Project: Visualizing Race and Gender Representation In American Movies

In this project you'll use data visualization techniques to analyze how the top 50 movies of 2016 performed according to the Bechdel Test and other representation-based tests.

The Bechdel-Wallace Test — often abbreviated to the "Bechdel Test" — evaluates movies based on two simple questions:

1. Does the movie have at least two named female characters?
2. And do those characters have at least one conversation that is not about a man?

To perform your visualization, you'll use a dataset called "The Next Bechdel Test" from fivethirtyeight.com. You can read more about the dataset at the following link:

The Next Bechdel Test

In addition to results of the Bechdel test, this dataset includes the results of similar representation tests developed by members of the movie industry and evaluated by fivethirtyeight.com.

The Waithe Test (Lena Waithe)

A movie passes if:

- There's a black woman in the work
- Who's in a position of power
- And she's in a healthy relationship

The Ko Test (Naomi Ko)

A movie passes if:

- There's a non-white, female-identifying person in the film
- Who speaks in five or more scenes
- And speaks english

1. SETUP

Import matplotlib.pyplot as plt and pandas as pd.

In [1]:

```
import matplotlib.pyplot as plt
import pandas as pd
```

Use Pandas `pd.read_csv()` to load the dataset **bechdelExpanded.csv** into a DataFrame and save the results to a variable. You can name the variable anything you like. `df` is often used as the variable name for a DataFrame.

In [2]:

```
df = pd.read_csv('bechdelExpanded.csv')
```

2. Learn About Your Data

Inspect the DataFrame using `.head()`

The first column in the dataset is the name of a movie. Each column that follows represents one of the tests being applied to each movie. Each row represents whether that movie passed, or failed each test. A `1` represents a passing score, and a `0` represents a failing score.

In [4]:

```
print(df.head())
```

```
                          movie  bechdel  peirce  landau  feldman  villareal  \
0                      Bad Moms        0       0       0        1          0   
1                Hidden Figures        1       0       0        0          0   
2  Independence Day: Resurgence        0       0       1        0          0   
3                  Finding Dory        0       0       1        0          0   
4                  Ghostbusters        0       0       0        0          0   

   hagen  ko  villarobos  waithe  koeze_dottle  uphold  white  rees-davies  
0      0   0           1       0             0       1      1            1  
1      1   1           1       1             1       1      1            1  
2      1   0           1       0             0       1      1            1  
3      0   1           1       1             1       1      1            0  
4      1   0           1       1             1       1      1            1  
```

Call `.info()` on your DataFrame and print the result. This will display a summary of the basic information about your DataFrame and its data.

In [5]:

```
print(df.info)
```

```
<bound method DataFrame.info of                                                movie  bechdel  peirce  la
au  \
0                                           Bad Moms        0       0       0
1                                     Hidden Figures        1       0       0
2                       Independence Day: Resurgence        0       0       1
3                                       Finding Dory        0       0       1
4                                       Ghostbusters        0       0       0
5                                          Allegiant        0       0       1
6                                            Arrival        0       0       1
7                          Ice Age: Collision Course        0       1       0
8                                    Kung Fu Panda 3        0       1       0
9        Miss Peregrine's Home for Peculiar Children        0       0       0
10                                              Sing        0       0       1
11                                          The Boss        0       0       0
12                             The Girl on the Train        0       0       1
13                             Boo! A Madea Halloween        0       0       1
14                   Alice Through the Looking Glass        0       0       1
15           Fantastic Beasts and Where to Find Them        0       0       1
16                                        La La Land        0       0       1
17                                  Now You See Me 2        1       0       0
18                                        Passengers        1       0       0
19                                     Pete's Dragon        0       1       1
20                                     Sausage Party        0       0       1
21                                            Storks        1       0       1
22                                     Suicide Squad        0       0       1
23                                   The Conjuring 2        0       0       1
24                          The Purge: Election Year        0       0       1
25                                 X-Men: Apocalypse        0       0       0
26                               10 Cloverfield Lane        0       0       1
27                Batman v Superman: Dawn of Justice        1       0       1
28                        Captain America: Civil War        0       0       1
29                              Central Intelligence        1       0       1
30                                     Don't Breathe        0       0       1
31                                     Hacksaw Ridge        1       1       0
32                                        Lights Out        0       0       1
33                                             Moana        1       0       1
34                                      Ride Along 2        1       0       0
35                                  Star Trek Beyond        1       0       1
36                                             Sully        1       1       0
37  Teenage Mutant Ninja Turtles: Out of the Shadows        0       1       0
38                             The Angry Birds Movie        1       1       0
39                             The Magnificent Seven        1       0       0
40                                            Trolls        0       0       1
41                                          Zootopia        0       0       1
42                                      Jason Bourne        1       0       1
43                                         Rogue One        0       0       1
44                                    The Accountant        1       0       1
45                                   The Jungle Book        1       1       0
46                              The Legend of Tarzan        1       0       1
47                                          Deadpool        1       1       1
48                                    Doctor Strange        1       1       1
49                           The Secret Life of Pets        1       0       1

    feldman  villareal  hagen  ko  villarobos  waithe  koeze_dottle  uphold  \
0         1          0      0   0           1       0             0       1
1         0          0      1   1           1       1             1       1
2         0          0      1   0           1       0             0       1
3         0          0      0   1           1       1             1       1
4         0          0      1   0           1       1             1       1
5         0          0      1   0           1       1             1       1
6         0          0      1   1           1       1             0       1
7         1          1      1   0           1       1             0       1
8         0          1      1   0           1       1             1       1
9         0          0      1   1           1       1             1       1
10        1          1      1   0           1       1             0       1
11        0          1      1   1           1       1             0       1
12        0          0      1   1           1       1             0       1
13        1          1      1   0           1       0             0       1
14        0          1      1   1           1       1             1       1
15        1          0      1   0           1       1             1       1
16        1          0      1   1           1       1             0       1
17        1          0      1   0           1       1             1       1
18        1          0      1   1           1       1             0       1
19        0          0      1   1           1       1             1       1
```

```
20        1          1      1   0           1       1              1       1
21        1          0      1   1           1       1              0       1
22        1          1      1   0           1       1              1       1
23        1          0      1   1           1       1              0       1
24        1          0      1   0           1       1              1       1
25        1          0      1   1           1       1              1       1
26        1          0      1   1           1       1              1       1
27        1          0      1   1           1       1              1       1
28        1          1      1   1           1       1              1       1
29        1          1      1   0           1       0              1       1
30        1          1      1   1           1       1              0       1
31        1          0      1   1           1       1              1       1
32        1          1      1   1           1       1              0       1
33        1          1      1   1           1       1              1       1
34        1          1      1   0           1       1              1       1
35        1          1      1   0           1       1              1       1
36        1          0      1   1           1       1              0       1
37        1          0      1   1           1       1              1       1
38        1          0      1   0           1       1              1       1
39        1          0      1   1           1       1              1       1
40        1          1      1   1           1       1              0       1
41        1          0      1   1           1       1              1       1
42        1          1      1   1           1       1              1       1
43        1          1      1   1           1       1              1       1
44        1          1      1   0           1       1              1       1
45        1          1      1   0           1       1              1       1
46        1          0      1   1           1       1              1       1
47        1          1      1   1           1       1              0       1
48        1          1      1   1           1       1              1       1
49        1          1      1   1           1       1              1       1

    white  rees-davies
0       1            1
1       1            1
2       1            1
3       1            0
4       1            1
5       1            1
6       1            1
7       1            0
8       1            0
9       1            1
10      1            0
11      1            1
12      1            1
13      1            1
14      1            0
15      1            1
16      1            1
17      1            1
18      1            1
19      1            0
20      1            0
21      1            0
22      1            0
23      1            1
24      1            1
25      1            1
26      1            1
27      1            0
28      1            0
29      1            1
30      1            1
31      1            0
32      1            1
33      1            1
34      1            1
35      1            0
36      1            1
37      1            1
38      1            1
39      1            1
40      1            1
41      1            1
42      1            0
43      1            1
44      1            1
```

```
45      1             1
46      1             1
47      1             1
48      1             0
49      1             1  >
```

3. Data Manipulation

Create a column for total_score and set the value of each of its entries equal to the sum of the three columns: bechdel, waithe, ko. This will give us the total score each movie received based on these three tests.

In [6]:

```
df['total_score'] = df.bechdel + df.waithe + df.ko
```

Check your DataFrame again with `.head()` to see the new column.

In [7]:

```
print(df.head())
```

```
                          movie  bechdel  peirce  landau  feldman  villareal  \
0                      Bad Moms        0       0       0        1          0   
1                Hidden Figures        1       0       0        0          0   
2  Independence Day: Resurgence        0       0       1        0          0   
3                  Finding Dory        0       0       1        0          0   
4                  Ghostbusters        0       0       0        0          0   

   hagen  ko  villarobos  waithe  koeze_dottle  uphold  white  rees-davies  \
0      0   0           1       0             0       1      1            1   
1      1   1           1       1             1       1      1            1   
2      1   0           1       0             0       1      1            1   
3      0   1           1       1             1       1      1            0   
4      1   0           1       1             1       1      1            1   

   total_score  
0            0  
1            3  
2            0  
3            2  
4            1  
```

4. Sorting Data

It will be easier to visualize our data if it is organized by each movie's total score.

Using Pandas `.sort_values()`, create a new DataFrame called `df_sorted` that is a copy of your current DataFrame, sorted by `total_score`. Make sure to reset the index of the new DataFrame using `.reset_index(drop = True)`.

Hint: The syntax for sorting by a column in Pandas is: `df.sort_values("column_name").reset_index(drop = True)`

In [9]:

```
df_sorted = df.sort_values('total_score').reset_index(drop = True)
```

Use .head() to check your new `df_sorted` DataFrame.

In [10]:

```
print(df_sorted.head())
```

```
                          movie  bechdel  peirce  landau  feldman  villareal  \
0                      Bad Moms        0       0       0        1          0   
1  Independence Day: Resurgence        0       0       1        0          0   
2        Boo! A Madea Halloween        0       0       1        1          1   
3          Central Intelligence        1       0       1        1          1   
4                 Suicide Squad        0       0       1        1          1   

   hagen  ko  villarobos  waithe  koeze_dottle  uphold  white  rees-davies  \
0      0   0           1       0             0       1      1            1   
1      1   0           1       0             0       1      1            1   
2      1   0           1       0             0       1      1            1   
3      1   0           1       0             1       1      1            1   
4      1   0           1       1             1       1      1            0   

   total_score  
0            0  
1            0  
2            0  
3            1  
4            1  
```

5. Isolating the Data

For this project, we have selected three of the representation tests to focus on. To make it easier for us to look at the relevant data, create a new DataFrame containing only the columns 'movie', 'bechdel', 'waithe', 'ko', 'total_score' from the `df_sorted` DataFrame.

In [17]:

```
df_partial = df_sorted[[ 'movie', 'bechdel', 'waithe', 'ko', 'total_score']]
```

Use .head() to check the new DataFrame.

In [16]:

```
print(df_partial.head())
```

```
                          movie  bechdel  waithe  ko  total_score
0                      Bad Moms        0       0   0            0
1  Independence Day: Resurgence        0       0   0            0
2        Boo! A Madea Halloween        0       0   0            0
3          Central Intelligence        1       0   0            1
4                 Suicide Squad        0       1   0            1
```

6. Plot DataFrame with Matplotlib

Next, use Pandas to create a new DataFrame using only the data you want to use in a plot. Then, create a bar chart with that data using Matplotlib.

6.A

Using `[[]]` notation, select the columns `movie` and `total_score` from the DataFrame `df_partial` , then using `.set_index()` , set the `index` to the columns `movie` , and save it all to a variable named `ax` .

In [18]:

```
ax = df_partial[['movie', 'total_score']].set_index('movie')
```

6.B

Create a plot of the `ax` DataFrame with the Matplotlib method `.plot()` . Include the following arguments inside of `.plot()` . You can change these on your own and run the plot again if you would like.

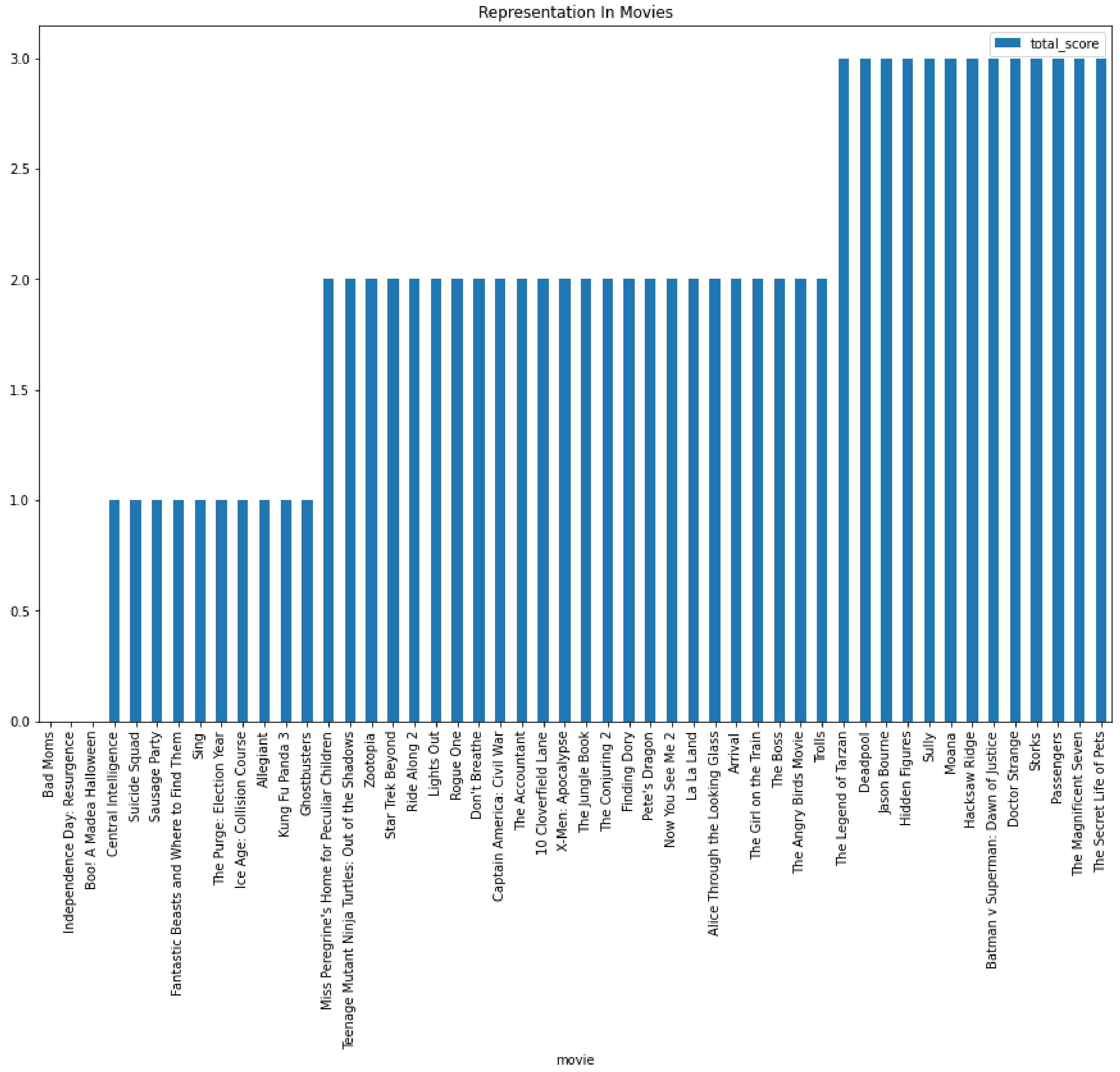
- `kind = 'bar'`
- `title ='Representation In Movies'`
- `figsize=(15, 10)`
- `legend=True`

In [19]:

```
ax.plot(kind = 'bar', title ='Representation In Movies', figsize=(15, 10), legend=True)
```

Out[19]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7feead17c730>
```



7. Iterate and Discover Meaning

There are many aspects of a Matplotlib plot that can be customized to make it easier to visualize data. Try a few of them below:

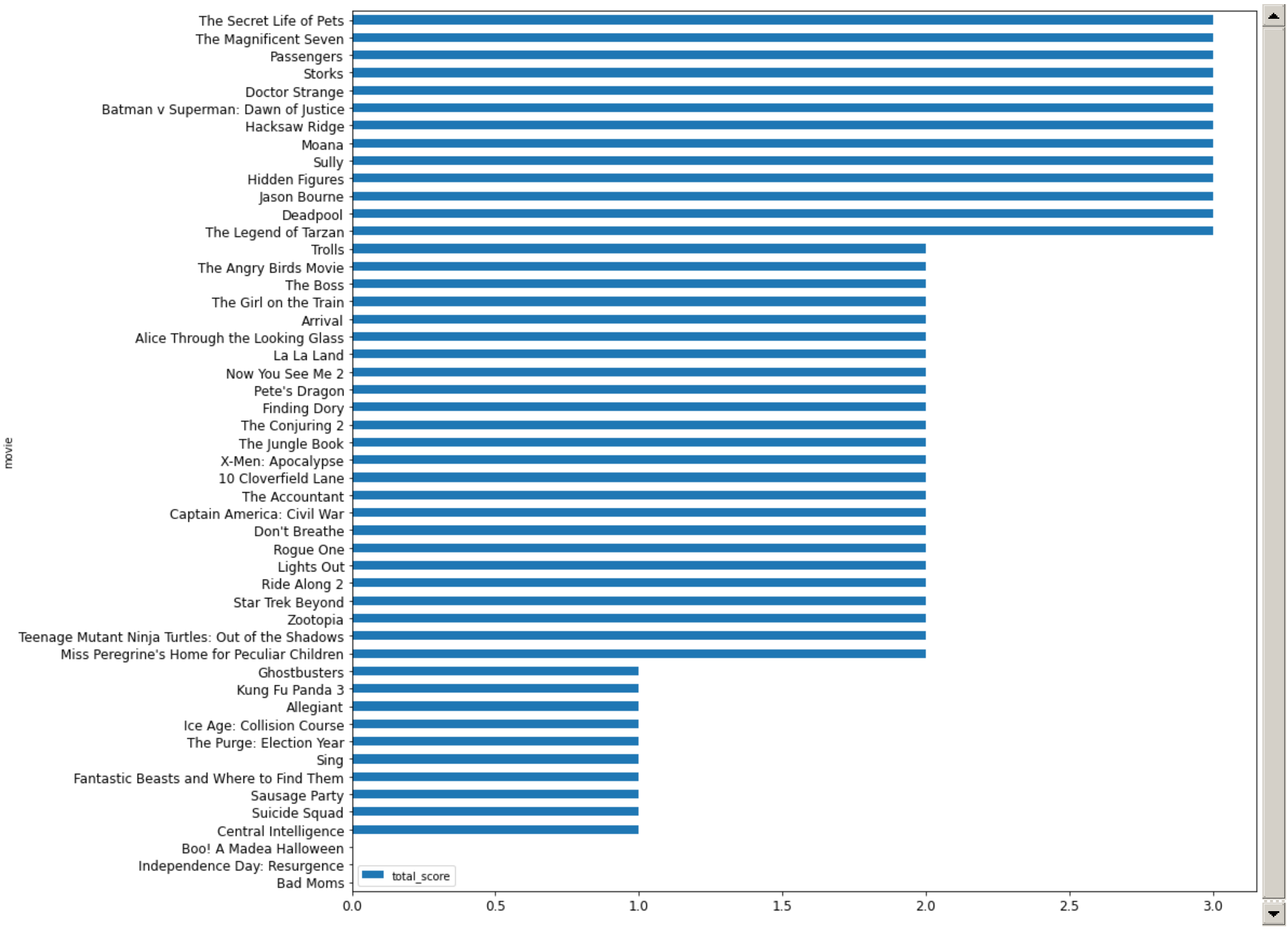
Set the `kind` argument of `.plot()` to `barh` to make the plot a horizontal bar chart.

Add the following argument to `.plot()` so it is easier to see the names of each movie: `fontsize=12`.

Make the visualization taller, and even the spread by changing the figsize argument to `figsize=(15, 15)`.

In [23]:

```
ax = df_partial[['movie', 'total_score']].set_index('movie').plot(kind='barh', fontsize=12, figsize=(15,
```



You're done!

Feel free to use this notebook to continue experimenting.