MOVIES

"Data analysis to movies's table "

CONTENT : OUR FAVOURITE COLOUR

what i tried to do ??

1- STAGE OF UPLOAD THE DATA

stages of choosing the code runner:

1. use pycharm — couldn't upload pandas
2. vscode — so complicated
3. vscodiom — ugly windows colors
4. jupyter — don't help in writing

IMPORTING CODE

[1]:
```
import pandas as pd
```

[3]:
```
import numpy as np
```

[317]:
```
x=pd.read_csv("project films.csv")
x
```

2- DATA CLEANING :

1

[333]:
```
x.isnull().sum()
x.head()
x.isnull().sum()
x.dropna(inplace=True)
x.isnull().sum()
x.dropna(how='any',subset=['genres','director'],inplace=True)
x.isnull().sum()
x
```

2

[43]:
```
xx = x.drop(columns=['release_year', 'release_date', 'day'])
xx
```

3

[331]:
```
x.dtypes
```

DATE SPLITING:

[347]:
```
x['release_date'] = pd.to_datetime(x['release_date'], format='%m/%d/%y')

# extract the day, month, and year components
x['day'] = x['release_date'].dt.day
x['month'] =x['release_date'].dt.month
x['year'] = x['release_date'].dt.year
x

# show the modified data frame
```

4

[343]:
```
x['month'] = x['release_date'].dt.month
x
```

5

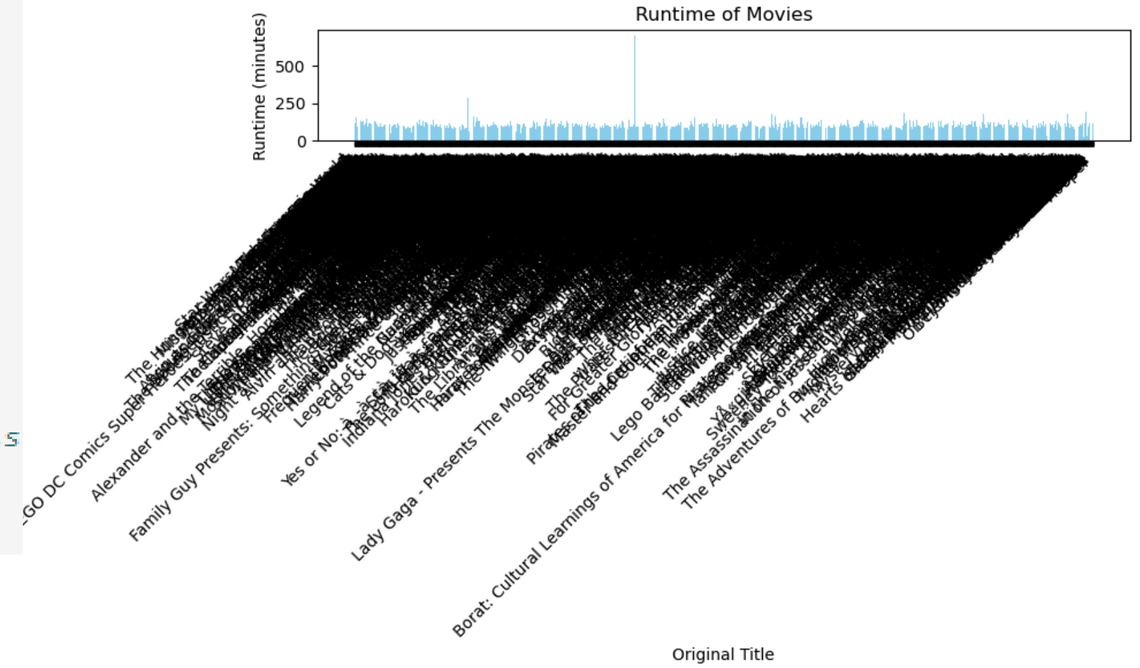
3-THE FAILED QUESTIONS:

The names of movies and it's runtime

[357]:

```
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))  # Set the figure size
plt.bar(x['original_title'], x['runtime'], color='skyblue')  # Create a bar plot

# Add title and labels
plt.title('Runtime of Movies')
plt.xlabel('Original Title')
plt.ylabel('Runtime (minutes)')

# Rotate x-axis labels for better readability
plt.xticks(rotation=45, ha='right')

# Show the plot
plt.tight_layout()  # Adjust layout to make room for the labels
plt.show()
```

1



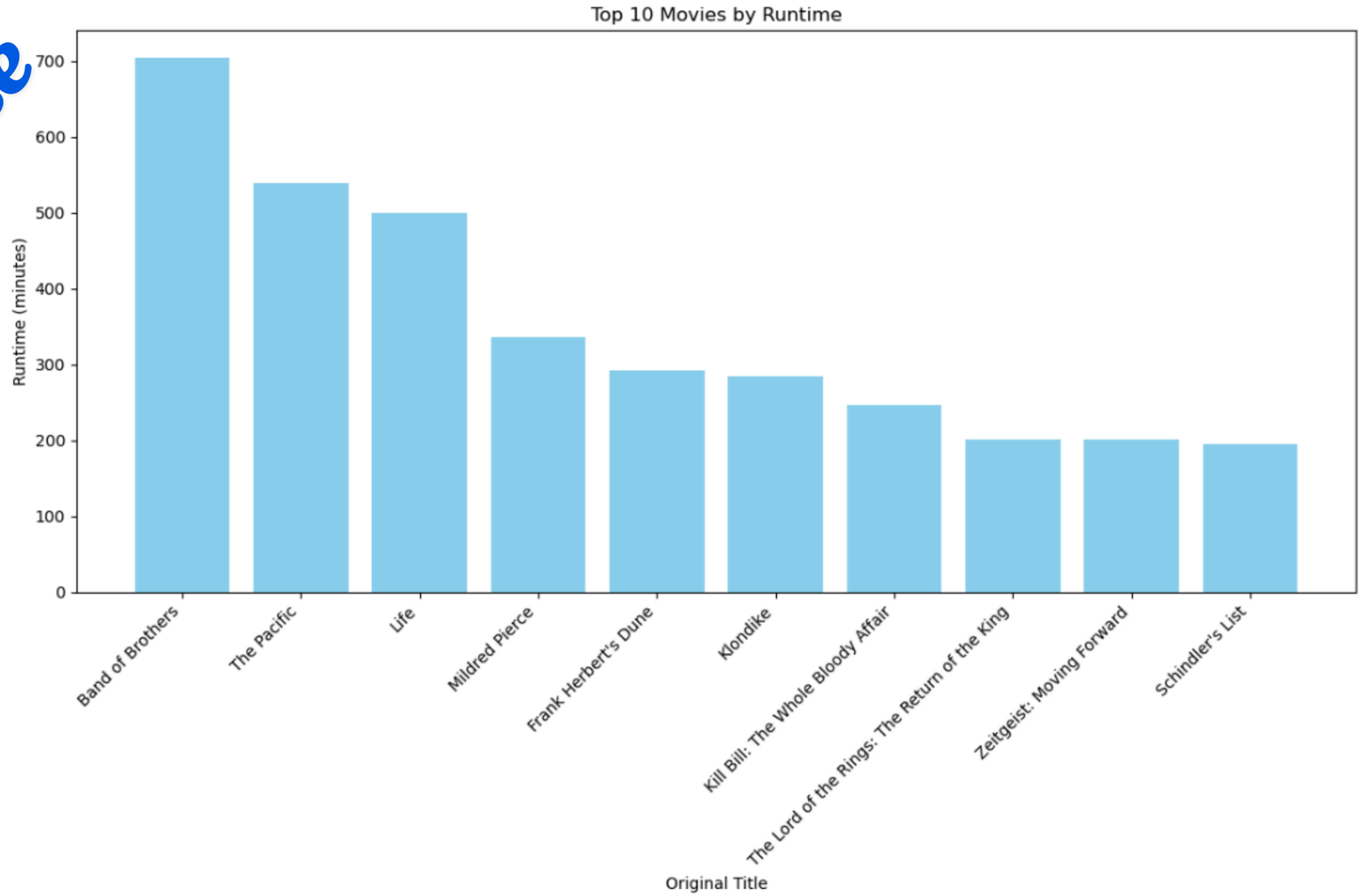
SAME BUT TOP 10 MOVIES :

2

[361]:
```
import pandas as pd
import matplotlib.pyplot as plt

top_10_movies = x.sort_values(by='runtime', ascending=False).head(10)


# Create a bar plot for the top 10 movies
plt.figure(figsize=(12, 8))  # Set the figure size
plt.bar(top_10_movies['original_title'], top_10_movies['runtime'], color='skyblue')  # Create a bar plot

# Add title and labels
plt.title('Top 10 Movies by Runtime')
plt.xlabel('Original Title')
plt.ylabel('Runtime (minutes)')

# Rotate x-axis labels for better readability
plt.xticks(rotation=45, ha='right')

# Show the plot
plt.tight_layout()  # Adjust layout to make room for the labels
plt.show()
```

i know there is no purpose

PICTURE OF GRAPH

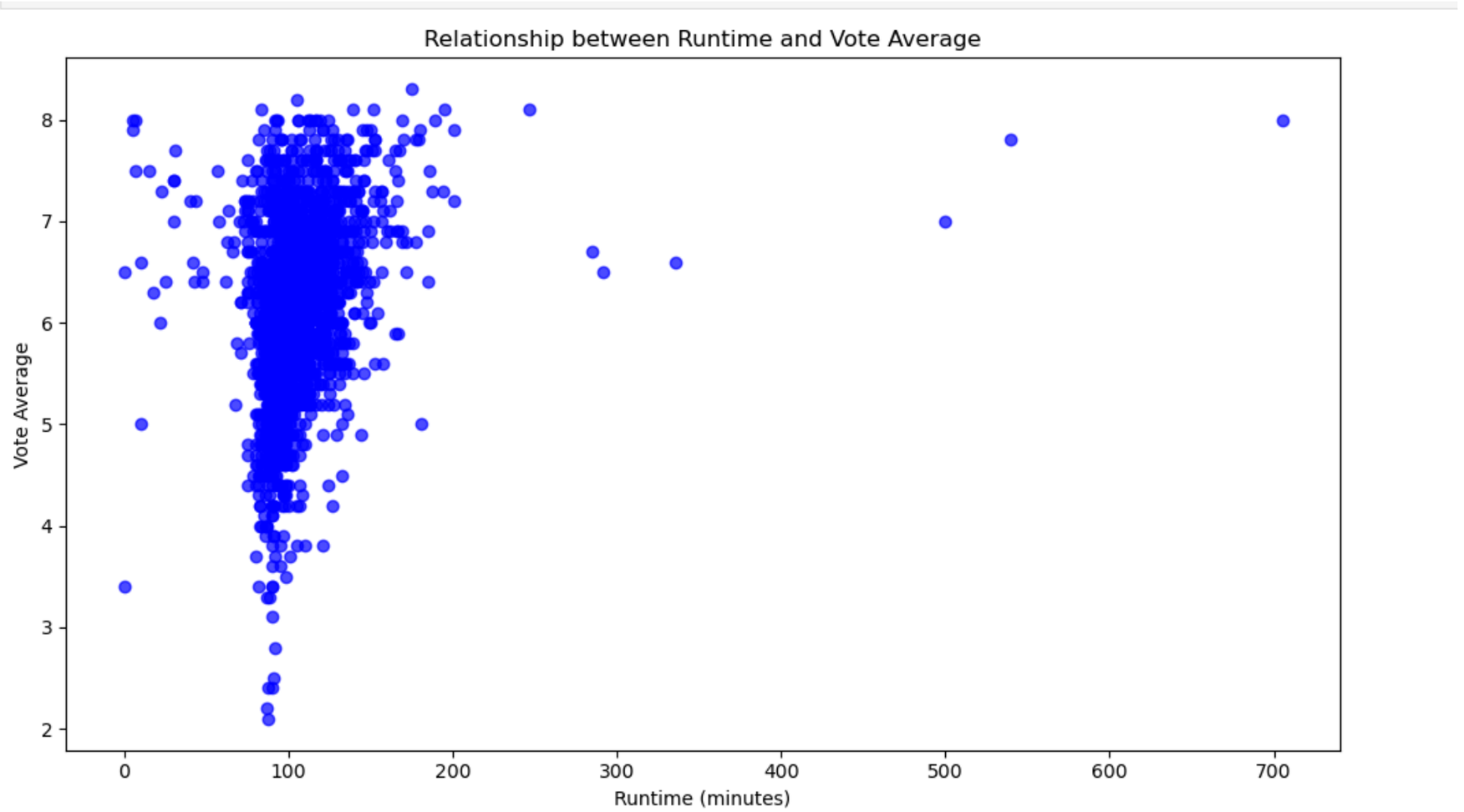


4- THE RELATION BETWEEN REVIEWS AND RUNTIME

[379]:

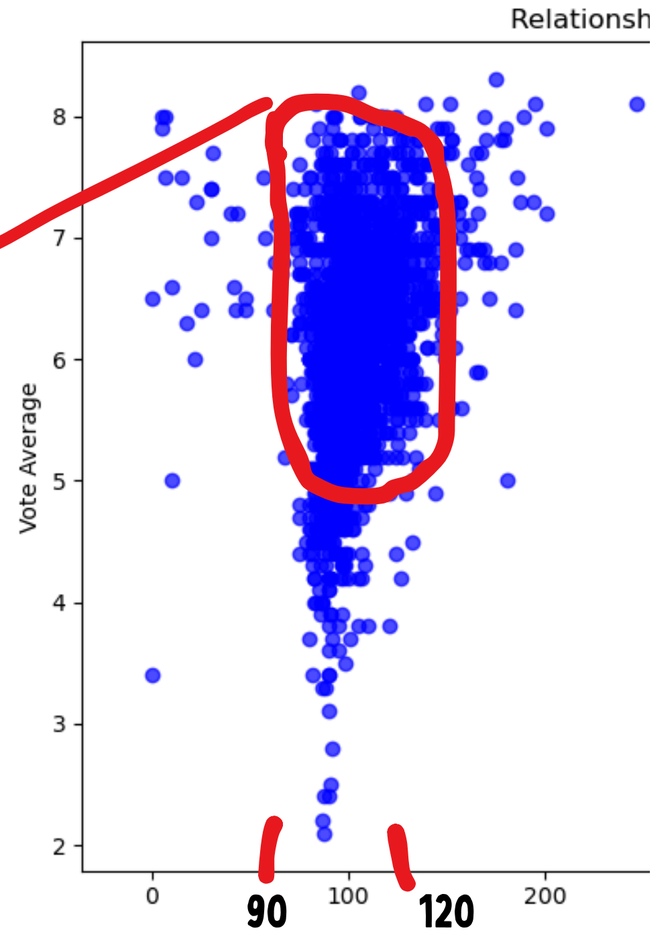
```
# Create a scatter plot
plt.figure(figsize=(10, 6))  # Set the figure size
plt.scatter(x['runtime'], x['vote_average'], color='blue', alpha=0.7)  # Create a scatter plot

# Add title and labels
plt.title('Relationship between Runtime and Vote Average')
plt.xlabel('Runtime (minutes)')
plt.ylabel('Vote Average')

# Show the plot
plt.tight_layout()  # Adjust layout to make room for the labels
plt.show()
```

IT`S PLOT:


Relationship between Runtime and Vote Average
Vote Average
Runtime (minutes)
2
3
4
5
6
7
8
0
100
200
300
400
500
600
700

THE OUT COME:

more the movie between
90 and 120 min
gives more good reviews

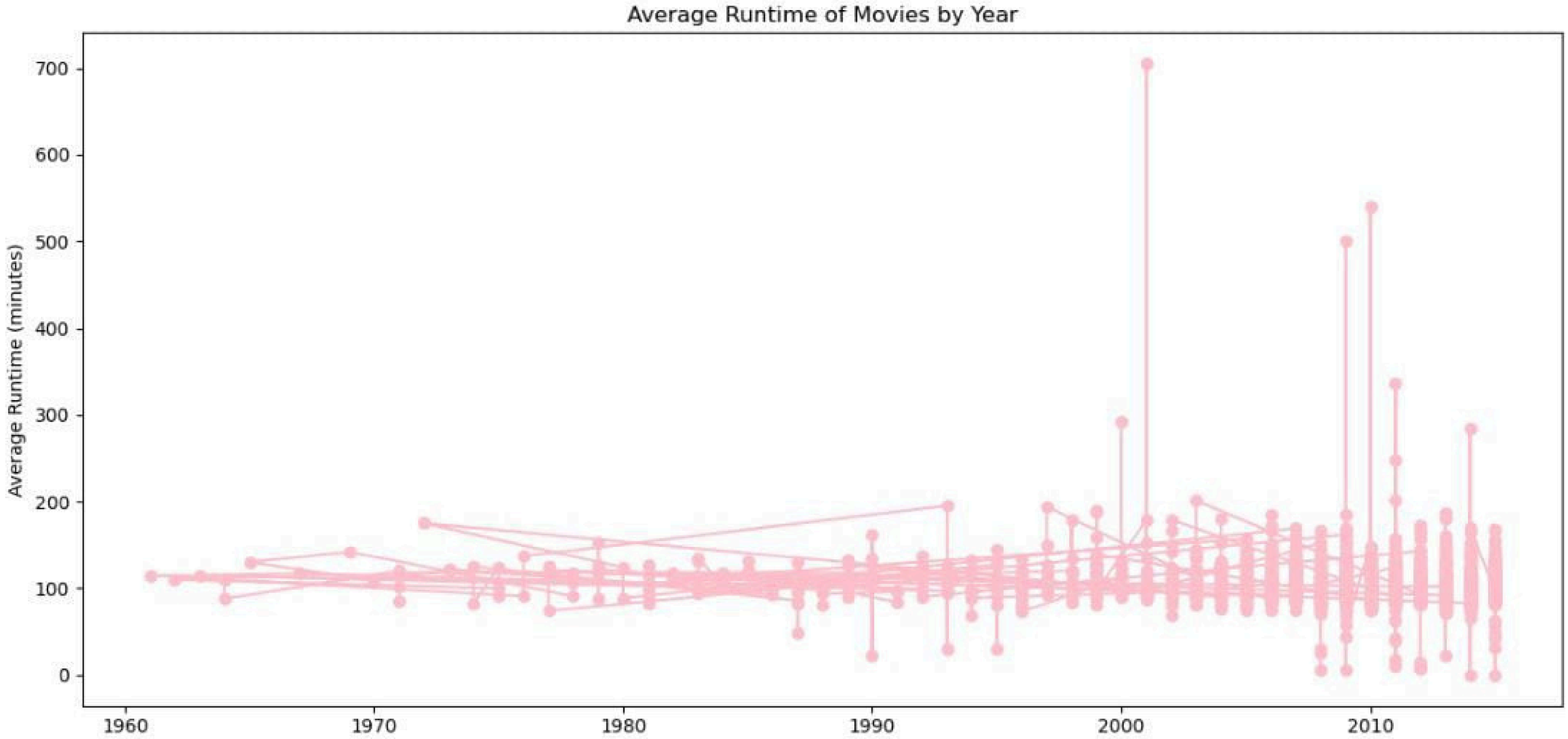


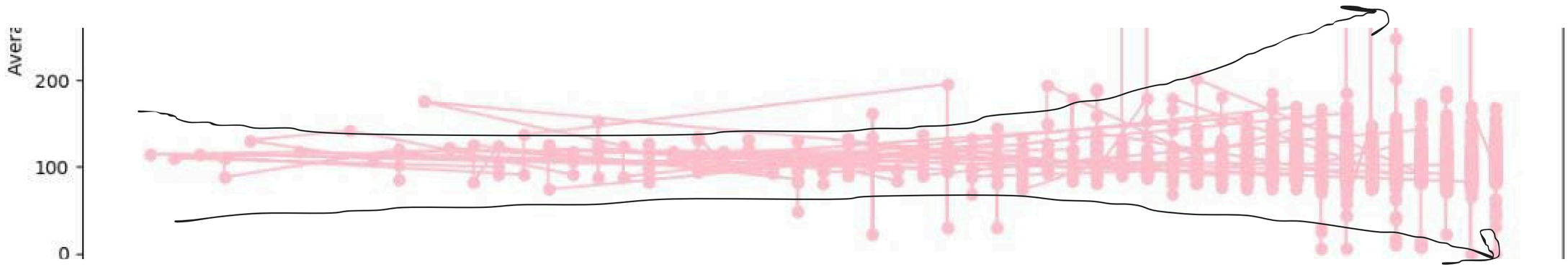
5-HOW RUNTIME CHANGES BY YEARS :

377]:

```
# Create a line plot
plt.figure(figsize=(12, 6))  # Set the figure size
plt.plot(xx['year'], xx['runtime'], marker='o', color='pink')  # Line plot

# Add title and labels
plt.title('Average Runtime of Movies by Year')
plt.xlabel('Year')
plt.ylabel('Average Runtime (minutes)')

# Show the plot
plt.tight_layout()  # Adjust layout to make room for the labels
plt.show()
```

IT`S PLOT:


Average Runtime of Movies by Year
Average Runtime (minutes)
700
600
500
400
300
200
100
0
1960
1970
1980
1990
2000
2010



The runtime kind of come longer

Could you Please give me your feedback

THANKS EVRYBODY