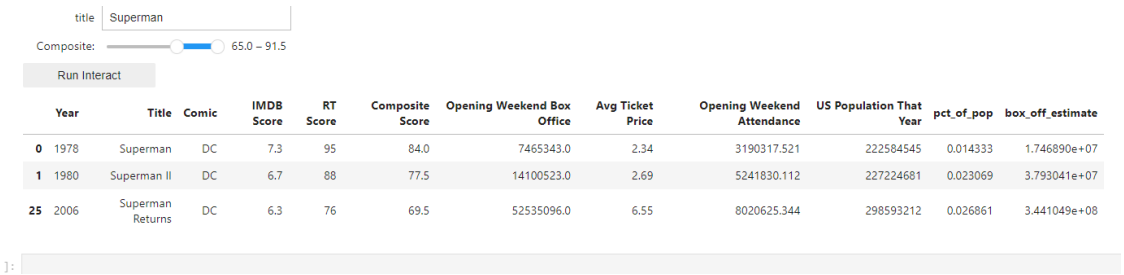
Practice-Superhero-Movies

September 27, 2024

0.1 PRACTICE - Superhero Movies

- UI to search for a title
  - use the `.str` property on the Series to access the string methods!
- Select a range for the Composite score based on the movie.
  - use the FloatRangeSlider widget https://ipywidgets.readthedocs.io/en/latest/examples/Widget%20List
- Output the Results using `display()`



	Year	Title	Comic	IMDB Score	RT Score	Composite Score	Opening Weekend Box Office	Avg Ticket Price	Opening Weekend Attendance	US Population That Year	pct_of_pop	box_off_estimate
0	1978	Superman	DC	7.3	95	84.0	7465343.0	2.34	3190317.521	222584545	0.014333	1.746890e+07
1	1980	Superman II	DC	6.7	88	77.5	14100523.0	2.69	5241830.112	227224681	0.023069	3.793041e+07
25	2006	Superman Returns	DC	6.3	76	69.5	52535096.0	6.55	8020625.344	298593212	0.026861	3.441049e+08

Here is the interface example

[1]:
```
from ipywidgets import interact_manual, widgets
from IPython.display import display
import pandas as pd
import numpy as np
```

[2]:
```
#Reading dataset

superheros = pd.read_csv("superhero2.csv")
```

0.1.1 Show 10 random rows of the dataset

[3]:
```
#using .sample to get random 10 data samples from the dataset
superheros.sample(10)
```

[3]:

	Year	Title	Comic	IMDB Score	RT Score	\
18	2004	Blade: Trinity	Marvel	5.8	26	
4	1986	Howard the Duck	Marvel	4.3	16	
15	2003	Daredevil	Marvel	5.4	45	
37	2010	Jonah Hex	DC	4.6	13	
24	2005	Fantastic Four	Marvel	5.7	27	

```
38  2011  Captain America: The First Avenger  Marvel         6.8        79
36  2010                        Iron Man 2  Marvel         7.1        74
1   1980                       Superman II      DC         6.7        88
7   1992                     Batman Returns      DC         7.0        78
16  2003                               Hulk  Marvel         5.7        62

    Composite Score  Opening Weekend  Box Office  Avg Ticket Price  \
18             42.0                  16061271.0              6.21
4              29.5                   5070136.0              3.71
15             49.5                  40310419.0              6.03
37             29.5                   5379365.0              7.89
24             42.0                  56061504.0              6.41
38             73.5                  65058524.0              7.93
36             72.5                 128122480.0              7.89
1              77.5                  14100523.0              2.69
7              74.0                  45687711.0              4.15
16             59.5                  62128420.0              6.03

    Opening Weekend Attendance  US Population That Year  pct_of_pop  \
18                2.586356e+06                293045739    0.008826
4                 1.366613e+06                240132887    0.005691
15                6.684978e+06                290326418    0.023026
37                6.817953e+05                308745538    0.002208
24                8.745944e+06                295753151    0.029572
38                8.204101e+06                311591917    0.026330
36                1.623859e+07                308745538    0.052595
1                 5.241830e+06                227224681    0.023069
7                 1.100909e+07                255029699    0.043168
16                1.030322e+07                290326418    0.035488

    box_off_estimate
18      9.974049e+07
4       1.881020e+07
15      2.430718e+08
37      4.244319e+07
24      3.593542e+08
38      5.159141e+08
36      1.010886e+09
1       3.793041e+07
7       1.896040e+08
16      3.746344e+08
```

[4]:
```
#understanding the dataset
superheros.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 46 entries, 0 to 45
```

```
Data columns (total 12 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Year                          46 non-null     int64
 1   Title                         46 non-null     object
 2   Comic                         46 non-null     object
 3   IMDB Score                    46 non-null     float64
 4   RT Score                      46 non-null     int64
 5   Composite Score               46 non-null     float64
 6   Opening Weekend  Box Office   46 non-null     float64
 7   Avg Ticket Price              46 non-null     float64
 8   Opening Weekend Attendance    46 non-null     float64
 9   US Population That Year       46 non-null     int64
 10  pct_of_pop                    46 non-null     float64
 11  box_off_estimate              46 non-null     float64
dtypes: float64(7), int64(3), object(2)
memory usage: 4.4+ KB
```

0.1.2 Search for Superman movies

[5]:
```
#creating a search dropdown for superman movies

# Make widgets
movies_list = superheros['Title'].tolist()

movies_dropdown = widgets.Dropdown(options=movies_list, description="Movies")
movies_dropdown
```

[5]:
```
Dropdown(description='Movies', options=('Superman', 'Superman II', 'Superman
III', 'Supergirl', 'Howard the Du…
```

[ ]:
```

```

Build the range slider of the composite score. Determine tha max and min values

Here is a start min_comp = sh[‘Composite Score’].min() max_comp = sh[‘Composite Score’].max() print(min_comp, max_comp)

[6]:
```
min_comp = superheros['Composite Score'].min()
max_comp = superheros['Composite Score'].max()
print(min_comp, max_comp)
```

```
19.5 91.5
```

[7]:
```
widgets.FloatRangeSlider(
    value=[5, 7.5],
    min=min_comp,
```

```
    max=max_comp,
    step=0.1,
    description='Composite Score:',
    disabled=False,
    continuous_update=False,
    orientation='horizontal',
    readout=True,
    readout_format='.1f',
)
```

[7]:
```
FloatRangeSlider(value=(19.5, 19.5), continuous_update=False,
description='Composite Score:', max=91.5, min=19…
```

0.2 Complete working code

[8]:
```
from ipywidgets import interact_manual, widgets
from IPython.display import display
import pandas as pd
import numpy as np

#Reading dataset
superheros = pd.read_csv("superhero2.csv")


#using .sample to get random 10 data samples from the dataset
superheros.sample(10)

#creating a search dropdown for superman movies
movies_list = superheros['Title'].tolist()
movies_selection = widgets.Dropdown(options=movies_list, description="Movies")


#Creating slider for composite scores by ranging values from min and max of the␣
 ↪composite scores
min_comp = superheros['Composite Score'].min()
max_comp = superheros['Composite Score'].max()
cs=widgets.FloatRangeSlider(
    value=[min_comp, max_comp],
    min=min_comp,
    max=max_comp,
    step=0.1,
    description='Composite Score:',
    disabled=False,
    continuous_update=False,
    orientation='horizontal',
    readout=True,
    readout_format='.1f',
```

```
)

#creating on click interact function to pass the input values of movies and the␣
 ↪composite scores to get the filtered list of movies
@interact_manual(Composite_score=cs, movie=movies_selection)
def on_click(Composite_score, movie):
    filtered_movies = superheros[
         (superheros["Title"].str.contains(movie, case=False, na=False)) &
        (superheros["Composite Score"] >= Composite_score[0]) &
        (superheros["Composite Score"] <= Composite_score[1])
    ]
    display(filtered_movies)
```

```
interactive(children=(FloatRangeSlider(value=(19.5, 91.5),␣
 ↪continuous_update=False, description='Composite Sco…
```