data_science

December 1, 2024

[1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:

```
df = pd.read_csv('./vgsales.csv')
```

[3]:

```
df.head(10)
```

[3]:

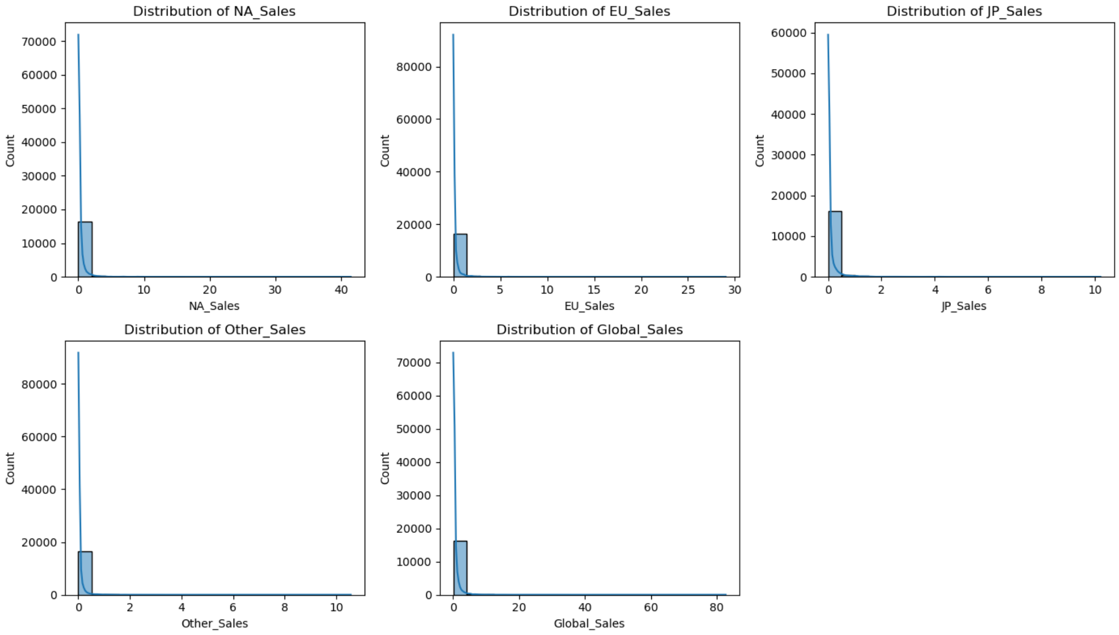
```
   Rank                       Name Platform    Year         Genre Publisher  \
0     1                Wii Sports      Wii  2006.0        Sports  Nintendo
1     2         Super Mario Bros.      NES  1985.0      Platform  Nintendo
2     3            Mario Kart Wii      Wii  2008.0        Racing  Nintendo
3     4         Wii Sports Resort      Wii  2009.0        Sports  Nintendo
4     5  Pokemon Red/Pokemon Blue       GB  1996.0  Role-Playing  Nintendo
5     6                    Tetris       GB  1989.0        Puzzle  Nintendo
6     7     New Super Mario Bros.       DS  2006.0      Platform  Nintendo
7     8                  Wii Play      Wii  2006.0          Misc  Nintendo
8     9 New Super Mario Bros. Wii      Wii  2009.0      Platform  Nintendo
9    10                 Duck Hunt      NES  1984.0       Shooter  Nintendo

   NA_Sales  EU_Sales  JP_Sales  Other_Sales  Global_Sales
0     41.49     29.02      3.77         8.46         82.74
1     29.08      3.58      6.81         0.77         40.24
2     15.85     12.88      3.79         3.31         35.82
3     15.75     11.01      3.28         2.96         33.00
4     11.27      8.89     10.22         1.00         31.37
5     23.20      2.26      4.22         0.58         30.26
6     11.38      9.23      6.50         2.90         30.01
7     14.03      9.20      2.93         2.85         29.02
8     14.59      7.06      4.70         2.26         28.62
9     26.93      0.63      0.28         0.47         28.31
```

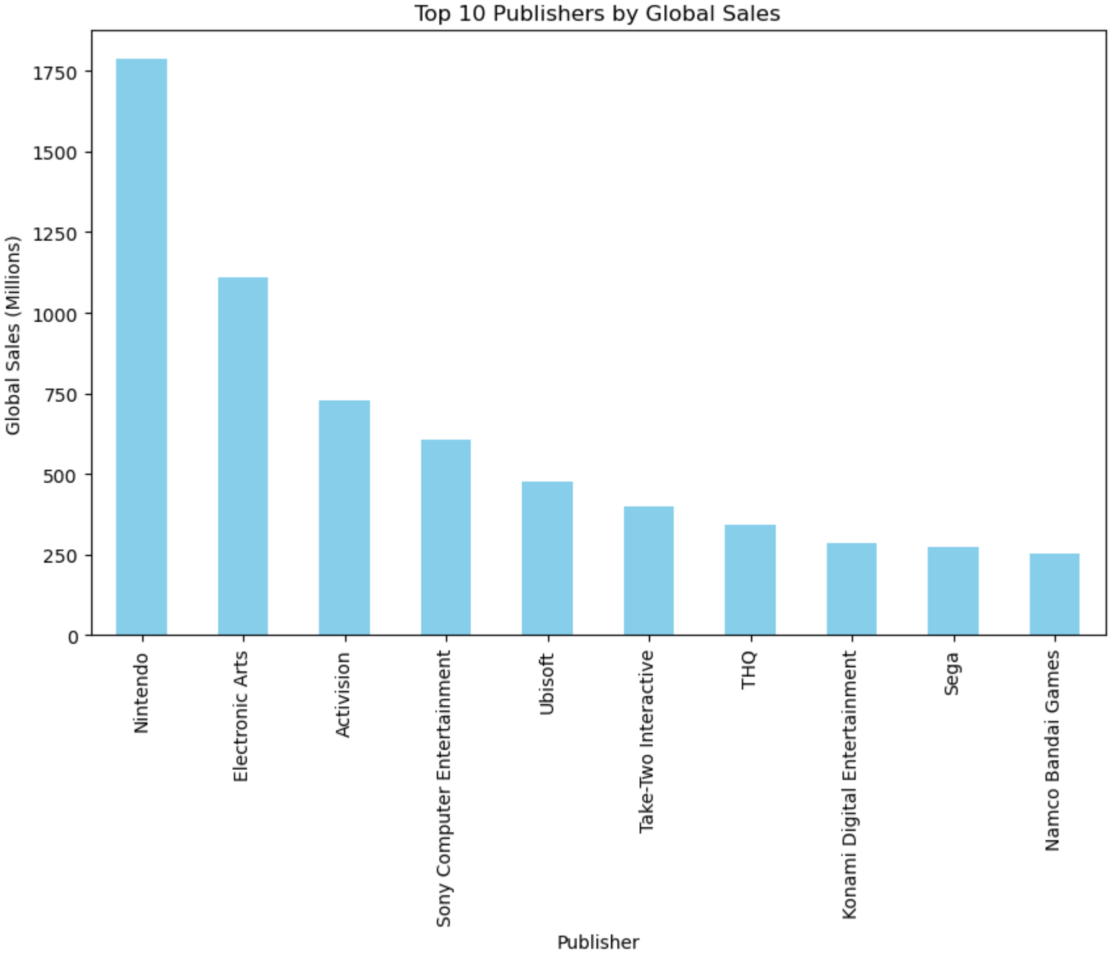
[4]:

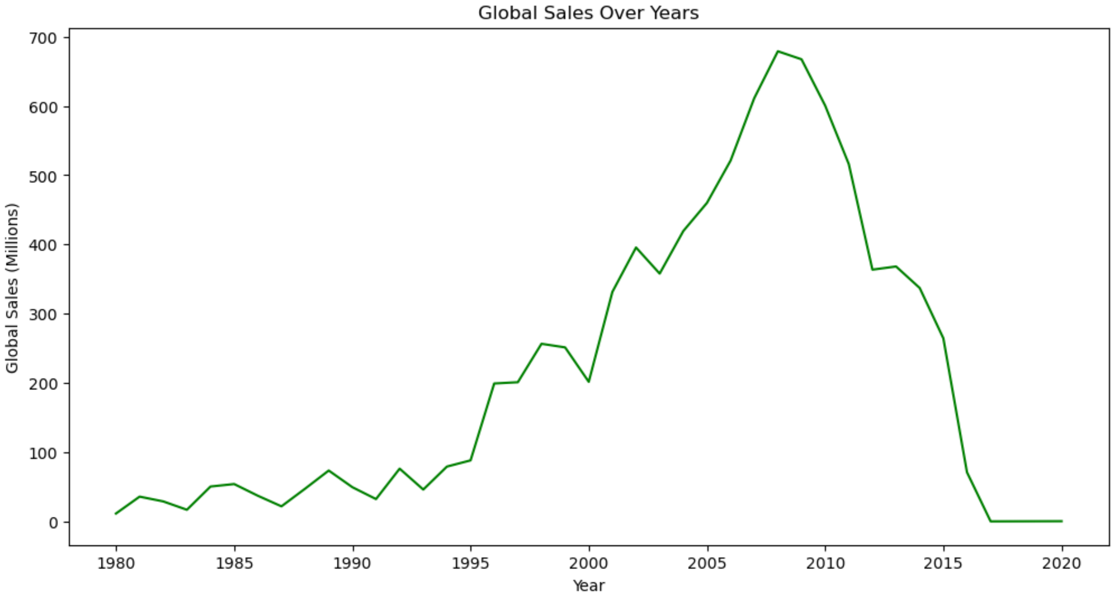
```
# Display the first few rows
print("Dataset Preview:")
display(df.head())

# 1. Dataset Overview
print("\nDataset Information:")
```

```
df.info()

print("\nChecking for Missing Values:")
print(df.isnull().sum())

# 2. Statistical Summary
print("\nStatistical Summary of Numerical Columns:")
print(df.describe())

# 3. Distribution of Numerical Data
numerical_cols = ['NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales',␣
 ↪'Global_Sales']
plt.figure(figsize=(14, 8))
for i, col in enumerate(numerical_cols, 1):
    plt.subplot(2, 3, i)
    sns.histplot(df[col], kde=True, bins=20)
    plt.title(f'Distribution of {col}')
plt.tight_layout()
plt.show()

# 4. Top 10 Publishers by Global Sales
top_publishers = df.groupby('Publisher')['Global_Sales'].sum().nlargest(10)
plt.figure(figsize=(10, 6))
top_publishers.plot(kind='bar', color='skyblue')
plt.title('Top 10 Publishers by Global Sales')
plt.xlabel('Publisher')
plt.ylabel('Global Sales (Millions)')
plt.show()

# 5. Sales Trend Over Years
# Drop rows with missing 'Year' values
df_yearly_sales = df.dropna(subset=['Year'])
yearly_sales = df_yearly_sales.groupby('Year')['Global_Sales'].sum()

plt.figure(figsize=(12, 6))
yearly_sales.plot(kind='line', color='green')
plt.title('Global Sales Over Years')
plt.xlabel('Year')
plt.ylabel('Global Sales (Millions)')
plt.show()

# 6. Genre Distribution
plt.figure(figsize=(12, 6))
sns.countplot(data=df, x='Genre', order=df['Genre'].value_counts().index,␣
 ↪palette='viridis')
plt.title('Distribution of Game Genres')
plt.xticks(rotation=45)
plt.show()

# 7. Platform Analysis - Top 10 Platforms by Total Sales
top_platforms = df.groupby('Platform')['Global_Sales'].sum().nlargest(10)
plt.figure(figsize=(10, 6))
top_platforms.plot(kind='bar', color='salmon')
plt.title('Top 10 Platforms by Global Sales')
plt.xlabel('Platform')
plt.ylabel('Global Sales (Millions)')
plt.show()

# 8. Correlation Analysis
plt.figure(figsize=(10, 6))
sns.heatmap(df[numerical_cols].corr(), annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Matrix of Sales Data')
plt.show()
```

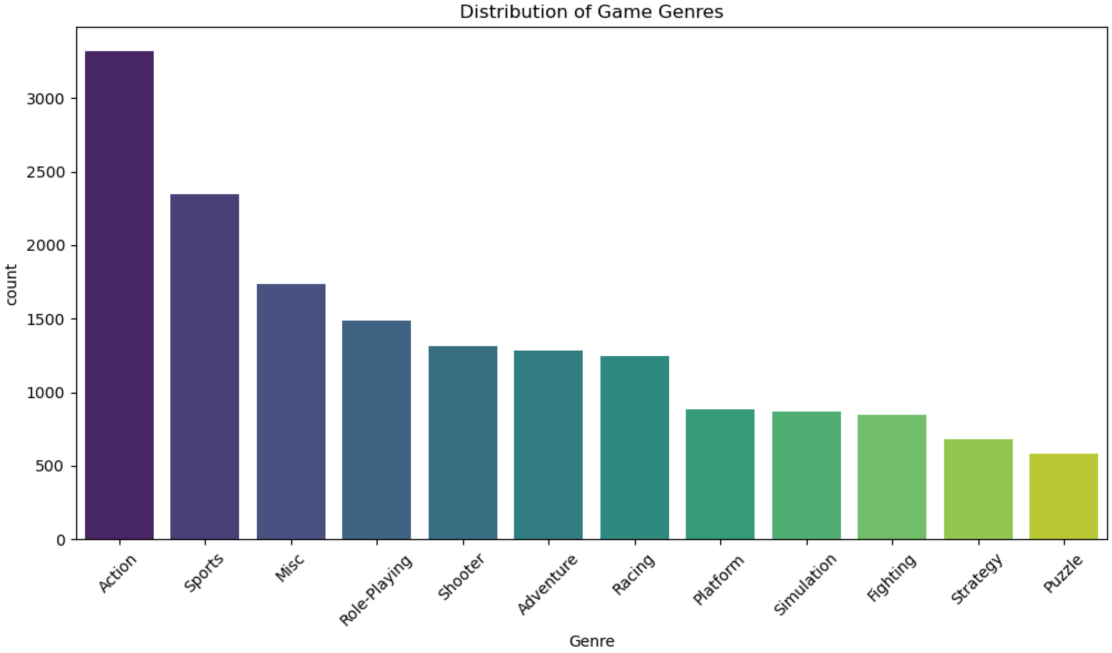
```
Dataset Preview:

   Rank                      Name Platform    Year         Genre Publisher  \
0     1                Wii Sports      Wii  2006.0        Sports  Nintendo
1     2         Super Mario Bros.      NES  1985.0      Platform  Nintendo
2     3            Mario Kart Wii      Wii  2008.0        Racing  Nintendo
3     4         Wii Sports Resort      Wii  2009.0        Sports  Nintendo
4     5  Pokemon Red/Pokemon Blue       GB  1996.0  Role-Playing  Nintendo

   NA_Sales  EU_Sales  JP_Sales  Other_Sales  Global_Sales
0     41.49     29.02      3.77         8.46         82.74
1     29.08      3.58      6.81         0.77         40.24
2     15.85     12.88      3.79         3.31         35.82
3     15.75     11.01      3.28         2.96         33.00
4     11.27      8.89     10.22         1.00         31.37


Dataset Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16598 entries, 0 to 16597
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Rank          16598 non-null  int64
 1   Name          16598 non-null  object
 2   Platform      16598 non-null  object
 3   Year          16327 non-null  float64
 4   Genre         16598 non-null  object
 5   Publisher     16540 non-null  object
 6   NA_Sales      16598 non-null  float64
 7   EU_Sales      16598 non-null  float64
 8   JP_Sales      16598 non-null  float64
 9   Other_Sales   16598 non-null  float64
 10  Global_Sales  16598 non-null  float64
dtypes: float64(6), int64(1), object(4)
memory usage: 1.4+ MB

Checking for Missing Values:
Rank              0
Name              0
Platform          0
Year            271
Genre             0
Publisher        58
NA_Sales          0
EU_Sales          0
JP_Sales          0
Other_Sales       0
Global_Sales      0
dtype: int64

Statistical Summary of Numerical Columns:
               Rank          Year      NA_Sales      EU_Sales      JP_Sales  \
count  16598.000000  16327.000000  16598.000000  16598.000000  16598.000000   
mean    8300.605254   2006.406443      0.264667      0.146652      0.077782   
std     4791.853933      5.828981      0.816683      0.505351      0.309291   
min        1.000000   1980.000000      0.000000      0.000000      0.000000   
25%     4151.250000   2003.000000      0.000000      0.000000      0.000000   
50%     8300.500000   2007.000000      0.080000      0.020000      0.000000   
75%    12449.750000   2010.000000      0.240000      0.110000      0.040000   
max    16600.000000   2020.000000     41.490000     29.020000     10.220000   

        Other_Sales  Global_Sales  
count  16598.000000  16598.000000  
mean       0.048063      0.537441  
std        0.188588      1.555028  
min        0.000000      0.010000  
25%        0.000000      0.060000  
50%        0.010000      0.170000  
75%        0.040000      0.470000  
max       10.570000     82.740000  
```

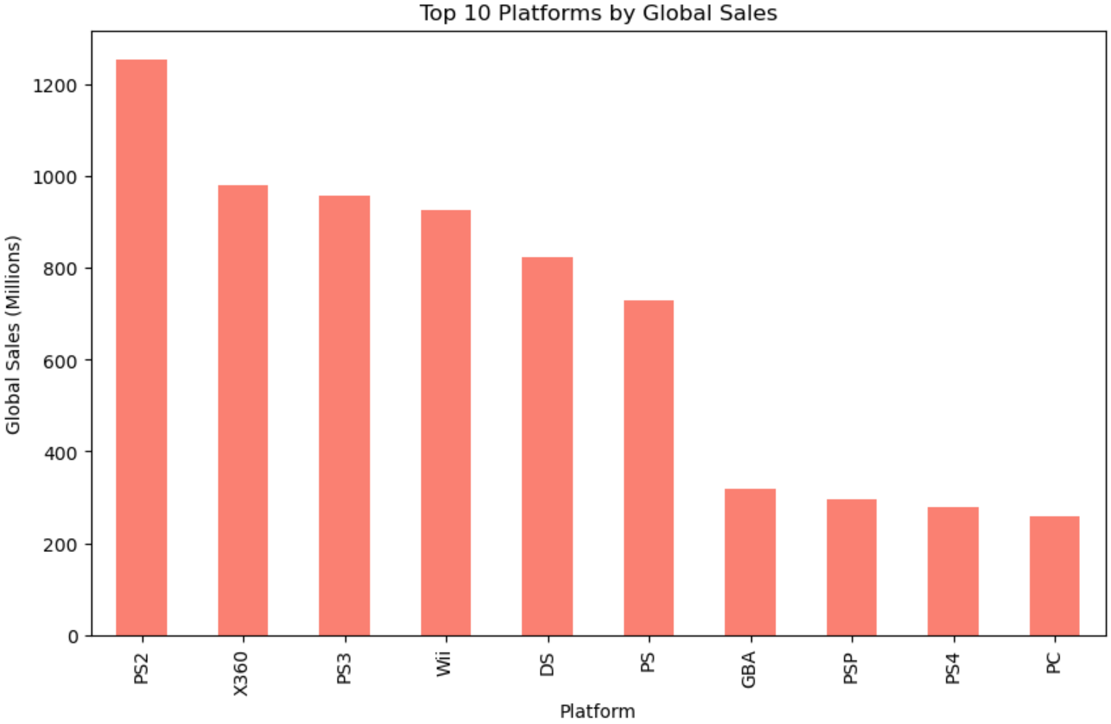

Distribution of NA_Sales
Count
70000
60000
50000
40000
30000
20000
10000
0
0
10
20
30
40
NA_Sales
Distribution of EU_Sales
Count
80000
60000
40000
20000
0
0
5
10
15
20
25
30
EU_Sales
Distribution of JP_Sales
Count
60000
50000
40000
30000
20000
10000
0
0
2
4
6
8
10
JP_Sales
Distribution of Other_Sales
Count
80000
60000
40000
20000
0
0
2
4
6
8
10
Other_Sales
Distribution of Global_Sales
Count
70000
60000
50000
40000
30000
20000
10000
0
0
20
40
60
80
Global_Sales

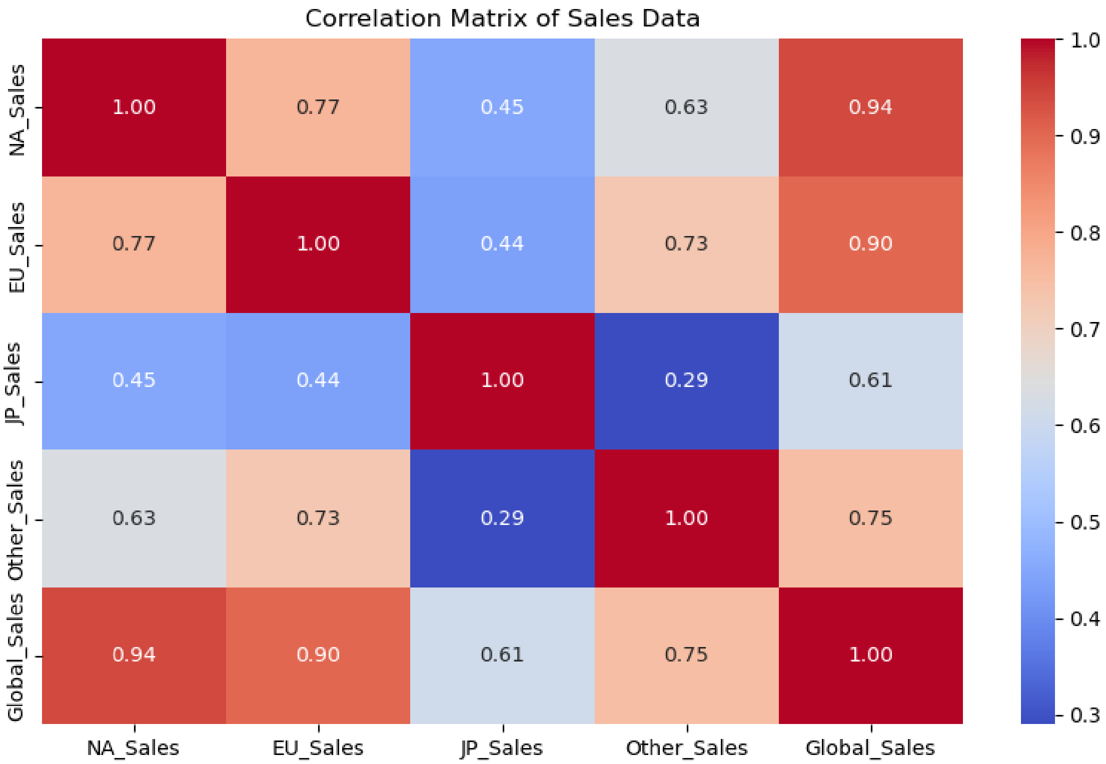

Top 10 Publishers by Global Sales
Global Sales (Millions)
1750
1500
1250
1000
750
500
250
0
Nintendo
Electronic Arts
Activision
Sony Computer Entertainment
Ubisoft
Take-Two Interactive
THQ
Konami Digital Entertainment
Sega
Namco Bandai Games
Publisher



```
/tmp/ipykernel_636145/2204207068.py:49: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.countplot(data=df, x='Genre', order=df['Genre'].value_counts().index,
palette='viridis')
```


Distribution of Game Genres
count
0
500
1000
1500
2000
2500
3000
Action
Sports
Misc
Role-Playing
Shooter
Adventure
Racing
Platform
Simulation
Fighting
Strategy
Puzzle
Genre


Top 10 Platforms by Global Sales
Global Sales (Millions)
0
200
400
600
800
1000
1200
PS2
X360
PS3
Wii
DS
PS
GBA
PSP
PS4
PC
Platform


Correlation Matrix of Sales Data
NA_Sales
EU_Sales
JP_Sales
Other_Sales
Global_Sales
1.00
0.77
0.45
0.63
0.94
0.77
1.00
0.44
0.73
0.90
0.45
0.44
1.00
0.29
0.61
0.63
0.73
0.29
1.00
0.75
0.94
0.90
0.61
0.75
1.00
NA_Sales
EU_Sales
JP_Sales
Other_Sales
Global_Sales
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3