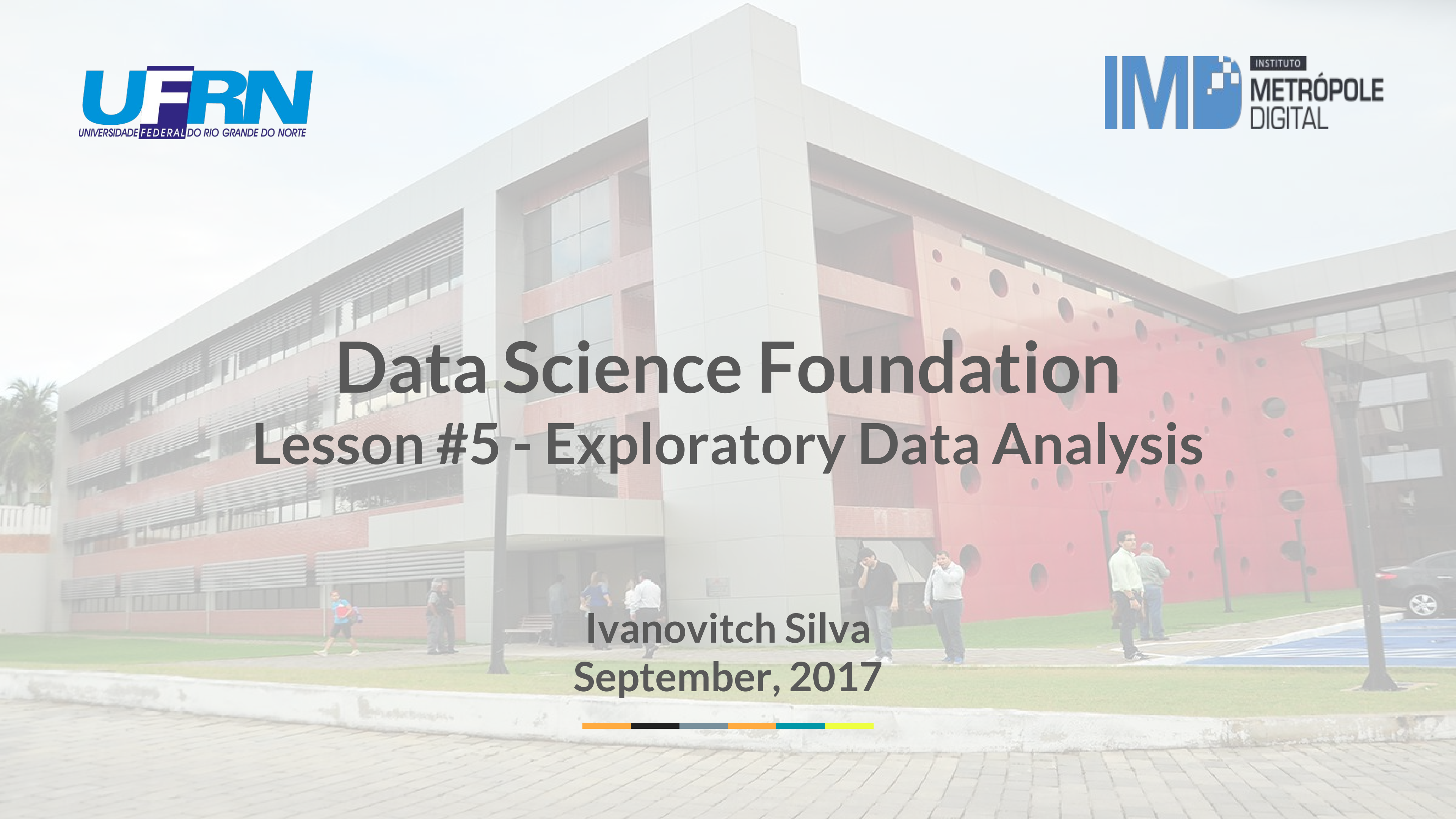


Data Science Foundation

Lesson #5 - Exploratory Data Analysis

Ivanovitch Silva

September, 2017

Agenda

- Case study: unemployment rate
- Tabular vs Visual representation
- Matplotlib
- Line plots
- Case study: movie ratings
- Bar & Scatter plots

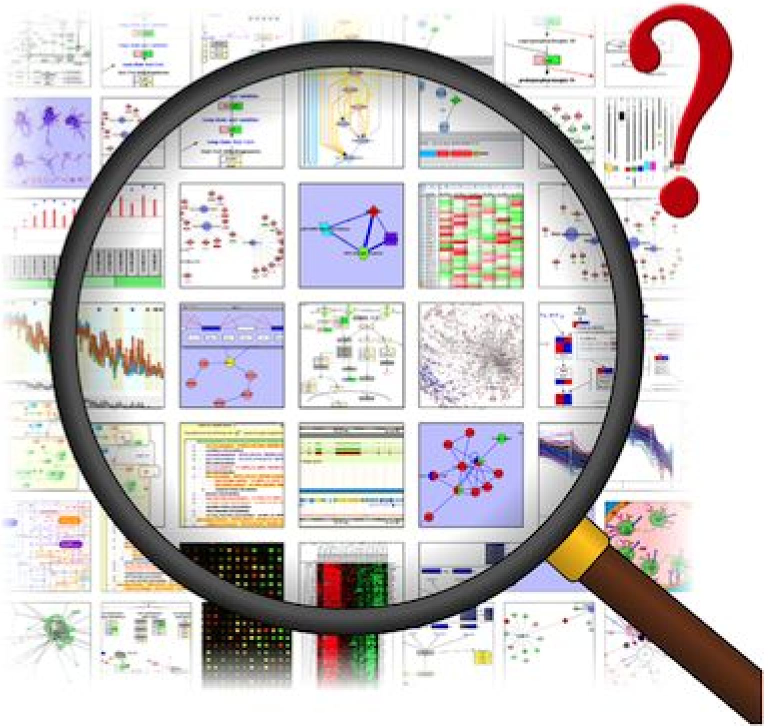
Update the repository

```
git clone https://github.com/ivanovitchm/EEC2006.git
```

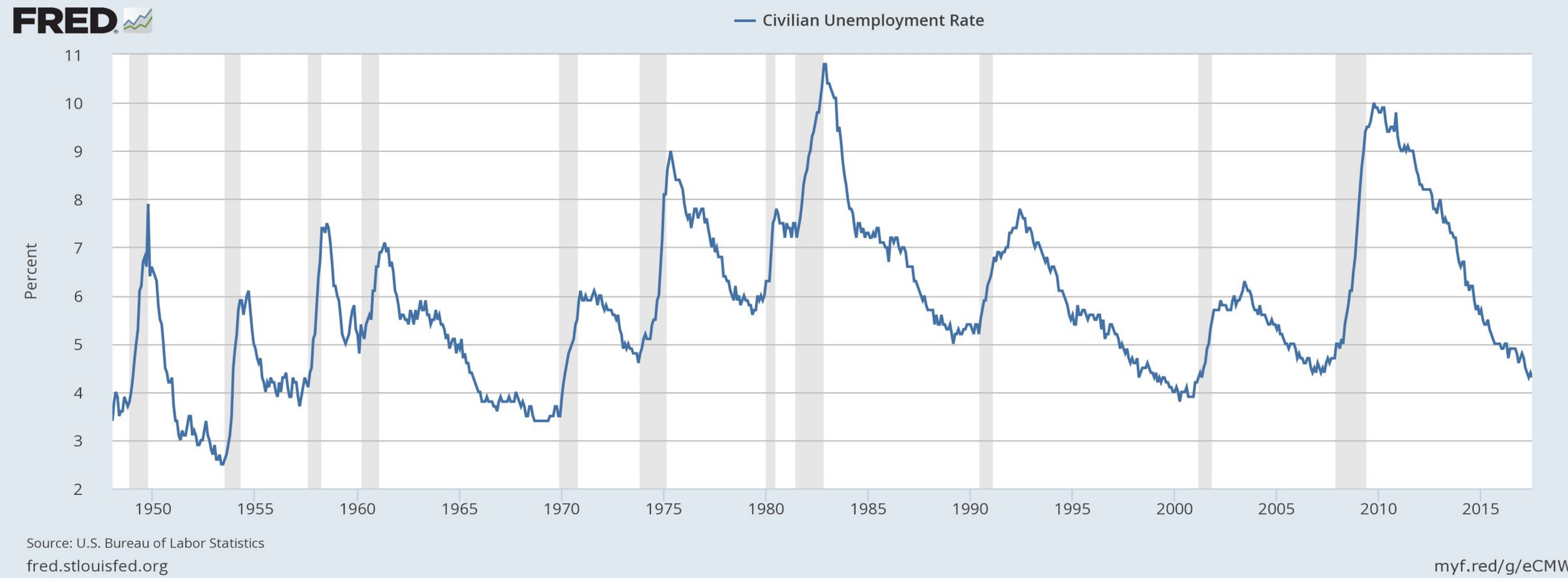
Or ….

```
git pull
```

Motivation

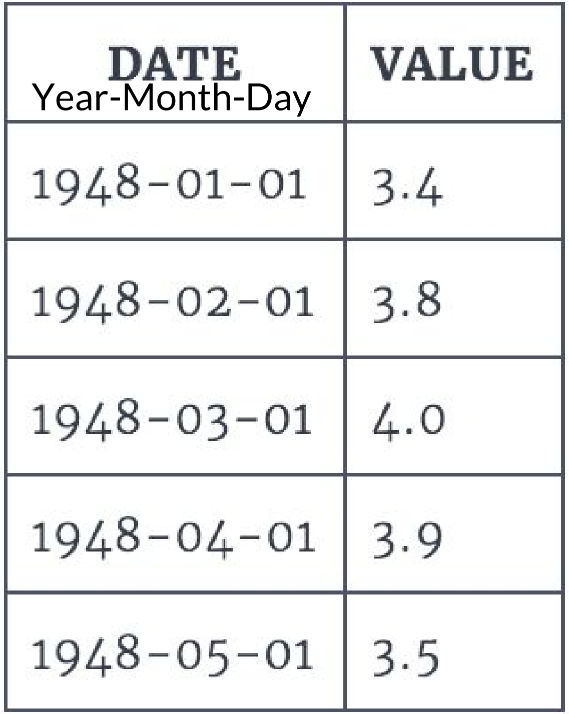


Case study: unemployment rate (US)



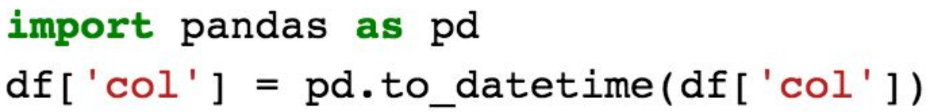
Source: U.S. Bureau of Labor Statistics

Investigating the dataset



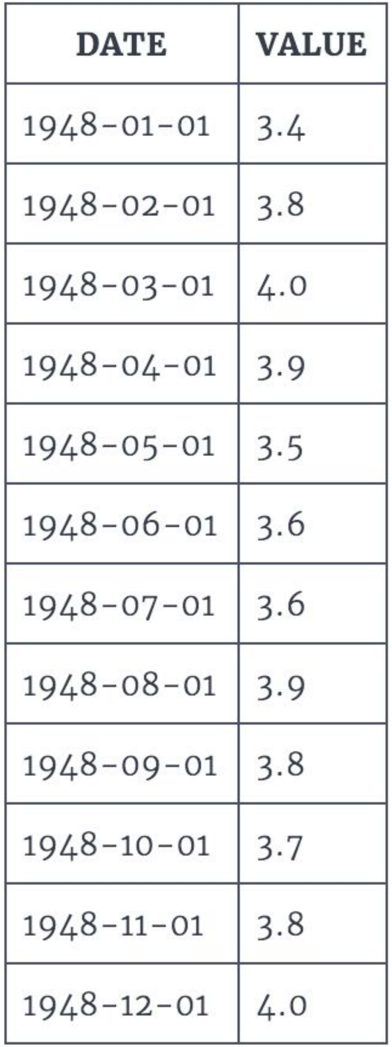
DATE Year-Month-Day	VALUE
1948-01-01	3.4
1948-02-01	3.8
1948-03-01	4.0
1948-04-01	3.9
1948-05-01	3.5

Conversion of types (Object to Datetime)



```
import pandas as pd
df['col'] = pd.to_datetime(df['col'])
```

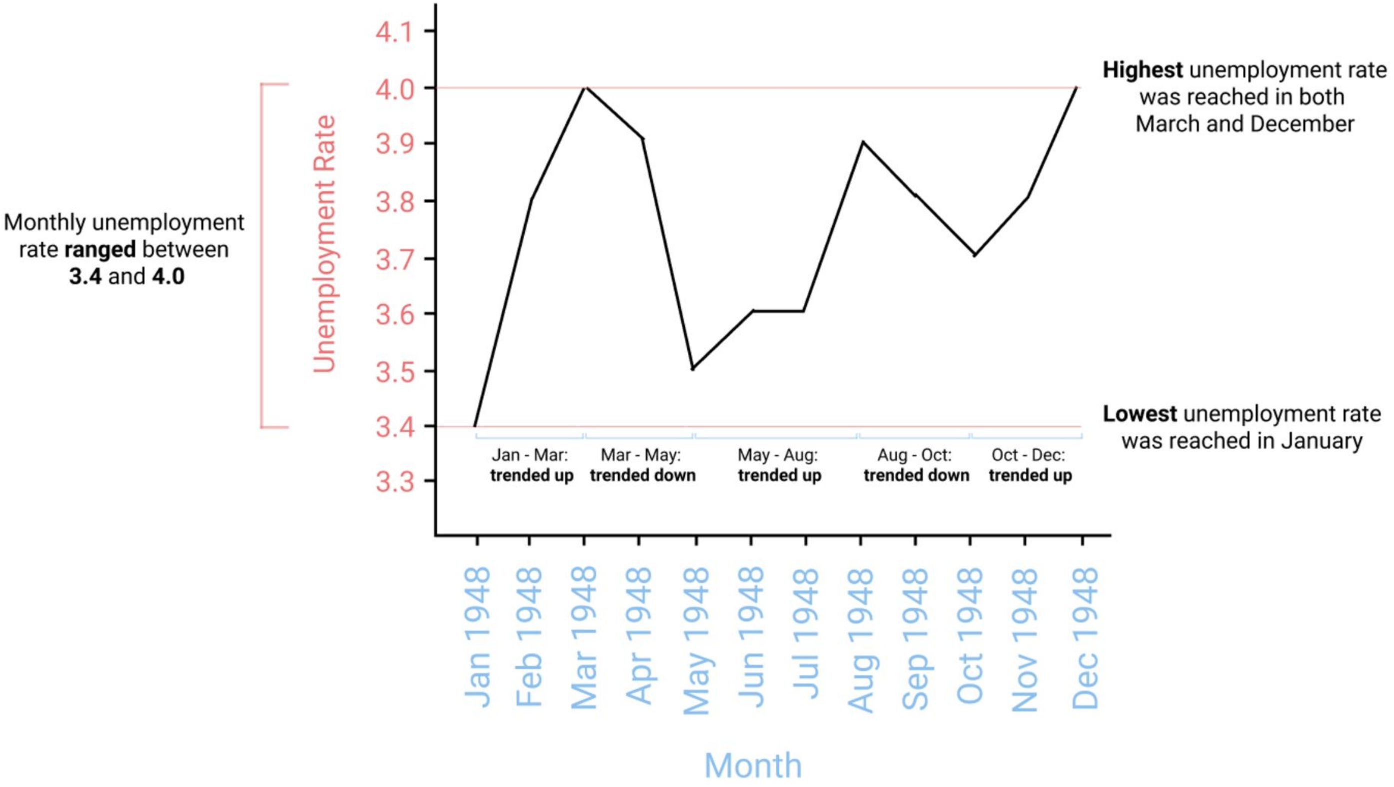
Observation from the table representation



DATE	VALUE
1948-01-01	3.4
1948-02-01	3.8
1948-03-01	4.0
1948-04-01	3.9
1948-05-01	3.5
1948-06-01	3.6
1948-07-01	3.6
1948-08-01	3.9
1948-09-01	3.8
1948-10-01	3.7
1948-11-01	3.8
1948-12-01	4.0

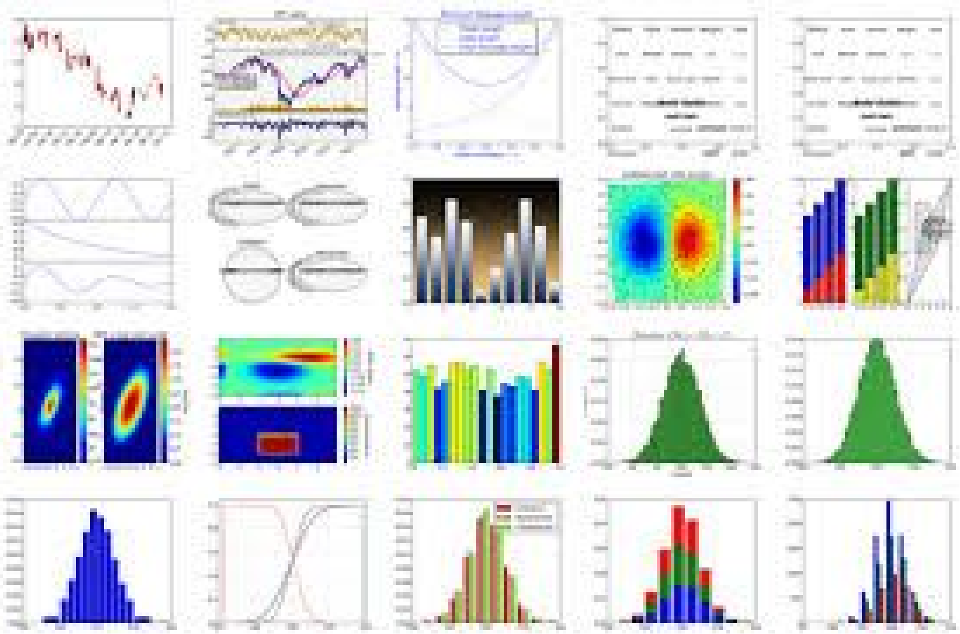
- What is the minimum value?
- What is the maximum value?
- Is there seasonality?
- What are the trend up periods?
- What are the trend down periods?
- Is the table representation really useful?

Visual representation



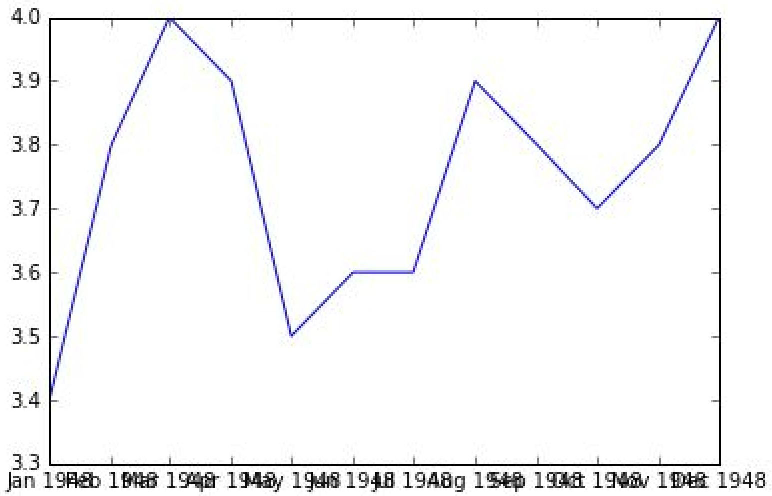
matplotlib

```
import matplotlib.pyplot as plt
plt.plot()
plt.show()
```

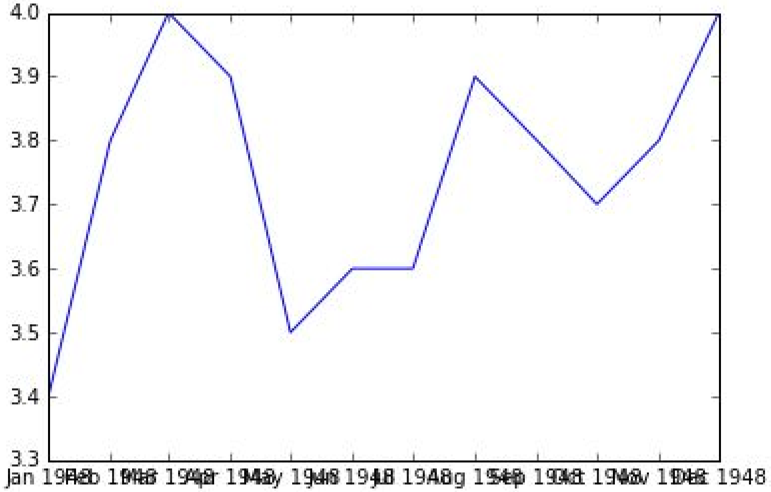


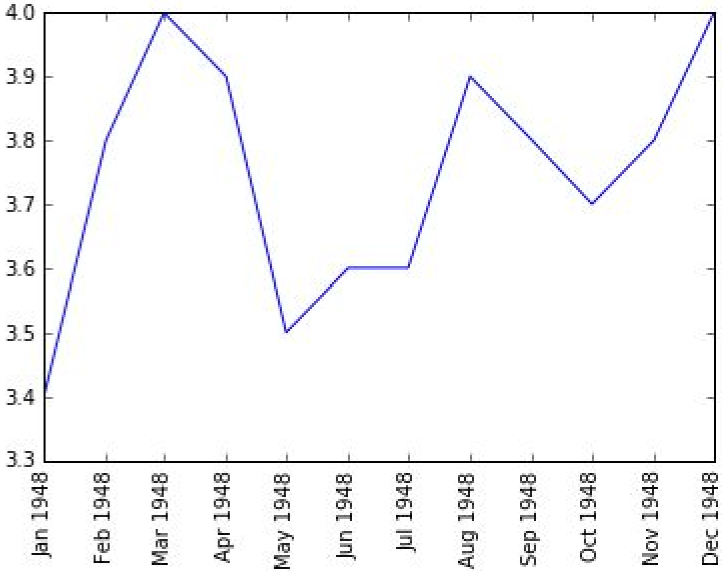
Adding data

```
plt.plot(x_values, y_values)
```



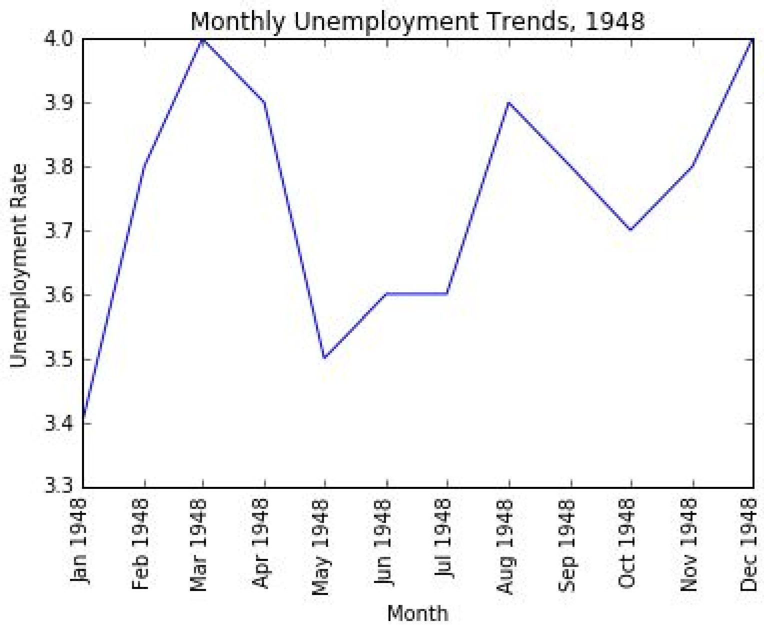
Fixing axis ticks





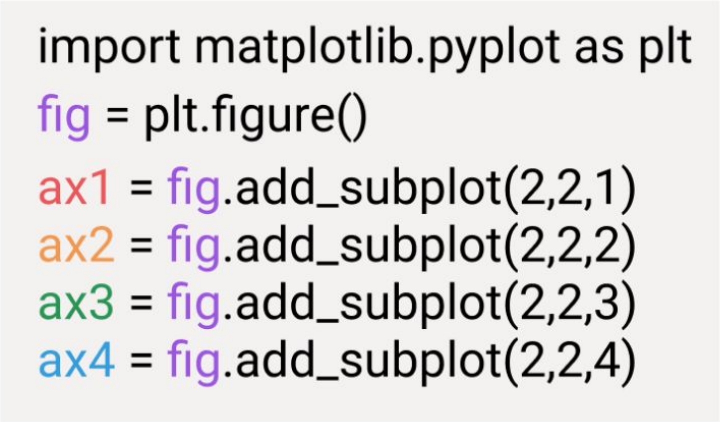
```
plt.xticks(rotation=90)
```

Additional information

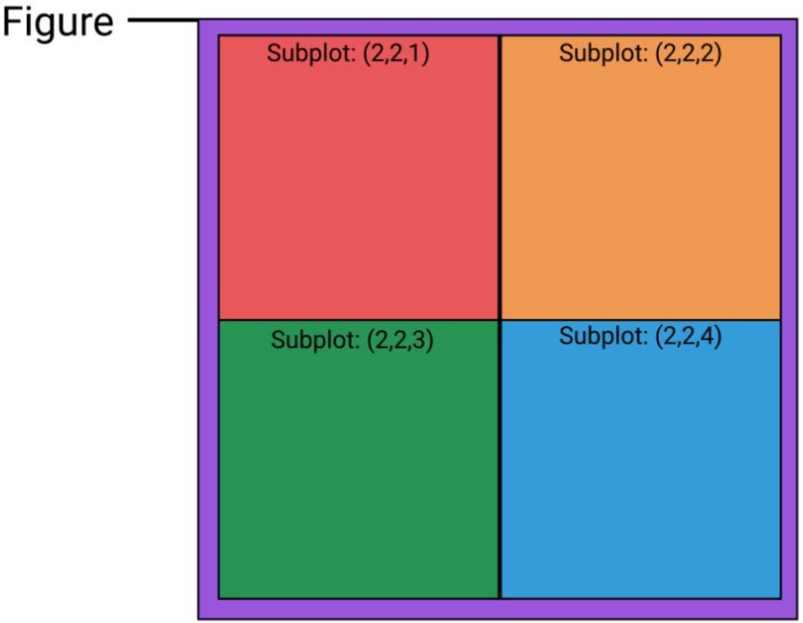


```
plt.xlabel("Month")
plt.ylabel("Unemployment Rate")
plt.title("Monthly Unemployment Trends, 1948")
```

Grid positioning



```
import matplotlib.pyplot as plt
fig = plt.figure()
ax1 = fig.add_subplot(2,2,1)
ax2 = fig.add_subplot(2,2,2)
ax3 = fig.add_subplot(2,2,3)
ax4 = fig.add_subplot(2,2,4)
```



Grid positioning

fig.add_subplot(4, 1, x)

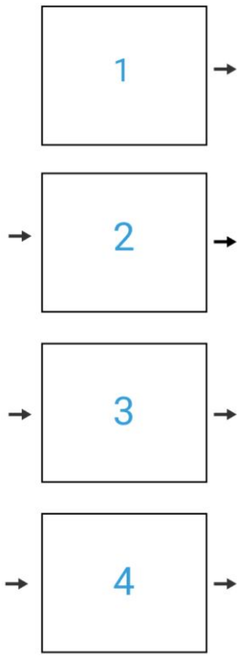


fig.add_subplot(2, 2, x)

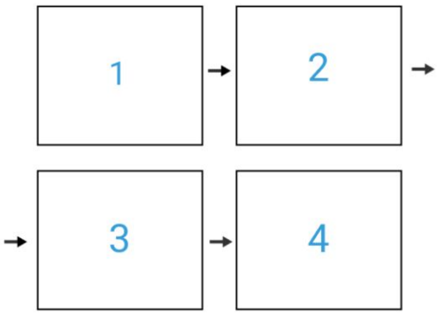
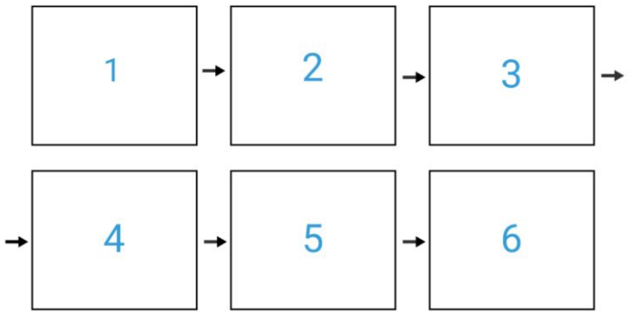
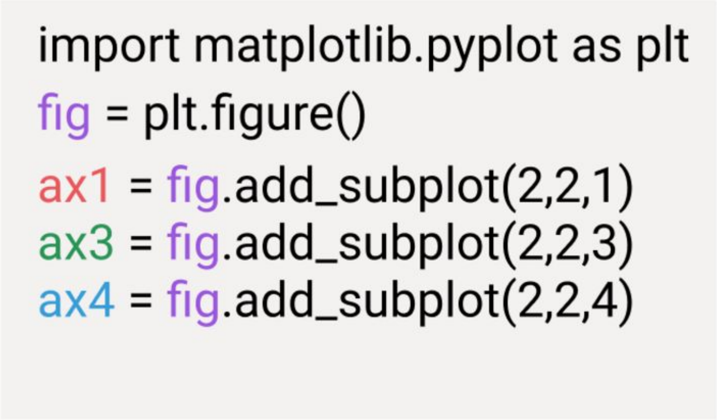


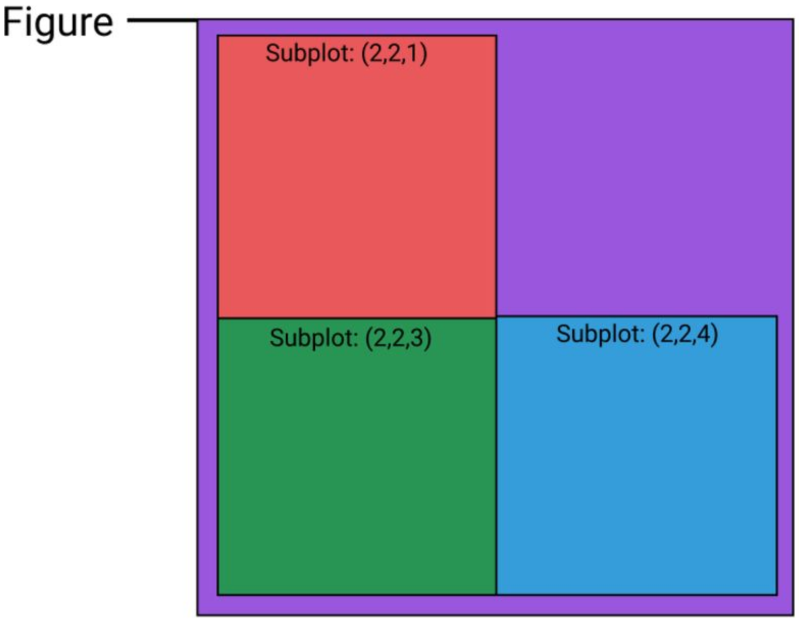
fig.add_subplot(2, 3, x)



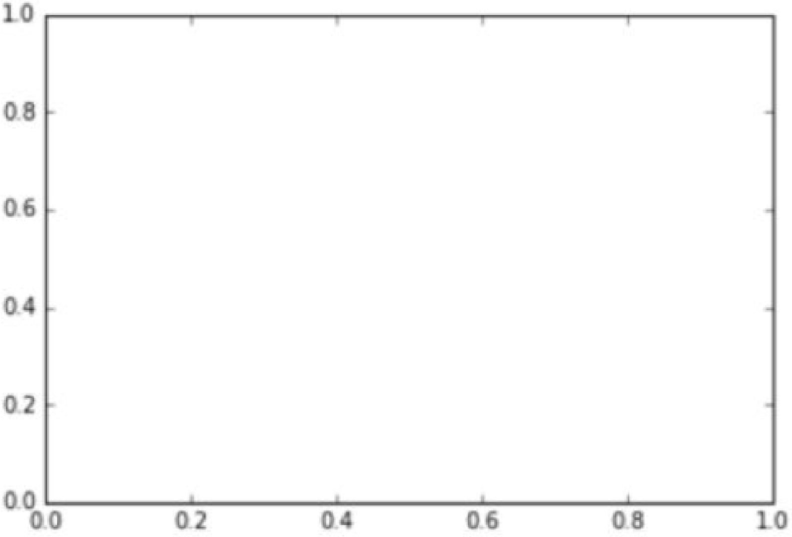
Grid positioning

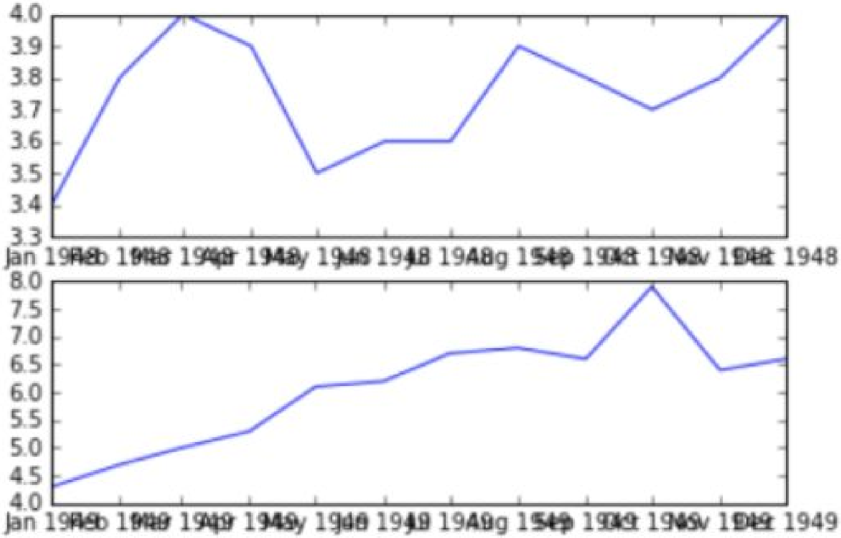


```
import matplotlib.pyplot as plt
fig = plt.figure()
ax1 = fig.add_subplot(2,2,1)
ax3 = fig.add_subplot(2,2,3)
ax4 = fig.add_subplot(2,2,4)
```



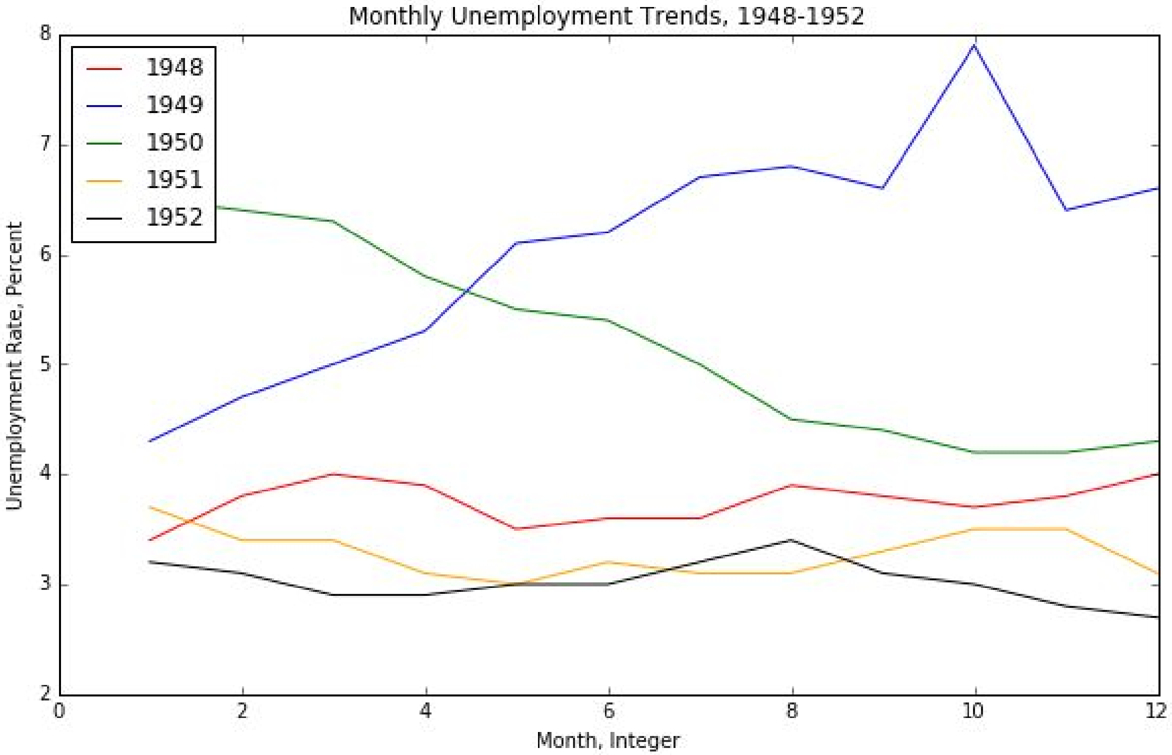
Formatting and spacing

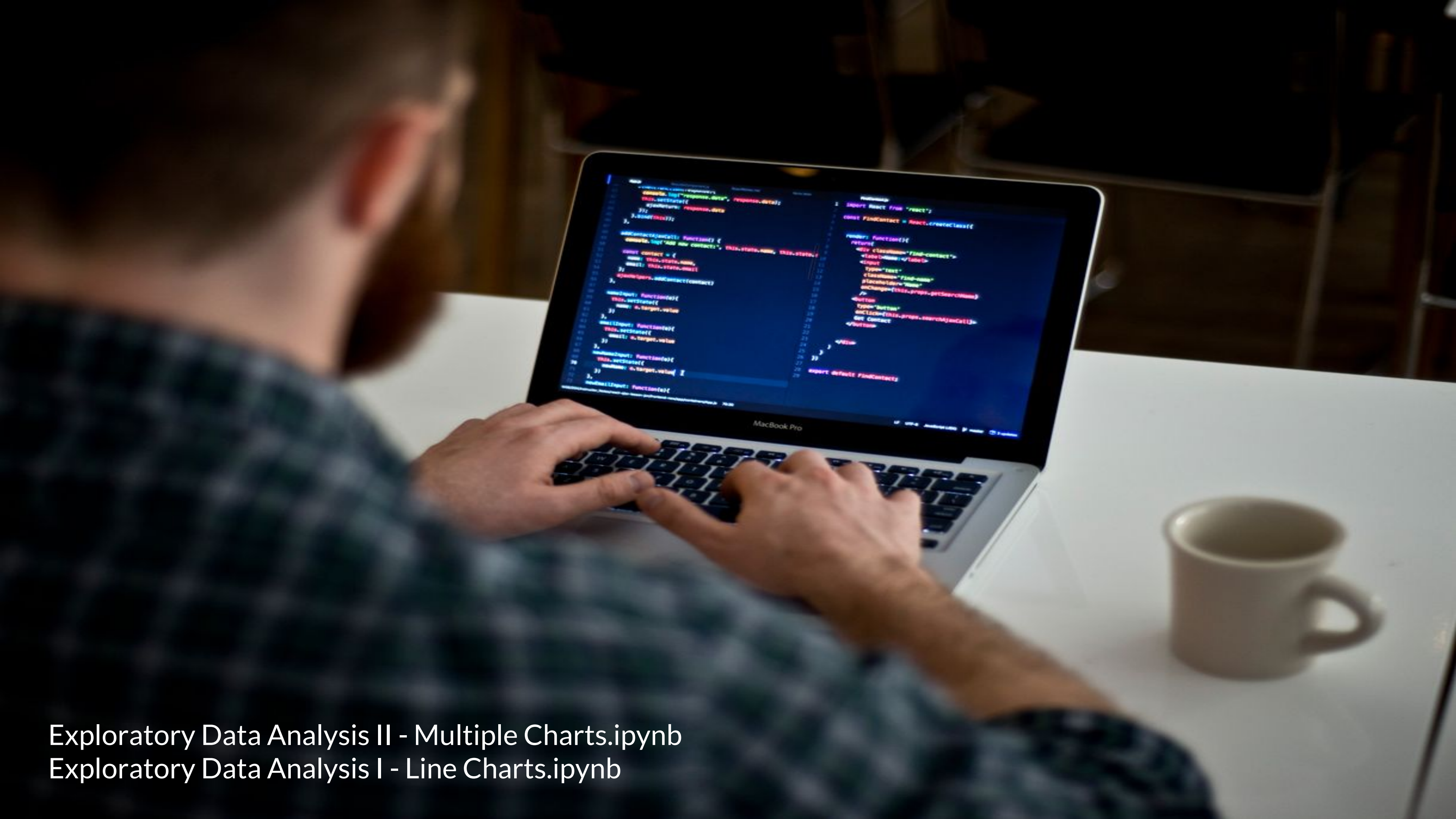




```
fig = plt.figure(figsize=(width, height))
```

Overlaying line charts

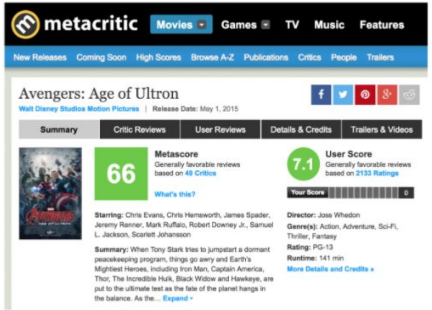




Exploratory Data Analysis II - Multiple Charts.ipynb
Exploratory Data Analysis I - Line Charts.ipynb

Case study: movie ratings

Metacritic



Avengers: Age of Ultron

Walt Disney Studios Motion Pictures | Release Date: May 1, 2015

66 **Metascore** Generally favorable reviews based on 49 Critics

7.1 **User Score** Generally favorable reviews based on 2133 Ratings

Starring: Chris Evans, Chris Hemsworth, James Spader, Jeremy Renner, Mark Ruffalo, Robert Downey Jr., Samuel L. Jackson, Scarlett Johansson

Summary: When Tony Stark tries to jumpstart a dormant peacekeeping program, things go awry and Earth's Mightiest Heroes, including Iron Man, Captain America, Thor, The Incredible Hulk, Black Widow and Hawkeye, are put to the ultimate test as the fate of the planet hangs in the balance. As the... Expand

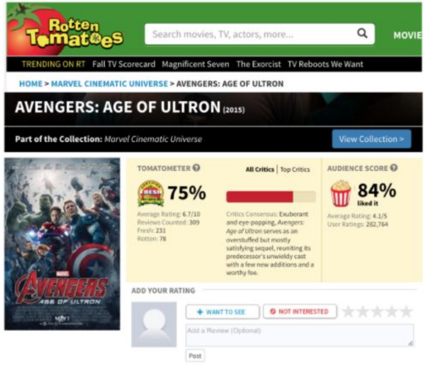
Director: Joss Whedon

Genre(s): Action, Adventure, Sci-Fi, Thriller, Fantasy

Rating: PG-13

Runtime: 141 min

Rotten Tomatoes



HOME > MARVEL CINEMATIC UNIVERSE > AVENGERS: AGE OF ULTRON

AVENGERS: AGE OF ULTRON (2015)

Part of the Collection: *Marvel Cinematic Universe*

TOMATOMETER 75%

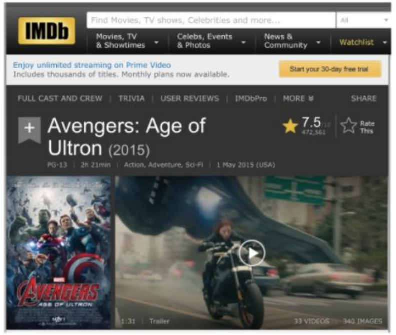
Average Rating: 6.7/10
Reviews Counted: 309
Fresh: 231
Rotten: 78

Critics Consensus: Exuberant and eye-popping, *Avengers: Age of Ultron* serves as an overstuffed but mostly satisfying sequel, reuniting its predecessor's unwieldy cast with a few new additions and a worthy foe.

AUDIENCE SCORE 84% liked it

Average Rating: 4.1/5
User Ratings: 282,764

IMDB

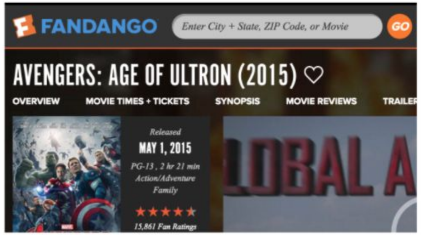


Enjoy unlimited streaming on Prime Video
Includes thousands of titles. Monthly plans now available.

Avengers: Age of Ultron (2015) 7.5/10 472,561

PG-13 | 2h 21min | Action, Adventure, Sci-Fi | 1 May 2015 (USA)

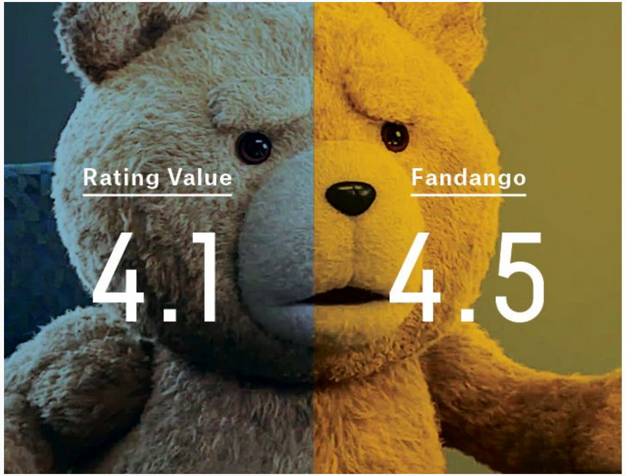
Fandango

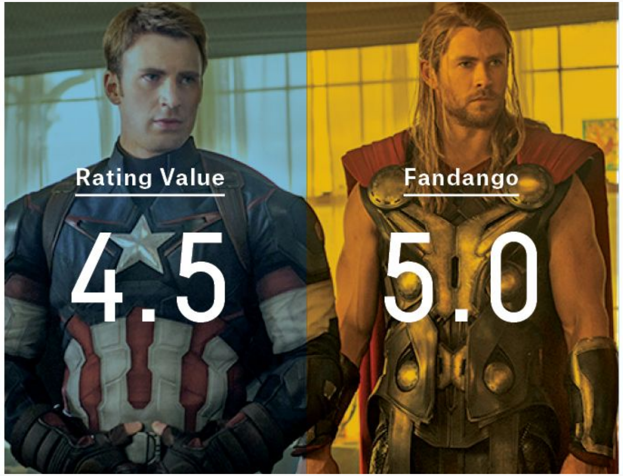


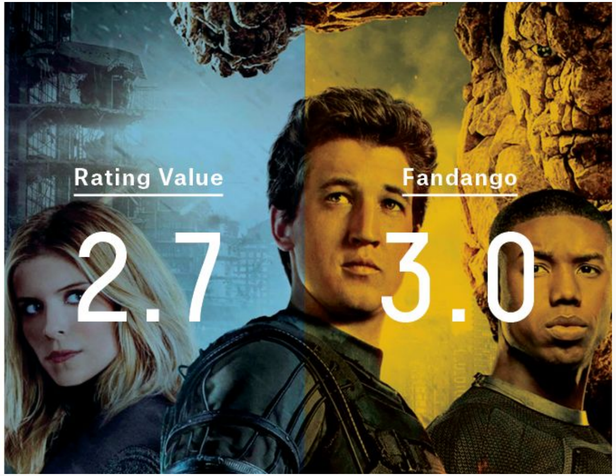
AVENGERS: AGE OF ULTRON (2015)

Released
MAY 1, 2015
PG-13 , 2 hr 21 min
Action/Adventure
Family
15,861 Fan Ratings

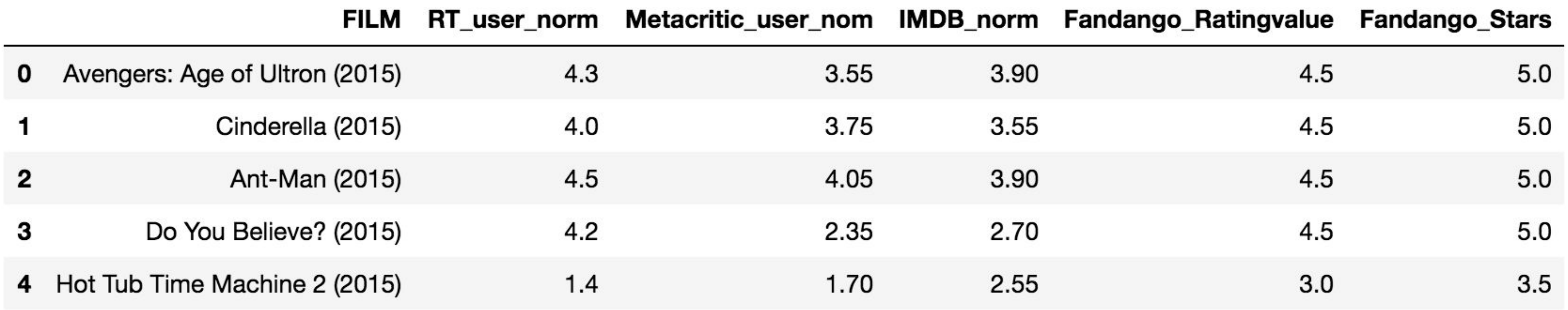
Bias in movie ratings







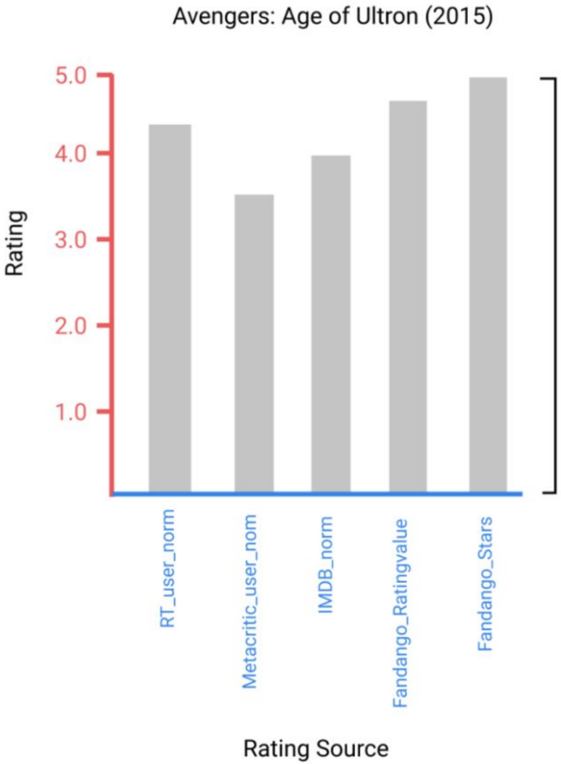
Introduction to the data



	FILM	RT_user_norm	Metacritic_user_nom	IMDB_norm	Fandango_Ratingvalue	Fandango_Stars
0	Avengers: Age of Ultron (2015)	4.3	3.55	3.90	4.5	5.0
1	Cinderella (2015)	4.0	3.75	3.55	4.5	5.0
2	Ant-Man (2015)	4.5	4.05	3.90	4.5	5.0
3	Do You Believe? (2015)	4.2	2.35	2.70	4.5	5.0
4	Hot Tub Time Machine 2 (2015)	1.4	1.70	2.55	3.0	3.5

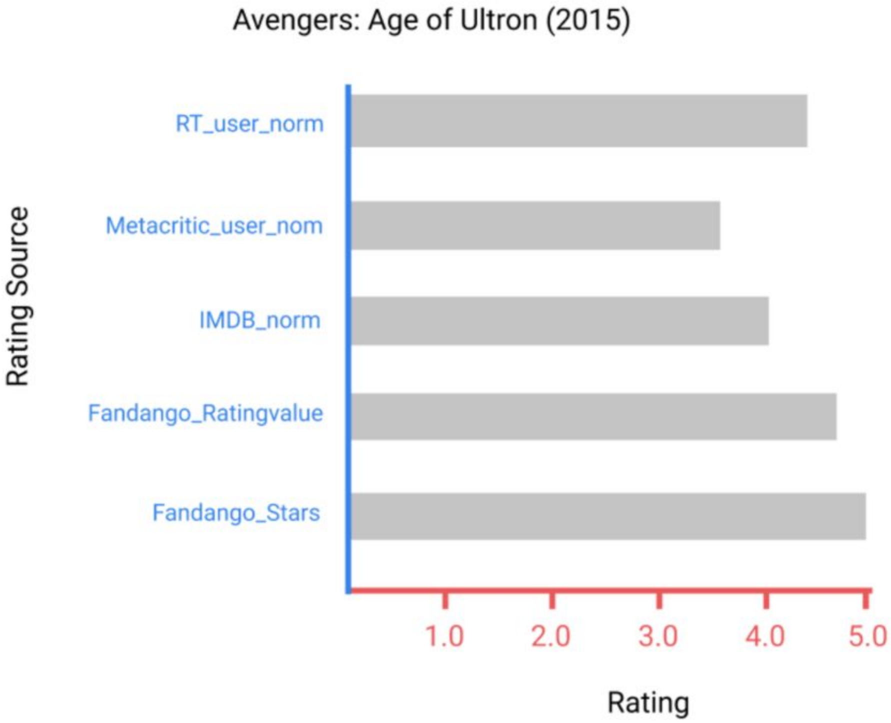
https://github.com/fivethirtyeight/data/tree/master/fandango

Bar plot



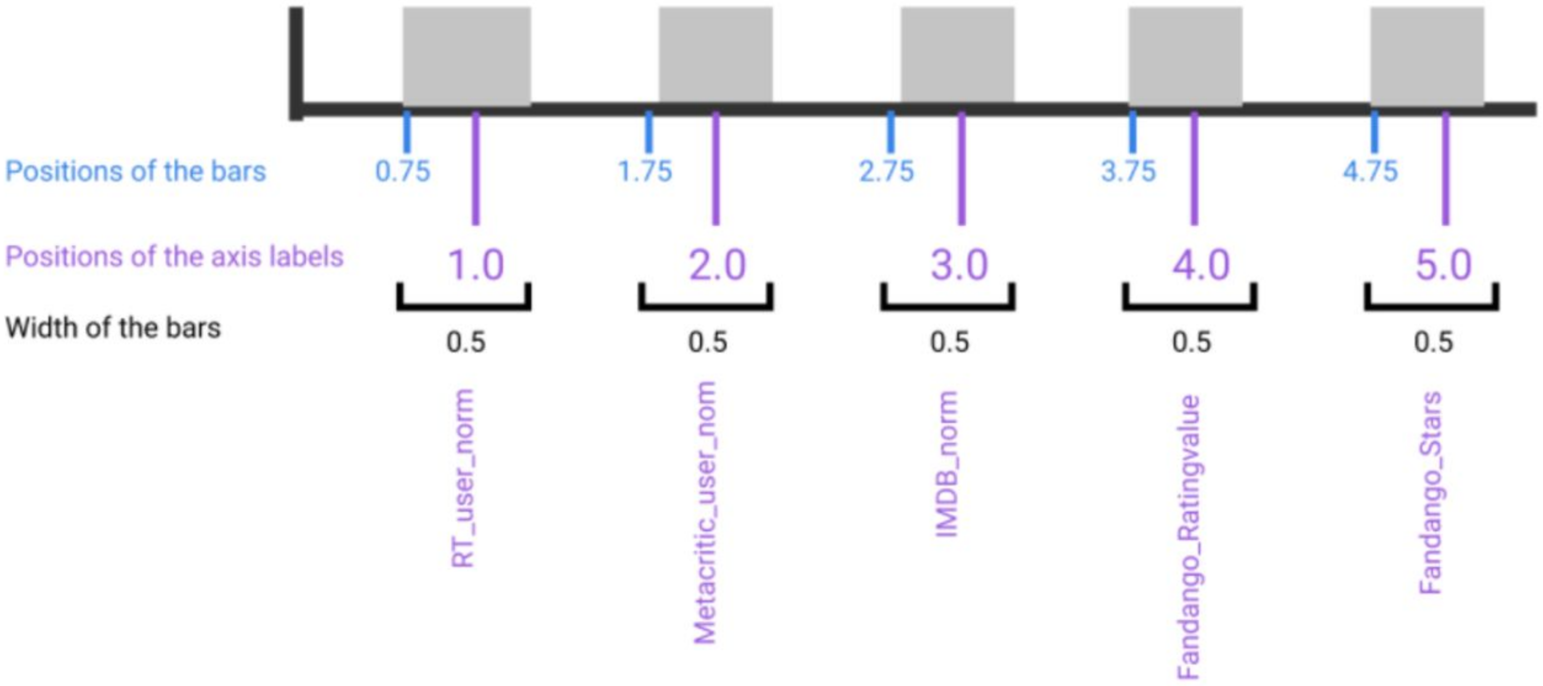
Avengers: Age of Ultron (2015)

Length of each bar corresponds to numerical value

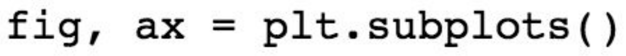


Avengers: Age of Ultron (2015)

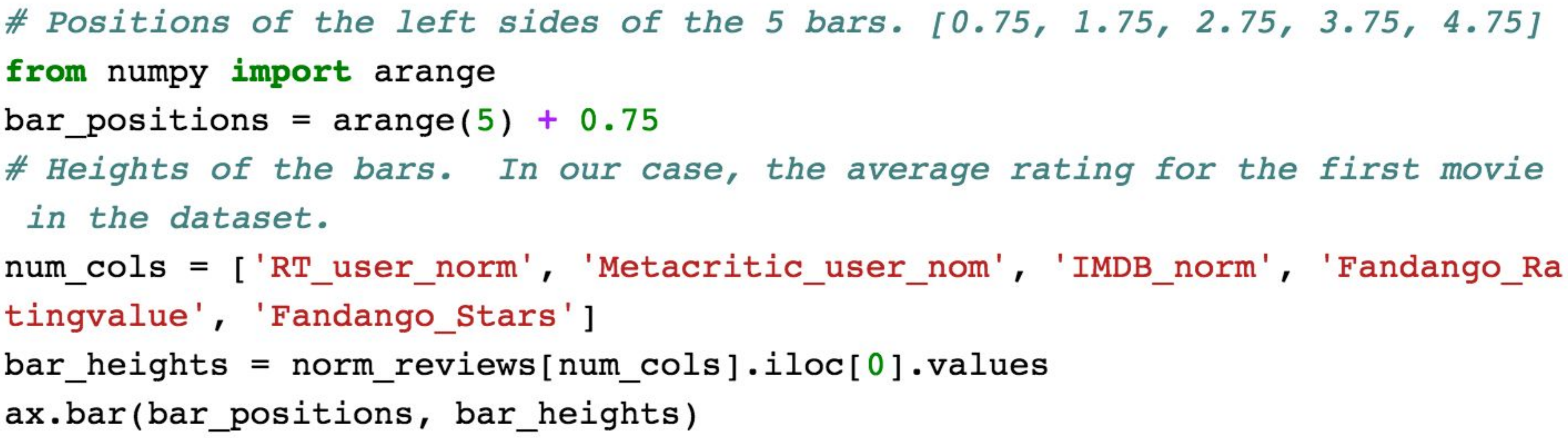
Creating bars



Creating bars



```
fig, ax = plt.subplots()
```



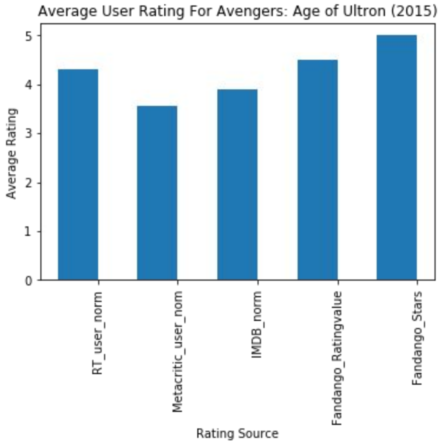
```
# Positions of the left sides of the 5 bars. [0.75, 1.75, 2.75, 3.75, 4.75]
from numpy import arange
bar_positions = arange(5) + 0.75
# Heights of the bars.  In our case, the average rating for the first movie
 in the dataset.
num_cols = ['RT_user_norm', 'Metacritic_user_nom', 'IMDB_norm', 'Fandango_Ra
tingvalue', 'Fandango_Stars']
bar_heights = norm_reviews[num_cols].iloc[0].values
ax.bar(bar_positions, bar_heights)
```

Aligning axis ticks and labels

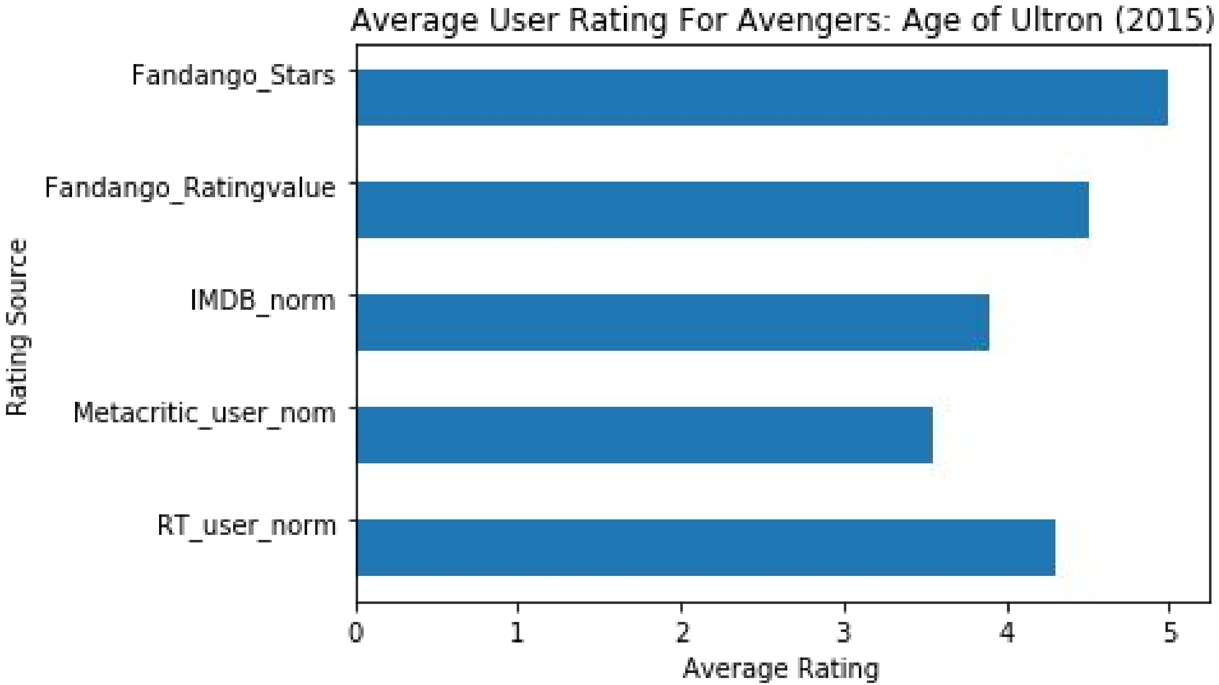
```
tick_positions = range(1,6)
ax.set_xticks(tick_positions)
```

```
num_cols = ['RT_user_norm', 'Metacritic_user_nom', 'IMDB_norm', 'Fandango_Ratingvalue', 'Fandango_Stars']
ax.set_xticklabels(num_cols)
```

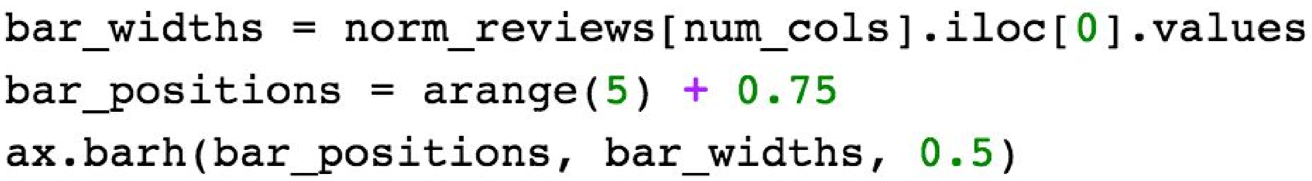
```
ax.set_xticklabels(num_cols, rotation=90)
```



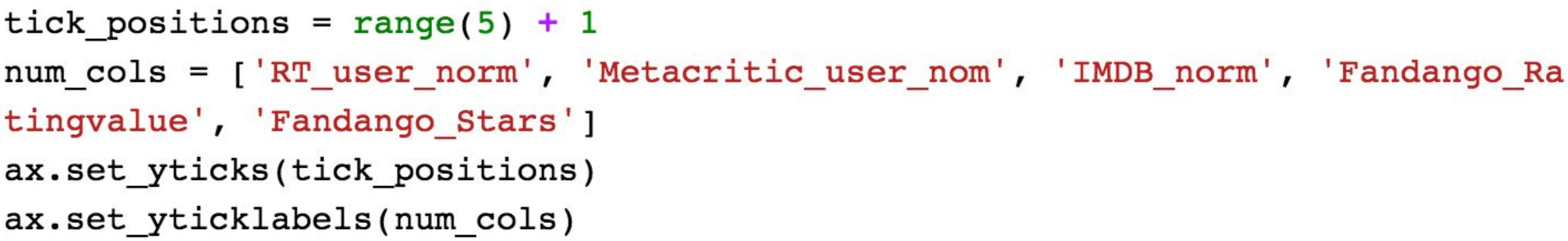
Horizontal bar plots



Horizontal bar plots

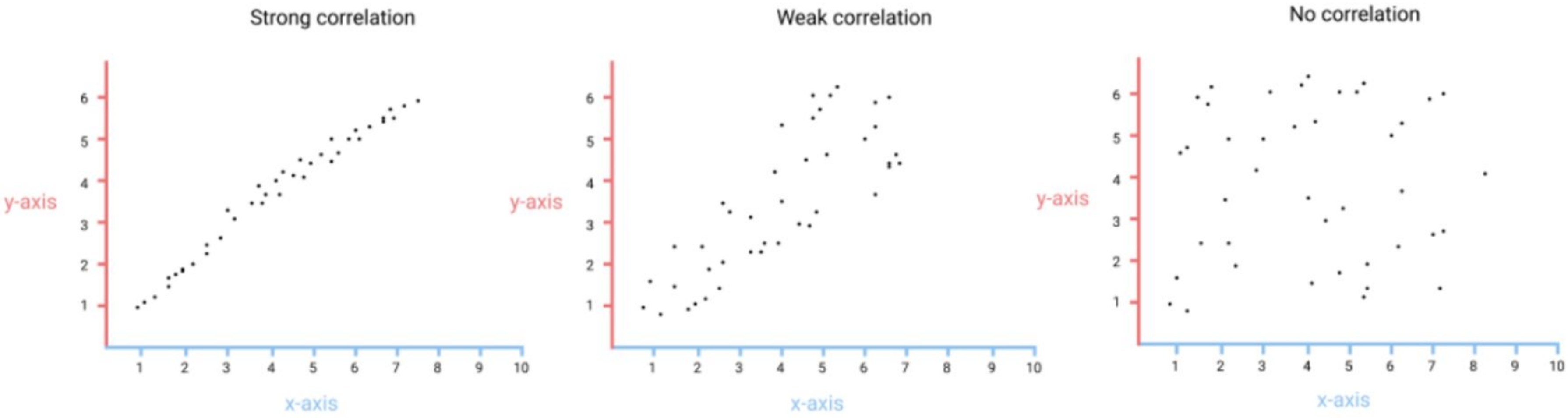


```
bar_widths = norm_reviews[num_cols].iloc[0].values
bar_positions = arange(5) + 0.75
ax.barh(bar_positions, bar_widths, 0.5)
```

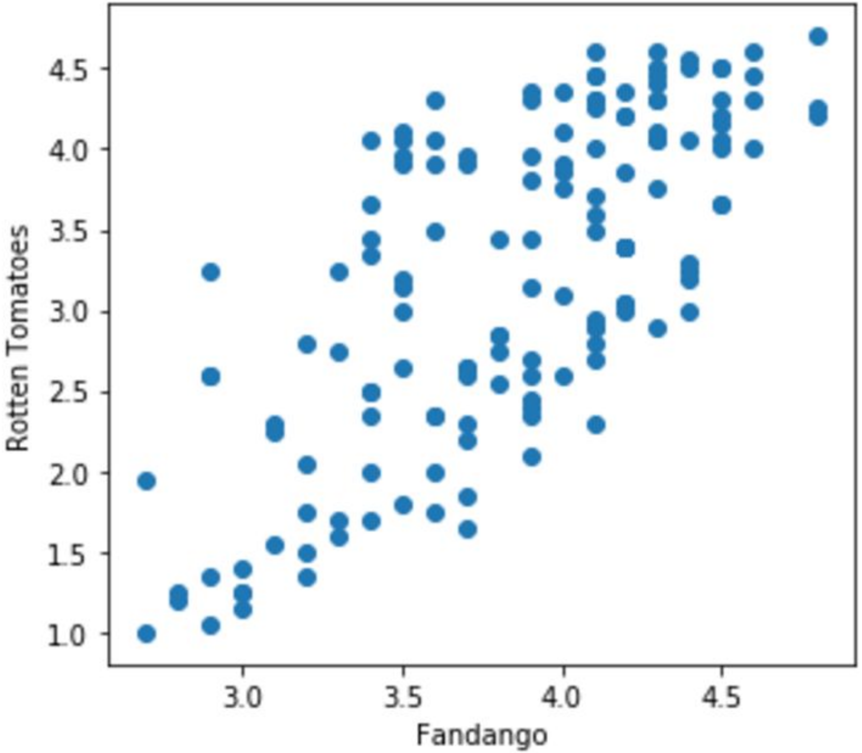


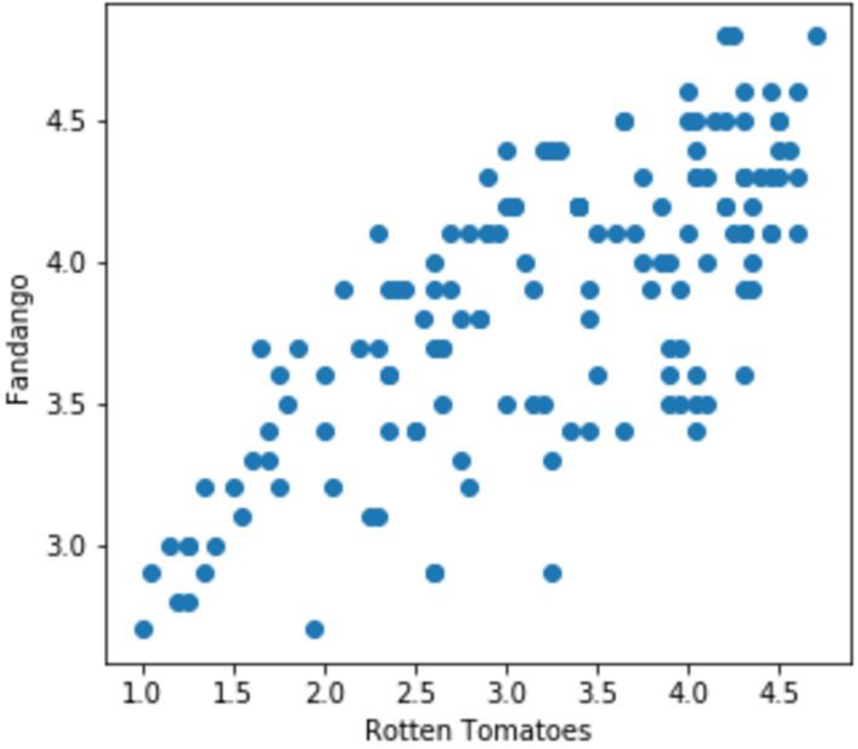
```
tick_positions = range(5) + 1
num_cols = ['RT_user_norm', 'Metacritic_user_nom', 'IMDB_norm', 'Fandango_Ra
tingvalue', 'Fandango_Stars']
ax.set_yticks(tick_positions)
ax.set_yticklabels(num_cols)
```

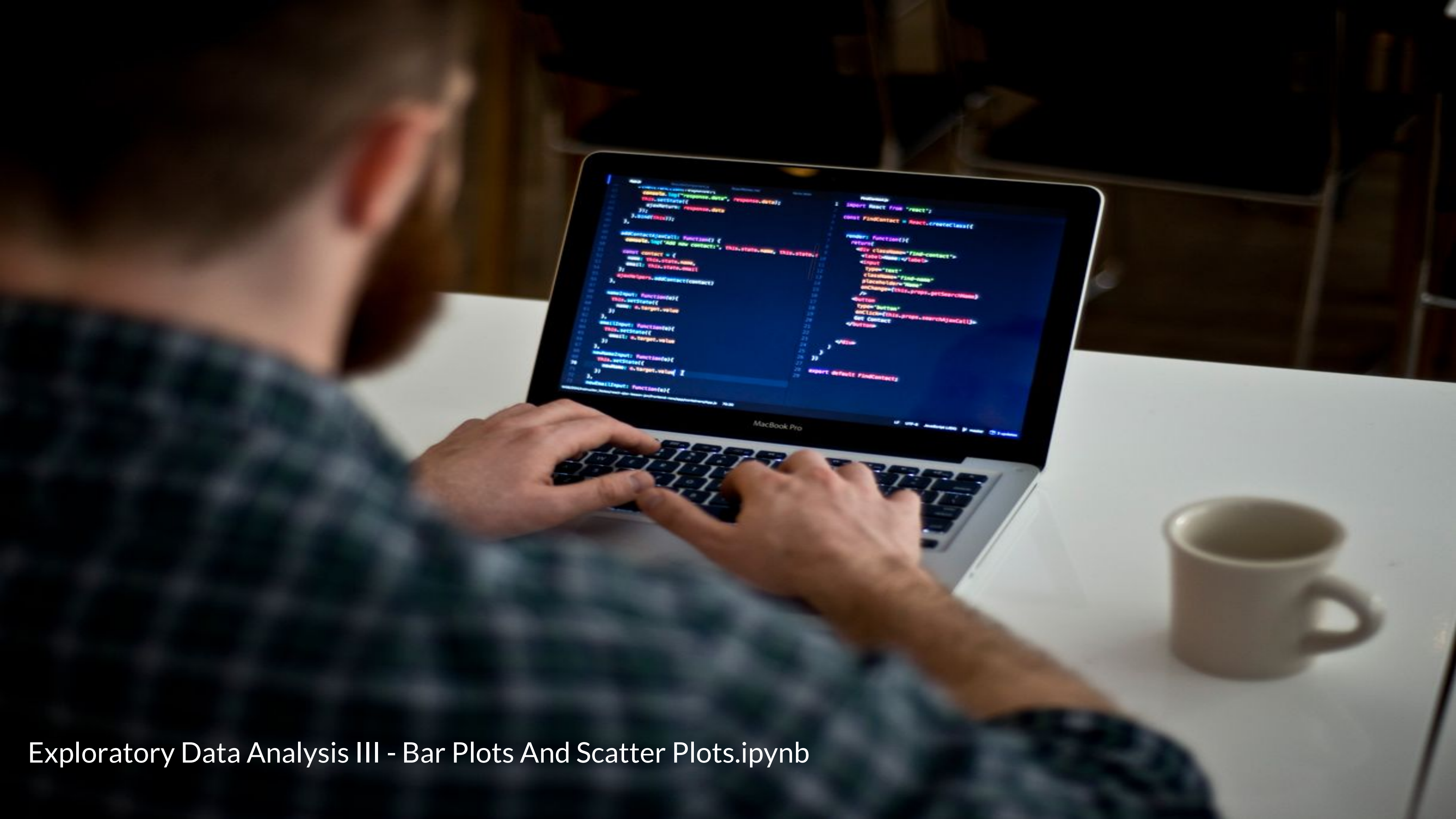
Scatter plot



Switching axes







Exploratory Data Analysis III - Bar Plots And Scatter Plots.ipynb