Miles A. Peña

DSC 550

Week 1

March 17, 2024

1. Load the dataset as a Pandas data frame.

In [3]:
```
import pandas as pd
```

In [4]:
```
# load csv file through read_csv

dataframe = pd.read_csv('Video Game Sales with Ratings.csv')
```

2. Display the first ten rows of data.

In [5]:
```
# display first 10 rows

dataframe.head(10)
```

Out[5]:

	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	82.0	73.0	8.3	709.0	Nintendo	E
3	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.61	10.93	3.28	2.95	32.77	80.0	73.0	8	192.0	Nintendo	E
4	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	31.37	NaN	NaN	NaN	NaN	NaN	NaN
5	Tetris	GB	1989.0	Puzzle	Nintendo	23.20	2.26	4.22	0.58	30.26	NaN	NaN	NaN	NaN	NaN	NaN
6	New Super Mario Bros.	DS	2006.0	Platform	Nintendo	11.28	9.14	6.50	2.88	29.80	89.0	65.0	8.5	431.0	Nintendo	E
7	Wii Play	Wii	2006.0	Misc	Nintendo	13.96	9.18	2.93	2.84	28.92	58.0	41.0	6.6	129.0	Nintendo	E
8	New Super Mario Bros. Wii	Wii	2009.0	Platform	Nintendo	14.44	6.94	4.70	2.24	28.32	87.0	80.0	8.4	594.0	Nintendo	E
9	Duck Hunt	NES	1984.0	Shooter	Nintendo	26.93	0.63	0.28	0.47	28.31	NaN	NaN	NaN	NaN	NaN	NaN

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?

In [6]:
```
dataframe.shape
```

Out[6]:
```
(16719, 16)
```

In [66]:
```
# number of rows

print("Number of Rows: ", len(dataframe))
```

```
Number of Rows:  16719
```

In [67]:
```
# number of columns; actual number of original columns is 16, number changed once I ran the cell after the percetage cell was added

print("Number of Columns: ", len(dataframe.columns))
```

```
Number of Columns:  17
```

In the context of the data, the number of rows means that there are 16,719 video games that were included in the dataset and that there are 16 variables associated with each game that are used to measure sales as well as other identifiers specific to the game.

4. Find the top five games by critic score.

In [23]:
```
# sort values by critic score and sort from high to low

dataframe.sort_values('Critic_Score', ascending = False).head(5)
```

Out[23]:

	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
227	Tony Hawk's Pro Skater 2	PS	2000.0	Sports	Activision	3.05	1.41	0.02	0.20	4.68	98.0	19.0	7.7	299.0	Neversoft Entertainment	T
57	Grand Theft Auto IV	PS3	2008.0	Action	Take-Two Interactive	4.76	3.69	0.44	1.61	10.50	98.0	64.0	7.5	2833.0	Rockstar North	M
51	Grand Theft Auto IV	X360	2008.0	Action	Take-Two Interactive	6.76	3.07	0.14	1.03	11.01	98.0	86.0	7.9	2951.0	Rockstar North	M
5350	SoulCalibur	DC	1999.0	Fighting	Namco Bandai Games	0.00	0.00	0.34	0.00	0.34	98.0	24.0	8.8	200.0	Namco	T
165	Grand Theft Auto V	XOne	2014.0	Action	Take-Two Interactive	2.81	2.19	0.00	0.47	5.48	97.0	14.0	7.9	764.0	Rockstar North	M

In [26]:
```
# another way to get top 5 games by critic score using 'nlargest'

dataframe.nlargest(n = 5, columns = 'Critic_Score')
```

Out[26]:

	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
51	Grand Theft Auto IV	X360	2008.0	Action	Take-Two Interactive	6.76	3.07	0.14	1.03	11.01	98.0	86.0	7.9	2951.0	Rockstar North	M
57	Grand Theft Auto IV	PS3	2008.0	Action	Take-Two Interactive	4.76	3.69	0.44	1.61	10.50	98.0	64.0	7.5	2833.0	Rockstar North	M
227	Tony Hawk's Pro Skater 2	PS	2000.0	Sports	Activision	3.05	1.41	0.02	0.20	4.68	98.0	19.0	7.7	299.0	Neversoft Entertainment	T
5350	SoulCalibur	DC	1999.0	Fighting	Namco Bandai Games	0.00	0.00	0.34	0.00	0.34	98.0	24.0	8.8	200.0	Namco	T
16	Grand Theft Auto V	PS3	2013.0	Action	Take-Two Interactive	7.02	9.09	0.98	3.96	21.04	97.0	50.0	8.2	3994.0	Rockstar North	M

5. Find the number of video games in the data frame in each genre.

In [35]:
```
# get counts by genre

genre_counts = dataframe['Genre'].value_counts()
```

In [36]:
```
genre_counts
```

Out[36]:
```
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
```

6. Find the first five games in the data frame on the SNES platform.

In [37]:
```
# find games specifically on SNES platform

dataframe[dataframe['Platform'] == 'SNES'].head(5)
```

Out[37]:

	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
18	Super Mario World	SNES	1990.0	Platform	Nintendo	12.78	3.75	3.54	0.55	20.61	NaN	NaN	NaN	NaN	NaN	NaN
56	Super Mario All-Stars	SNES	1993.0	Platform	Nintendo	5.99	2.15	2.12	0.29	10.55	NaN	NaN	NaN	NaN	NaN	NaN
71	Donkey Kong Country	SNES	1994.0	Platform	Nintendo	4.36	1.71	3.00	0.23	9.30	NaN	NaN	NaN	NaN	NaN	NaN
76	Super Mario Kart	SNES	1992.0	Racing	Nintendo	3.54	1.24	3.81	0.18	8.76	NaN	NaN	NaN	NaN	NaN	NaN
137	Street Fighter II: The World Warrior	SNES	1992.0	Fighting	Capcom	2.47	0.83	2.87	0.12	6.30	NaN	NaN	NaN	NaN	NaN	NaN

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.

In [14]:
```
global_sales = dataframe.groupby('Publisher')['Global_Sales'].sum()
```

In [32]:
```
global_sales.sort_values(ascending = False).head(5)
```

Out[32]:
```
Publisher
Nintendo                  1788.81
Electronic Arts           1116.96
Activision                 731.16
Sony Computer Entertainment    606.48
Ubisoft                    471.61
Name: Global_Sales, dtype: float64
```

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.

In [45]:
```
# calculate percentage by dividing NA sales by global sales and multiplying by 100

dataframe['Percentage_NA_Sales'] = dataframe['NA_Sales'] / dataframe['Global_Sales'] * 100
dataframe.head(5)
```

Out[45]:

	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	Percentage_NA_S
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	50.11
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	72.26
2	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.68	12.76	3.79	3.29	35.52	82.0	73.0	8.3	709.0	Nintendo	E	44.14
3	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.61	10.93	3.28	2.95	32.77	80.0	73.0	8	192.0	Nintendo	E	47.63
4	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	31.37	NaN	NaN	NaN	NaN	NaN	NaN	35.92

9. Find the number NaN entries (missing data values) in each column.

In [50]:
```
# utilize isna() or isnull() to find missing data values

dataframe.isna().sum()
```

Out[50]:
```
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
Percentage_NA_Sales      0
dtype: int64
```

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.

In [51]:
```
# attempt to find median (will error due to str)

dataframe['User_Score'].median()
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
File /Applications/anaconda3/lib/python3.11/site-packages/pandas/core/nanops.py:786, in nanmedian(values, axis, skipna, mask)
    785 try:
--> 786     values = values.astype("f8")
    787 except ValueError as err:
    788     # e.g. "could not convert string to float: 'a'"

ValueError: could not convert string to float: 'tbd'

The above exception was the direct cause of the following exception:

TypeError                                 Traceback (most recent call last)
Cell In[51], line 1
----> 1 dataframe['User_Score'].median()

File /Applications/anaconda3/lib/python3.11/site-packages/pandas/core/generic.py:11623, in NDFrame._add_numeric_operations.<locals>.median(self, axis, skipna, numeric_only, **kwargs)
  11606 @doc(
  11607     _num_doc,
  11608     desc="Return the median of the values over the requested axis.",
  (...)
  11621     **kwargs,
  11622 ):
> 11623     return NDFrame.median(self, axis, skipna, numeric_only, **kwargs)

File /Applications/anaconda3/lib/python3.11/site-packages/pandas/core/generic.py:11212, in NDFrame.median(self, axis, skipna, numeric_only, **kwargs)
  11205 def median(
  11206     self,
  11207     axis: Axis | None = 0,
  (...)
  11210     **kwargs,
  11211 ) -> Series | float:
> 11212     return self._stat_function(
  11213         "median", nanops.nanmedian, axis, skipna, numeric_only, **kwargs
  11214     )

File /Applications/anaconda3/lib/python3.11/site-packages/pandas/core/generic.py:11158, in NDFrame._stat_function(self, name, func, axis, skipna, numeric_only, **kwargs)
  11154     nv.validate_stat_func((), kwargs, fname=name)
  11156 validate_bool_kwarg(skipna, "skipna", none_allowed=False)
> 11158 return self._reduce(
  11159     func, name=name, axis=axis, skipna=skipna, numeric_only=numeric_only
  11160 )

File /Applications/anaconda3/lib/python3.11/site-packages/pandas/core/series.py:4670, in Series._reduce(self, op, name, axis, skipna, numeric_only, filter_type, **kwds)
   4665     raise TypeError(
   4666         f"Series.{name} does not allow {kwd_name}={numeric_only} "
   4667         "with non-numeric dtypes."
   4668     )
   4669 with np.errstate(all="ignore"):
-> 4670     return op(delegate, skipna=skipna, **kwds)

File /Applications/anaconda3/lib/python3.11/site-packages/pandas/core/nanops.py:158, in bottleneck_switch.__call__.<locals>.f(values, axis, skipna, **kwds)
    156         result = alt(values, axis=axis, skipna=skipna, **kwds)
    157     else:
--> 158         result = alt(values, axis=axis, skipna=skipna, **kwds)
    160 return result

File /Applications/anaconda3/lib/python3.11/site-packages/pandas/core/nanops.py:789, in nanmedian(values, axis, skipna, mask)
    786         values = values.astype("f8")
    787     except ValueError as err:
    788         # e.g. "could not convert string to float: 'a'"
--> 789         raise TypeError(str(err)) from err
    790 if mask is not None:
    791     values[mask] = np.nan

TypeError: could not convert string to float: 'tbd'
```

In [53]:
```
# replace strings with NaN

dataframe['User_Score'] = pd.to_numeric(dataframe['User_Score'], errors = 'coerce')
```

In [64]:
```
# find median

print("Median: ", dataframe['User_Score'].median())
```

```
Median:  7.5
```

In [59]:
```
# replace NaN with median

dataframe['User_Score'].fillna(median, inplace = True)
```

In [63]:
```
# check that median has replaced NaN in dataframe

dataframe.head(5)
```

Out[63]:

	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	Percentage_NA_S
0	Wii Sports	Wii	2006.0	Sports	Nintendo	41.36	28.96	3.77	8.45	82.53	76.0	51.0	8.0	322.0	Nintendo	E	50.11
1	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24	NaN	NaN	7.5	NaN	NaN	NaN	72.26
2	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.68	12.76	3.79	3.29	35.52	82.0	73.0	8.3	709.0	Nintendo	E	44.14
3	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.61	10.93	3.28	2.95	32.77	80.0	73.0	8.0	192.0	Nintendo	E	47.63
4	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	31.37	NaN	NaN	7.5	NaN	NaN	NaN	35.92