In [186]:
```python
import pandas as pd
import missingno as msno
import plotly.express as px
import matplotlib.pyplot as plt
import plotly.graph_objects as go
import seaborn as sns
```

In [198]:
```python
df = pd.read_csv('Video_Games_Sales_as_at_22_Dec_2016.csv')
```

In [17]:
```python
df.head()
```

Out[17]:

	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	Develope
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	Nintend
1	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24	NaN	NaN	NaN	NaN	Na
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	Nintend
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	Nintend
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	Na

In [18]:
```python
df.describe()
```

Out[18]:

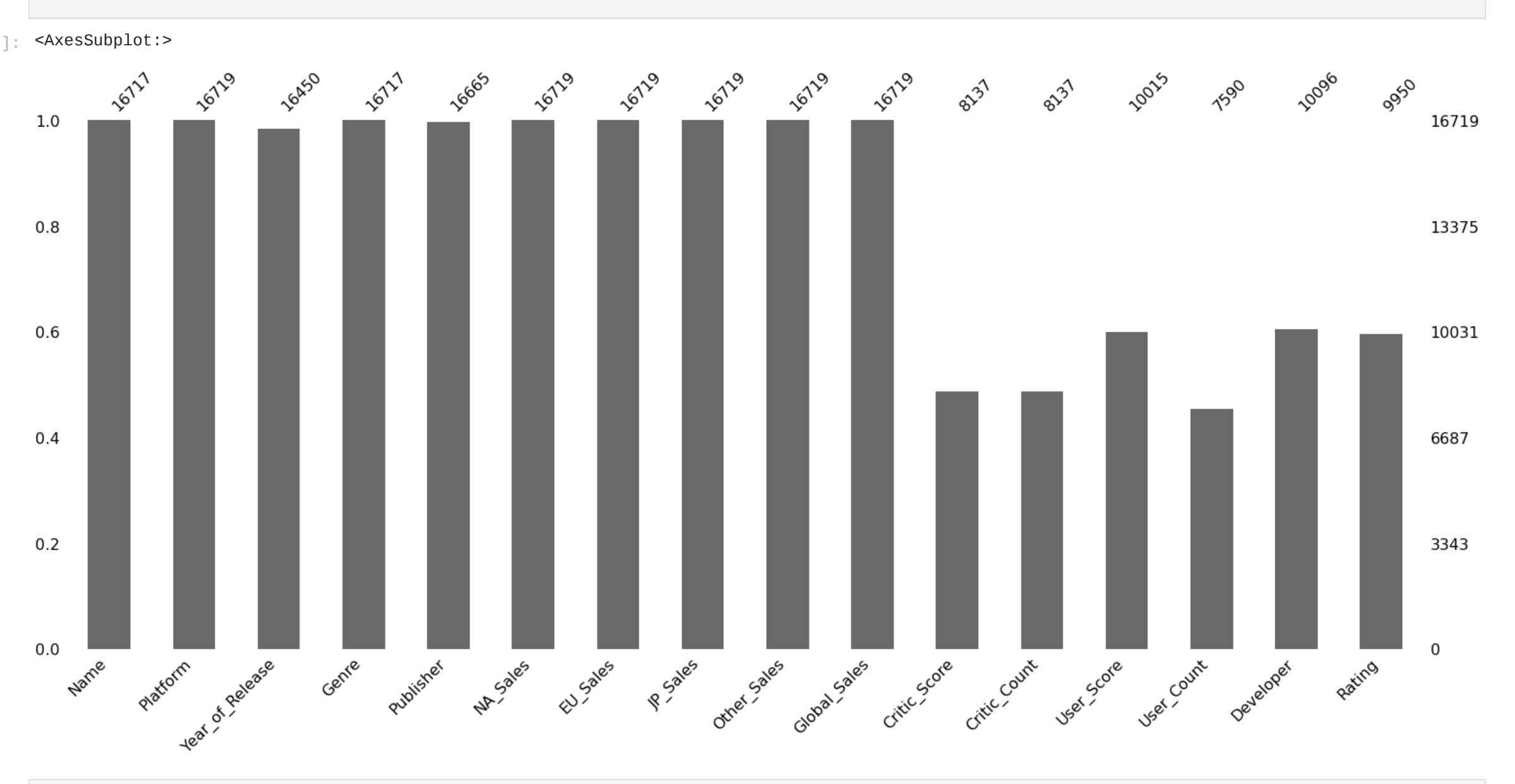
	Year_of_Release	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Critic_Score	Critic_Count	User_Count
count	16450.000000	16719.000000	16719.000000	16719.000000	16719.000000	16719.000000	8137.000000	8137.000000	7590.000000
mean	2006.487356	0.263330	0.145025	0.077602	0.047332	0.533543	68.967679	26.360821	162.229908
std	5.878995	0.813514	0.503283	0.308818	0.186710	1.547935	13.938165	18.980495	561.282326
min	1980.000000	0.000000	0.000000	0.000000	0.000000	0.010000	13.000000	3.000000	4.000000
25%	2003.000000	0.000000	0.000000	0.000000	0.000000	0.060000	60.000000	12.000000	10.000000
50%	2007.000000	0.080000	0.020000	0.000000	0.010000	0.170000	71.000000	21.000000	24.000000
75%	2010.000000	0.240000	0.110000	0.040000	0.030000	0.470000	79.000000	36.000000	81.000000
max	2020.000000	41.360000	28.960000	10.220000	10.570000	82.530000	98.000000	113.000000	10665.000000

In [203]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16719 entries, 0 to 16718
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   name             16717 non-null  object
 1   platform         16719 non-null  object
 2   year_of_release  16450 non-null  float64
 3   genre            16717 non-null  object
 4   publisher        16665 non-null  object
 5   na_sales         16719 non-null  float64
 6   eu_sales         16719 non-null  float64
 7   jp_sales         16719 non-null  float64
 8   other_sales      16719 non-null  float64
 9   global_sales     16719 non-null  float64
 10  critic_score     8137 non-null   float64
 11  critic_count     8137 non-null   float64
 12  user_score       10015 non-null  object
 13  user_count       7590 non-null   float64
 14  developer        10096 non-null  object
 15  rating           9950 non-null   object
dtypes: float64(9), object(7)
memory usage: 2.0+ MB
```

In [23]:
```python
msno.bar(df)
```

Out[23]: <AxesSubplot:>



In [206]:
```python
df.columns = df.columns.str.lower()
df.columns
```

Out[206]:
```
Index(['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'],
      dtype='object')
```

In [25]:
```python
ct = pd.crosstab(df['platform'],
                 df['year_of_release'], margins=True,
                 margins_name="Total",
                ).T
ct
```

Out[25]:

platform	2600	3DO	3DS	DC	DS	GB	GBA	GC	GEN	GG	...	SCD	SNES	TG16	WS	Wii	WiiU	X360	XB	XOne	Total
year_of_release																					
1980.0	9	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	9
1981.0	46	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	46
1982.0	36	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	36
1983.0	11	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	17
1984.0	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	14
1985.0	1	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	14
1986.0	2	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	21
1987.0	6	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	16
1988.0	2	0	0	0	0	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	15
1989.0	2	0	0	0	0	10	0	0	0	0	...	0	0	0	0	0	0	0	0	0	17
1990.0	0	0	0	0	0	3	0	0	1	0	...	0	4	0	0	0	0	0	0	0	16
1991.0	0	0	0	0	0	9	0	0	1	0	...	0	24	0	0	0	0	0	0	0	41
1992.0	0	0	0	0	0	7	0	0	6	1	...	0	21	0	0	0	0	0	0	0	43
1993.0	0	0	0	0	0	0	0	0	9	0	...	1	46	0	0	0	0	0	0	0	62
1994.0	0	1	0	0	0	3	0	0	12	0	...	5	53	0	0	0	0	0	0	0	121
1995.0	0	2	0	0	0	4	0	0	0	0	...	0	54	2	0	0	0	0	0	0	219
1996.0	0	0	0	0	0	6	0	0	0	0	...	0	30	0	0	0	0	0	0	0	263
1997.0	0	0	0	0	0	8	0	0	0	0	...	0	4	0	0	0	0	0	0	0	289
1998.0	0	0	0	7	0	8	0	0	0	0	...	0	2	0	0	0	0	0	0	0	379
1999.0	0	0	0	14	0	11	0	0	0	0	...	0	1	0	2	0	0	0	0	0	338
2000.0	0	0	0	20	0	17	1	0	0	0	...	0	0	0	2	0	0	0	1	0	350
2001.0	0	0	0	8	0	10	107	22	0	0	...	0	0	0	2	0	0	0	33	0	482
2002.0	0	0	0	1	0	0	198	152	0	0	...	0	0	0	0	0	0	0	158	0	829
2003.0	0	0	0	0	0	0	150	144	0	0	...	0	0	0	0	0	0	0	189	0	775
2004.0	0	0	0	0	23	0	175	83	0	0	...	0	0	0	0	0	0	0	177	0	762
2005.0	0	0	0	0	118	0	135	97	0	0	...	0	0	0	0	0	0	18	179	0	939
2006.0	0	0	0	0	201	0	39	40	0	0	...	0	0	0	0	44	0	93	62	0	1006
2007.0	0	0	0	1	376	0	6	4	0	0	...	0	0	0	0	185	0	123	3	0	1197
2008.0	0	0	0	1	492	0	0	0	0	0	...	0	0	0	0	282	0	146	1	0	1427
2009.0	0	0	0	0	403	0	0	0	0	0	...	0	0	0	0	325	0	172	0	0	1426
2010.0	0	0	0	0	323	0	0	0	0	0	...	0	0	0	0	253	0	182	0	0	1255
2011.0	0	0	116	0	153	0	0	0	0	0	...	0	0	0	0	143	0	206	0	0	1136
2012.0	0	0	93	0	23	0	0	0	0	0	...	0	0	0	0	31	32	106	0	0	653
2013.0	0	0	91	0	8	0	0	0	0	0	...	0	0	0	0	12	42	75	0	19	544
2014.0	0	0	80	0	0	0	0	0	0	0	...	0	0	0	0	6	31	63	0	61	581
2015.0	0	0	86	0	0	0	0	0	0	0	...	0	0	0	0	4	28	35	0	80	606
2016.0	0	0	46	0	0	0	0	0	0	0	...	0	0	0	0	1	14	13	0	87	502
2017.0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	3
2020.0	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1
Total	116	3	512	52	2122	97	811	542	29	1	...	6	239	2	6	1286	147	1232	803	247	16450

40 rows × 32 columns

In [31]:
```python
ct.drop('Total', axis=1)
```

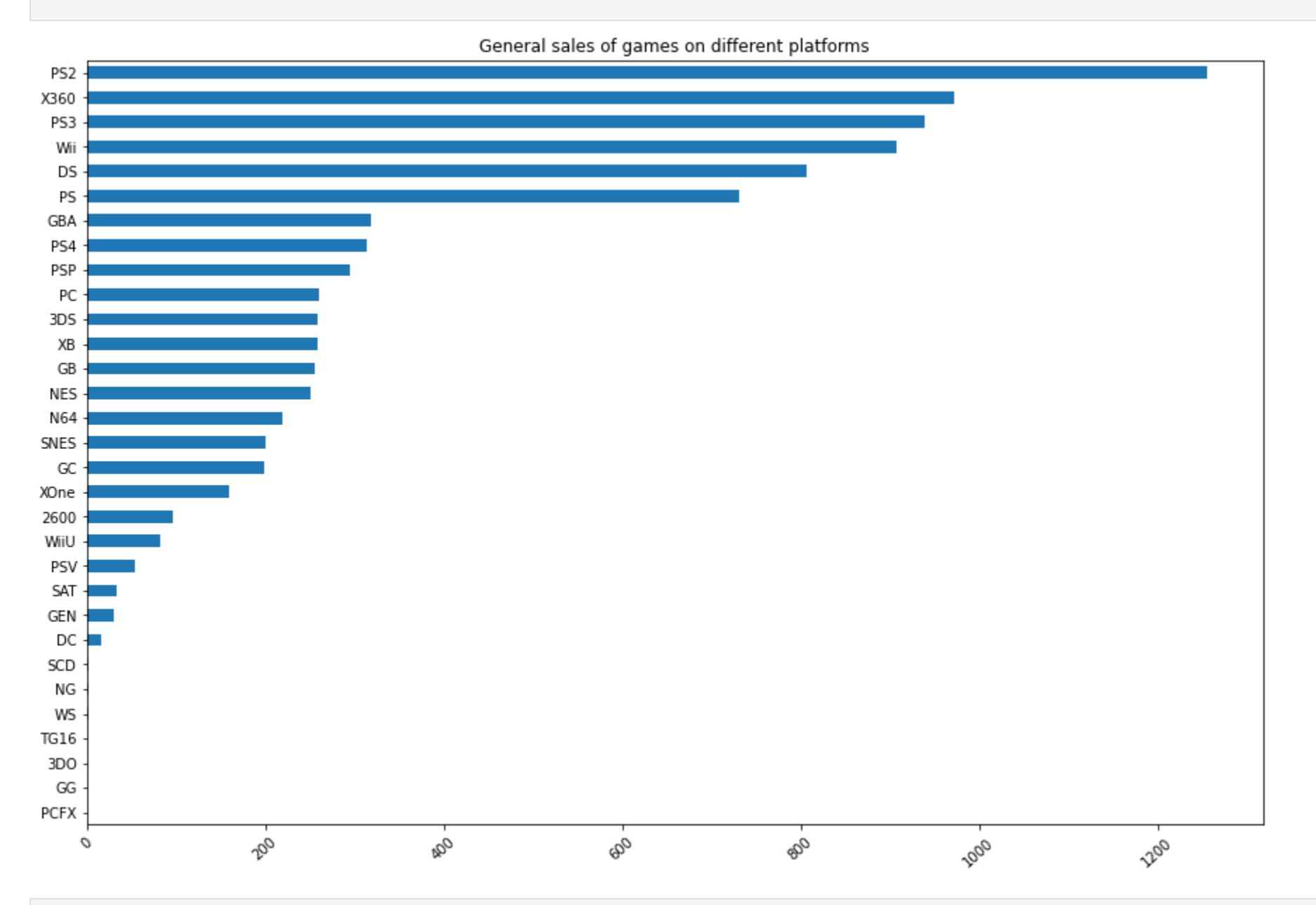
Out[31]:

platform	2600	3DO	3DS	DC	DS	GB	GBA	GC	GEN	GG	...	SAT	SCD	SNES	TG16	WS	Wii	WiiU	X360	XB	XOne
year_of_release																					
1980.0	9	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1981.0	46	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1982.0	36	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1983.0	11	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1984.0	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1985.0	1	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1986.0	2	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1987.0	6	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1988.0	2	0	0	0	0	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1989.0	2	0	0	0	0	10	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1990.0	0	0	0	0	0	3	0	0	1	0	...	0	0	4	0	0	0	0	0	0	0
1991.0	0	0	0	0	0	9	0	0	1	0	...	0	0	24	0	0	0	0	0	0	0
1992.0	0	0	0	0	0	7	0	0	6	1	...	0	0	21	0	0	0	0	0	0	0
1993.0	0	0	0	0	0	0	0	0	9	0	...	0	1	46	0	0	0	0	0	0	0
1994.0	0	1	0	0	0	3	0	0	12	0	...	19	5	53	0	0	0	0	0	0	0
1995.0	0	2	0	0	0	4	0	0	0	0	...	52	0	54	2	0	0	0	0	0	0
1996.0	0	0	0	0	0	6	0	0	0	0	...	38	0	30	0	0	0	0	0	0	0
1997.0	0	0	0	0	0	8	0	0	0	0	...	34	0	4	0	0	0	0	0	0	0
1998.0	0	0	0	7	0	8	0	0	0	0	...	29	0	2	0	0	0	0	0	0	0
1999.0	0	0	0	14	0	11	0	0	0	0	...	1	0	1	0	2	0	0	0	0	0
2000.0	0	0	0	20	0	17	1	0	0	0	...	0	0	0	0	2	0	0	0	1	0
2001.0	0	0	0	8	0	10	107	22	0	0	...	0	0	0	0	2	0	0	0	33	0
2002.0	0	0	0	1	0	0	198	152	0	0	...	0	0	0	0	0	0	0	0	158	0
2003.0	0	0	0	0	0	0	150	144	0	0	...	0	0	0	0	0	0	0	0	189	0
2004.0	0	0	0	0	23	0	175	83	0	0	...	0	0	0	0	0	0	0	0	177	0
2005.0	0	0	0	0	118	0	135	97	0	0	...	0	0	0	0	0	0	0	18	179	0
2006.0	0	0	0	0	201	0	39	40	0	0	...	0	0	0	0	0	44	0	93	62	0
2007.0	0	0	0	1	376	0	6	4	0	0	...	0	0	0	0	0	185	0	123	3	0
2008.0	0	0	0	1	492	0	0	0	0	0	...	0	0	0	0	0	282	0	146	1	0
2009.0	0	0	0	0	403	0	0	0	0	0	...	0	0	0	0	0	325	0	172	0	0
2010.0	0	0	0	0	323	0	0	0	0	0	...	0	0	0	0	0	253	0	182	0	0
2011.0	0	0	116	0	153	0	0	0	0	0	...	0	0	0	0	0	143	0	206	0	0
2012.0	0	0	93	0	23	0	0	0	0	0	...	0	0	0	0	0	31	32	106	0	0
2013.0	0	0	91	0	8	0	0	0	0	0	...	0	0	0	0	0	12	42	75	0	19
2014.0	0	0	80	0	0	0	0	0	0	0	...	0	0	0	0	0	6	31	63	0	61
2015.0	0	0	86	0	0	0	0	0	0	0	...	0	0	0	0	0	4	28	35	0	80
2016.0	0	0	46	0	0	0	0	0	0	0	...	0	0	0	0	0	1	14	13	0	87
2017.0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2020.0	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
Total	116	3	512	52	2122	97	811	542	29	1	...	173	6	239	2	6	1286	147	1232	803	247

40 rows × 31 columns

In [32]:
```python
fig = px.imshow(ct)
fig.show()
```

In [35]:
```python
df.groupby('platform')['global_sales'].sum().sort_values(ascending=True).plot(
            kind='barh', y='global_sales', figsize=(15,10)
)
plt.title('General sales of games on different platforms')
plt.xticks(rotation=42)
plt.xlabel('')
plt.ylabel('')
plt.show()
```



In [53]:
```python
current_gen_platforms = ['PC', 'PS4', 'XOne', '3DS', 'DS']
dead_platforms = df.query('platform not in @current_gen_platforms').copy()
born = dead_platforms.groupby('platform')['year_of_release'].agg(min)
death = dead_platforms.groupby('platform')['year_of_release'].agg(max)
active_years = death - born
active_years
```

Out[53]:
```
platform
2600     9.0
3DO      1.0
DC      10.0
GB      13.0
GBA      7.0
GC       6.0
GEN      4.0
GG       0.0
N64      6.0
NES     11.0
NG       3.0
PCFX     0.0
PS       9.0
PS2     11.0
PS3     10.0
PSP     11.0
PSV      6.0
SAT      5.0
SCD      1.0
SNES     9.0
TG16     0.0
WS       2.0
Wii     10.0
WiiU     4.0
X360    11.0
XB       8.0
Name: year_of_release, dtype: float64
```

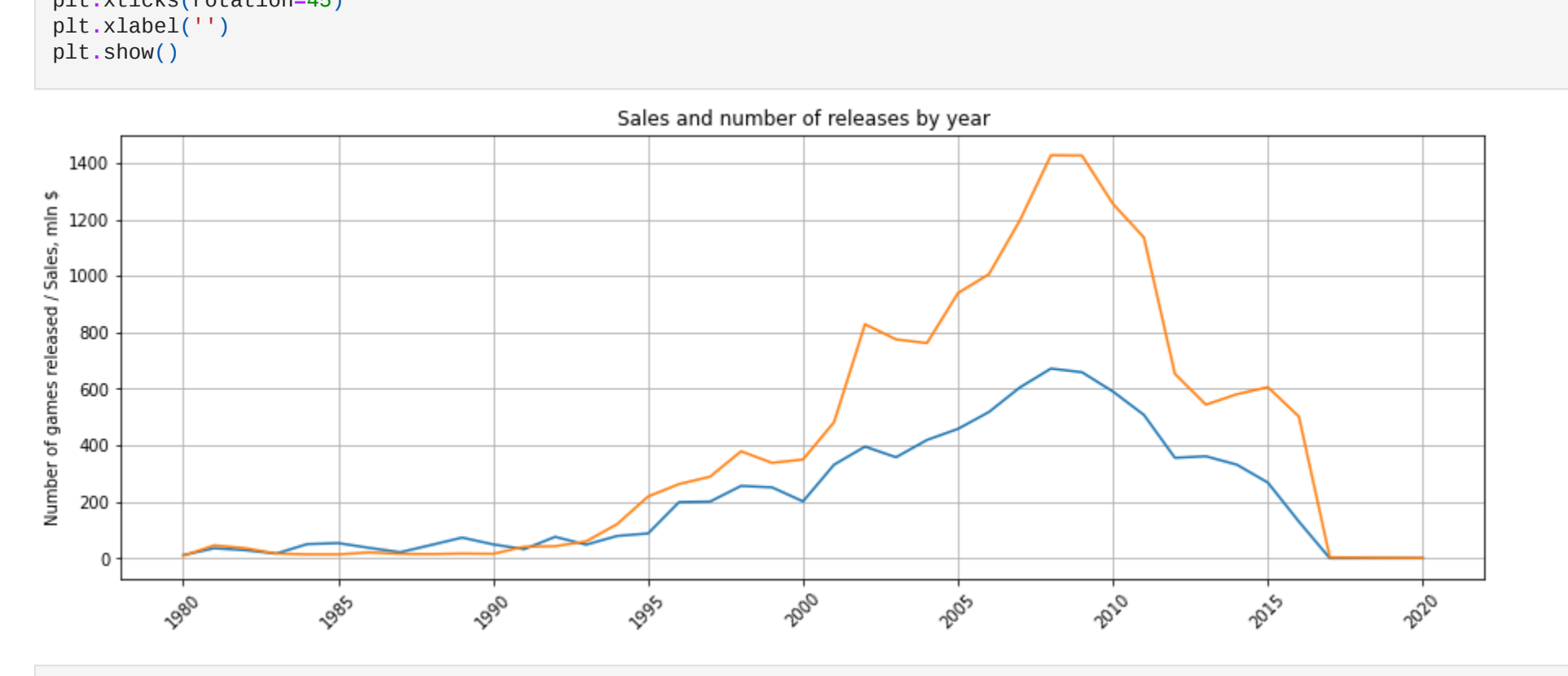
In [57]:
```python
ay = active_years.to_frame()
```

In [65]:
```python
ay.year_of_release.sort_values(ascending=False).head(10)
```

Out[65]:
```
platform
GB      13.0
PS2     11.0
X360    11.0
PSP     11.0
NES     11.0
DC      10.0
Wii     10.0
PS3     10.0
PS       9.0
SNES     9.0
Name: year_of_release, dtype: float64
```

In [214…
```python
plt.figure(figsize=(15, 5))
ax = plt.gca()
df.groupby('year_of_release')['global_sales'].sum().plot(title='Sales and number of releases by year')
df.groupby('year_of_release')['name'].count().plot(grid=True)

plt.ylabel('Number of games released / Sales, mln $')
plt.xticks(rotation=45)
plt.xlabel('')
plt.show()
```



In [184…
```python
ct2 = pd.crosstab(index=df['platform'], columns=df['year_of_release'], margins=True,
            margins_name="Total",
           ).T
ct2=ct2.drop('Total', axis=1)
ct2=ct2.drop('Total')
ct2
```

Out[184…

platform	2600	3DO	3DS	DC	DS	GB	GBA	GC	GEN	GG	...	SAT	SCD	SNES	TG16	WS	Wii	WiiU	X360	XB	XOne
year_of_release																					
1980.0	9	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1981.0	46	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1982.0	36	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1983.0	11	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1984.0	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1985.0	1	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1986.0	2	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1987.0	6	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1988.0	2	0	0	0	0	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1989.0	2	0	0	0	0	10	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1990.0	0	0	0	0	0	3	0	0	1	0	...	0	0	4	0	0	0	0	0	0	0
1991.0	0	0	0	0	0	9	0	0	1	0	...	0	0	24	0	0	0	0	0	0	0
1992.0	0	0	0	0	0	7	0	0	6	1	...	0	0	21	0	0	0	0	0	0	0
1993.0	0	0	0	0	0	0	0	0	9	0	...	0	1	46	0	0	0	0	0	0	0
1994.0	0	1	0	0	0	3	0	0	12	0	...	19	5	53	0	0	0	0	0	0	0
1995.0	0	2	0	0	0	4	0	0	0	0	...	52	0	54	2	0	0	0	0	0	0
1996.0	0	0	0	0	0	6	0	0	0	0	...	38	0	30	0	0	0	0	0	0	0
1997.0	0	0	0	0	0	8	0	0	0	0	...	34	0	4	0	0	0	0	0	0	0
1998.0	0	0	0	7	0	8	0	0	0	0	...	29	0	2	0	0	0	0	0	0	0
1999.0	0	0	0	14	0	11	0	0	0	0	...	1	0	1	0	2	0	0	0	0	0
2000.0	0	0	0	20	0	17	1	0	0	0	...	0	0	0	0	2	0	0	0	1	0
2001.0	0	0	0	8	0	10	107	22	0	0	...	0	0	0	0	2	0	0	0	33	0
2002.0	0	0	0	1	0	0	198	152	0	0	...	0	0	0	0	0	0	0	0	158	0
2003.0	0	0	0	0	0	0	150	144	0	0	...	0	0	0	0	0	0	0	0	189	0
2004.0	0	0	0	0	23	0	175	83	0	0	...	0	0	0	0	0	0	0	0	177	0
2005.0	0	0	0	0	118	0	135	97	0	0	...	0	0	0	0	0	0	0	18	179	0
2006.0	0	0	0	0	201	0	39	40	0	0	...	0	0	0	0	0	44	0	93	62	0
2007.0	0	0	0	1	376	0	6	4	0	0	...	0	0	0	0	0	185	0	123	3	0
2008.0	0	0	0	1	492	0	0	0	0	0	...	0	0	0	0	0	282	0	146	1	0
2009.0	0	0	0	0	403	0	0	0	0	0	...	0	0	0	0	0	325	0	172	0	0
2010.0	0	0	0	0	323	0	0	0	0	0	...	0	0	0	0	0	253	0	182	0	0
2011.0	0	0	116	0	153	0	0	0	0	0	...	0	0	0	0	0	143	0	206	0	0
2012.0	0	0	93	0	23	0	0	0	0	0	...	0	0	0	0	0	31	32	106	0	0
2013.0	0	0	91	0	8	0	0	0	0	0	...	0	0	0	0	0	12	42	75	0	19
2014.0	0	0	80	0	0	0	0	0	0	0	...	0	0	0	0	0	6	31	63	0	61
2015.0	0	0	86	0	0	0	0	0	0	0	...	0	0	0	0	0	4	28	35	0	80
2016.0	0	0	46	0	0	0	0	0	0	0	...	0	0	0	0	0	1	14	13	0	87
2017.0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2020.0	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

39 rows × 31 columns

In [175]:
```python
fig = px.imshow(ct2)

fig.show()
```

In [192…
```python
games_after_11 = df.query('year_of_release >=2012').reset_index(drop=True)
games_after_11.head()
```

Out[192…

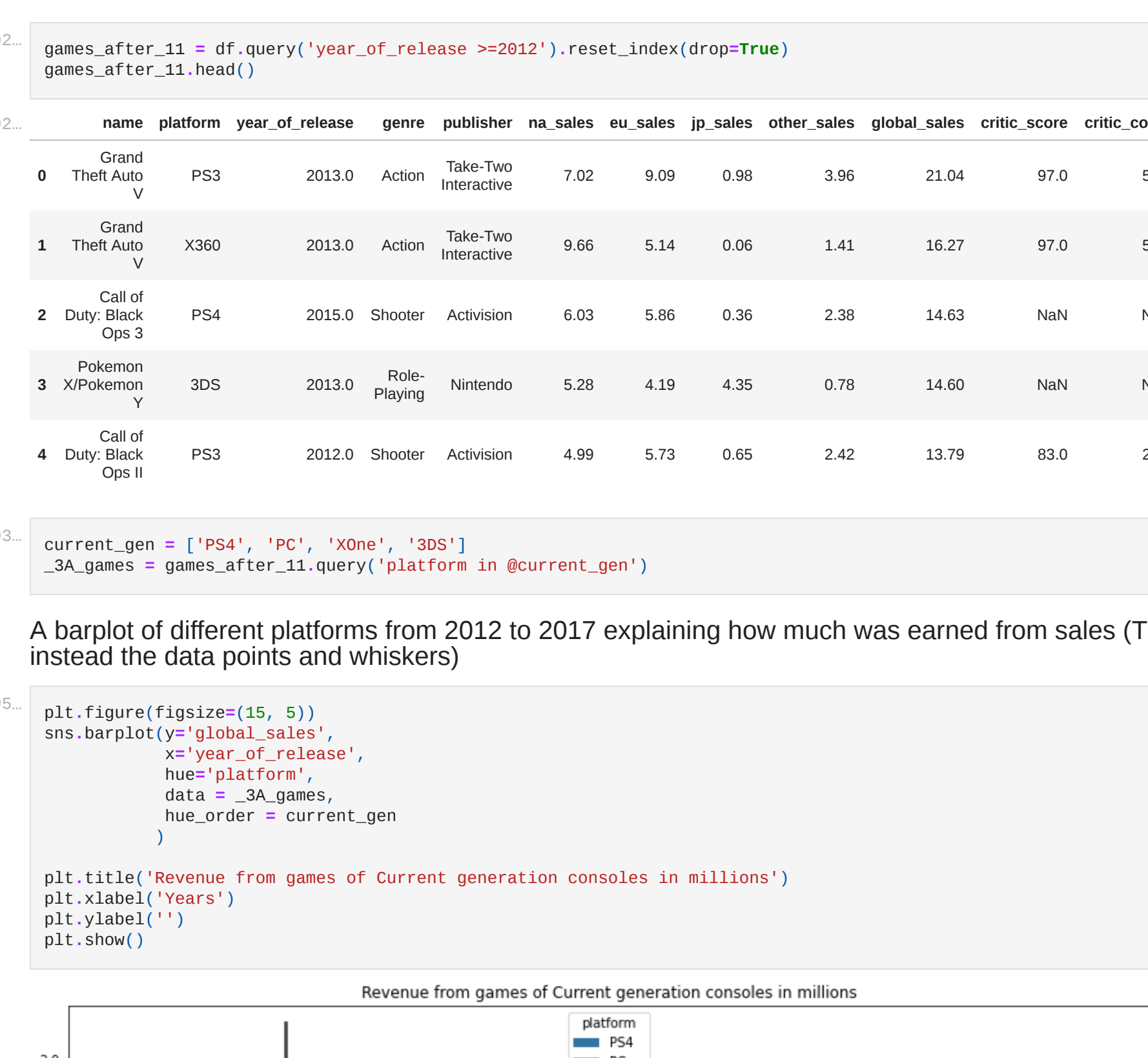
	name	platform	year_of_release	genre	publisher	na_sales	eu_sales	jp_sales	other_sales	global_sales	critic_score	critic_count	user_score	user_count	developer	rating
0	Grand Theft Auto V	PS3	2013.0	Action	Take-Two Interactive	7.02	9.09	0.98	3.96	21.04	97.0	50.0	8.2	3994.0	Rockstar North	M
1	Grand Theft Auto V	X360	2013.0	Action	Take-Two Interactive	9.66	5.14	0.06	1.41	16.27	97.0	58.0	8.1	3711.0	Rockstar North	M
2	Call of Duty: Black Ops 3	PS4	2015.0	Shooter	Activision	6.03	5.86	0.36	2.38	14.63	NaN	NaN	NaN	NaN	NaN	NaN
3	Pokemon X/Pokemon Y	3DS	2013.0	Role-Playing	Nintendo	5.28	4.19	4.35	0.78	14.60	NaN	NaN	NaN	NaN	NaN	NaN
4	Call of Duty: Black Ops II	PS3	2012.0	Shooter	Activision	4.99	5.73	0.65	2.42	13.79	83.0	21.0	5.3	922.0	Treyarch	M

In [193]:
```python
current_gen = ['PS4', 'PC', 'XOne', '3DS']
_3A_games = games_after_11.query('platform in @current_gen')
```

A barplot of different platforms from 2012 to 2017 explaining how much was earned from sales (This is not the total revenue instead the data points and whiskers)

In [195]:
```python
plt.figure(figsize=(15, 5))
sns.barplot(y='global_sales',
            x='year_of_release',
            hue='platform',
            data = _3A_games,
            hue_order = current_gen
           )

plt.title('Revenue from games of Current generation consoles in millions')
plt.xlabel('Years')
plt.ylabel('')
plt.show()
```



In [212…
```python
_3A_games.dropna()
```

Out[212…

	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	de
6	Grand Theft Auto V	PS4	2014.0	Action	Take-Two Interactive	3.96	6.31	0.38	1.97	12.61	97.0	66.0	8.3	2899.0	Rocks
9	New Super Mario Bros. 2	3DS	2012.0	Platform	Nintendo	3.66	3.14	2.47	0.63	9.90	78.0	70.0	7.2	424.0	N
13	Animal Crossing: New Leaf	3DS	2012.0	Simulation	Nintendo	2.03	2.36	4.39	0.39	9.16	88.0	70.0	8.7	626.0	N
14	FIFA 16	PS4	2015.0	Sports	Electronic Arts	1.12	6.12	0.06	1.28	8.57	82.0	42.0	4.3	896.0	E.
18	Call of Duty: Advanced Warfare	PS4	2014.0	Shooter	Activision	2.81	3.48	0.14	1.23	7.66	83.0	39.0	5.7	1443.0	Sledge
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2819	Carmageddon: Max Damage	PS4	2016.0	Action	Stainless Games	0.01	0.00	0.00	0.00	0.01	51.0	28.0	5.5	39.0	S
2837	Transformers: Fall of Cybertron	PC	2012.0	Action	Activision	0.01	0.00	0.00	0.00	0.01	78.0	13.0	7.9	121.0	Hi Me Tech
2844	Saints Row: Gat out of Hell	XOne	2015.0	Action	Deep Silver	0.00	0.01	0.00	0.00	0.01	65.0	16.0	6	27.0	Vo
2858	Sébastien Loeb Rally Evo	XOne	2016.0	Racing	Milestone S.r.l	0.00	0.01	0.00	0.00	0.01	63.0	8.0	8.2	22.0	Milest
2883	Metal Gear Solid V: Ground Zeroes	PC	2014.0	Action	Konami Digital Entertainment	0.00	0.01	0.00	0.00	0.01	80.0	20.0	7.6	412.0	Pro

672 rows × 16 columns

In [231]:
```python
fig = px.scatter_matrix(_3A_games,
    dimensions=["global_sales", "user_score", "critic_score"], color='platform')
fig.show()
```

In [215…
```python
top_games = _3A_games.pivot_table(index='genre',columns='year_of_release'
                                  ,values='global_sales',aggfunc='sum').copy()
top_games
```

Out[215…

year_of_release	2012.0	2013.0	2014.0	2015.0	2016.0	2017.0
genre						
Action	17.31	24.66	66.88	53.38	25.08	NaN
Adventure	1.33	1.88	2.60	5.33	3.10	NaN
Fighting	0.28	1.28	7.83	7.10	3.17	NaN
Misc	2.37	5.01	11.05	6.18	1.20	NaN
Platform	10.40	7.42	6.47	0.80	2.80	NaN
Puzzle	0.95	0.85	0.19	0.71	NaN	NaN
Racing	0.56	5.19	6.98	7.91	2.83	NaN
Role-Playing	20.70	27.37	34.46	33.92	16.04	0.04
Shooter	4.53	19.17	43.63	58.25	37.81	NaN
Simulation	11.52	7.28	5.50	4.89	1.88	NaN
Sports	3.72	10.24	24.74	30.19	21.00	NaN
Strategy	0.97	3.60	0.88	1.80	1.06	NaN

In [223…
```python
top_list = ['Action', 'Role-Playing', 'Shooter', 'Sports', 'Platform', 'Misc']
```

In [224]:
```python
top_games.query('genre in @top_list').plot(kind='bar', figsize=(15,5), edgecolor='black')
plt.title('')
plt.show()
```


In [ ]: