

# Super Hero Data Analysis and Visualization

July 28, 2018

```
In [1]: import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.plotly as py1
import plotly.offline as py
py.init_notebook_mode(connected=True)
from plotly.offline import init_notebook_mode, iplot
init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.offline as offline
offline.init_notebook_mode()
from plotly import tools
import plotly.graph_objs as go

pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
```

## 0.0.1 Reading in and cleaning data

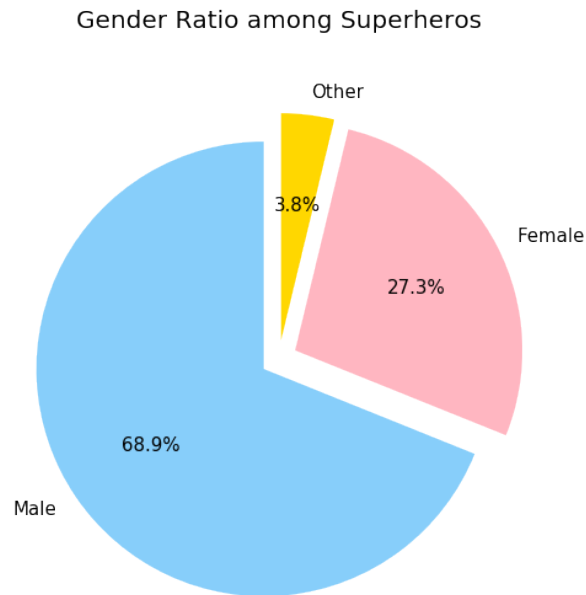
```
In [2]: data = pd.read_csv('SuperheroDataset.csv')
data.replace(to_replace='-', value='Other', inplace=True)
data['Creator'].fillna('Other', inplace=True)
```

## 0.0.2 Finding out and visualising gender ratio of all collected superheroes

```
In [3]: hero_g = data.Gender.value_counts()

In [4]: plt.figure(figsize=(16,8))
plt.title('Gender Ratio among Superheros', fontsize=20, y=1.1,)
labels = 'Male', 'Female', 'Other'
colors = ['lightskyblue', 'lightpink', 'gold']
explode=(0.08, 0.08, 0.08)
plt.rcParams['font.size'] = 15.0
plt.pie(hero_g.values, colors=colors,
        explode=explode, labels=labels,
        autopct='%1.1f%%', startangle=90)
```

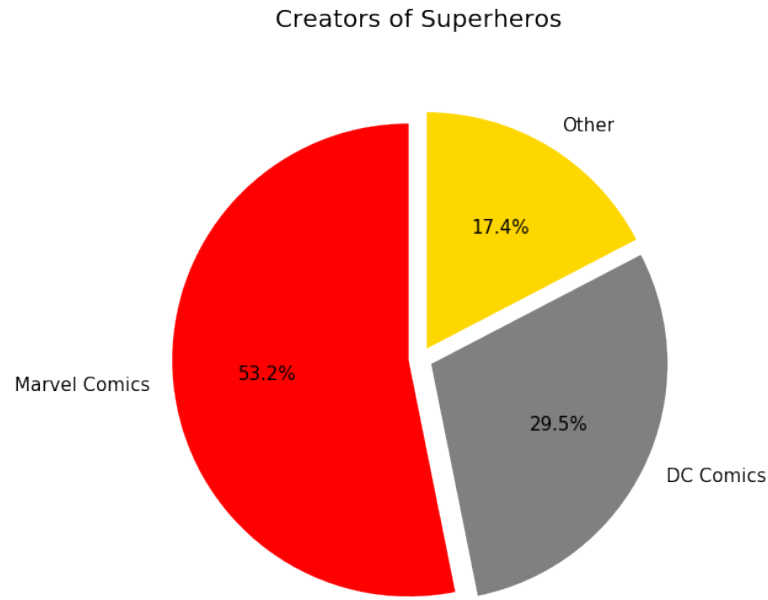
```
plt.axis('equal')
plt.show()
```



### 0.03 Visualizing Publishers

```
In [5]: hero_p = data.Creator.value_counts()
        other = hero_p.values[2:].sum()
        hero_p = hero_p[:2]
        hero_p['Other'] = other

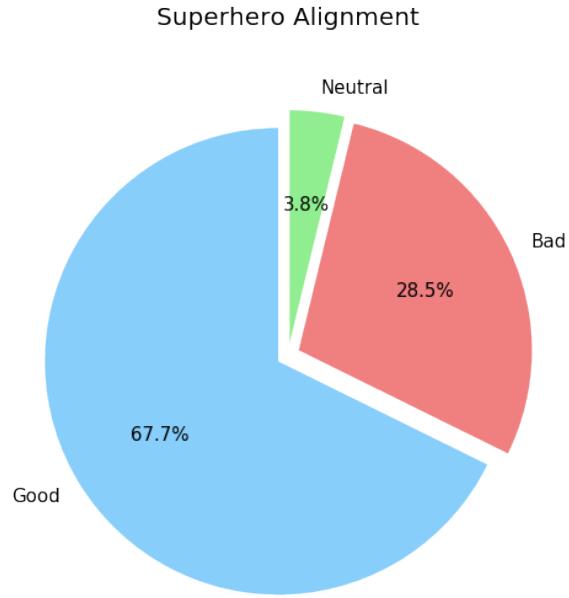
In [6]: plt.figure(figsize=(16,8))
        plt.title('Creators of Superheros', fontsize=20, y=1.1,)
        labels = hero_p.index
        colors = ['red', 'gray', 'gold']
        explode=(0.05, 0.05, 0.05)
        plt.rcParams['font.size'] = 15.0
        plt.pie(hero_p.values, colors=colors,
                explode=explode, labels=labels,
                autopct='%1.1f%%', startangle=90)
        plt.axis('equal')
        plt.show()
```



#### 0.0.4 Visualising Superhero Alignments

```
In [7]: hero_align = data.Alignment.value_counts()
        hero_align = hero_align[:3]

In [8]: plt.figure(figsize=(16,8))
        plt.title('Superhero Alignment', fontsize=20, y=1.1,)
        labels = ['Good', 'Bad', 'Neutral']
        colors = ['lightskyblue', 'lightcoral', 'lightgreen']
        explode=(0.05, 0.05, 0.05)
        plt.rcParams['font.size'] = 15.0
        plt.pie(hero_align.values, colors=colors,
                explode=explode, labels=labels,
                autopct='%1.1f%%', startangle=90)
        plt.axis('equal')
        plt.show()
```



### 0.0.5 Finding the most powerful superheroes in each universe

```
In [9]: data_marvel = data.loc[data['Creator'] == 'Marvel Comics']
data_dc = data.loc[data['Creator'] == 'DC Comics']
data_marvel = data_marvel.sort_values('Total Power', ascending=False)
data_dc = data_dc.sort_values('Total Power', ascending=False)
top_10_dc = data_dc[:10]
top_10_marvel = data_marvel[:10]
```

```
In [10]: top_10_dc[['Name', 'Total Power']]
```

```
Out[10]:
```

	Name	Total Power
668	The Presence	600.0
275	General Zod	595.0
471	Monarch	590.0
653	Superman	585.0
292	Granny Goodness	585.0
651	Superboy-Prime	585.0
646	Steppenwolf	585.0
416	Lucifer Morningstar	580.0
652	Supergirl	575.0
529	Power Girl	575.0

```
In [11]: top_10_marvel[['Name', 'Total Power']]
```

```
Out[11]:
```

	Name	Total Power
506	One-Above-All	600.0

87	Binary	595.0
80	Beyonder	585.0
670	Thor	570.0
522	Phoenix	565.0
161	Captain Universe	565.0
640	Stardust	565.0
341	Hyperion	560.0
231	Dormammu	555.0
335	Hulk	545.0

### 0.0.6 Comparing total strength of DC vs Marvel characters

```
In [12]: strength_dc = data_dc['Total Power'].sum()/hero_p['DC Comics']
        strength_marvel = data_marvel['Total Power'].sum()/hero_p['Marvel Comics']
```

```
In [13]: df=pd.DataFrame({'Universe':['Marvel', 'DC'],
                          'Strength':[strength_marvel, strength_dc]})
fig = plt.figure(figsize=(12,7))
fig.add_subplot(1,1,1)
sns.barplot(x='Universe',y='Strength',data=df)
plt.xticks(rotation=0)
plt.show()
```

