C550-T301-Data-Mining_2241-1_week1_Samanta_Rajib

September 3, 2023

0.1 Class : C550-T301 Data Mining (2241-1)

0.2 Name : Rajib Samanta

0.2.1 Assignment : Week 1

Download the Video Game Sales with Ratings dataset from this link: Video Game Sales with Ratings. https://www.kaggle.com/code/jayagopal20/video-game-stats

1. Load the dataset as a Pandas data frame.
2. Display the first ten rows of data.
3. Find the dimensions (number of rows and columns) in the data frame. What do these two numbers represent in the context of the data?
4. Find the top five games by critic score.
5. Find the number of video games in the data frame in each genre.
6. Find the first five games in the data frame on the SNES platform.
7. Find the five publishers with the highest total global sales. Note: You will need to calculate the total global sales for each publisher to do this.
8. Create a new column in the data frame that calculates the percentage of global sales from North America. Display the first five rows of the new data frame.
9. Find the number NaN entries (missing data values) in each column.
10. Try to calculate the median user score of all the video games. You will likely run into an error because some of the user score entries are a non-numerical string that cannot be converted to a float. Find and replace this string with NaN and then calculate the median. Then, replace all NaN entries in the user score column with the median value.

[23]:

```
# Load the Libraries
import os
import pandas as pd
```

[24]:

```
# 1. Load the dataset as a Pandas data frame.
# 2. Display the first ten rows of data.
# Read in the Video Game Sales with Ratings data file␣
  ↪('Video_Games_Sales_as_at_22_Dec_2016.csv') from local:
directory = '/Users/rajibsamanta/Documents/Rajib/College/Sem6_fall_2023/week1'
# Set the working directory
os.chdir(directory)
print(os.getcwd())
dataset1_csv = pd.read_csv("Video_Games_Sales_as_at_22_Dec_2016.csv")
dataset1_csv.head(10)
```

```
# Display the DataFrame 10 rows
```

```
/Users/rajibsamanta/Documents/Rajib/College/Sem6_fall_2023/week1
```

[24]:

```
                         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   
5                      Tetris       GB           1989.0        Puzzle   
6       New Super Mario Bros.       DS           2006.0      Platform   
7                    Wii Play      Wii           2006.0          Misc   
8   New Super Mario Bros. Wii      Wii           2009.0      Platform   
9                   Duck Hunt      NES           1984.0       Shooter   

  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   
5  Nintendo     23.20      2.26      4.22         0.58         30.26   
6  Nintendo     11.28      9.14      6.50         2.88         29.80   
7  Nintendo     13.96      9.18      2.93         2.84         28.92   
8  Nintendo     14.44      6.94      4.70         2.24         28.32   
9  Nintendo     26.93      0.63      0.28         0.47         28.31   

   Critic_Score  Critic_Count User_Score  User_Count Developer Rating  
0          76.0          51.0          8       322.0  Nintendo      E  
1           NaN           NaN        NaN         NaN       NaN    NaN  
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  
```

[25]:

```
# describe the data set
dataset1_csv.describe()
```

[25]:

```
       Year_of_Release      NA_Sales      EU_Sales      JP_Sales  \
count     16450.000000  16719.000000  16719.000000  16719.000000   
mean       2006.487356      0.263330      0.145025      0.077602   
std           5.878995      0.813514      0.503283      0.308818   
min        1980.000000      0.000000      0.000000      0.000000   
```

```
25%        2003.000000      0.000000      0.000000      0.000000
50%        2007.000000      0.080000      0.020000      0.000000
75%        2010.000000      0.240000      0.110000      0.040000
max        2020.000000     41.360000     28.960000     10.220000

        Other_Sales  Global_Sales  Critic_Score  Critic_Count    User_Count
count  16719.000000  16719.000000   8137.000000   8137.000000   7590.000000
mean       0.047332      0.533543     68.967679     26.360821    162.229908
std        0.186710      1.547935     13.938165     18.980495    561.282326
min        0.000000      0.010000     13.000000      3.000000      4.000000
25%        0.000000      0.060000     60.000000     12.000000     10.000000
50%        0.010000      0.170000     71.000000     21.000000     24.000000
75%        0.030000      0.470000     79.000000     36.000000     81.000000
max       10.570000     82.530000     98.000000    113.000000  10665.000000
```

[26]:

```
# 3. Find the dimensions (number of rows and columns) in the data frame.
#    What do these two numbers represent in the context of the data?

# DataFrame is named dataset1_csv
num_rows, num_columns = dataset1_csv.shape
print(' Number of rows '+ str(num_rows))
print(' Number of Columns '+ str(num_columns))

# Number of Rows (num_rows): This represents the total number of observations␣
 ↪or data points in your dataset. Each row typically corresponds to a single␣
 ↪data entry or record.
```

```
 Number of rows 16719
 Number of Columns 16
```

Number of Rows (num_rows): 16719

```
    This represents the total number of observations or data points in your dataset.
    Each row typically corresponds to a single data entry or record.
```

Number of Columns (num_columns): 16

```
    This represents the number of variables or features in your dataset. Each column typically
    corresponds to a different attribute or characteristic of the data. Columns can contain va
    types of information, such as numerical values, text, dates, or categorical data.
Understanding the dimensions of DataFrame is essential for data analysis because it helps to g
size and structure of  dataset, which is fundamental for making data-driven decisions and perf
various data manipulation and analysis tasks.
```

[27]:

```
# 4 - Find the top five games by critic score.
# Critic_score - Aggregate score compiled by Metacritic staff
# Sort the DataFrame by 'Critic_Score' column in descending order and get the␣
 ↪top 5 rows
```

```
top_five_games = dataset1_csv.nlargest(5, 'Critic_Score')

# Display the top five games by critic score in table format
#print(top_five_games.to_string)
top_five_games.head()
```

[27]:

```
                          Name Platform  Year_of_Release     Genre  \
51         Grand Theft Auto IV     X360           2008.0    Action
57         Grand Theft Auto IV      PS3           2008.0    Action
227   Tony Hawk's Pro Skater 2       PS           2000.0    Sports
5350               SoulCalibur       DC           1999.0  Fighting
16          Grand Theft Auto V      PS3           2013.0    Action

                 Publisher  NA_Sales  EU_Sales  JP_Sales  Other_Sales  \
51    Take-Two Interactive      6.76      3.07      0.14         1.03
57    Take-Two Interactive      4.76      3.69      0.44         1.61
227             Activision      3.05      1.41      0.02         0.20
5350    Namco Bandai Games      0.00      0.00      0.34         0.00
16    Take-Two Interactive      7.02      9.09      0.98         3.96

      Global_Sales  Critic_Score  Critic_Count User_Score  User_Count  \
51           11.01          98.0          86.0        7.9      2951.0
57           10.50          98.0          64.0        7.5      2833.0
227           4.68          98.0          19.0        7.7       299.0
5350          0.34          98.0          24.0        8.8       200.0
16           21.04          97.0          50.0        8.2      3994.0

                    Developer Rating
51             Rockstar North      M
57             Rockstar North      M
227   Neversoft Entertainment      T
5350                    Namco      T
16             Rockstar North      M
```

[28]:

```
# 5 - Find the number of video games in the data frame in each genre.
## count based on 'Genre' column
genre_counts = dataset1_csv['Genre'].value_counts()

# Display the number of video games in each genre
print(genre_counts)
```

```
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
```

[29]:

```
# 6. Find the first five games in the data frame on the SNES platform.
snes_games = dataset1_csv[dataset1_csv['Platform'] == 'SNES'].head(5)

# Display the first five games on the SNES platform
snes_games.head()
```

[29]:

```
                                     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
```

[30]:

```
# 7. Find the five publishers with the highest total global sales.
#.    Note: You will need to calculate the total global sales for each␣
 ↪publisher to do this.

# Group the DataFrame by 'Publisher' and calculate the sum of 'Global_Sales'␣
 ↪for each publisher
publisher_sales = dataset1_csv.groupby('Publisher')['Global_Sales'].sum()

# Sort the publishers by total global sales in descending order and get the top␣
 ↪5
top_publishers = publisher_sales.sort_values(ascending=False).head(5)

# Display the top 5 publishers with the highest total global sales
```

```
print(top_publishers)
```

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

[31]:

```
# 8. Create a new column in the data frame that calculates the percentage of␣
 ↪global sales from North America.
#.  Display the first five rows of the new data frame.
# --> New column name is 'NA_Sales_Percentage'

# Calculate the percentage of global sales from North America and store it in a␣
 ↪new column 'NA_Sales_Percentage'
dataset1_csv['NA_Sales_Percentage'] = (dataset1_csv['NA_Sales'] /␣
 ↪dataset1_csv['Global_Sales']) * 100

# Display the first five rows of the DataFrame with the new column
dataset1_csv.head()
```

[31]:

```
                       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
```

[32]:

```
# 9. Find the number NaN entries (missing data values) in each column.
# Count the number of NaN entries in each column
nan_counts = dataset1_csv.isna().sum()
```

```
# Display the number of NaN entries in each column
print(nan_counts)
```

```
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
```

[33]:

```
# 10 . Try to calculate the median user score of all the video games.
#   You will likely run into an error because some of the user score entries␣
 ↪are a non-numerical string that cannot be converted to a float.
#.  Find and replace this string with NaN and then calculate the median.
#  Then, replace all NaN entries in the user score column with the median value.

# Replace non-numeric strings with NaN in the 'User_Score' column
dataset1_csv['User_Score'] = pd.to_numeric(dataset1_csv['User_Score'],␣
 ↪errors='coerce')

# Calculate the median of the 'User_Score' column, excluding NaN values
median_user_score = dataset1_csv['User_Score'].median()

# Replace NaN entries in the 'User_Score' column with the median value
dataset1_csv['User_Score'].fillna(median_user_score, inplace=True)

# Display the median user score and  few rows of the DataFrame
print(f"Median User Score: {median_user_score}")
dataset1_csv.head(10)
```

```
Median User Score: 7.5
```

[33]:

```
                       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
5                     Tetris       GB           1989.0        Puzzle
6      New Super Mario Bros.       DS           2006.0      Platform
7                   Wii Play      Wii           2006.0          Misc
8  New Super Mario Bros. Wii      Wii           2009.0      Platform
9                  Duck Hunt      NES           1984.0       Shooter

  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
5  Nintendo     23.20      2.26      4.22         0.58         30.26
6  Nintendo     11.28      9.14      6.50         2.88         29.80
7  Nintendo     13.96      9.18      2.93         2.84         28.92
8  Nintendo     14.44      6.94      4.70         2.24         28.32
9  Nintendo     26.93      0.63      0.28         0.47         28.31

   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
5           NaN           NaN         7.5         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         7.5         NaN       NaN    NaN

   NA_Sales_Percentage
0            50.115110
1            72.266402
2            44.144144
3            47.635032
4            35.926044
5            76.668870
6            37.852349
7            48.271093
8            50.988701
9            95.125397
```

```
[ ]:
```