

Nintendo Video Game Sales - Exploratory Data Analysis

Christopher Heimbuch - <https://github.com/chrisheimbuch>

Dataset: <https://www.kaggle.com/datasets/thedevastator/global-video-game-sales>



[Image link](#)

Dataset Overview

This dataset has been sourced from vgchartz.com. The data was scraped from numerous sources to compile a dataset of many video game titles for various video game consoles with global sales over 100,000. Video game titles span all the way from 1980 - 2020!

The columns that this dataset includes are:

- **Name** - The name of the video game titles.
- **Platform** - The Console associated with the videogame title. (Example: Gamecube, PS1, PS2 etc.)
- **Year** - The year the video game was released.
- **Genre** - What type of category/genre the video game is.
- **Publisher** - Which company has published the video game.
- **NA_Sales** - How many sales were in the United States, in millions.

- **EU_Sales** - How many sales were in the Europe, in millions.
- **JP_Sales** - How many sales were in the Japan, in millions.
- **Other_Sales** - How many sales were in the other countries/regions, in millions.
- **Global_Sales** - Total sales across all regions, in millions.

Note: This dataset analysis only contains data up to 2017 and does not include data on the Nintendo Switch.

Section 1: Read, Clean, and Learn about the Data

In [38]: *#Import in standard libraries to help with the analysis.*

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [39]: *#Set 'df' as our dataframe variable, reading in the 'vgsales.csv' document.*

```
df = pd.read_csv("vgsales.csv", encoding='latin-1')
```

In [40]: *#Check to make sure information loaded correctly and start to get familiar with our data*

```
df.head()
```

Out[40]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77	8.46
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	0.77
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79	3.31
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3.28	2.96
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	1.00

In [41]: *#Call info() method to check for data types and potential null values. Null values spotted*

```
df.info()
```

```
<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
```

```
In [42]: #Check for total amount of null values and in their respective categories.

df.isna().sum()
```

```
Out[42]: 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
```

```
In [43]: #Inspect the "Year" columns NaN values to determine how to handle them! This could be us

df[df['Year'].isna()]
```

Out[43]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales
179	180	Madden NFL 2004	PS2	NaN	Sports	Electronic Arts	4.26	0.26	0.01	
377	378	FIFA Soccer 2004	PS2	NaN	Sports	Electronic Arts	0.59	2.36	0.04	
431	432	LEGO Batman: The Videogame	Wii	NaN	Action	Warner Bros. Interactive Entertainment	1.86	1.02	0.00	
470	471	wwe Smackdown vs. Raw 2006	PS2	NaN	Fighting	NaN	1.57	1.02	0.00	
607	608	Space Invaders	2600	NaN	Shooter	Atari	2.36	0.14	0.00	
...
16307	16310	Freaky Flyers	GC	NaN	Racing	Unknown	0.01	0.00	0.00	
16327	16330	Inversion	PC	NaN	Shooter	Namco Bandai Games	0.01	0.00	0.00	
16366	16369	Hakuouki: Shinsengumi	PS3	NaN	Adventure	Unknown	0.01	0.00	0.00	

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other
		Kitan								
16427	16430	Virtua Quest	GC	NaN	Role-Playing	Unknown	0.01	0.00	0.00	
16493	16496	The Smurfs	3DS	NaN	Action	Unknown	0.00	0.01	0.00	

271 rows × 11 columns

In [44]:

#Inspect the 'Publisher' columns NaN values. These could potentially serve useful for our analysis.
df[df['Publisher'].isna()]

Out[44]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other
		wwe								
470	471	Smackdown vs. Raw 2006	PS2	NaN	Fighting	NaN	1.57	1.02	0.00	
1303	1305	Triple Play 99	PS	NaN	Sports	NaN	0.81	0.55	0.00	
		Shrek / Shrek 2 2-in-1								
1662	1664	Gameboy Advance Video	GBA	2007.0	Misc	NaN	0.87	0.32	0.00	
2222	2224	Bentley's Hackpack	GBA	2005.0	Misc	NaN	0.67	0.25	0.00	
		Nicktoons Collection: Game Boy Advance Video V...								
3159	3161		GBA	2004.0	Misc	NaN	0.46	0.17	0.00	
		SpongeBob SquarePants: Game Boy Advance Video ...								
3166	3168		GBA	2004.0	Misc	NaN	0.46	0.17	0.00	
		SpongeBob SquarePants: Game Boy Advance Video ...								
3766	3768		GBA	2004.0	Misc	NaN	0.38	0.14	0.00	
4145	4147	Sonic the Hedgehog	PS3	NaN	Platform	NaN	0.00	0.48	0.00	
		The Fairly Odd Parents: Game Boy Advance Video...								
4526	4528		GBA	2004.0	Misc	NaN	0.31	0.11	0.00	
		The Fairly Odd Parents: Game Boy								
4635	4637		GBA	2004.0	Misc	NaN	0.30	0.11	0.00	

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Othe
		Advance Video...								
5302	5304	Dragon Ball Z: Budokai Tenkaichi 2 (JP sales)	Wii	NaN	Action	NaN	0.15	0.05	0.14	
5647	5649	Cartoon Network Collection: Game Boy Advance V...	GBA	2005.0	Misc	NaN	0.23	0.08	0.00	
6272	6274	The Legend of Zelda: The Minish Cap(weekly JP ...	GBA	NaN	Action	NaN	0.00	0.00	0.27	
6437	6439	Sonic X: Game Boy Advance Video Volume 1	GBA	2004.0	Misc	NaN	0.19	0.07	0.00	
6562	6564	Dora the Explorer: Game Boy Advance Video Volu...	GBA	2004.0	Misc	NaN	0.18	0.07	0.00	
6648	6650	Cartoon Network Collection: Game Boy Advance V...	GBA	2004.0	Misc	NaN	0.18	0.07	0.00	
6849	6851	All Grown Up!: Game Boy Advance Video Volume 1	GBA	2004.0	Misc	NaN	0.17	0.06	0.00	
7208	7210	Nicktoons Collection: Game Boy Advance Video V...	GBA	2004.0	Misc	NaN	0.16	0.06	0.00	
7351	7353	Yu Yu Hakusho: Dark Tournament	PS2	NaN	Fighting	NaN	0.10	0.08	0.00	
7470	7472	SpongeBob SquarePants: Game Boy Advance Video ...	GBA	2004.0	Misc	NaN	0.15	0.05	0.00	
7953	7955	Thomas the Tank Engine &	GBA	2004.0	Adventure	NaN	0.13	0.05	0.00	

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Othe
		Friends								
		Dragon Ball GT: Game Boy Advance Video Volume 1	GBA	2004.0	Misc	NaN	0.12	0.05	0.00	
		Codename: Kids Next Door: Game Boy Advance Vid...	GBA	2004.0	Misc	NaN	0.12	0.05	0.00	
		Teenage Mutant Ninja Turtles: Game Boy Advance...	GBA	2004.0	Misc	NaN	0.12	0.04	0.00	
		Stronghold 3	PC	2011.0	Strategy	NaN	0.06	0.10	0.00	
		Cartoon Network Collection: Game Boy Advance V...	GBA	2005.0	Misc	NaN	0.11	0.04	0.00	
		PokÃ©mon: Johto Photo Finish: Game Boy Advance...	GBA	2004.0	Misc	NaN	0.11	0.04	0.00	
		Strawberry Shortcake: Game Boy Advance Video V...	GBA	2004.0	Misc	NaN	0.11	0.04	0.00	
		Farming Simulator 2011	PC	2010.0	Simulation	NaN	0.00	0.13	0.00	
		Super Robot Wars OG Saga: Masou Kishin II - Re...	PSP	NaN	Strategy	NaN	0.00	0.00	0.12	
		Disney Channel Collection Vol. 1	GBA	2004.0	Misc	NaN	0.08	0.03	0.00	
		Atsumare! Power Pro Kun no DS Koushien	DS	NaN	Sports	NaN	0.00	0.00	0.10	
		Action Man- Operation Extreme	PS	NaN	Action	NaN	0.05	0.03	0.00	

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Othe
11526	11528	Cartoon Network Collection: Game Boy Advance V...	GBA	2004.0	Misc	NaN	0.06	0.02	0.00	
12487	12489	Chou Soujuu Mecha MG	DS	NaN	Simulation	NaN	0.00	0.00	0.06	
12517	12519	Prinny: Can I Really Be The Hero? (US sales)	PSP	NaN	Action	NaN	0.06	0.00	0.00	
13278	13280	Monster Hunter Frontier Online	PS3	NaN	Role-Playing	NaN	0.00	0.00	0.05	
13672	13674	B.L.U.E.: Legend of Water	PS	NaN	Adventure	NaN	0.00	0.00	0.04	
13962	13964	World of Tanks	X360	NaN	Shooter	NaN	0.00	0.03	0.00	
14087	14089	Housekeeping	DS	NaN	Action	NaN	0.00	0.00	0.04	
14296	14299	Bikkuriman Daijiten	DS	NaN	Misc	NaN	0.00	0.00	0.03	
14311	14314	Silverlicious	DS	2012.0	Action	NaN	0.03	0.00	0.00	
14698	14701	UK Truck Simulator	PC	2010.0	Simulation	NaN	0.00	0.03	0.00	
14942	14945	Umineko no Naku Koro ni San: Shinjitsu to Gens...	PS3	NaN	Adventure	NaN	0.00	0.00	0.02	
15056	15059	Xia-Xia	DS	2012.0	Platform	NaN	0.00	0.02	0.00	
15261	15264	Mario Tennis	3DS	NaN	Sports	NaN	0.00	0.00	0.02	
15325	15328	Nicktoons Collection: Game Boy Advance Video V...	GBA	2005.0	Misc	NaN	0.01	0.01	0.00	
15353	15356	Demolition Company: Gold Edition	PC	2011.0	Simulation	NaN	0.00	0.02	0.00	
15788	15791	Moshi, Kono Sekai ni Kami-sama ga Iru to suru ...	PSV	2016.0	Adventure	NaN	0.00	0.00	0.02	
15915	15918	Dream Dancer	DS	NaN	Misc	NaN	0.01	0.00	0.00	

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Othe
16191	16194	Homeworld Remastered Collection	PC	NaN	Strategy	NaN	0.00	0.01	0.00	
16198	16201	AKB1/48: Idol to Guam de Koishitara...	X360	NaN	Misc	NaN	0.00	0.00	0.01	
16208	16211	Super Robot Monkey Team: Game Boy Advance Vide...	GBA	2005.0	Misc	NaN	0.01	0.00	0.00	
16229	16232	Brothers in Arms: Furious 4	X360	NaN	Shooter	NaN	0.01	0.00	0.00	
16367	16370	Dance with Devils	PSV	2016.0	Action	NaN	0.00	0.00	0.01	
16494	16497	Legends of Oz: Dorothy's Return	3DS	2014.0	Puzzle	NaN	0.00	0.01	0.00	
16543	16546	Driving Simulator 2011	PC	2011.0	Racing	NaN	0.00	0.01	0.00	
16553	16556	Bound By Flame	X360	2014.0	Role-Playing	NaN	0.00	0.01	0.00	

In [45]: *#Create a copy of original dataset - just incase we need to reference.*

```
df_copy = df.copy()
```

In [46]: *#Sanity check to make sure it worked properly.*

```
df_copy.head()
```

Out[46]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77	8.46
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	0.77
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79	3.31
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3.28	2.96
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	1.00


```
In [47]: #Fill in the 'Publisher' columns null values with "Unknown" string, in the "df_copy" da
df_copy['Publisher'].fillna("Unknown", inplace=True)
```

```
In [48]: #Sanity check to examine that null values have filled in properly.

df_copy.info()
```

```
<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       16598 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
```

```
In [49]: #Fill in "Year" columns null values with 0 int.

df_copy['Year'].fillna(0, inplace=True)
```

```
In [53]: #Sanity check to see years were converted. Should convert all years to not be a float.

df_copy['Year'].value_counts()
```

```
Out[53]: 2009    1431
2008    1428
2010    1259
2007    1202
2011    1139
2006    1008
2005     941
2002     829
2003     775
2004     763
2012     657
2015     614
2014     582
2013     546
2001     482
1998     379
2000     349
2016     344
1999     338
1997     289
0         271
1996     263
1995     219
1994     121
1993      60
1981      46
1992      43
```

1991	41
1982	36
1986	21
1989	17
1983	17
1990	16
1987	16
1988	15
1984	14
1985	14
1980	9
2017	3
2020	1

Name: Year, dtype: int64

```
In [54]: #Make all years a int data type and not float.

df_copy['Year'] = df_copy['Year'].astype(int)
```

```
In [55]: #Sanity check.

df_copy['Year'].value_counts()
```

```
Out[55]: 2009    1431
          2008    1428
          2010    1259
          2007    1202
          2011    1139
          2006    1008
          2005     941
          2002     829
          2003     775
          2004     763
          2012     657
          2015     614
          2014     582
          2013     546
          2001     482
          1998     379
          2000     349
          2016     344
          1999     338
          1997     289
          0       271
          1996     263
          1995     219
          1994     121
          1993      60
          1981      46
          1992      43
          1991      41
          1982      36
          1986      21
          1989      17
          1983      17
          1990      16
          1987      16
          1988      15
          1984      14
          1985      14
          1980       9
          2017       3
          2020       1
          Name: Year, dtype: int64
```

```
In [56]: #Check to see if null values still exist in "df_copy" dataframe, and they don't!
```

```
df_copy.info()
```

```
<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            16598 non-null  int32
4   Genre           16598 non-null  object
5   Publisher       16598 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(5), int32(1), int64(1), object(4)
memory usage: 1.3+ MB
```

```
In [57]: #Clean up a few columnds data for clarity.
df_copy['Platform'] = df_copy['Platform'].replace("GB", "Gameboy")
df_copy['Platform'] = df_copy['Platform'].replace("GBA", "Gameboy Advance")
df_copy['Platform'] = df_copy['Platform'].replace("GC", "GameCube")
df_copy['Publisher'] = df_copy['Publisher'].replace("Warner Bros. Interactive Entertainment", "Warner Bros.")
```

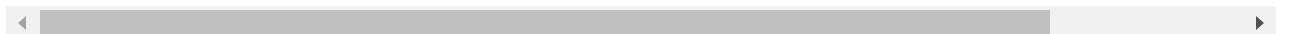
```
In [58]: #Drop the 2020 data, as this project I am only focusing up to 2017, where most data goes.
index_to_drop = df_copy[df_copy['Year'] == 2020].index
df_copy.drop(index_to_drop, inplace=True)
```

```
In [59]: #Confirming changes completed!
df_copy
```

```
Out[59]:
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other
0	1	Wii Sports	Wii	2006	Sports	Nintendo	41.49	29.02	3.77	
1	2	Super Mario Bros.	NES	1985	Platform	Nintendo	29.08	3.58	6.81	
2	3	Mario Kart Wii	Wii	2008	Racing	Nintendo	15.85	12.88	3.79	
3	4	Wii Sports Resort	Wii	2009	Sports	Nintendo	15.75	11.01	3.28	
4	5	Pokemon Red/Pokemon Blue	Gameboy	1996	Role-Playing	Nintendo	11.27	8.89	10.22	
...
16593	16596	Woody Woodpecker in Crazy Castle 5	Gameboy Advance	2002	Platform	Kemco	0.01	0.00	0.00	
16594	16597	Men in Black II: Alien Escape	GameCube	2003	Shooter	Infogrames	0.01	0.00	0.00	
16595	16598	SCORE International Baja 1000: The Official Game	PS2	2008	Racing	Activision	0.00	0.00	0.00	
16596	16599	Know How 2	DS	2010	Puzzle	7G//AMES	0.00	0.01	0.00	
16597	16600	Spirits & Spells	Gameboy Advance	2003	Platform	Wanadoo	0.01	0.00	0.00	

16597 rows × 11 columns



```
In [60]: df_copy['Year'].unique()
```

```
Out[60]: array([2006, 1985, 2008, 2009, 1996, 1989, 1984, 2005, 1999, 2007, 2010,
        2013, 2004, 1990, 1988, 2002, 2001, 2011, 1998, 2015, 2012, 2014,
```

1992, 1997, 1993, 1994, 1982, 2003, 1986, 2000, 0, 1995, 2016,
1991, 1981, 1987, 1980, 1983, 2017])

In [61]: *#Check unique publishers - there are many. This can help answer questions for Nintendo.*

```
df_copy['Publisher'].unique()
```

```
Out[61]: array(['Nintendo', 'Microsoft Game Studios', 'Take-Two Interactive',
'Sony Computer Entertainment', 'Activision', 'Ubisoft',
'Bethesda Softworks', 'Electronic Arts', 'Sega', 'SquareSoft',
'Atari', '505 Games', 'Capcom', 'GT Interactive',
'Konami Digital Entertainment',
'Sony Computer Entertainment Europe', 'Square Enix', 'LucasArts',
'Virgin Interactive', 'Warner Bros', 'Universal Interactive',
'Eidos Interactive', 'RedOctane', 'Vivendi Games',
'Enix Corporation', 'Namco Bandai Games', 'Palcom',
'Hasbro Interactive', 'THQ', 'Fox Interactive',
'Acclaim Entertainment', 'MTV Games', 'Disney Interactive Studios',
'Unknown', 'Majesco Entertainment', 'Codemasters', 'Red Orb',
'Level 5', 'Arena Entertainment', 'Midway Games', 'JVC',
'Deep Silver', '989 Studios', 'NCSoft', 'UEP Systems',
'Parker Bros.', 'Maxis', 'Imagic', 'Tecmo Koei', 'Valve Software',
'ASCII Entertainment', 'Mindscape', 'Infogrames', 'Square',
'Valve', 'Activision Value', 'Banpresto', 'D3Publisher',
'Oxygen Interactive', 'Red Storm Entertainment', 'Video System',
'Hello Games', 'Global Star', 'Gotham Games', 'Westwood Studios',
'GungHo', 'Crave Entertainment', 'Hudson Soft', 'Coleco',
'Rising Star Games', 'Atlus', 'TDK Mediactive', 'ASC Games',
'Zoo Games', 'Accolade', 'Sony Online Entertainment', '3DO', 'RTL',
'Natsume', 'Focus Home Interactive', 'Alchemist',
'Black Label Games', 'SouthPeak Games', 'Mastertronic', 'Ocean',
'Zoo Digital Publishing', 'Psygnosis', 'City Interactive',
'Empire Interactive', 'Success', 'Compile', 'Russel', 'Taito',
'Agetec', 'GSP', 'Microprose', 'Play It', 'Slightly Mad Studios',
'Tomy Corporation', 'Sammy Corporation', 'Koch Media',
'Game Factory', 'Titus', 'Marvelous Entertainment', 'Genki',
'Mojang', 'Pinnacle', 'CTO SpA', 'TalonSoft', 'Crystal Dynamics',
'SCI', 'Quelle', 'mixi, Inc', 'Rage Software', 'Ubisoft Anney',
'Scholastic Inc.', 'Interplay', 'Mystique', 'ChunSoft',
'Square EA', '20th Century Fox Video Games', 'Avanquest Software',
'Hudson Entertainment', 'Nordic Games', 'Men-A-Vision', 'Nobilis',
'Big Ben Interactive', 'Touchstone', 'Spike', 'Jester Interactive',
'Nippon Ichi Software', 'LEGO Media', 'Quest',
'Illusion Softworks', 'Tigervision', 'Funbox Media',
'Rocket Company', 'Metro 3D', 'Mattel Interactive', 'IE Institute',
'Rondomedia', 'Sony Computer Entertainment America',
'Universal Gamex', 'Ghostlight', 'Wizard Video Games',
'BMG Interactive Entertainment', 'PQube', 'Trion Worlds', 'Laguna',
'Ignition Entertainment', 'Takara', 'Kadokawa Shoten', 'Destineer',
'Enterbrain', 'Xseed Games', 'Imagineer',
'System 3 Arcade Software', 'CPG Products', 'Aruze Corp',
'Gamebridge', 'Midas Interactive Entertainment', 'Jaleco',
'Answer Software', 'XS Games', 'Activision Blizzard',
'Pack In Soft', 'Rebellion', 'Xplosiv', 'Ultravision',
'GameMill Entertainment', 'Wanadoo', 'NovaLogic', 'Telltale Games',
'Epoch', 'BAM! Entertainment', 'Knowledge Adventure', 'Mastiff',
'Tetris Online', 'Harmonix Music Systems', 'ESP', 'TYO',
'Telegames', 'Mud Duck Productions', 'Screenlife', 'Pioneer LDC',
'Magical Company', 'Mentor Interactive', 'Kemco',
'Human Entertainment', 'Avanquest', 'Data Age',
'Electronic Arts Victor', 'Black Bean Games', 'Jack of All Games',
'989 Sports', 'Takara Tomy', 'Media Rings', 'Elf', 'Kalypso Media',
'Starfish', 'Zushi Games', 'Jorudan', 'Destination Software, Inc',
'New', 'Brash Entertainment', 'ITT Family Games', 'PopCap Games',
```

'Home Entertainment Suppliers', 'Ackkstudios', 'Starpath Corp.',
'P2 Games', 'BPS', 'Gathering of Developers', 'NewKidCo',
'Storm City Games', 'CokeM Interactive', 'CBS Electronics',
'Magix', 'Marvelous Interactive', 'Nihon Falcom Corporation',
'Wargaming.net', 'Angel Studios', 'Arc System Works', 'Playmates',
'SNK Playmore', 'Hamster Corporation', 'From Software',
'Nippon Columbia', 'Nichibutsu', 'Little Orbit',
'Conspiracy Entertainment', 'DTP Entertainment', 'Hect',
'Mumbo Jumbo', 'Pacific Century Cyber Works', 'Indie Games',
'Liquid Games', 'NEC', 'Axela', 'ArtDink', 'Sunsoft', 'Gust',
'SNK', 'NEC Interchannel', 'FuRyu', 'Xing Entertainment',
'ValuSoft', 'Victor Interactive', 'Detn8 Games',
'American Softworks', 'Nordcurrent', 'Bomb', 'Falcom Corporation',
'AQ Interactive', 'CCP', 'Milestone S.r.l.', 'Sears',
'JoWood Productions', 'Seta Corporation', 'On Demand', 'NCS',
'Aspyr', 'Gremlin Interactive Ltd', 'Agatsuma Entertainment',
'Compile Heart', 'Culture Brain', 'Mad Catz', 'Shogakukan',
'Merscom LLC', 'Rebellion Developments', 'Nippon Telenet',
'TDK Core', 'bitComposer Games', 'Foreign Media Games', 'Astragon',
'SSI', 'Kadokawa Games', 'Idea Factory',
'Performance Designed Products', 'Asylum Entertainment',
'Core Design Ltd.', 'PlayV', 'UFO Interactive',
'Idea Factory International', 'Playlogic Game Factory',
'Essential Games', 'Adeline Software', 'Funcom',
'Panther Software', 'Blast! Entertainment Ltd', 'Game Life',
'DSI Games', 'Avalon Interactive', 'Popcorn Arcade',
'Neko Entertainment', 'Vir2L Studios', 'Aques', 'Syscom',
'White Park Bay Software', 'System 3', 'Vatical Entertainment',
'Daedalic', 'EA Games', 'Media Factory', 'Vic Tokai',
'The Adventure Company', 'Game Arts', 'Broccoli', 'Acquire',
'General Entertainment', 'Excalibur Publishing', 'Imadio',
'Swing! Entertainment', 'Sony Music Entertainment', 'Aqua Plus',
'Paradox Interactive', 'Hip Interactive',
'DreamCatcher Interactive', 'Tripwire Interactive', 'Sting',
'Yacht Club Games', 'SCS Software', 'Bigben Interactive',
'Havas Interactive', 'Slitherine Software', 'Graffiti', 'Funsta',
'Telstar', 'U.S. Gold', 'DreamWorks Interactive',
'Data Design Interactive', 'MTO', 'DHM Interactive', 'FunSoft',
'SPS', 'Bohemia Interactive', 'Reef Entertainment',
'Tru Blu Entertainment', 'Moss', 'T&E Soft', 'O-Games',
'Aksys Games', 'NDA Productions', 'Data East',
'Time Warner Interactive', 'Gainax Network Systems', 'Daito',
'O3 Entertainment', 'Gameloft', 'Xicat Interactive',
'Simon & Schuster Interactive', 'Valcon Games', 'PopTop Software',
'TOHO', 'HMM Interactive', '5pb', 'Cave',
'CDV Software Entertainment', 'Microids', 'PM Studios', 'Paon',
'Micro Cabin', 'GameTek', 'Benesse', 'Type-Moon',
'Enjoy Gaming ltd.', 'Asmik Corp', 'Interplay Productions',
'Asmik Ace Entertainment', 'inXile Entertainment', 'Image Epoch',
'Phantom EFX', 'Evolved Games', 'responDESIGN',
'Culture Publishers', 'Griffin International', 'Hackberry',
'Hearty Robin', 'Nippon Amuse', 'Origin Systems', 'Seventh Chord',
'Mitsui', 'Milestone', 'Abylight', 'Flight-Plan', 'Glams', 'Locus',
'Warp', 'Daedalic Entertainment', 'Alternative Software',
'Myelin Media', 'Mercury Games', 'Irem Software Engineering',
'Sunrise Interactive', 'Elite', 'Evolution Games', 'Tivola',
'Global A Entertainment', 'Edia', 'Athena', 'Aria', 'Gamecock',
'Tommo', 'Altron', 'Happinet', 'iWin', 'Media Works', 'Fortyfive',
'Revolution Software', 'Imax', 'Crimson Cow', '10TACLE Studios',
'Groove Games', 'Pack-In-Video', 'Insomniac Games',
'Ascaron Entertainment GmbH', 'Asgard', 'Ecole', 'Yumedia',
'Phenomedia', 'HAL Laboratory', 'Grand Prix Games', 'DigiCube',
'Creative Core', 'Kaga Create', 'WayForward Technologies',
'LSP Games', 'ASCII Media Works', 'Coconuts Japan', 'Arika',
'Ertain', 'Marvel Entertainment', 'Prototype',

```
'TopWare Interactive', 'Phantagram', '1C Company',
'The Learning Company', 'TechnoSoft', 'Vap', 'Misawa', 'Tradewest',
'Team17 Software', 'Yeti', 'Pow', 'Navarre Corp', 'MediaQuest',
'Max Five', 'Comfort', 'Monte Christo Multimedia', 'Pony Canyon',
'Riverhillsoft', 'Summitsoft', 'Milestone S.r.l', 'Playmore',
'MLB.com', 'Kool Kizz', 'Flashpoint Games', '49Games',
'Legacy Interactive', 'Alawar Entertainment', 'CyberFront',
'Cloud Imperium Games Corporation', 'Societa',
'Virtual Play Games', 'Interchannel', 'Sonnet', 'Experience Inc.',
'Zenrin', 'Iceberg Interactive', 'Ivolgamus', '2D Boy',
'MC2 Entertainment', 'Kando Games', 'Just Flight', 'Office Create',
'Mamba Games', 'Fields', 'Princess Soft', 'Maximum Family Games',
'Berkeley', 'Fuji', 'Dusenberry Martin Racing', 'imageepoch Inc.',
'Big Fish Games', 'Her Interactive', 'Kamui', 'ASK',
'Headup Games', 'KSS', 'Cygames', 'KID', 'Quinrose', 'Sunflowers',
'dramatic create', 'TGL', 'Encore', 'Extreme Entertainment Group',
'Intergrow', 'G.Rev', 'Sweets', 'Kokopeli Digital Studios',
'Number None', 'Nexon', 'id Software', 'BushiRoad', 'Tryfirst',
'Strategy First', '7G//AMES', 'GN Software', "Yuke's",
'Easy Interactive', 'Licensed 4U', 'FuRyu Corporation',
'Lexicon Entertainment', 'Paon Corporation', 'Kids Station', 'GOA',
'Graphsim Entertainment', 'King Records', 'Introversion Software',
'Minato Station', 'Devolver Digital', 'Blue Byte', 'Gaga',
'Yamasa Entertainment', 'Plenty', 'Views', 'fonfun', 'NetRevo',
'Codemasters Online', 'Quintet', 'Phoenix Games', 'Dorart',
'Marvelous Games', 'Focus Multimedia', 'Imageworks',
'Karin Entertainment', 'Aerosoft', 'Technos Japan Corporation',
'Gakken', 'Mirai Shounen', 'Datam Polystar', 'Saurus', 'HuneX',
'Revolution (Japan)', 'Giza10', 'Visco', 'Alvion', 'Mycom', 'Giga',
'Warashi', 'System Soft', 'Sold Out', 'Lighthouse Interactive',
'Masque Publishing', 'RED Entertainment', 'Michaelsoft',
'Media Entertainment', 'New World Computing', 'Genterprise',
'Interworks Unlimited, Inc.', 'Boost On', 'Stainless Games',
'EON Digital Entertainment', 'Epic Games', 'Naxat Soft',
'Ascaron Entertainment', 'Piacchi', 'Nitroplus',
'Paradox Development', 'Otomate', 'Ongakukan', 'Commseed',
'Inti Creates', 'Takuyo', 'Interchannel-Holon', 'Rain Games',
'UIG Entertainment'], dtype=object)
```

In [62]: *#Check unique platforms. This also can help answer questions for Nintendo.*

```
df_copy['Platform'].unique()
```

Out[62]: array(['Wii', 'NES', 'Gameboy', 'DS', 'X360', 'PS3', 'PS2', 'SNES',
'Gameboy Advance', '3DS', 'PS4', 'N64', 'PS', 'XB', 'PC', '2600',
'PSP', 'XOne', 'GameCube', 'WiiU', 'GEN', 'DC', 'PSV', 'SAT',
'SCD', 'WS', 'NG', 'TG16', '3DO', 'GG', 'PCFX'], dtype=object)

In [63]: *#Check unique genre's. This can help us as well.*

```
df_copy['Genre'].unique()
```

Out[63]: array(['Sports', 'Platform', 'Racing', 'Role-Playing', 'Puzzle', 'Misc',
'Shooter', 'Simulation', 'Action', 'Fighting', 'Adventure',
'Strategy'], dtype=object)

Section 2. Descriptive Questions to Learn More!

In [64]: *#Global Mask Functions for helping with our analysis.*

```
def MASK_GLOBAL_SALES(number=5):
```

```

"""Takes an int input and filters the dataframe based on global sales greater than
return df_copy[df_copy['Global_Sales'] > number]

def MASK_PUBLISHER(publisher):
    """Takes an input as a string and filters the dataframe based on a particular publi
    return df_copy[df_copy['Publisher'] == publisher]

def MASK_NAME_TITLE(game_title):
    """
    Takes an input as a string and filters the dataframes 'Name' column to
    return any games that contain the string provided.
    """
    return df_copy[df_copy['Name'].str.contains(game_title)]

def MASK_PUBLISHER_GENRE(publisher, genre):
    """
    Filters the dataframe based on publisher and genre columns.

    Args:
        publisher (str): The first string.
        genre (str): The second string.

    Returns:
        Dataframe filtered for those cocolumns
    """
    return df_copy[(df_copy['Publisher'] == publisher) & (df_copy['Genre'] == genre)]

def MASK_PUBLISHER_TWO(publisher):
    """Takes an input as a string and filters the dataframe so it does not include a pa
    return df_copy[df_copy['Publisher'] != publisher]

```

1. What are the top 10 games with the highest sales?

In [65]: *#Check out data and see if it's what we want to plot.*
 MASK_GLOBAL_SALES(15)[:10]

Out[65]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales
0	1	Wii Sports	Wii	2006	Sports	Nintendo	41.49	29.02	3.77	8.46
1	2	Super Mario Bros.	NES	1985	Platform	Nintendo	29.08	3.58	6.81	0.77
2	3	Mario Kart Wii	Wii	2008	Racing	Nintendo	15.85	12.88	3.79	3.31
3	4	Wii Sports Resort	Wii	2009	Sports	Nintendo	15.75	11.01	3.28	2.96
4	5	Pokemon Red/Pokemon Blue	Gameboy	1996	Role-Playing	Nintendo	11.27	8.89	10.22	1.00
5	6	Tetris	Gameboy	1989	Puzzle	Nintendo	23.20	2.26	4.22	0.58
6	7	New Super Mario Bros.	DS	2006	Platform	Nintendo	11.38	9.23	6.50	2.90
7	8	Wii Play	Wii	2006	Misc	Nintendo	14.03	9.20	2.93	2.85

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales
8	9	New Super Mario Bros. Wii	Wii	2009	Platform	Nintendo	14.59	7.06	4.70	2.26
9	10	Duck Hunt	NES	1984	Shooter	Nintendo	26.93	0.63	0.28	0.47

```
In [66]: #Data sorting and setting up for plotting.
Top_Ten_Games = MASK_GLOBAL_SALES(15)[:10]

#Create Label making function for labeling each bar in graph.
def addlabels(games,sales):
    for game in range(len(games)):
        plt.text(game, sales[game] - 3, sales[game], ha = 'center', va='center', color=

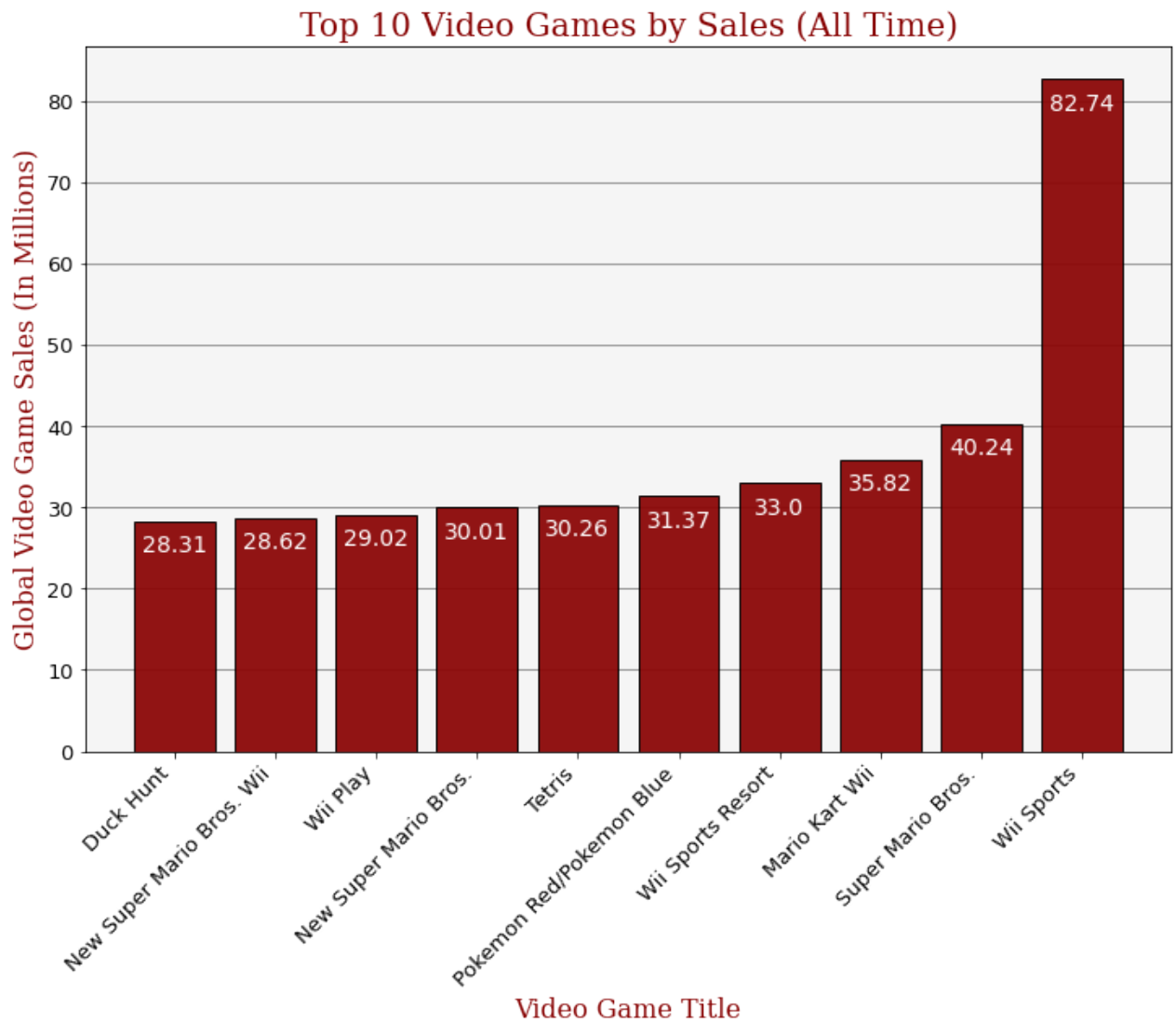
#Custom Fonts
font1 = {'family':'serif','color':'darkred','size':20}
font2 = {'family':'serif','color':'darkred','size':16}

#Create the plot, add y axis gridlines, change background color.
fig, ax = plt.subplots(figsize=(12,8))
ax.grid(axis = 'y', color='grey')
ax.set_facecolor('whitesmoke')

#Model data and set the name of the title, x, and y axis.
ax.bar(x=Top_Ten_Games['Name'], height=Top_Ten_Games['Global_Sales'],alpha=0.9, zorder=
ax.set_title("Top 10 Video Games by Sales (All Time)", fontdict=font1)
ax.set_xlabel("Video Game Title", fontdict=font2, ha="center")
ax.set_ylabel("Global Video Game Sales (In Millions)", fontdict=font2, ha="center")

#Adjust the order of the x axis from Least to most, rotate the x axis names by 45 degree
plt.xticks(rotation=45, ha='right', fontsize=13)
plt.yticks(fontsize=13)
plt.gca().invert_xaxis()
addlabels(Top_Ten_Games['Name'], Top_Ten_Games['Global_Sales'])

plt.show()
```



2. Which Nintendo Platform has the highest video game sales?

```
In [67]: #Still need to style

#Data sorting and setting up for plotting.
Platform_Sales = MASK_PUBLISHER('Nintendo')
Total_Platform_Sales = Platform_Sales.groupby(['Platform'], as_index=False)['Global_Sal

#Custom Fonts
font1 = {'family': 'verdana', 'color': '#000000', 'size': 20}
font2 = {'family': 'verdana', 'color': '#000000', 'size': 16}

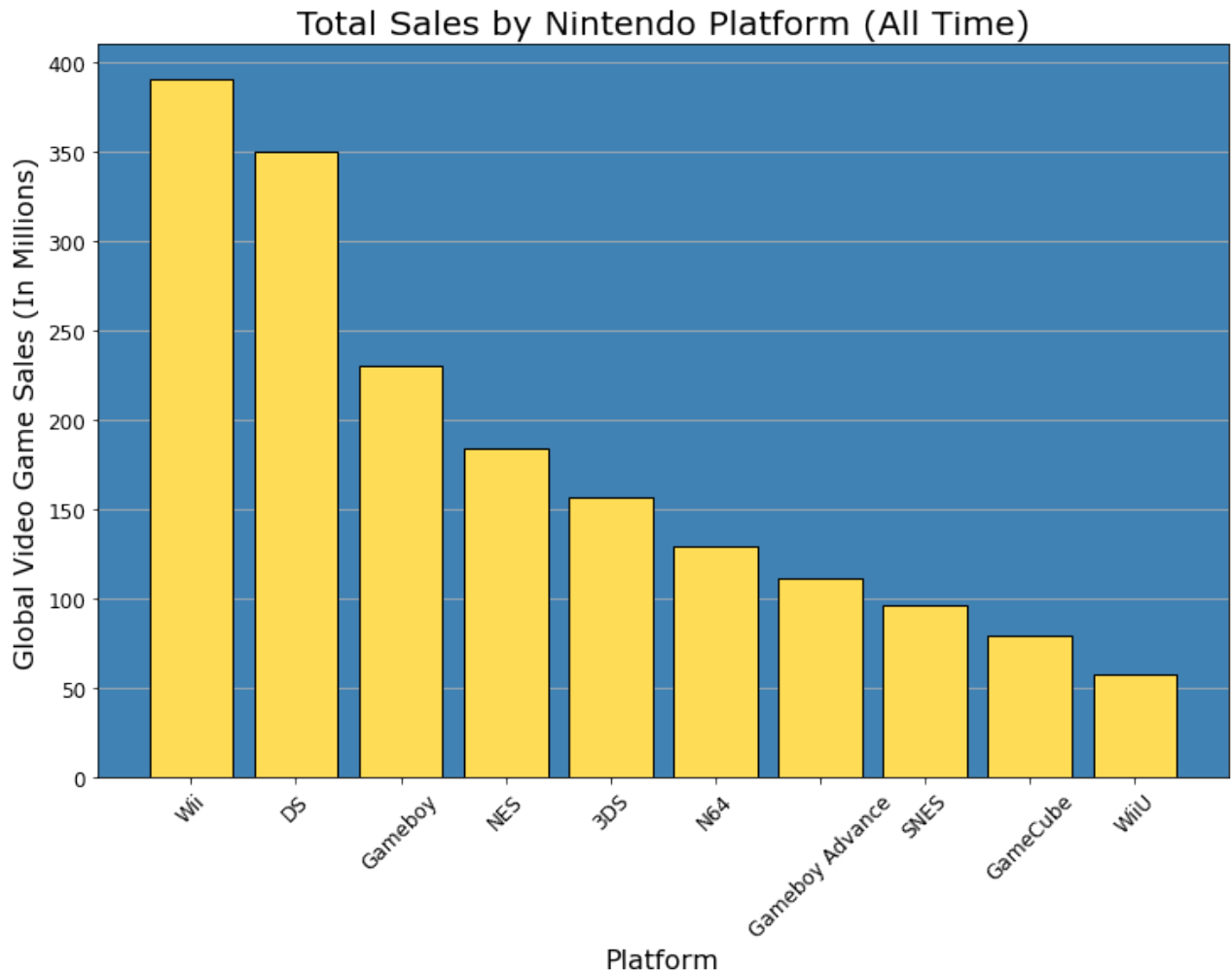
#Create the plot, set x & y axis titles, and graph title.
fig, ax = plt.subplots(figsize=(12,8))
ax.bar(x=Total_Platform_Sales['Platform'], height=Total_Platform_Sales['Global_Sales'],
ax.set_title("Total Sales by Nintendo Platform (All Time)", fontdict=font1)
ax.set_xlabel("Platform", fontdict=font2)
ax.set_ylabel("Global Video Game Sales (In Millions)", fontdict=font2)

#Plot Styling for axes ticks, background image.
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
```

```
plt.gca().invert_xaxis()
ax.set_facecolor('#4584b6')

ax.grid(axis='y')
plt.xticks(rotation=45, ha='center')

plt.show()
```

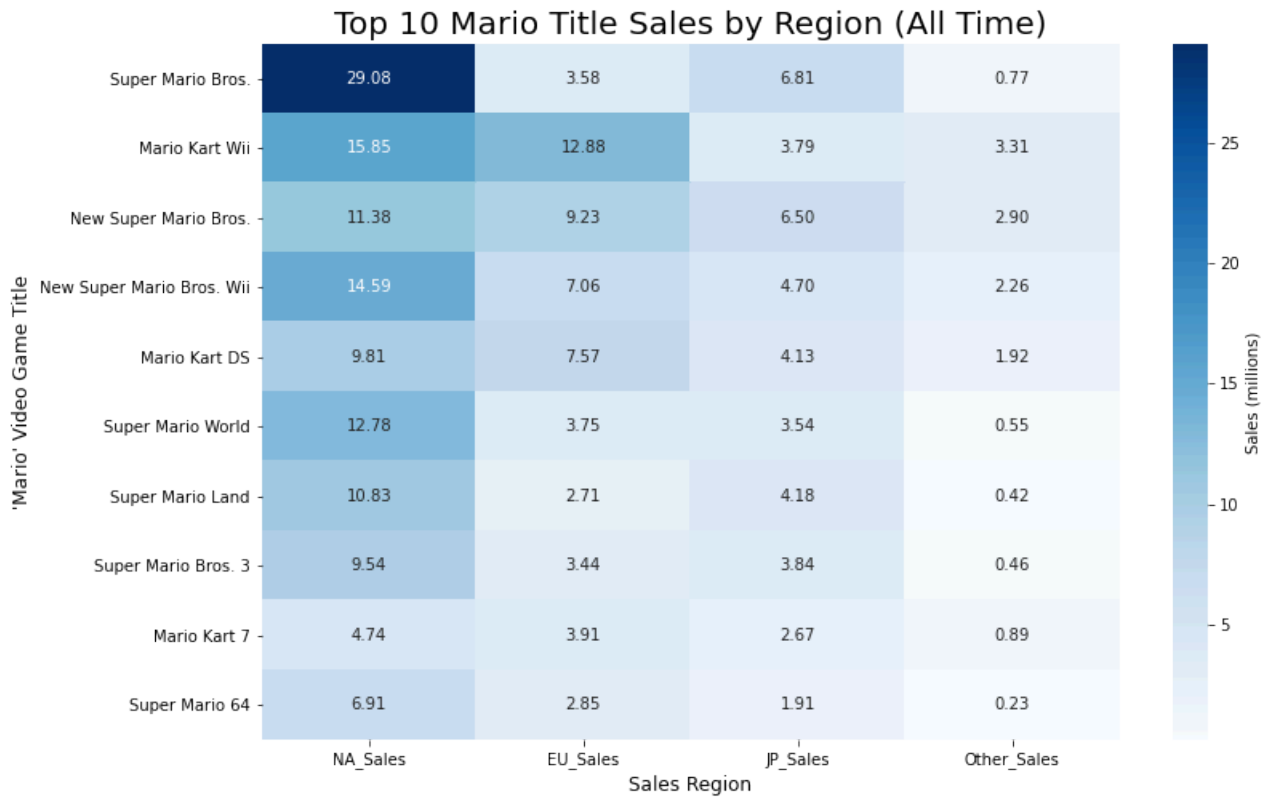


3. Which region has the best performing sales for "Mario" titles?

```
In [68]: #Data sorting and setting up for plotting.
Mario_Sales_by_Region = (MASK_NAME_TITLE("Mario")[:10])["Name", "NA_Sales", "EU_Sales"]
Mario_Sales_by_Region.set_index('Name', inplace=True)

#Create the heatmap
fig, ax = plt.subplots(figsize=(12,8))
sns.heatmap(Mario_Sales_by_Region, annot=True, cmap='Blues', fmt='.2f', cbar_kws={'labe

#Customize the axes
plt.title("Top 10 Mario Title Sales by Region (All Time)", fontsize=20)
plt.xlabel("Sales Region", fontsize=12)
plt.ylabel("'Mario' Video Game Title", fontsize=12)
plt.show()
```



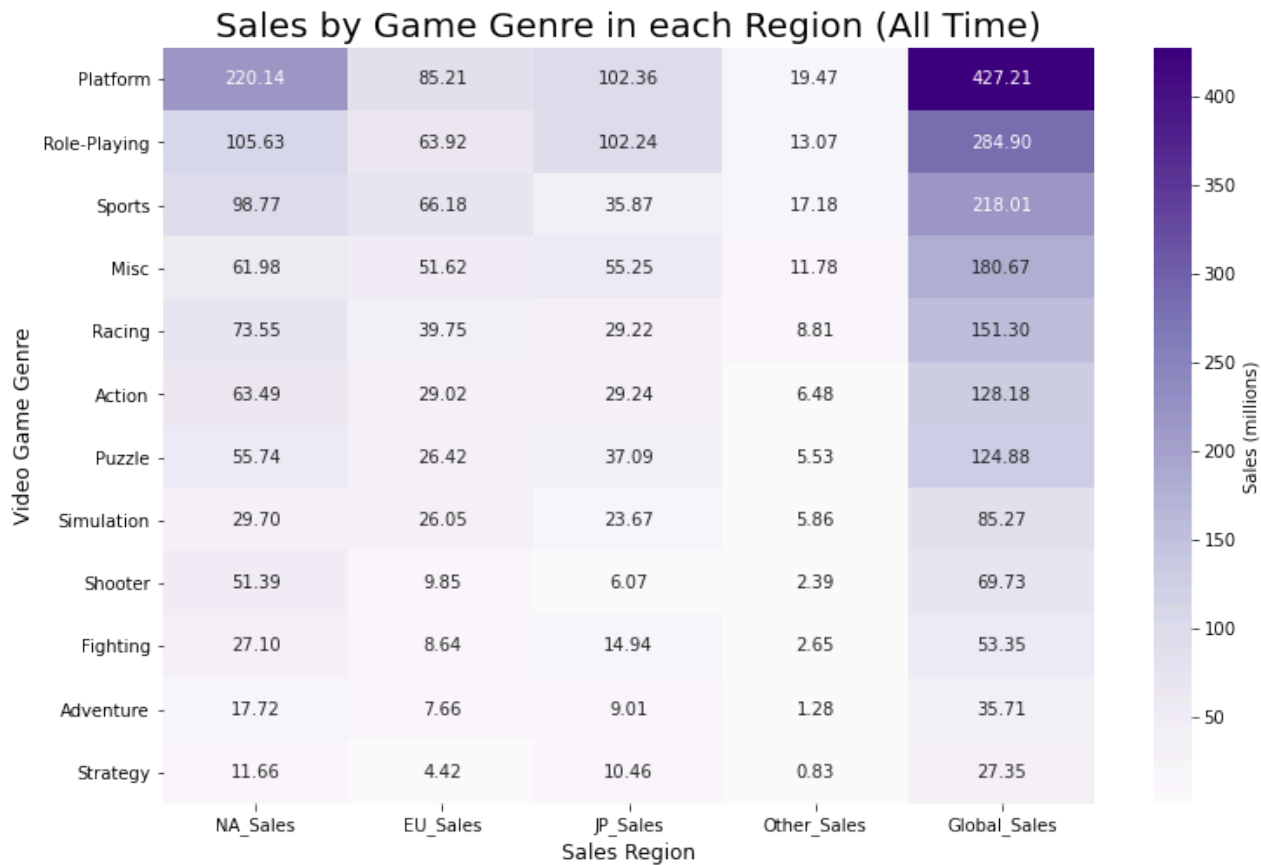
4. Which Genre has the highest sales for Nintendo?

```
In [69]: #Create the variable slicing for modeling the genre's and their sales.
Total_Genre_Sales = Platform_Sales.groupby(['Genre'], as_index=True)[['NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales']]

#Create the heatmap.
fig, ax = plt.subplots(figsize=(12,8))
sns.heatmap(Total_Genre_Sales, annot=True, cmap='Purples', fmt='.2f', cbar_kws={'label': 'Sales (millions)'})

#Customize the axes.
plt.title("Sales by Game Genre in each Region (All Time)", fontsize=20)
plt.xlabel("Sales Region", fontsize=12)
plt.ylabel("Video Game Genre", fontsize=12)

plt.show()
```



5. Which competitors have high sales compared to Nintendo from in recent years?

```
In [70]: #ALL data in tabular format where year is greater than 2015 (most recent).  
Greater_Than_2015
```

Out[70]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other
33	34	Call of Duty: Black Ops 3	PS4	2015	Shooter	Activision	5.77	5.81	0.35	
77	78	FIFA 16	PS4	2015	Sports	Electronic Arts	1.11	6.06	0.06	
92	93	Star Wars Battlefront (2015)	PS4	2015	Shooter	Electronic Arts	2.93	3.29	0.22	
101	102	Call of Duty: Black Ops 3	XOne	2015	Shooter	Activision	4.52	2.09	0.01	
109	110	Fallout 4	PS4	2015	Role-Playing	Bethesda Softworks	2.47	3.15	0.24	
...	
16565	16568	Teslagrad	PSV	2015	Platform	Rain Games	0.00	0.01	0.00	

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other
16569	16572	Resident Evil 4 HD	XOne	2016	Shooter	Capcom	0.01	0.00	0.00	
16570	16573	Farming 2017 - The Simulation	PS4	2016	Simulation	UIG Entertainment	0.00	0.01	0.00	
16576	16579	Rugby Challenge 3	XOne	2016	Sports	Alternative Software	0.00	0.01	0.00	
16589	16592	Chou Ezaru wa Akai Hana: Koi wa Tsuki ni Shiru...	PSV	2016	Action	dramatic create	0.00	0.00	0.01	

961 rows × 11 columns

```
In [71]: #Data sorting and setting up for plotting.
Greater_Than_2015 = df_copy[df_copy['Year'] > 2015]
Sales_By_Publisher_2015_And_Greater = Greater_Than_2015[:50].groupby(['Publisher'], as_

#Custom Fonts & Color Map for Bars in Graph.
font1 = {'family':'serif','color':'red','size':20}
font2 = {'family':'serif','color':'red','size':16}
color_mapping = ['darkgray' for bar in range(15)]
color_mapping[10] = ('#e4000f')

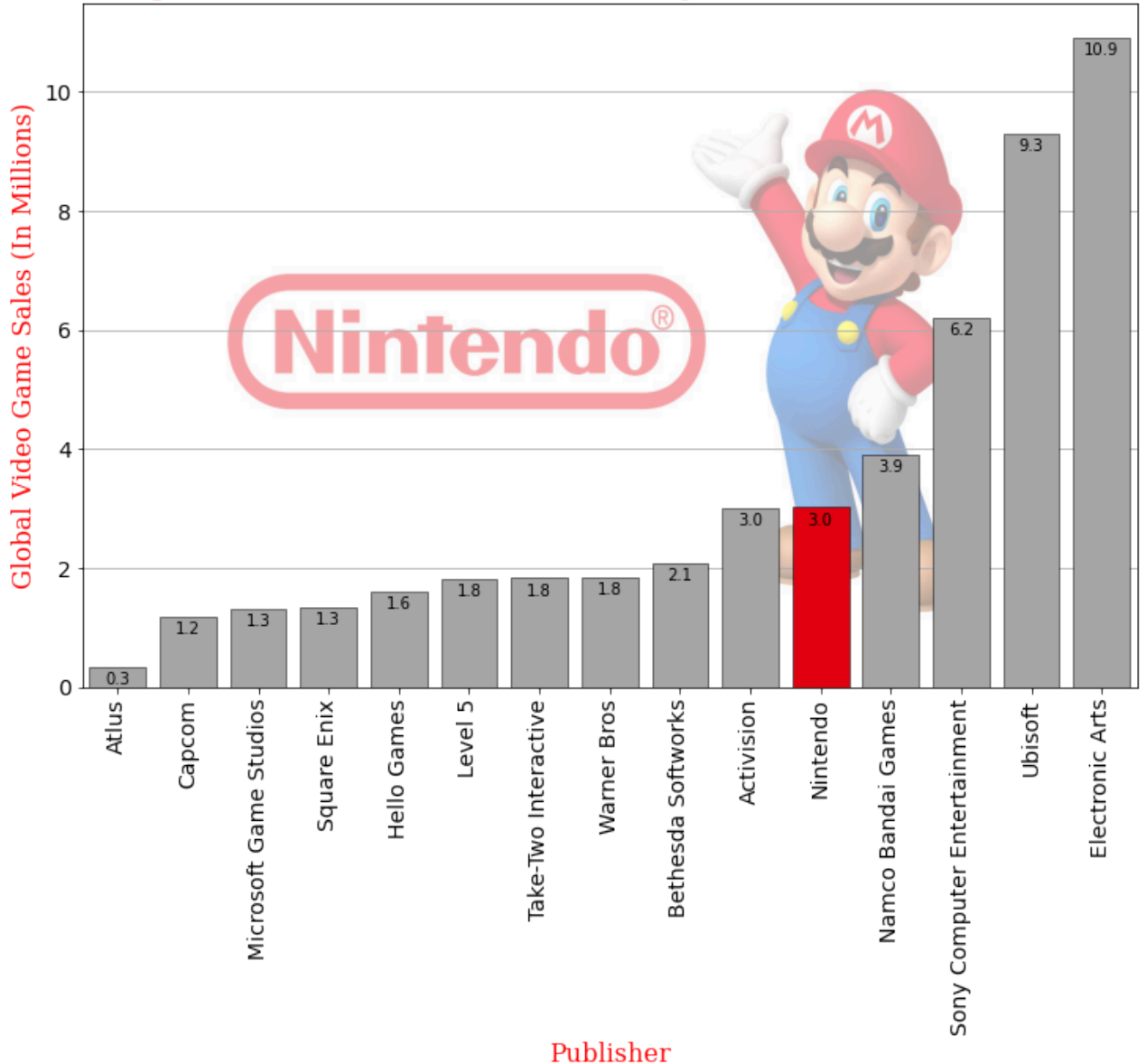
#Create the plot, set x & y axis titles, and graph title.
fig, ax = plt.subplots(figsize=(12,8))
ax.bar(data=Sales_By_Publisher_2015_And_Greater, x=Sales_By_Publisher_2015_And_Greater[
ax.set_xlabel("Publisher", fontdict=font2)
ax.set_ylabel("Global Video Game Sales (In Millions)", fontdict=font2)
ax.set_title("Top 50 Video Games Global Sales by Publisher (2016 & 2017)", fontdict=fon

#Plot Styling for axes ticks, background image.
ax.invert_xaxis()
plt.xticks(rotation=90, ha='center', fontsize=14)
plt.yticks(fontsize=14)
ax.grid(axis='y')
background = plt.imread(r'C:\Users\Chris\Desktop\nintendo_logo.jpg')
ax.imshow(background, extent=[-0.5, 14.5, 0, 11.5], aspect='auto', alpha=0.4)

# Annotate each bar
for bar in ax.patches:
    height = bar.get_height()
    ax.text(bar.get_x() + bar.get_width() / 2,
            height - 0.08,
            f'{height:.1f}',
            ha='center',
            va='top',
            color='black',
            fontsize=10)
```

```
plt.show()
```

Top 50 Video Games Global Sales by Publisher (2016 & 2017)



```
In [72]: #Data sorting and setting up for plotting.
Greater_Than_2015 = df_copy[df_copy['Year'] >= 2015]
Platform_Sales_From_2015 = Greater_Than_2015.groupby('Platform', as_index=False)[['Platform', 'Sales']]

#Create a color map variable for bar styling.
color_map = ['cyan' for bar in range(10)]
color_map[0] = color_map[6] = color_map[7] = ('#DFFF00')

#Create the plot and barchart, set axis names and styling.
fig, ax = plt.subplots(figsize=(12,8))
ax.bar(data=Platform_Sales_From_2015, x=Platform_Sales_From_2015['Platform'], height=Platform_Sales_From_2015['Sales'])
ax.set_xlabel("Platforms", fontsize=16)
ax.set_ylabel("Global Video Game Sales (In Millions)", fontsize=16)
ax.set_title("Global Video Game Sales by Platform (2015-2017)", fontsize=20)

#Change styling of x and y axis ticks, add in grid background, adjust grid face color, and font color.
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)
```

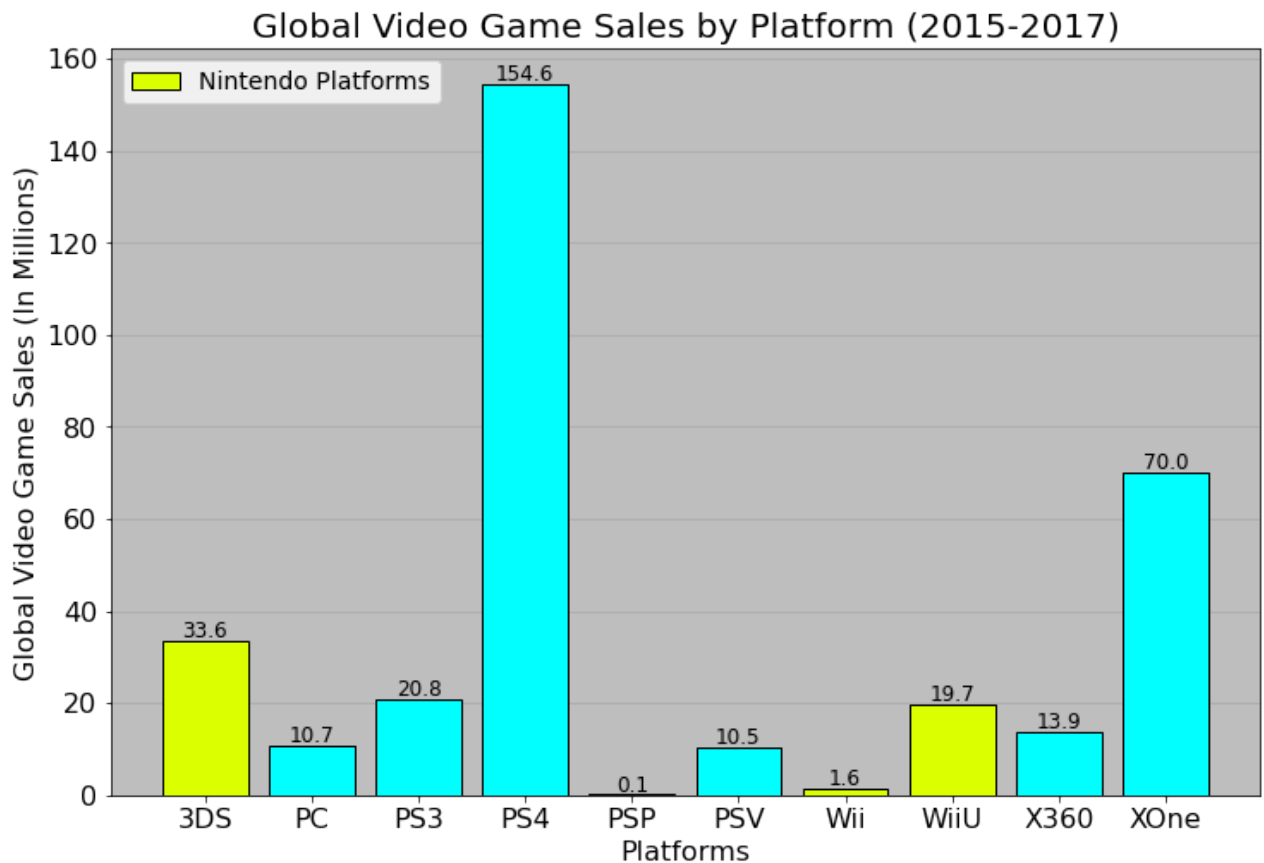
```

ax.grid(axis='y', color='gray', alpha=0.3)
ax.set_facecolor('#C0C0C0')
plt.legend(['Nintendo Platforms'], loc='upper left', fontsize=14)

# Annotate each bar with the global sales above the top of the bar.
for bar in ax.patches:
    height = bar.get_height()
    ax.text(bar.get_x() + bar.get_width() / 2,
            height + 4.5,
            f'{height:.1f}',
            ha='center',
            va='top',
            color='black',
            fontsize=12)

plt.show()

```



Section 3: Analysis & Recommendation

Analysis

Nintendo has been one of the best selling video game companies of all time, with success dating back to 1985 with the super successful launch of the Nintendo Entertainment System and the Super Mario Bros Brand that remains immensely competitive to this day. It is clear from our data analysis that "Mario" branded video games is a dominant revenue stream for Nintendo, having 4 titles in the top 10 video games with best global sales. Subsequently, this was also indicated as most "Mario" branded video games are platformers, which our data

visualization has shown that platformer genres are the best performing genre for every region in which we have data for, in our dataset. The trend for Nintendo platforms appears that handheld consoles (I.E. Gameboy, Gameboy Advance, 3DS, etc.) have had video games with increasing sales as time goes on year over year when compared to traditional "at-home" platforms, such as the Gamecube and N64. This may suggest that the target market is moving towards hand-held devices for their gaming experience as time continues. Additionally, when looking at the top 50 global video game sales and comparing Nintendo's competitors, the marketplace is very competitive with competition such as the Playstation 4 and Xbox One which is pulling sales numbers and revenue away from Nintendo. Among the top 15 companies with video game sales ranging more recently from 2016 - 2017, Nintendo ranks #5 of 15, which is very competitive. However, Nintendo is in a great position and I believe has opportunity to grow.

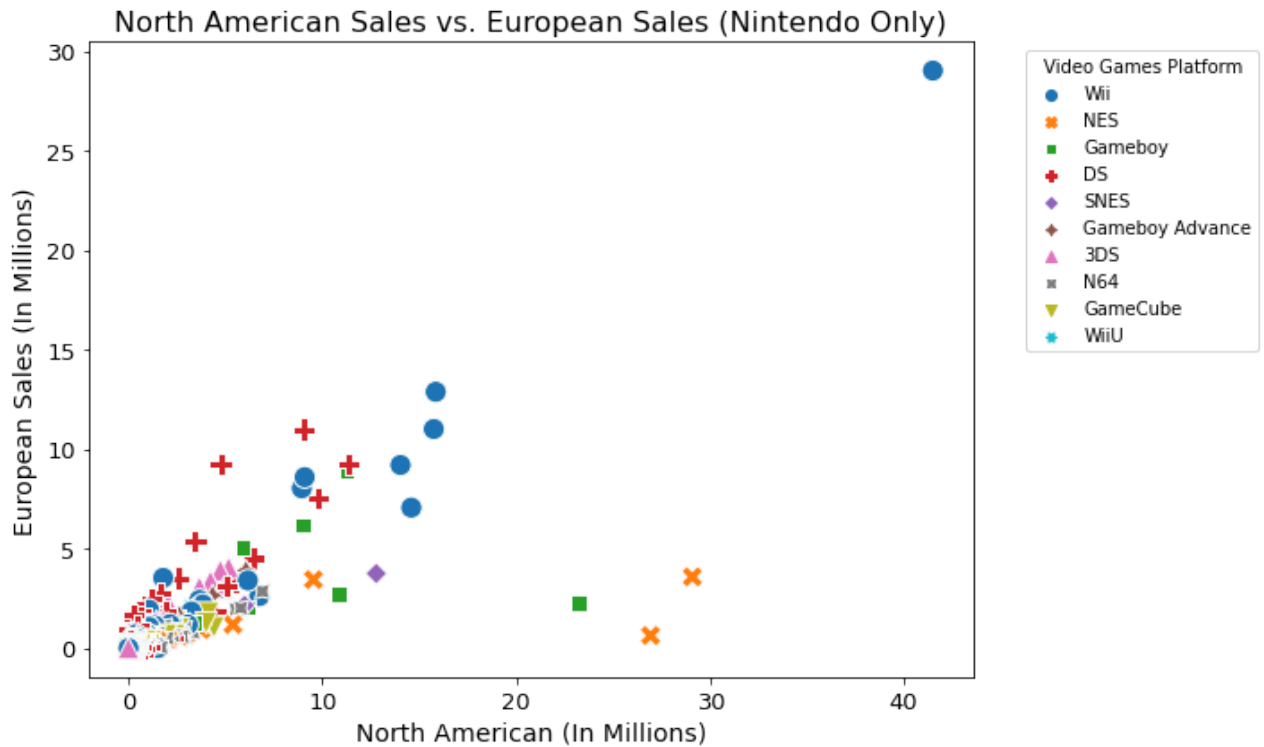
Recommendation

With video game sales increasing with hand-held consoles as time increases and with Wii video game sales being so high, this could be a good indicator on the next generation console. I recommend investing more money into Research and Development for the next generation console. The data suggests that there is still a huge demand for 'at-home' consoles and there is also a huge demand for 'hand-held' consoles. A hybrid approach may work best for the next generation console, where the user experience can be at home if desired, or on the go. 'Mario' themed games are a huge revenue source for Nintendo, and I recommend to continue building upon the brand and franchise, as historical sales data for Platform, Role-Playing, and Sports genres are the best performing genres in North America, Europe, and Japan. Subsequently, I recommend putting additional capital into market research for the European and Japanese markets, as sales are not performing as well when compared to North America. We need to understand why, and change our strategy accordingly. Additionally, funding for market research in North America is appropriate as Nintendo is very competitive, with their highest sales numbers being in the North American market. Nintendo can continue to keep their competitive edge and strengthen it by identifying new trends within the target market and strategize towards them. In addition, I recommend Nintendo to create new titles in the Roleplaying and Platformer genres, not just for "Mario" titles. These have been the best performing genres historically of all video games, which indicates that these are very popular genres for the target market in all regions. Focusing on creating new video games in these categories may help bring even more revenue in for Nintendo. Finally, I recommend to negotiate with more game studios to bring their video games to Nintendo platforms. Playstation 4 and Xbox One have substantially more sales when compared to the Nintendo 3DS, Wii, and WiiU. Building and leveraging existing relationships to increase sales by creating new titles for Nintendo platforms can help increase sales and revenue for the company.

Section 4: Appendix

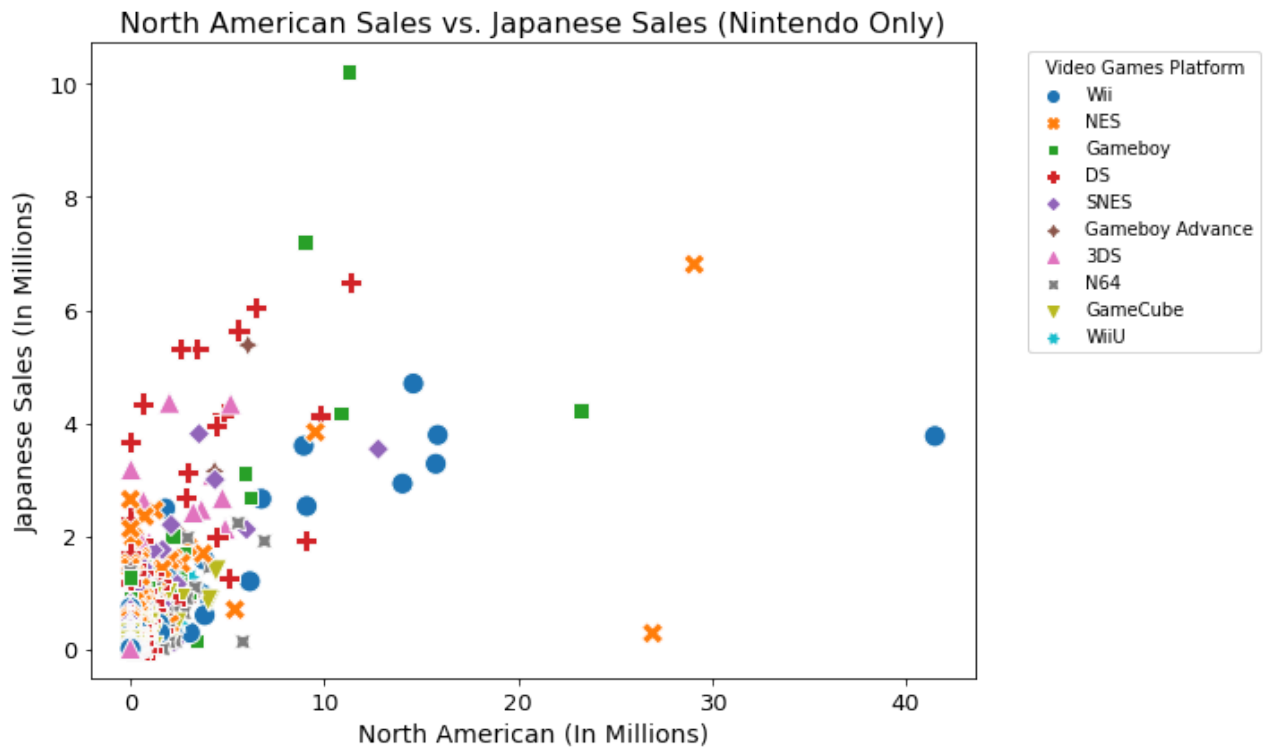
```
In [73]: # Relationship analysis for Nintendo's sales by region
plt.figure(figsize=(10, 6))
sns.scatterplot(data=Platform_Sales, x='NA_Sales', y='EU_Sales', hue='Platform', style=
plt.title("North American Sales vs. European Sales (Nintendo Only)", fontsize=16)
plt.xlabel('North American (In Millions)', fontsize=14)
plt.ylabel('European Sales (In Millions)', fontsize=14)
plt.xticks(fontsize=13)
plt.yticks(fontsize=13)
plt.legend(title='Video Games Platform', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()

plt.show()
```



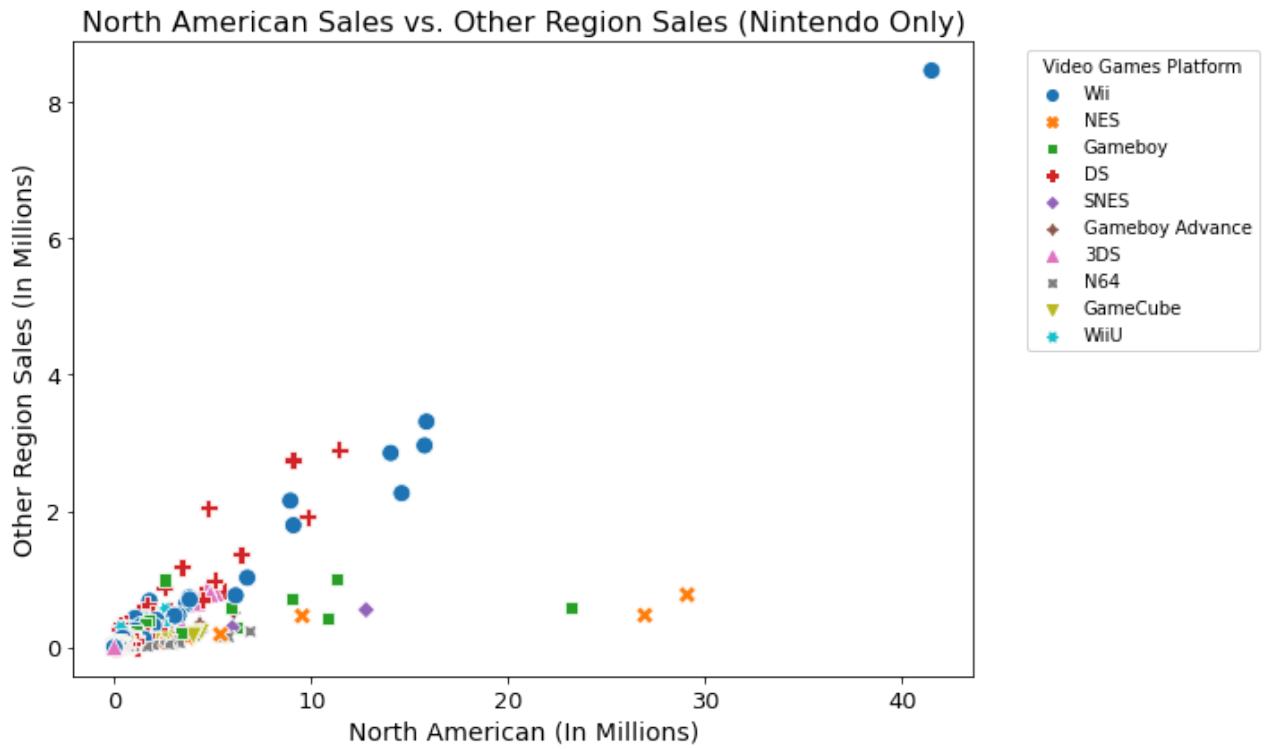
```
In [74]: # Relationship analysis for Nintendo's sales by region
plt.figure(figsize=(10, 6))
sns.scatterplot(data=Platform_Sales, x='NA_Sales', y='JP_Sales', hue='Platform', style=
plt.title("North American Sales vs. Japanese Sales (Nintendo Only)", fontsize=16)
plt.xlabel('North American (In Millions)', fontsize=14)
plt.ylabel('Japanese Sales (In Millions)', fontsize=14)
plt.xticks(fontsize=13)
plt.yticks(fontsize=13)
plt.legend(title='Video Games Platform', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()

plt.show()
```



```
In [75]: # Relationship analysis for Nintendo's sales by region
plt.figure(figsize=(10, 6))
sns.scatterplot(data=Platform_Sales, x='NA_Sales', y='Other_Sales', hue='Platform', style='Platform')
plt.title("North American Sales vs. Other Region Sales (Nintendo Only)", fontsize=16)
plt.xlabel('North American (In Millions)', fontsize=14)
plt.ylabel('Other Region Sales (In Millions)', fontsize=14)
plt.legend(title='Video Games Platform', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.xticks(fontsize=13)
plt.yticks(fontsize=13)
plt.tight_layout()

plt.show()
```



```
In [37]: plt.figure(figsize=(10, 6))
sns.scatterplot(data=Platform_Sales, x='Year', y='Global_Sales', hue='Platform', style=
plt.title("Year vs Global Sales", fontsize=20)
plt.xlabel('Year', fontsize=14)
plt.ylabel('Global Sales (In Millions)', fontsize=14)
plt.xlim(1982, 2017)
plt.legend(title='Video Games Platform', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.xticks(fontsize=13)
plt.yticks(fontsize=13)
plt.tight_layout()

plt.show()
```

