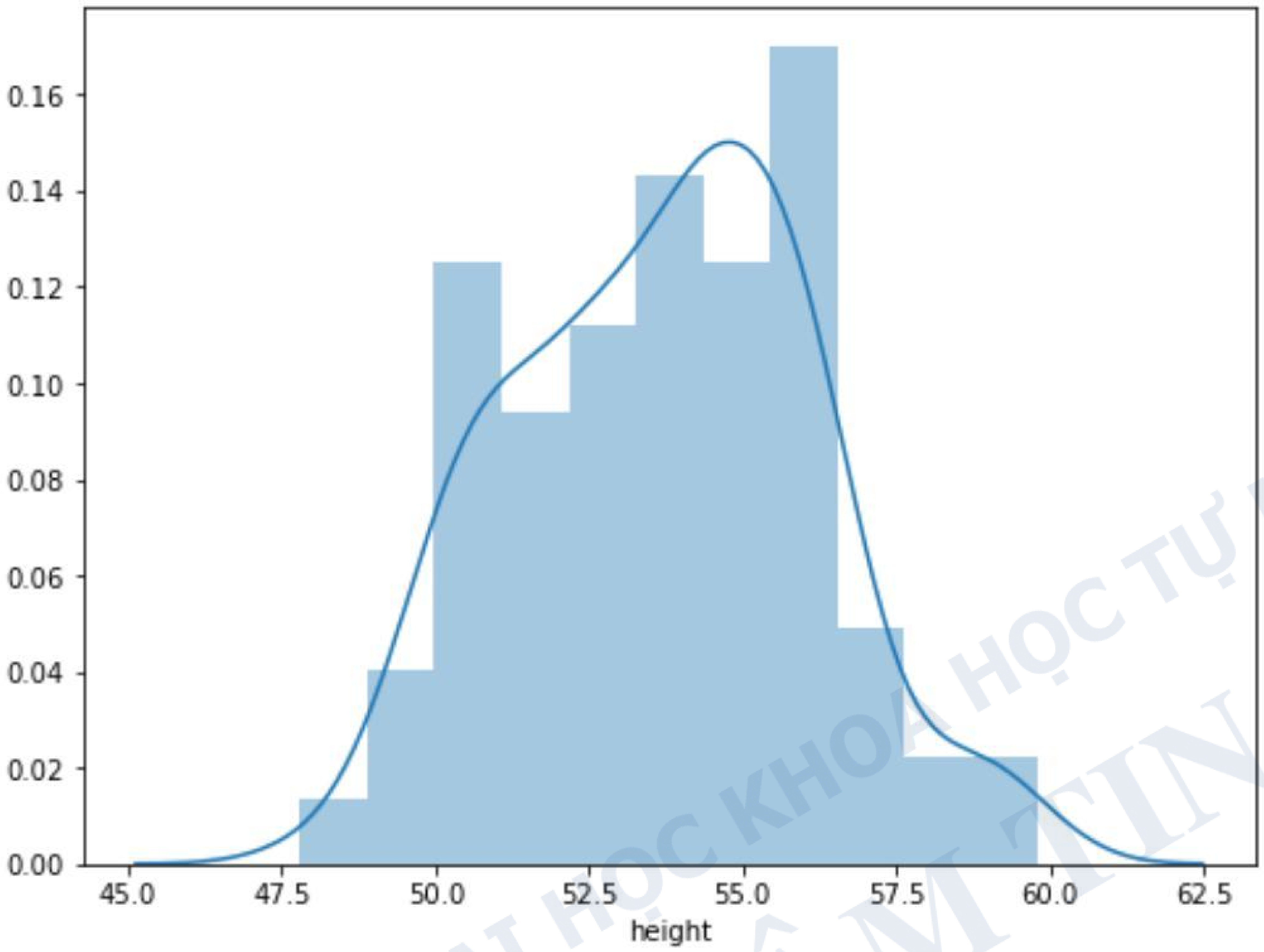
```
53.72 54.1 0    50.8
dtype: float64
```

In [ ]:

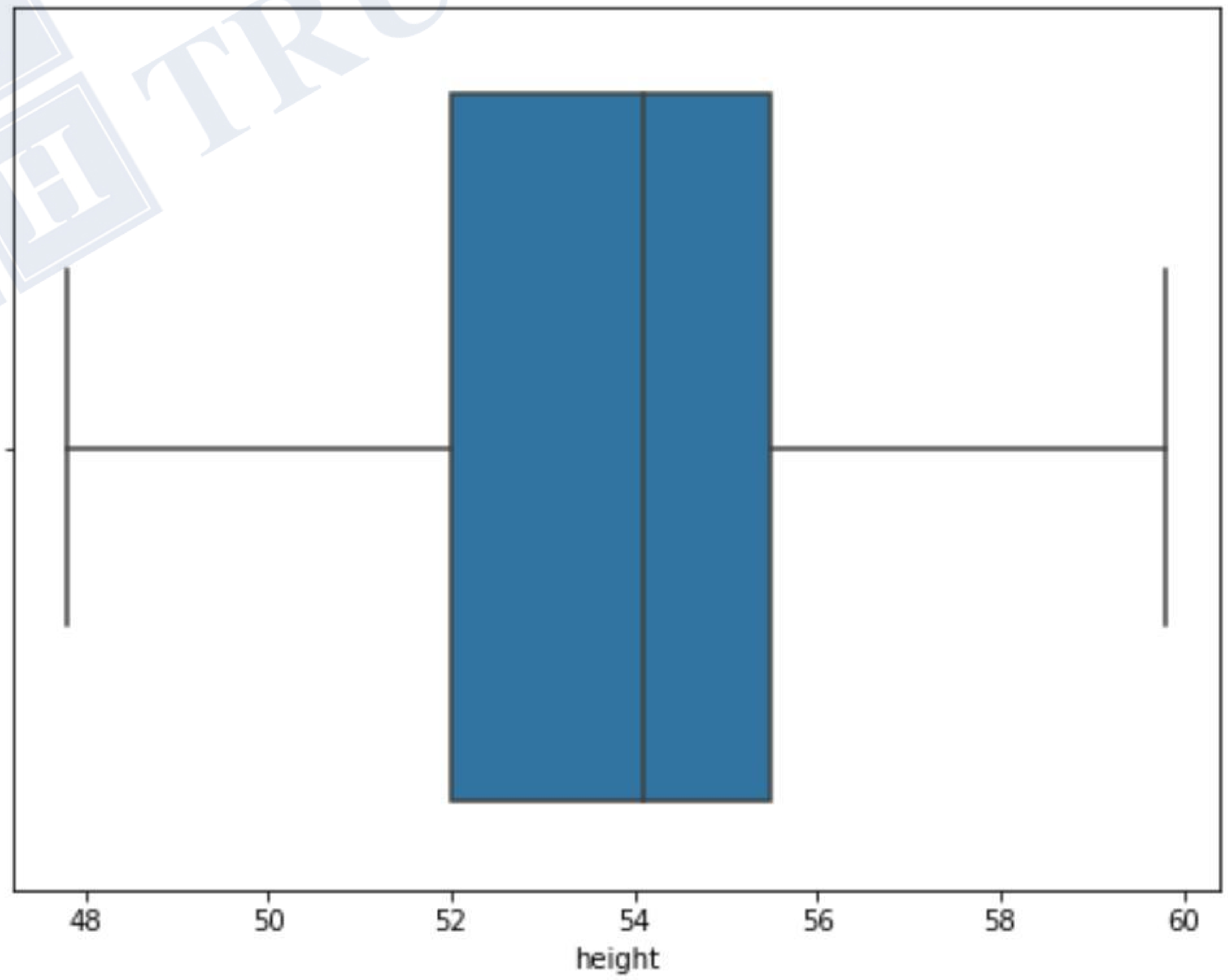
```
#calculate mean, median and mode of price
mean = df["price"].mean()
median =df["price"].median()
mode = df["price"].mode()
print(round(mean,2) , median, mode)
```

```
13150.31 10295.0 0    10295.0
dtype: float64
```

```
In [ ]: #6.
        plt.figure(figsize=(8,6))
        sns.distplot(df["height"])
        plt.show()
```



```
In [ ]: # boxplot for height
        plt.figure(figsize=(8,6))
        sns.boxplot(x="height",data=df)
        plt.show()
```



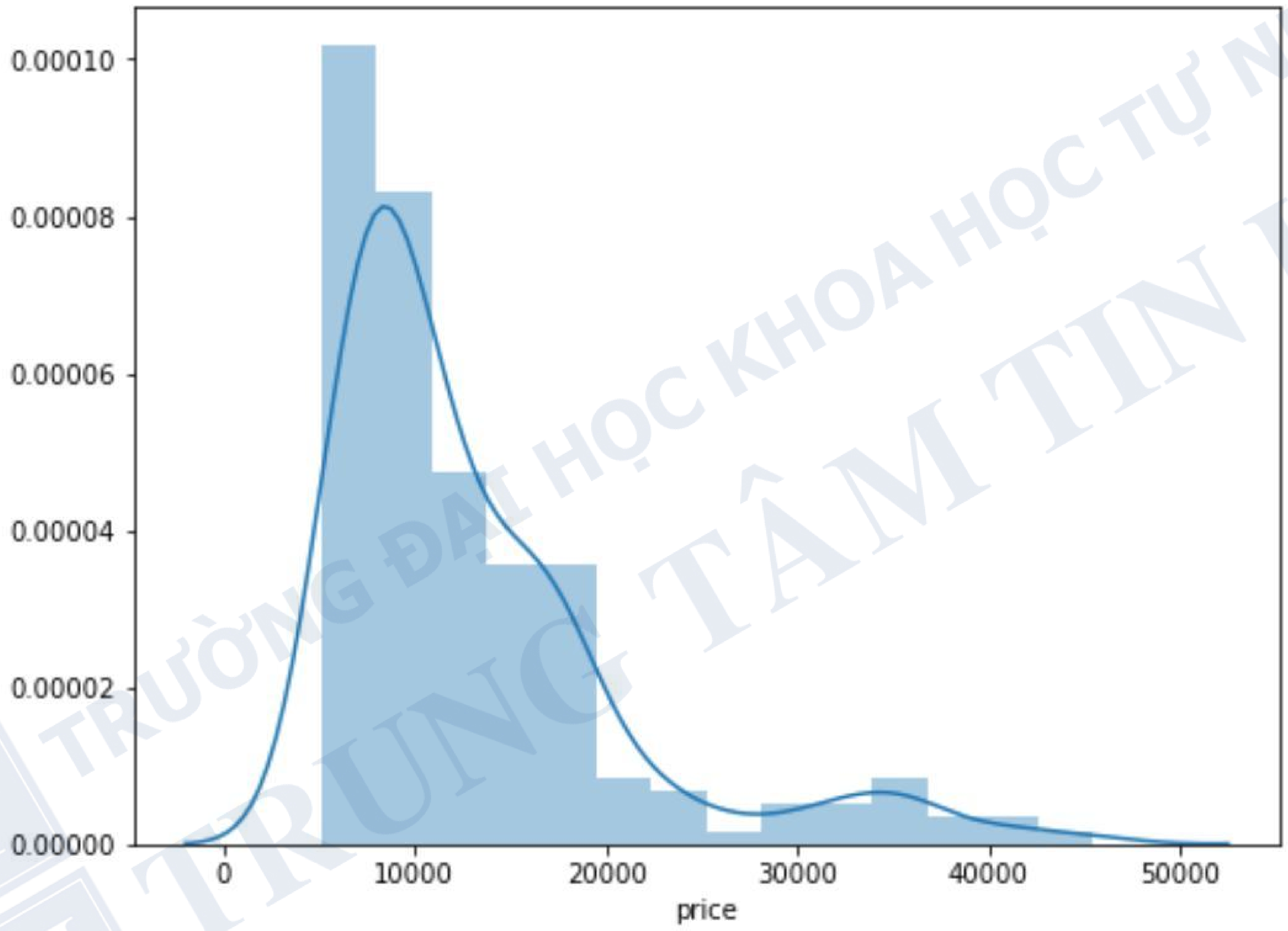
```
In [ ]: df["height"].skew()
```

```
Out[16]: 0.06312273247192804
```

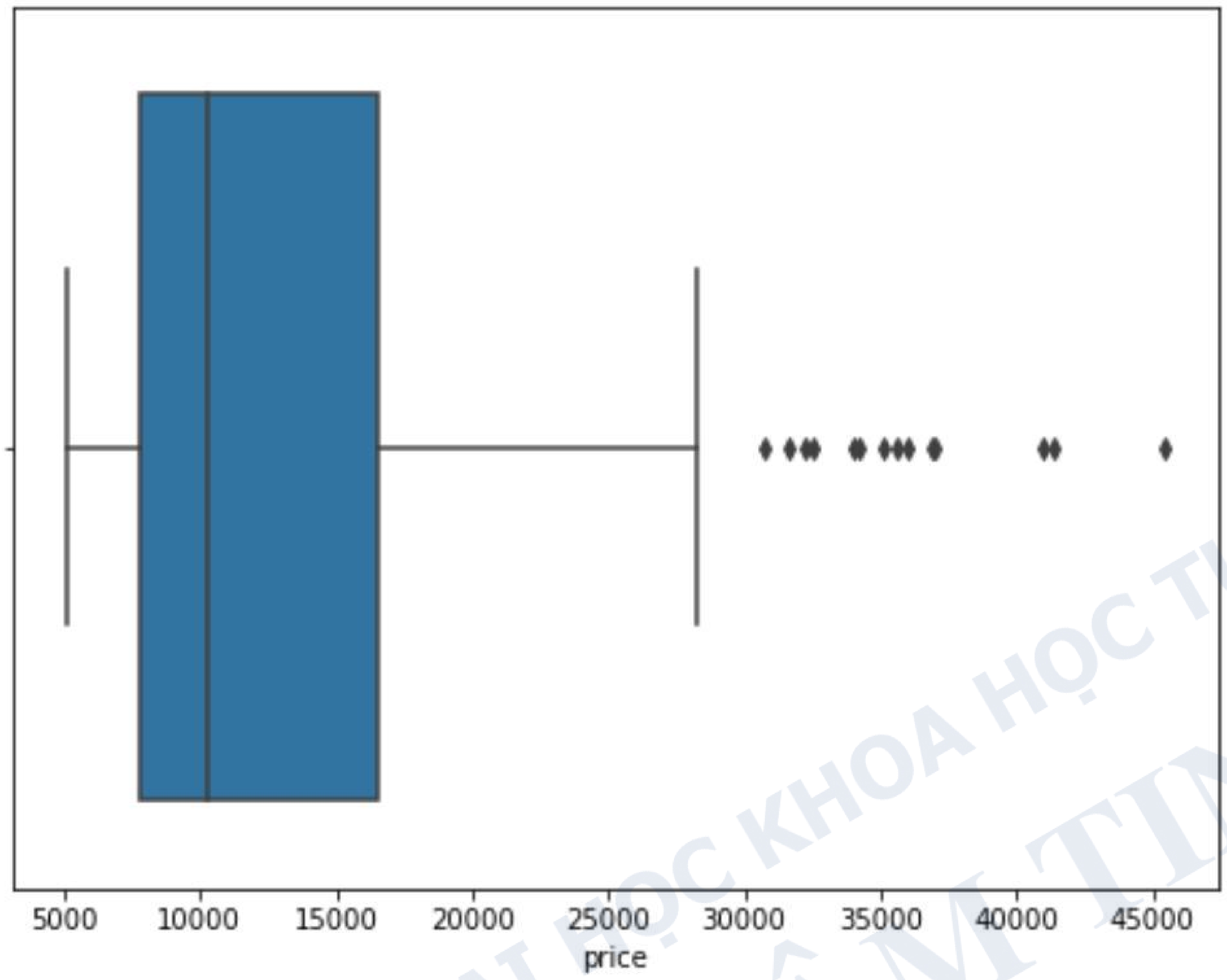
```
In [ ]: df["height"].kurtosis()
```

```
Out[17]: -0.4438123650575503
```

```
In [ ]: plt.figure(figsize=(8,6))
        sns.distplot(df["price"])
        plt.show()
```



```
In [ ]: # boxplot for price of cars
        plt.figure(figsize=(8,6))
        sns.boxplot(x="price",data=df)
        plt.show()
```



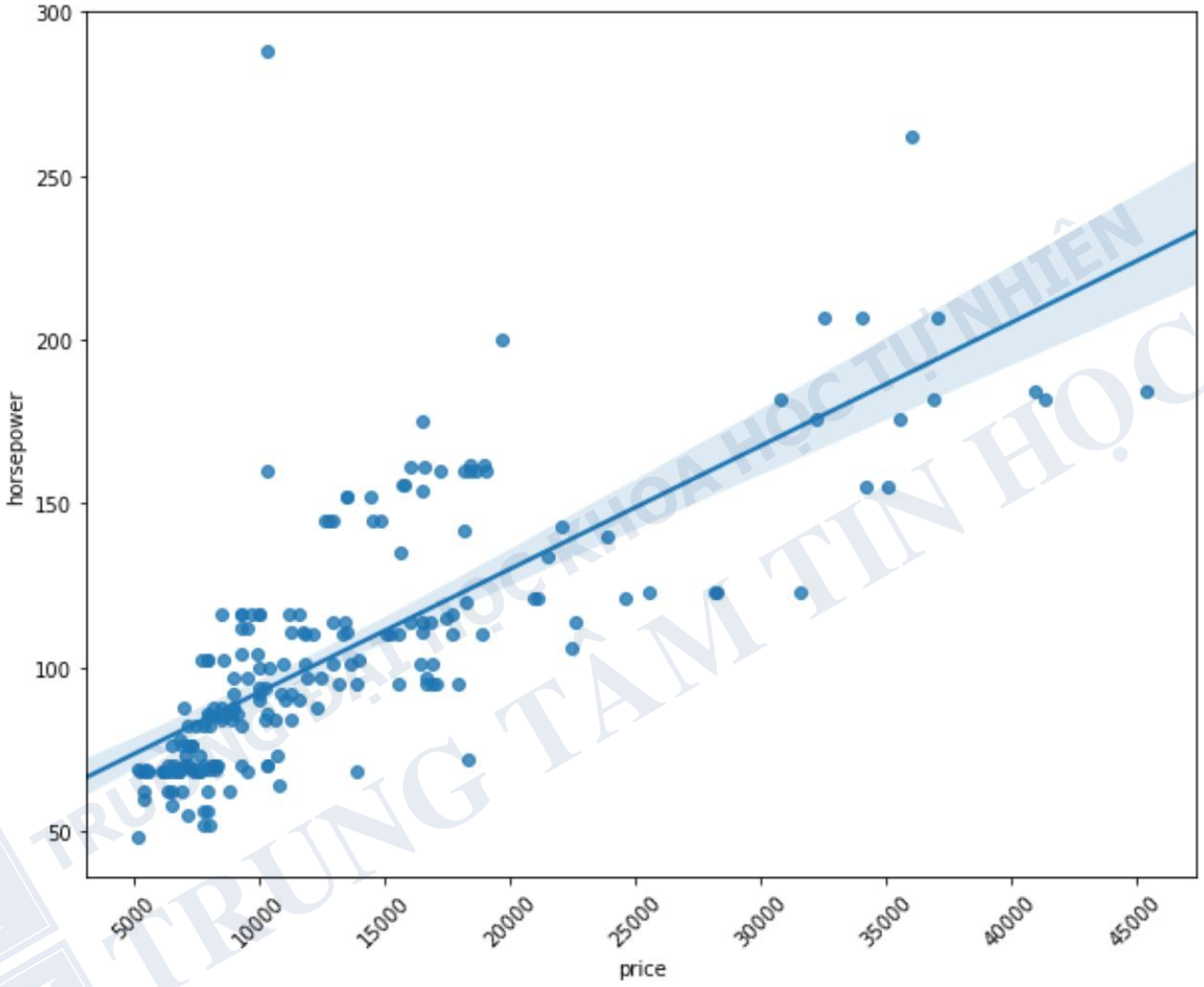
```
In [ ]: df["price"].skew()

Out[20]: 1.8409793088634683
```

```
In [ ]: df["price"].kurtosis()

Out[21]: 3.374863565224175
```

```
In [ ]: #7. Plot the relationship between 'horsepower' and 'price'
        plt.figure(figsize=(10,8))
        sns.regplot(x='price', y='horsepower', data=df)
        plt.xticks(rotation=45)
        plt.show()
```


horsepower
price