DataMiningWeek1

March 17, 2024

[3]:
```
import numpy as np
import pandas as pd

df = pd.read_csv('/Users/nickblackford/Desktop/Python/
 ↪Video_Games_Sales_as_at_22_Dec_2016.csv')
```

[7]:
```
# Display the first ten rows of data
first_ten_rows = df.head(10)
first_ten_rows
```

[7]:

	Name	Platform	Year_of_Release	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Critic_Score	Critic_Count	User_Score	User_Count	Developer	Rating
0	Wii Sports	Wii	2006.0	Sports	Nintendo	41.36	28.96	3.77	8.45	82.53	76.0	51.0	8	322.0	Nintendo	E
1	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24	NaN	NaN	NaN	NaN	NaN	NaN
2	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.68	12.76	3.79	3.29	35.52						
3	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.61	10.93	3.28	2.95	32.77						
4	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	31.37						
5	Tetris	GB	1989.0	Puzzle	Nintendo	23.20	2.26	4.22	0.58	30.26						
6	New Super Mario Bros.	DS	2006.0	Platform	Nintendo	11.28	9.14	6.50	2.88	29.80						
7	Wii Play	Wii	2006.0	Misc	Nintendo	13.96	9.18	2.93	2.84	28.92						
8	New Super Mario Bros. Wii	Wii	2009.0	Platform	Nintendo	14.44	6.94	4.70	2.24	28.32						
9	Duck Hunt	NES	1984.0	Shooter	Nintendo	26.93	0.63	0.28	0.47	28.31						

```
2          82.0          73.0        8.3       709.0  Nintendo      E
3          80.0          73.0          8       192.0  Nintendo      E
4           NaN           NaN        NaN         NaN       NaN    NaN
5           NaN           NaN        NaN         NaN       NaN    NaN
6          89.0          65.0        8.5       431.0  Nintendo      E
7          58.0          41.0        6.6       129.0  Nintendo      E
8          87.0          80.0        8.4       594.0  Nintendo      E
9           NaN           NaN        NaN         NaN       NaN    NaN
```

[9]:
```
# Find the dimensions (number of rows and columns) in the data frame
dimensions = df.shape
dimensions
```

[9]:
```
(16719, 16)
```

The dimensions represent 16,719 video game sales with 16 differnt attributes describing each sale.

[10]:
```
# Find the top five games by critic score
top_five_games_by_critic_score = df.sort_values(by='Critic_Score',␣
 ↪ascending=False).head(5)
top_five_games_by_critic_score
```

[10]:
```
                         Name Platform  Year_of_Release     Genre  \
227   Tony Hawk's Pro Skater 2       PS           2000.0    Sports
57         Grand Theft Auto IV      PS3           2008.0    Action
51         Grand Theft Auto IV     X360           2008.0    Action
5350               SoulCalibur       DC           1999.0  Fighting
165         Grand Theft Auto V     XOne           2014.0    Action

                Publisher  NA_Sales  EU_Sales  JP_Sales  Other_Sales  \
227            Activision      3.05      1.41      0.02         0.20
57    Take-Two Interactive      4.76      3.69      0.44         1.61
51    Take-Two Interactive      6.76      3.07      0.14         1.03
5350   Namco Bandai Games       0.00      0.00      0.34         0.00
165   Take-Two Interactive      2.81      2.19      0.00         0.47

      Global_Sales  Critic_Score  Critic_Count User_Score  User_Count  \
227           4.68          98.0          19.0        7.7       299.0
57           10.50          98.0          64.0        7.5      2833.0
51           11.01          98.0          86.0        7.9      2951.0
5350          0.34          98.0          24.0        8.8       200.0
165           5.48          97.0          14.0        7.9       764.0

                    Developer Rating
227   Neversoft Entertainment      T
57             Rockstar North      M
51             Rockstar North      M
```

```
5350                    Namco      T
165            Rockstar North      M
```

[11]:
```
# Find the number of video games in the data frame in each genre
games_in_each_genre = df['Genre'].value_counts()
games_in_each_genre
```

[11]:
```
Genre
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: count, dtype: int64
```

[13]:
```
# Find the first five games in the data frame on the SNES platform
first_five_games_snes = df[df['Platform'] == 'SNES'].head(5)
first_five_games_snes
```

[13]:
```
                                    Name Platform  Year_of_Release     Genre  \
18                     Super Mario World     SNES           1990.0  Platform
56                 Super Mario All-Stars     SNES           1993.0  Platform
71                   Donkey Kong Country     SNES           1994.0  Platform
76                      Super Mario Kart     SNES           1992.0    Racing
137  Street Fighter II: The World Warrior     SNES           1992.0  Fighting

    Publisher  NA_Sales  EU_Sales  JP_Sales  Other_Sales  Global_Sales  \
18   Nintendo     12.78      3.75      3.54         0.55         20.61
56   Nintendo      5.99      2.15      2.12         0.29         10.55
71   Nintendo      4.36      1.71      3.00         0.23          9.30
76   Nintendo      3.54      1.24      3.81         0.18          8.76
137    Capcom      2.47      0.83      2.87         0.12          6.30

     Critic_Score  Critic_Count User_Score  User_Count Developer Rating
18            NaN           NaN        NaN         NaN       NaN    NaN
56            NaN           NaN        NaN         NaN       NaN    NaN
71            NaN           NaN        NaN         NaN       NaN    NaN
76            NaN           NaN        NaN         NaN       NaN    NaN
137           NaN           NaN        NaN         NaN       NaN    NaN
```

[14]:
```
# Find the five publishers with the highest total global sales
total_global_sales_by_publisher = df.groupby('Publisher')['Global_Sales'].sum().
 ↪sort_values(ascending=False).head(5)
total_global_sales_by_publisher
```

[14]:
```
Publisher
Nintendo                        1788.81
Electronic Arts                 1116.96
Activision                       731.16
Sony Computer Entertainment      606.48
Ubisoft                          471.61
Name: Global_Sales, dtype: float64
```

[16]:
```
# Create a new column in the data frame that calculates the percentage of␣
 ↪global sales from North America
df['NA_Sales_Percentage'] = (df['NA_Sales'] / df['Global_Sales']) * 100
new_df_first_five_rows = df.head(5)
new_df_first_five_rows
```

[16]:

	Name	Platform	Year_of_Release	Genre	Publisher	\
0	Wii Sports	Wii	2006.0	Sports	Nintendo	
1	Super Mario Bros.	NES	1985.0	Platform	Nintendo	
2	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	
3	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	
4	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	

	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Critic_Score	\
0	41.36	28.96	3.77	8.45	82.53	76.0	
1	29.08	3.58	6.81	0.77	40.24	NaN	
2	15.68	12.76	3.79	3.29	35.52	82.0	
3	15.61	10.93	3.28	2.95	32.77	80.0	
4	11.27	8.89	10.22	1.00	31.37	NaN	

	Critic_Count	User_Score	User_Count	Developer	Rating	NA_Sales_Percentage
0	51.0	8	322.0	Nintendo	E	50.115110
1	NaN	NaN	NaN	NaN	NaN	72.266402
2	73.0	8.3	709.0	Nintendo	E	44.144144
3	73.0	8	192.0	Nintendo	E	47.635032
4	NaN	NaN	NaN	NaN	NaN	35.926044

[19]:
```
# Find the number NaN entries in each column
nan_entries_each_column = df.isna().sum()
nan_entries_each_column
```

[19]:
```
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
NA_Sales_Percentage        0
dtype: int64
```

[36]:
```
# Try to calculate the median user score of all the video games
import numpy as np

median_user_score = df['User_Score'].median()


# Find and replace this string with NaN and then calculate the median
df['User_Score'] = df['User_Score'].replace('tbd', np.nan).astype(float)
median_user_score_after_replacement = df['User_Score'].median()

# Replace all NaN entries in the user score column with the median value
df['User_Score'].fillna(median_user_score_after_replacement, inplace=True)
```

[30]:
```
median_user_score
```

[30]:
```
7.5
```

[31]:
```
median_user_score_after_replacement
```

[31]:
```
7.5
```

[37]:
```
df
```

[37]:
```
                              Name Platform  Year_of_Release         Genre  \
0                       Wii Sports      Wii           2006.0        Sports
1                Super Mario Bros.      NES           1985.0      Platform
2                   Mario Kart Wii      Wii           2008.0        Racing
3                Wii Sports Resort      Wii           2009.0        Sports
4          Pokemon Red/Pokemon Blue       GB           1996.0  Role-Playing
…                               …      …              …            …
16714  Samurai Warriors: Sanada Maru      PS3           2016.0        Action
```

```
16715               LMA Manager 2007     X360           2006.0        Sports
16716       Haitaka no Psychedelica      PSV           2016.0     Adventure
16717                Spirits & Spells      GBA           2003.0      Platform
16718            Winning Post 8 2016      PSV           2016.0    Simulation

          Publisher  NA_Sales  EU_Sales  JP_Sales  Other_Sales  Global_Sales  \
0          Nintendo     41.36     28.96      3.77         8.45         82.53
1          Nintendo     29.08      3.58      6.81         0.77         40.24
2          Nintendo     15.68     12.76      3.79         3.29         35.52
3          Nintendo     15.61     10.93      3.28         2.95         32.77
4          Nintendo     11.27      8.89     10.22         1.00         31.37
…               …       …        …         …            …             …
16714    Tecmo Koei      0.00      0.00      0.01         0.00          0.01
16715   Codemasters      0.00      0.01      0.00         0.00          0.01
16716  Idea Factory      0.00      0.00      0.01         0.00          0.01
16717       Wanadoo      0.01      0.00      0.00         0.00          0.01
16718    Tecmo Koei      0.00      0.00      0.01         0.00          0.01

       Critic_Score  Critic_Count  User_Score  User_Count Developer Rating  \
0              76.0          51.0         8.0       322.0  Nintendo      E
1               NaN           NaN         7.5         NaN       NaN    NaN
2              82.0          73.0         8.3       709.0  Nintendo      E
3              80.0          73.0         8.0       192.0  Nintendo      E
4               NaN           NaN         7.5         NaN       NaN    NaN
…               …             …           …           …    …
16714           NaN           NaN         7.5         NaN       NaN    NaN
16715           NaN           NaN         7.5         NaN       NaN    NaN
16716           NaN           NaN         7.5         NaN       NaN    NaN
16717           NaN           NaN         7.5         NaN       NaN    NaN
16718           NaN           NaN         7.5         NaN       NaN    NaN

       NA_Sales_Percentage
0                50.115110
1                72.266402
2                44.144144
3                47.635032
4                35.926044
…                      …
16714             0.000000
16715             0.000000
16716             0.000000
16717           100.000000
16718             0.000000

[16719 rows x 17 columns]
```

[ ]: