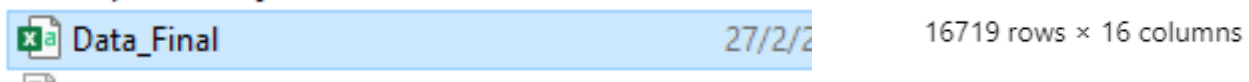
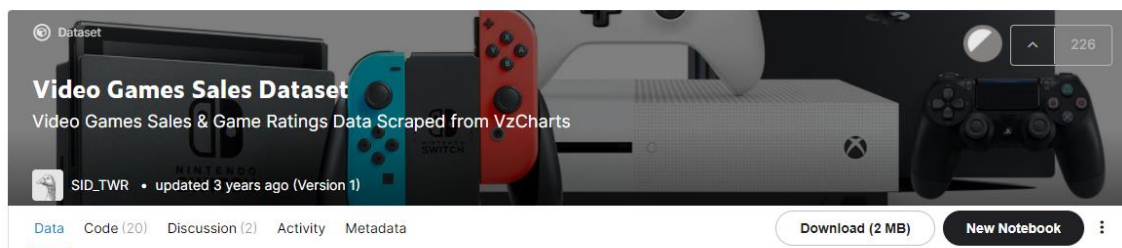


UMAP-Micro Credential: Data Visualization

Capstone Project

“ What genre would it be if I wanted to create a game? ”

This topic was chosen because I wanted to explore how the gaming market was doing and what genre a company would develop if they wanted to launch a brand new game. I began my hunt for information about game genres and the ratings they have received from both users and critics who have played them. After a while I got my dataset from Kaggle.com [Video Games Sales Dataset | Kaggle](#)



I extracted the valuable data from my dataset by eliminating irrelevant data (years that were less than 2010, N/A NaN), and unsuitable columns (Regional Sales, Publisher, Developer, and Year of Release).

```
[6]: data = data[data['Year_of_Release'] > 2010]
data
```

```
data.drop_duplicates(inplace=True)
data.reset_index(level=None, drop=True, inplace=True, col_level=0, col_fill='')
print(data['Year_of_Release'].unique())
print(data['Platform'].unique())
data
```

```
Globalsales = data.drop(['NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales', 'Publisher', 'Developer', 'Year_of_Release'], axis=1, inplace=False)
Globalsales
```

```
UsablePlatform = ['PS3', 'X360', 'PS4', 'Wii', 'XOne', 'PC']
SumarizedPlatform = pd.DataFrame()
for i in UsablePlatform:
    SumarizedPlatform = SumarizedPlatform.append(Globalsales[Globalsales['Platform'] == i])

print(SumarizedPlatform['Platform'].unique())
SumarizedPlatform
```

My new data that I'll use to visualize is now ready.

```
> ~
SumarizedPlatform.sort_values('Genre',inplace=True)
● SumarizedPlatform.set_index('Genre',inplace=True)
SumarizedPlatform
[15] ✓ 0.1s
```

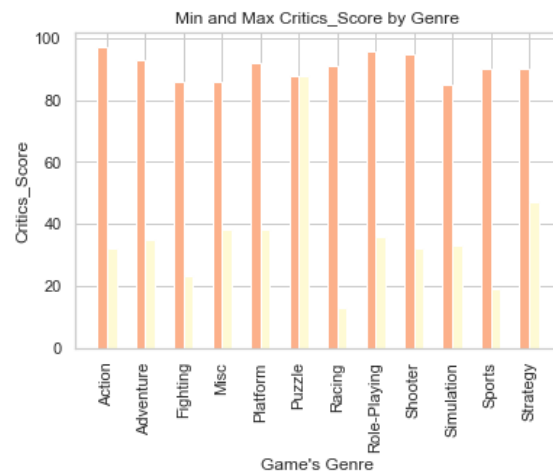
...

	Name	Platform	Global_Sales	Critic_Score	Critic_Count	User_Score	User_Count
Genre							
Action	Grand Theft Auto V	PS3	21.04	97.0	50.0	8.2	3994.0
Action	Thor: God of Thunder	X360	0.14	38.0	40.0	5.7	22.0
Action	Kung-Fu: High Impact	X360	0.13	56.0	18.0	7.8	20.0
Action	Rango: The Video Game	X360	0.13	68.0	25.0	8.0	9.0
Action	Green Lantern: Rise of the Manhunters	X360	0.13	59.0	17.0	7.2	23.0
...
Strategy	Warhammer 40,000: Dawn of War II - Retribution	PC	0.08	80.0	52.0	7.8	235.0
Strategy	Dungeons	PC	0.09	65.0	35.0	4.6	106.0
Strategy	Angry Birds Star Wars	PS4	0.21	47.0	5.0	2.0	87.0
Strategy	Angry Birds Star Wars	X360	0.28	59.0	3.0	5.6	25.0
Strategy	STORM: Frontline Nation	PC	0.01	60.0	12.0	7.2	13.0

1331 rows × 7 columns

So I start by visualizing the min and max critic scores by genre.

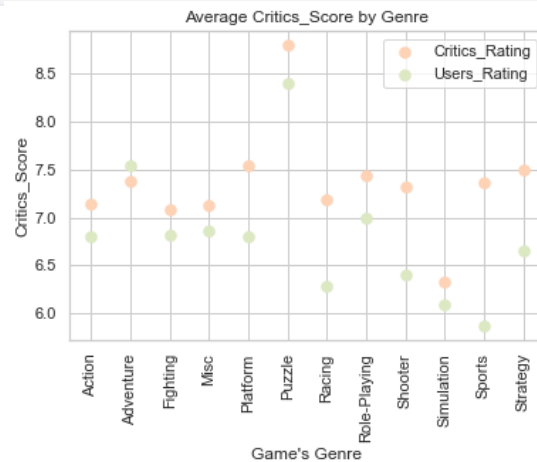
```
> ~
plt.bar([i for i in range(12)],tmp['Critic_Score'],width=0.25,color='#FCB08A')
plt.bar([i+0.25 for i in range(12)],tmpmin['Critic_Score'],width=0.25,color='#FEFAD4')
plt.xticks(rotation=90)
plt.xticks([i+0.125 for i in range(12)],tmp.index)
plt.xlabel('Game\'s Genre')
plt.ylabel('Critics_Score')
plt.title('Min and Max Critics_Score by Genre')
plt.show()
[97] ✓ 0.2s
```



Then I continued plotting critics' and users' scores, but now it's an average of each by game genre.

```
plt.scatter(tmpavg.index,tmpavg['Critic_Score'],color='#FDD2B5',linewidth=2.5,label='Critics_Rating')
plt.scatter(tmpavg.index,tmpavg['User_Score'],color= '#DBE8C2',linewidth=2.5,label='Users_Rating')
plt.legend()
plt.xticks(rotation=90)
plt.xlabel('Game\'s Genre')
plt.ylabel('Critics_Score')
plt.title('Average Critics_Score by Genre')
plt.show()
```

19] ✓ 0.6s

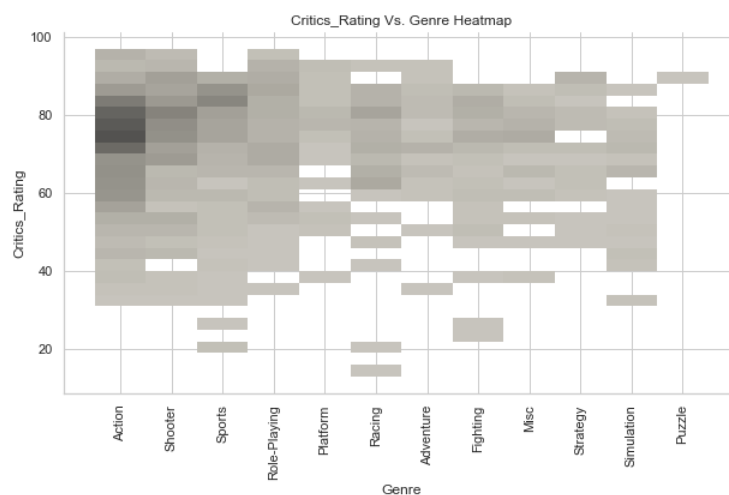


After I finished plotting the previous graph, I had an idea: what if I tried to map the whole data ratings that critics gave to each genre? Where will it lead me to? So I started plotting a distribution plot graph for both critic rating and user rating to examine how frequently each category's information appears.

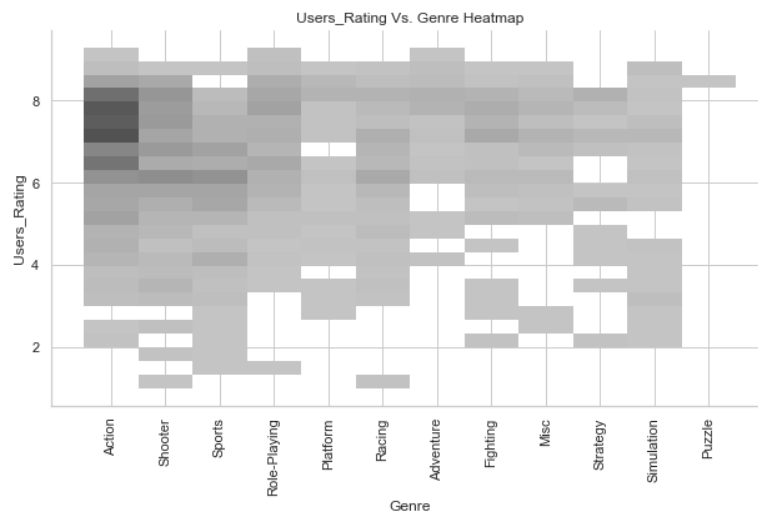
```
disploting = sns.displot(data=SumarizedPlatform, x='Genre', y='Critic_Score',legend=True,aspect=1.75,color='#EFEDE7')
disploting.set_ylabels(Label='Critics_Rating')
disploting.set_xticklabels(x='Genre', rotation=90)
disploting.set(title = "Critics_Rating Vs. Genre Heatmap")
```

[107] ✓ 0.3s

... <seaborn.axisgrid.FacetGrid at 0x2135bf22dc0>



```
✓
disploting = sns.displot(data=SumarizedPlatform, x='Genre', y='User_Score', Legend=True, aspect=1.75, color='grey')
disploting.set_xticklabels(x='Genre', rotation=90)
✓ 0.2s
<seaborn.axisgrid.FacetGrid at 0x21357b75df0>
```



However, after plotting the distribution graph, I recognized that my data was not evenly distributed. So I began tracking how many of each genre there were, and I discovered that the puzzle genre couldn't be used due to a lack of data. As a result, I decided to leave the puzzle out of this examination.

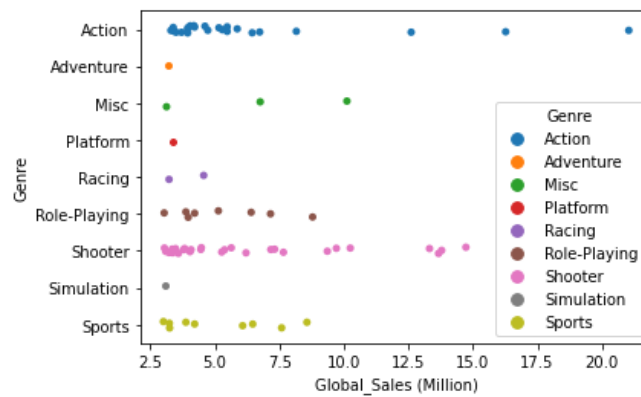
```
> ~
SumarizedPlatform.index.value_counts()
126] ✓ 0.5s
... Action      417
    Shooter    231
    Sports     161
    Role-Playing 131
    Racing      98
    Fighting    75
    Misc        55
    Adventure   44
    Simulation  40
    Platform    39
    Strategy    39
    Puzzle      1
    Name: Genre, dtype: int64
```

Combining two distribution graphs and the latest data, it told me that we can't rely on average data. And now I can see how both users and critics react to each genre by looking at the distribution graph that I've plotted, and I can also see that action is grouped together in 75 to 80 rating range. Now I'm going to see the genre's sales number.

```
> ~
sales3m = SumarizedPlatform[SumarizedPlatform['Global_Sales']>3]
globalsale = sns.stripplot(data=sales3m,x='Global_Sales',y='Genre',hue='Genre')
globalsale.set(xLabel='Global_Sales (Million)')
```

44] ✓ 0.9s

```
... [Text(0.5, 0, 'Global_Sales (Million)')]
```



In summary, I've concluded that Action is the most desirable game genre on the market, followed by Shooter games based on sales data and a heat map of user and critic ratings.