Week 1: Brandon Mather DSC550-T301

In [1]:
```
import pandas as pd
```

In [88]:
```
#Load the dataset as a Pandas data frame.
data = pd.read_csv('Video_Games_Sales_as_at_22_Dec_2016.csv')
```

In [4]:
```
#Display the first ten rows of data.
first_ten = data.head(10)
first_ten
```

Out[4]:

	Name	Platform	Year_of_Release	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_S
0	Wii Sports	Wii	2006.0	Sports	Nintendo	41.36	28.96	3.77	
1	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	
2	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.68	12.76	3.79	
3	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.61	10.93	3.28	
4	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	
5	Tetris	GB	1989.0	Puzzle	Nintendo	23.20	2.26	4.22	
6	New Super Mario Bros.	DS	2006.0	Platform	Nintendo	11.28	9.14	6.50	
7	Wii Play	Wii	2006.0	Misc	Nintendo	13.96	9.18	2.93	
8	New Super Mario Bros. Wii	Wii	2009.0	Platform	Nintendo	14.44	6.94	4.70	
9	Duck Hunt	NES	1984.0	Shooter	Nintendo	26.93	0.63	0.28	

In [5]:
```
#Find the dimensions (number of rows and columns) in the data frame. What do these two
#the data?
rows = len(data.axes[0])
cols = len(data.axes[1])
print("Number of Rows: ", rows)
print("Number of Columns: ", cols)
```

```
Number of Rows:  16719
Number of Columns:  16
```

This tells me that there are 16 categories and 16,719 games.

In [13]:
```
#Find the top five games by critic score.
top5_critic_score = data.nlargest(5, 'Critic_Score')
top5_critic_score
```

Out[13]:

	Name	Platform	Year_of_Release	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other
51	Grand Theft Auto IV	X360	2008.0	Action	Take-Two Interactive	6.76	3.07	0.14	
57	Grand Theft Auto IV	PS3	2008.0	Action	Take-Two Interactive	4.76	3.69	0.44	
227	Tony Hawk's Pro Skater 2	PS	2000.0	Sports	Activision	3.05	1.41	0.02	
5350	SoulCalibur	DC	1999.0	Fighting	Namco Bandai Games	0.00	0.00	0.34	
16	Grand Theft Auto V	PS3	2013.0	Action	Take-Two Interactive	7.02	9.09	0.98	

In [14]:
```
#Find the number of video games in the data frame in each genre.
number_eachgenre = data['Genre'].value_counts()
```

Out[14]:
```
Action          3370
Sports          2348
Misc            1750
Role-Playing    1500
Shooter         1323
Adventure       1303
Racing          1249
Platform         888
Simulation       874
Fighting         849
Strategy         683
Puzzle           580
Name: Genre, dtype: int64
```

In [33]:
```
#Find the first five games in the data frame on the SNES platform
snesplatform_df = data[data['Platform'] == 'SNES']
snes_five = snesplatform_df.head(5)
snes_five
```

Out[33]:

	Name	Platform	Year_of_Release	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sale
18	Super Mario World	SNES	1990.0	Platform	Nintendo	12.78	3.75	3.54	0.5
56	Super Mario All-Stars	SNES	1993.0	Platform	Nintendo	5.99	2.15	2.12	0.2
71	Donkey Kong Country	SNES	1994.0	Platform	Nintendo	4.36	1.71	3.00	0.2
76	Super Mario Kart	SNES	1992.0	Racing	Nintendo	3.54	1.24	3.81	0.1
137	Street Fighter II: The World Warrior	SNES	1992.0	Fighting	Capcom	2.47	0.83	2.87	0.1

In [56]:
```
#Find the five publishers with the highest total global sales. Note: You will need to
#each publisher to do this.
publisher_globalsales = data.groupby(['Publisher'])['Global_Sales'].sum().nlargest(5)
publisher_globalsales
```

Out[56]:
```
Publisher
Nintendo                        1788.81
Electronic Arts                 1116.96
Activision                       731.16
Sony Computer Entertainment      606.48
Ubisoft                          471.61
Name: Global_Sales, dtype: float64
```

In [81]:
```
#Create a new column in the data frame that calculates the percentage of global sales
#rows of the new data frame.
sale_list = ['NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales', 'Global_Sales']
data['total_sale'] = data[sale_list].sum(axis = 1)
```

In [82]:
```
data['NA_percentage'] = (data['NA_Sales'] /
                         data['total_sale'])
```

In [83]:
```
new_five = data.head(5)
new_five
```

Out[83]:

	Name	Platform	Year_of_Release	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_S
0	Wii Sports	Wii	2006.0	Sports	Nintendo	41.36	28.96	3.77	
1	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	
2	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.68	12.76	3.79	
3	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.61	10.93	3.28	
4	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	

In [21]:
```
#Find the number NaN entries (missing data values) in each column.
data.isna().sum()
```

Out[21]:
```
Name                 2
Platform             0
Year_of_Release    269
Genre                2
Publisher           54
NA_Sales             0
EU_Sales             0
JP_Sales             0
Other_Sales          0
Global_Sales         0
Critic_Score      8582
Critic_Count      8582
User_Score        6704
User_Count        9129
Developer         6623
Rating            6769
dtype: int64
```

In [89]:
```
#Try to calculate the median user score of all the video games. You will likely run ir
#entries are a non-numerical string that cannot be converted to a float. Find and repl
#calculate the median. Then, replace all NaN entries in the user score column with the
data['User_Score'] = pd.to_numeric(data['User_Score'], errors='coerce')
```

In [86]:
```
data['User_Score'].median()
```

Out[86]:
```
7.5
```

In [90]:
```
data['User_Score'].fillna(7.5)
```

Out[90]:
```
0        8.0
1        7.5
2        8.3
3        8.0
4        7.5
        ...
16714    7.5
16715    7.5
16716    7.5
16717    7.5
16718    7.5
Name: User_Score, Length: 16719, dtype: float64
```

In [ ]: