

A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light green. They are positioned diagonally, with the blue one partially covering the green one.

Video Game Data Analysis

Jarrin Flores, Ben Ahnen, Jonas Geerdes



Project Goal

- Video Game Data from Kaggle
- Determine why a game is critically and financially successful
- Discover overall trends and patterns



Packages and Functions Used

- Pandas
- Numpy
- Functions:
 - Total Sales by Genre
 - Fetching Top Performers
 - Plotting Sales vs. Critical Ratings
 - Comparing User vs. Critical Ratings

Total Sales by Genre

```
3 def total_sales():
4     import pandas as pd
5     import numpy as np
6     df_Games = pd.read_csv('Video_Games_Data1.csv')
7     sales = df_Games.groupby('Genre').agg({'Global_Sales': np.sum})
8     return sales
```

| Genre | |
|--------------|--------|
| Action | 199.58 |
| Adventure | 17.59 |
| Fighting | 28.19 |
| Misc | 37.54 |
| Platform | 18.05 |
| Puzzle | 2.22 |
| Racing | 27.54 |
| Role-Playing | 101.40 |
| Shooter | 170.98 |
| Simulation | 13.12 |
| Sports | 109.53 |
| Strategy | 3.98 |

ABOVE: Code

LEFT: Total Sales Output

RIGHT: Mean Sales Output

| Genre | |
|--------------|----------|
| Action | 0.321903 |
| Adventure | 0.095081 |
| Fighting | 0.469833 |
| Misc | 0.332212 |
| Platform | 0.475000 |
| Puzzle | 0.158571 |
| Racing | 0.399130 |
| Role-Playing | 0.454709 |
| Shooter | 1.335781 |
| Simulation | 0.298182 |
| Sports | 0.680311 |
| Strategy | 0.107568 |



5 Top Performers

Code

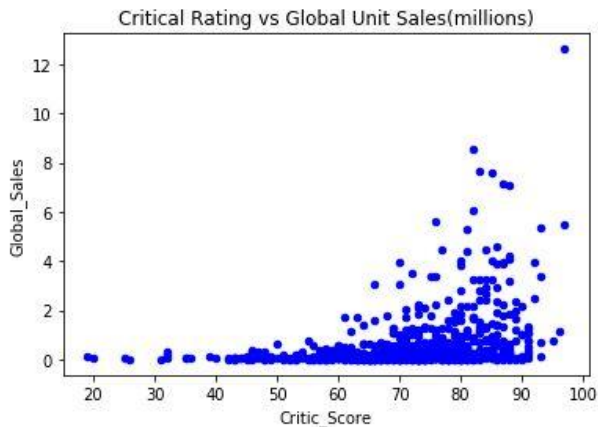
```
10 def top_performers():  
11     import pandas as pd  
12     df_Games = pd.read_csv('Video_Games_Data1.csv')  
13     top5 = df_Games.nlargest(5, 'Global_Sales')  
14     return top5[['Name', 'Platform', 'Global_Sales']]
```

Output

| | Name | Platform | Global_Sales |
|---|---|----------|--------------|
| 0 | Call of Duty: Black Ops 3 | PS4 | 14.63 |
| 1 | Grand Theft Auto V | PS4 | 12.61 |
| 2 | Pokemon Omega Ruby/Pokemon Alpha Sapphire | 3DS | 11.68 |
| 3 | FIFA 16 | PS4 | 8.57 |
| 4 | Star Wars Battlefront (2015) | PS4 | 7.98 |

Critical Ratings vs Sales

```
16
17 def rating_vs_sales():
18     import pandas as pd
19     df_Games = pd.read_csv('Video_Games_Data1.csv')
20     return df_Games.plot(x='Critic_Score', y='Global_Sales', kind='scatter', title='Critical Rating vs Global Unit Sales(millions)', color='b')
```



ABOVE: Code
LEFT: Output



Critical Ratings > 90

```
21
22 def critic_ratings_90_or_better():
23     import pandas as pd
24     df_Games = pd.read_csv('Video_Games_Data1.csv')
25     df = df_Games[df_Games['Critic_Score'] >= 90]
26     return df[['Name', 'Platform', 'Critic_Score']]
27
```

| Platform | Critic_Score | |
|----------|--------------|--------------------------------------|
| 1 | | Grand Theft Auto V |
| PS4 | 97.0 | |
| 14 | | Grand Theft Auto V |
| XOne | 97.0 | |
| 15 | | Uncharted 4: A Thief's End |
| PS4 | 93.0 | |
| 32 | | The Witcher 3: Wild Hunt |
| PS4 | 92.0 | |
| 40 | | Metal Gear Solid V: The Phantom Pain |
| PS4 | 93.0 | |
| 61 | | Bloodborne |
| PS4 | 92.0 | |
| 75 | | Overwatch |
| PS4 | 90.0 | |
| 130 | | The Witcher 3: Wild Hunt |
| XOne | 91.0 | |
| 139 | | Dark Souls II |
| PS3 | 91.0 | |
| 152 | | Grand Theft Auto V |
| PC | 96.0 | |
| 161 | | Forza Horizon 3 |

| | | |
|------|------|--------------------------------------|
| 139 | | Dark Souls II |
| PS3 | 91.0 | |
| 152 | | Grand Theft Auto V |
| PC | 96.0 | |
| 161 | | Forza Horizon 3 |
| XOne | 91.0 | |
| 164 | | Overwatch |
| XOne | 91.0 | |
| 167 | | NBA 2K17 |
| XOne | 90.0 | |
| 169 | | Rayman Legends |
| PS4 | 90.0 | |
| 212 | | Bayonetta 2 |
| WiiU | 91.0 | |
| 217 | | Metal Gear Solid V: The Phantom Pain |
| XOne | 95.0 | |
| 221 | | Dark Souls II |
| X360 | 91.0 | |
| 236 | | The Witcher 3: Wild Hunt |
| PC | 93.0 | |
| 311 | | Overwatch |
| PC | 91.0 | |

ABOVE: Code
LEFT: Output



Conclusion

- Shooter games sell the most per unit
- Platform does not indicate success
- High critic rating does not indicate financial success