Investigate_a_Dataset

January 6, 2021

> **Tip**: Welcome to the Investigate a Dataset project! You will find tips in quoted sections like this to help organize your approach to your investigation. Before submitting your project, it will be a good idea to go back through your report and remove these sections to make the presentation of your work as tidy as possible. First things first, you might want to double-click this Markdown cell and change the title so that it reflects your dataset and investigation.

1 Project: Looking at Video Game Sales

1.1 Table of Contents

> **Tip**: In this section of the report, provide a brief introduction to the dataset you've selected for analysis. At the end of this section, describe the questions that you plan on exploring over the course of the report. Try to build your report around the analysis of at least one dependent variable and three independent variables. If you're not sure what questions to ask, then make sure you familiarize yourself with the dataset, its variables and the dataset context for ideas of what to explore.
>
> If you haven't yet selected and downloaded your data, make sure you do that first before coming back here. In order to work with the data in this workspace, you also need to upload it to the workspace. To do so, click on the jupyter icon in the upper left to be taken back to the workspace directory. There should be an 'Upload' button in the upper right that will let you add your data file(s) to the workspace. You can then click on the .ipynb file name to come back here.

In [11]:
```
# Use this cell to set up import statements for all of the packages that you
#   plan to use.
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
% matplotlib inline
```

```
# Remember to include a 'magic word' so that your visualizations are plotted
#   inline with the notebook. See this page for more:
#   http://ipython.readthedocs.io/en/stable/interactive/magics.html
```

Data Wrangling

Tip: In this section of the report, you will load in the data, check for cleanliness, and then trim and clean your dataset for analysis. Make sure that you document your steps carefully and justify your cleaning decisions.

1.1.1 General Properties

In [12]:
```
# Load your data and print out a few lines. Perform operations to inspect data
#   types and look for instances of missing or possibly errant data.
# I got my dataset from Kaggle
df = pd.read_csv('vgsales.csv')
df.head(1000)
```

Out[12]:

	Rank	Name	Platform	Year	\
0	1	Wii Sports	Wii	2006	
1	2	Super Mario Bros.	NES	1985	
2	3	Mario Kart Wii	Wii	2008	
3	4	Wii Sports Resort	Wii	2009	
4	5	Pokemon Red/Pokemon Blue	GB	1996	
5	6	Tetris	GB	1989	
6	7	New Super Mario Bros.	DS	2006	
7	8	Wii Play	Wii	2006	
8	9	New Super Mario Bros. Wii	Wii	2009	
9	10	Duck Hunt	NES	1984	
10	11	Nintendogs	DS	2005	
11	12	Mario Kart DS	DS	2005	
12	13	Pokemon Gold/Pokemon Silver	GB	1999	
13	14	Wii Fit	Wii	2007	
14	15	Wii Fit Plus	Wii	2009	
15	16	Kinect Adventures!	X360	2010	
16	17	Grand Theft Auto V	PS3	2013	
17	18	Grand Theft Auto: San Andreas	PS2	2004	
18	19	Super Mario World	SNES	1990	
19	20	Brain Age: Train Your Brain in Minutes a Day	DS	2005	
20	21	Pokemon Diamond/Pokemon Pearl	DS	2006	
21	22	Super Mario Land	GB	1989	
22	23	Super Mario Bros. 3	NES	1988	
23	24	Grand Theft Auto V	X360	2013	
24	25	Grand Theft Auto: Vice City	PS2	2002	

25	26	Pokemon Ruby/Pokemon Sapphire	GBA	2002
26	27	Pokemon Black/Pokemon White	DS	2010
27	28	Brain Age 2: More Training in Minutes a Day	DS	2005
28	29	Gran Turismo 3: A-Spec	PS2	2001
29	30	Call of Duty: Modern Warfare 3	X360	2011
..	...	...	...	...
970	972	Kinect Star Wars	X360	2012
971	973	Midnight Club II	PS2	2003
972	974	Dragon Quest Monsters: Joker	DS	2006
973	975	SpongeBob SquarePants: SuperSponge	PS	2001
974	976	The Getaway: Black Monday	PS2	2004
975	977	Professor Layton and the Mask of Miracle	3DS	2011
976	978	Just Cause 2	PS3	2010
977	979	Dragon's Dogma	PS3	2012
978	980	The Legend of Zelda: The Wind Waker	WiiU	2013
979	981	50 Cent: Bulletproof	PS2	2005
980	982	High School Musical: Sing It!	Wii	2007
981	983	Wii Party U	WiiU	2013
982	984	Madden NFL 25	PS3	2013
983	985	Final Fantasy II	SNES	1991
984	986	Kirby 64: The Crystal Shards	N64	2000
985	987	Dead or Alive 3	XB	2001
986	988	UFC 2009 Undisputed	PS3	2009
987	989	Metroid II: Return of Samus	GB	1991
988	990	WWF Attitude	PS	1998
989	991	The SpongeBob SquarePants Movie	PS2	2004
990	992	Golden Sun	GBA	2001
991	993	Sonic the Hedgehog 3	GEN	1994
992	994	Kid Icarus	NES	1986
993	995	Def Jam: Fight for NY	PS2	2004
994	996	Tom Clancy's Ghost Recon	XB	2002
995	997	State of Emergency	PS2	2002
996	998	BioShock Infinite	PS3	2013
997	999	Hitman: Absolution	X360	2012
998	1000	2 Games in 1 Double Pack: The Incredibles / Fi...	GBA	2007
999	1001	Call of Duty: Black Ops 3	X360	2015

	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	\
0	Sports	Nintendo	41.49	29.02	3.77	
1	Platform	Nintendo	29.08	3.58	6.81	
2	Racing	Nintendo	15.85	12.88	3.79	
3	Sports	Nintendo	15.75	11.01	3.28	
4	Role-Playing	Nintendo	11.27	8.89	10.22	
5	Puzzle	Nintendo	23.20	2.26	4.22	
6	Platform	Nintendo	11.38	9.23	6.50	
7	Misc	Nintendo	14.03	9.20	2.93	
8	Platform	Nintendo	14.59	7.06	4.70	
9	Shooter	Nintendo	26.93	0.63	0.28	

10	Simulation	Nintendo	9.07	11.00	1.93
11	Racing	Nintendo	9.81	7.57	4.13
12	Role-Playing	Nintendo	9.00	6.18	7.20
13	Sports	Nintendo	8.94	8.03	3.60
14	Sports	Nintendo	9.09	8.59	2.53
15	Misc	Microsoft Game Studios	14.97	4.94	0.24
16	Action	Take-Two Interactive	7.01	9.27	0.97
17	Action	Take-Two Interactive	9.43	0.40	0.41
18	Platform	Nintendo	12.78	3.75	3.54
19	Misc	Nintendo	4.75	9.26	4.16
20	Role-Playing	Nintendo	6.42	4.52	6.04
21	Platform	Nintendo	10.83	2.71	4.18
22	Platform	Nintendo	9.54	3.44	3.84
23	Action	Take-Two Interactive	9.63	5.31	0.06
24	Action	Take-Two Interactive	8.41	5.49	0.47
25	Role-Playing	Nintendo	6.06	3.90	5.38
26	Role-Playing	Nintendo	5.57	3.28	5.65
27	Puzzle	Nintendo	3.44	5.36	5.32
28	Racing	Sony Computer Entertainment	6.85	5.09	1.87
29	Shooter	Activision	9.03	4.28	0.13
..	...	...	...	...	...
970	Action	Microsoft Game Studios	1.05	0.57	0.03
971	Racing	Take-Two Interactive	1.25	0.29	0.00
972	Role-Playing	Square Enix	0.23	0.03	1.49
973	Action	THQ	1.12	0.58	0.00
974	Action	Sony Computer Entertainment	0.39	1.01	0.02
975	Puzzle	Nintendo	0.32	0.95	0.36
976	Action	Square Enix	0.45	0.94	0.06
977	Role-Playing	Capcom	0.41	0.46	0.72
978	Action	Nintendo	0.93	0.57	0.14
979	Action	Vivendi Games	0.85	0.76	0.00
980	Misc	Disney Interactive Studios	1.16	0.45	0.00
981	Misc	Nintendo	0.31	0.54	0.84
982	Sports	Electronic Arts	1.59	0.03	0.00
983	Role-Playing	Square	0.24	0.09	1.33
984	Platform	Nintendo	0.63	0.06	1.03
985	Fighting	Microsoft Game Studios	1.19	0.29	0.24
986	Fighting	THQ	1.07	0.45	0.01
987	Adventure	Nintendo	0.85	0.31	0.56
988	Fighting	Acclaim Entertainment	1.27	0.42	0.00
989	Platform	THQ	1.06	0.54	0.00
990	Role-Playing	Nintendo	0.93	0.38	0.40
991	Platform	Sega	1.02	0.47	0.20
992	Platform	Nintendo	0.53	0.12	1.09
993	Fighting	Electronic Arts	0.86	0.67	0.00
994	Shooter	Ubisoft	1.23	0.46	0.00
995	Action	Take-Two Interactive	0.86	0.67	0.00
996	Shooter	Take-Two Interactive	0.72	0.69	0.04

997	Action	Square Enix	0.68	0.90	0.01	
998	Action	THQ	1.26	0.47	0.00	
999	Shooter	Activision	1.11	0.48	0.00	

	Other_Sales	Global_Sales
0	8.46	82.74
1	0.77	40.24
2	3.31	35.82
3	2.96	33.00
4	1.00	31.37
5	0.58	30.26
6	2.90	30.01
7	2.85	29.02
8	2.26	28.62
9	0.47	28.31
10	2.75	24.76
11	1.92	23.42
12	0.71	23.10
13	2.15	22.72
14	1.79	22.00
15	1.67	21.82
16	4.14	21.40
17	10.57	20.81
18	0.55	20.61
19	2.05	20.22
20	1.37	18.36
21	0.42	18.14
22	0.46	17.28
23	1.38	16.38
24	1.78	16.15
25	0.50	15.85
26	0.82	15.32
27	1.18	15.30
28	1.16	14.98
29	1.32	14.76
..	...	...
970	0.14	1.78
971	0.24	1.78
972	0.03	1.78
973	0.08	1.78
974	0.36	1.78
975	0.14	1.78
976	0.33	1.78
977	0.19	1.78
978	0.13	1.77
979	0.16	1.77
980	0.16	1.77
981	0.08	1.77

```
982         0.15          1.77
983         0.12          1.77
984         0.04          1.77
985         0.06          1.77
986         0.24          1.77
987         0.04          1.76
988         0.07          1.76
989         0.16          1.76
990         0.06          1.76
991         0.07          1.76
992         0.02          1.76
993         0.22          1.76
994         0.07          1.76
995         0.22          1.76
996         0.31          1.76
997         0.17          1.76
998         0.03          1.76
999         0.16          1.76

[1000 rows x 11 columns]
```

```
In [13]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16598 entries, 0 to 16597
Data columns (total 11 columns):
Rank            16598 non-null int64
Name            16598 non-null object
Platform        16598 non-null object
Year            16598 non-null int64
Genre           16598 non-null object
Publisher       16598 non-null object
NA_Sales        16598 non-null float64
EU_Sales        16598 non-null float64
JP_Sales        16598 non-null float64
Other_Sales     16598 non-null float64
Global_Sales    16598 non-null float64
dtypes: float64(5), int64(2), object(4)
memory usage: 1.4+ MB
```

```
In [14]: df.describe()
```

```
Out[14]:                Rank          Year      NA_Sales      EU_Sales      JP_Sales  \
         count  16598.000000  16598.000000  16598.000000  16598.000000  16598.000000
         mean    8300.605254   2006.138571      0.264667      0.146652      0.077782
         std     4791.853933      6.143743      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
```

```
In [15]: for i, v in enumerate(df.columns):
             print(i, v)
```

```
0 Rank
1 Name
2 Platform
3 Year
4 Genre
5 Publisher
6 NA_Sales
7 EU_Sales
8 JP_Sales
9 Other_Sales
10 Global_Sales
```

```
In [16]: df.nunique()
```

```
Out[16]: Rank            16598
         Name            11493
         Platform           31
         Year               39
         Genre              12
         Publisher         579
         NA_Sales          409
         EU_Sales          305
         JP_Sales          244
         Other_Sales       157
         Global_Sales      623
         dtype: int64
```

```
In [17]: sum(df.duplicated())
```

```
Out[17]: 0
```

Tip: You should *not* perform too many operations in each cell. Create cells freely to explore your data. One option that you can take with this project is to do a lot of explorations in an initial notebook. These don't have to be organized, but make sure you use enough comments to understand the purpose of each code cell. Then, after you're done with your analysis, create a duplicate notebook where you will trim the excess and organize your steps so that you have a flowing, cohesive report.

Tip: Make sure that you keep your reader informed on the steps that you are taking in your investigation. Follow every code cell, or every set of related code cells, with a markdown cell to describe to the reader what was found in the preceding cell(s). Try to make it so that the reader can then understand what they will be seeing in the following cell(s).

1.1.2 Data Cleaning (Making a few Modifications)

1.1.3 Base on ".info()" method, there isn't any null values in each of the columns but I will get rid of all the '_sales' of each of the columns since we know that we are dealing with sales.

In [18]:
```
# After discussing the structure of the data and any problems that need to be
#   cleaned, perform those cleaning steps in the second part of this section.
# I will get rid of all the '_sales' of each of the columns since we know that we are d
new_labels = []
for col in df.columns:
    if '_Sales' in col:
        new_labels.append(col[:-6])  # exclude last 6 characters
    else:
        new_labels.append(col)
new_labels
```

Out[18]:
```
['Rank',
 'Name',
 'Platform',
 'Year',
 'Genre',
 'Publisher',
 'NA',
 'EU',
 'JP',
 'Other',
 'Global']
```

In [19]:
```
#Check to see if the labels are correctly published.
df.columns = new_labels
df.head(1)
```

Out[19]:

	Rank	Name	Platform	Year	Genre	Publisher	NA	EU	JP	\
0	1	Wii Sports	Wii	2006	Sports	Nintendo	41.49	29.02	3.77	

```
   Other  Global
0   8.46   82.74
```

In [ ]:

In [ ]:

Exploratory Data Analysis

Tip: Now that you've trimmed and cleaned your data, you're ready to move on to exploration. Compute statistics and create visualizations with the goal of addressing the research questions that you posed in the Introduction section. It is recommended that you be systematic with your approach. Look at one variable at a time, and then follow it up by looking at relationships between variables.

1.1.4 Question: Does certain genre of games sell better in countries (Japan, North America, Europe)?

1.2 I wanted to campare the sales for each genre for Japan, North America, and Europe

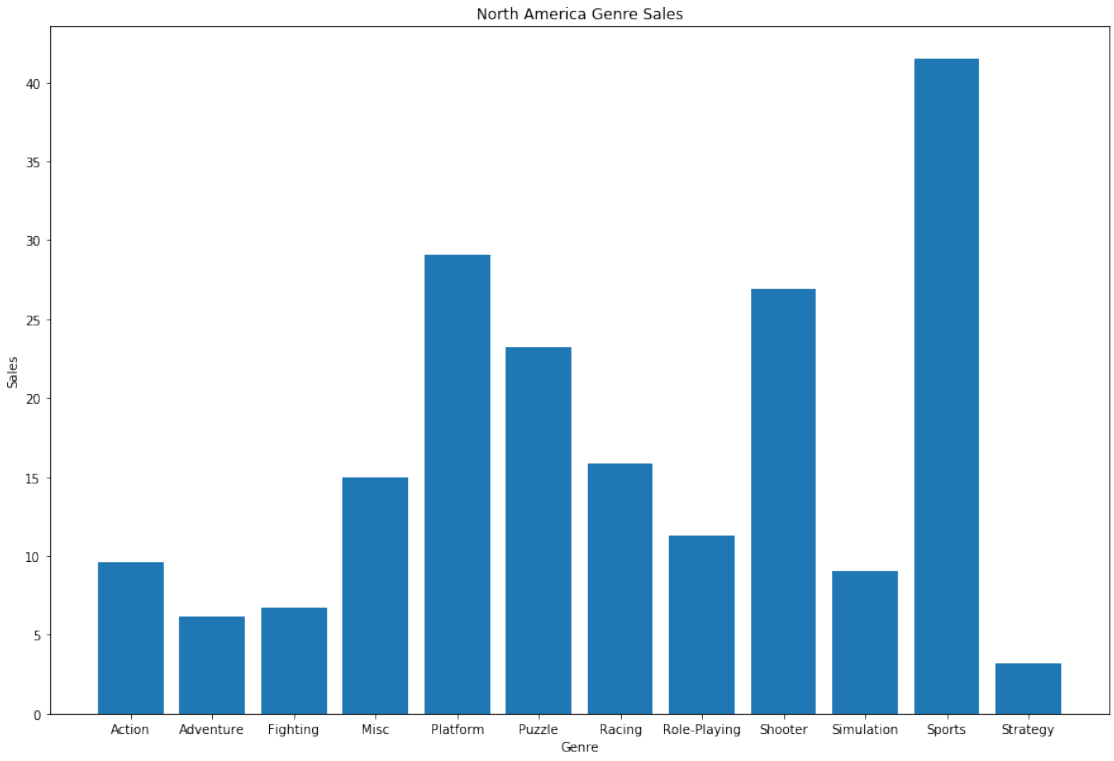
In [27]:
```
# Use this, and more code cells, to explore your data. Don't forget to add
#   Markdown cells to document your observations and findings.
x = df['Genre']
y = df['NA']



plt.figure(figsize= (15,10))

plt.bar(x , y)
plt.title('North America Genre Sales')
plt.xlabel('Genre')
plt.ylabel('Sales')
```

Out[27]:
```
Text(0,0.5,'Sales')
```



1.2.1 Sports has the highest sales with platform second.

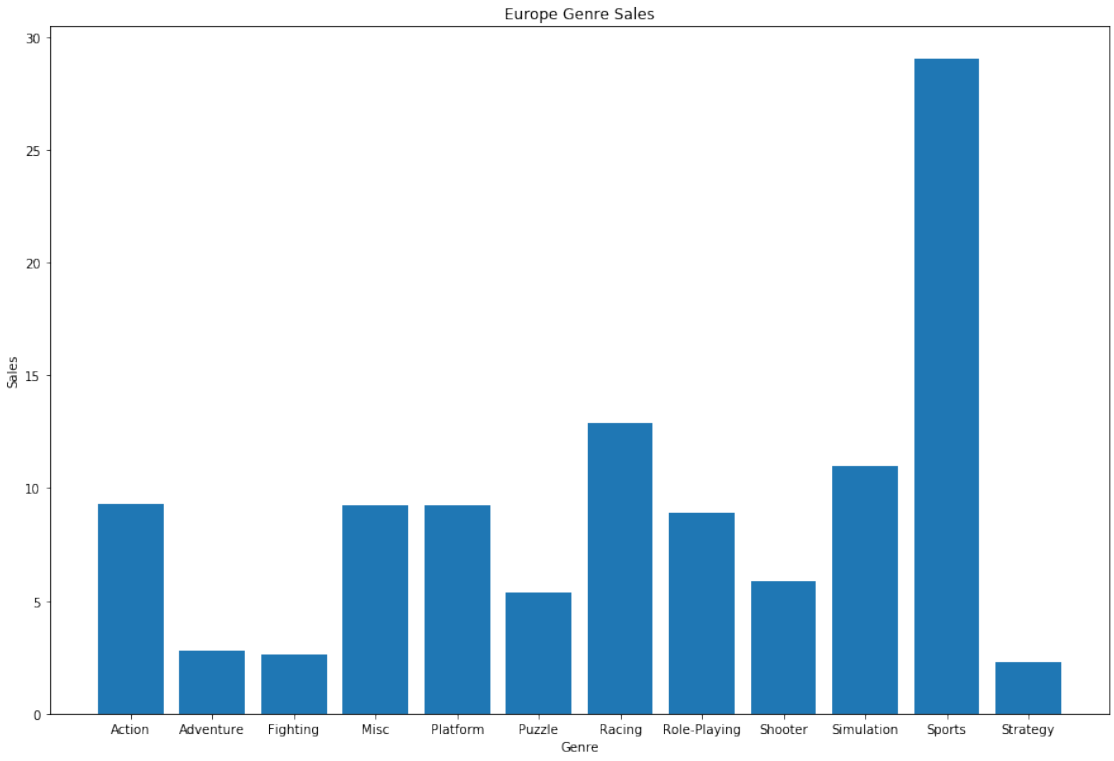
```
In [21]: x = df['Genre']
         y = df['EU']



         plt.figure(figsize= (15,10))

         plt.bar(x , y)
         plt.title('Europe Genre Sales')
         plt.xlabel('Genre')
         plt.ylabel('Sales')

Out[21]: Text(0,0.5,'Sales')
```



1.2.2 Sports games also reach very high sales with Racing in second for Europe.

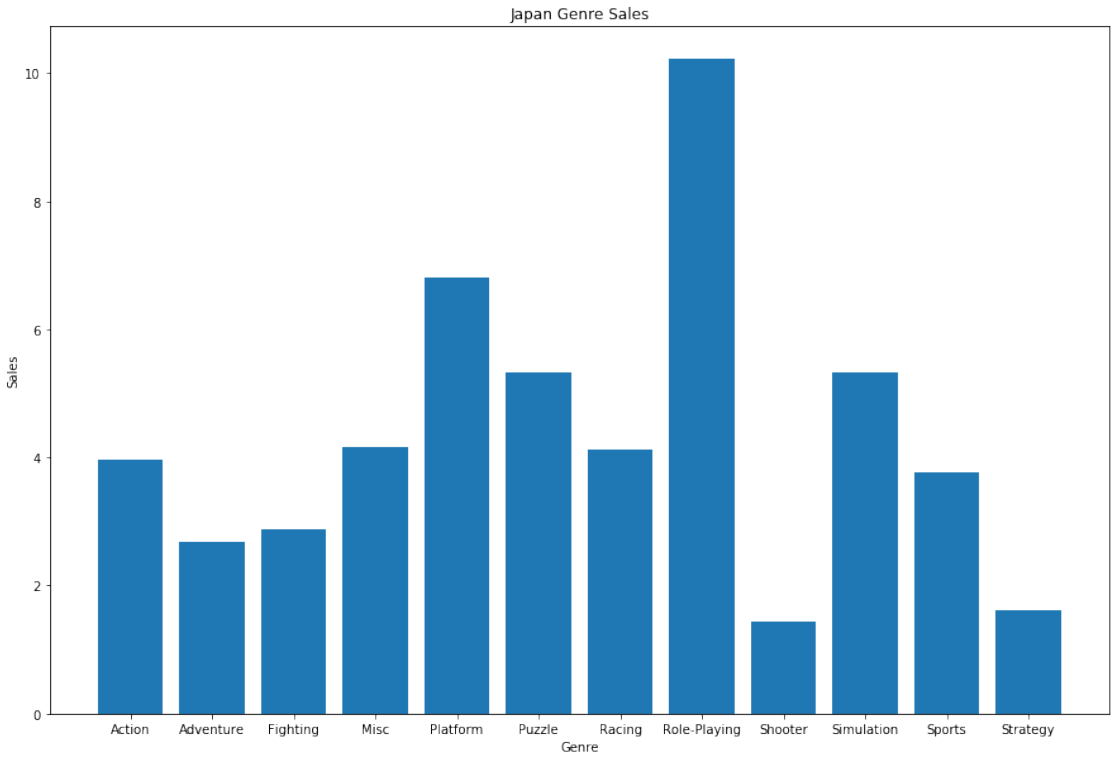
```
In [22]: x = df['Genre']
         y = df['JP']



         plt.figure(figsize= (15,10))

         plt.bar(x , y)
         plt.title('Japan Genre Sales')
         plt.xlabel('Genre')
         plt.ylabel('Sales')

Out[22]: Text(0,0.5,'Sales')
```



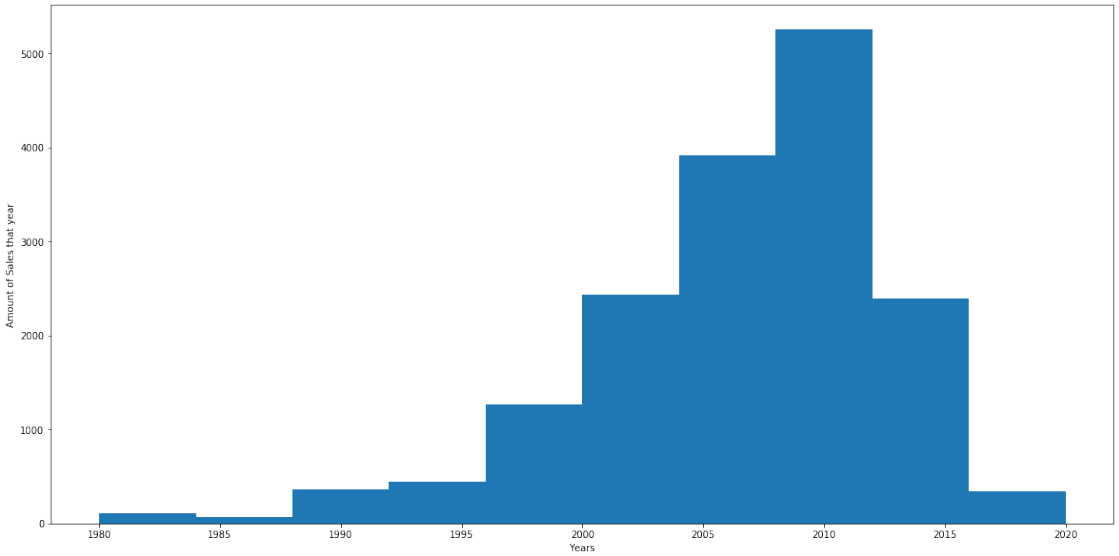
1.2.3 But in Japan, Role-Playing games are more popular with platform coming in second.

1.2.4 Quick Observation to note:

There a lot more higher sales in North America, with Europe in the middle, and Japan having the least sales overall.

1.2.5 Research Question 2: As the years go by and people sell of games increase, is there a increase of game development?

```
In [37]: # Continue to explore the data to address your additional research
         #   questions. Add more headers as needed if you have more questions to
         #   investigate.
         x= df['Year']
         y= df['NA']
         plt.figure(figsize=(20,10))
         plt.xlabel('Years')
         plt.ylabel('Amount of Sales that year')
         plt.hist(x,)
         plt.show()
```



1.3 Result showing a skewed left plot.

Conclusions

Tip: Finally, summarize your findings and the results that have been performed. Make sure that you are clear with regards to the limitations of your exploration. If you haven't done any statistical tests, do not imply any statistical conclusions. And make sure you avoid implying causation from correlation!

Tip: Once you are satisfied with your work here, check over your report to make sure that it is satisfies all the areas of the rubric (found on the project submission page at the end of the lesson). You should also probably remove all of the "Tips" like this one so that the presentation is as polished as possible.

1.4 Submitting your Project

Before you submit your project, you need to create a .html or .pdf version of this notebook in the workspace here. To do that, run the code cell below. If it worked correctly, you should get a return code of 0, and you should see the generated .html file in the workspace directory (click on the orange Jupyter icon in the upper left).

Alternatively, you can download this report as .html via the **File** > **Download as** submenu, and then manually upload it into the workspace directory by clicking on the orange Jupyter icon in the upper left, then using the Upload button.

Once you've done this, you can submit your project by clicking on the "Submit Project" button in the lower right here. This will create and submit a zip file with this .ipynb doc and the .html or .pdf version you created. Congratulations!

1.5 Base on my findings, it seems different genres of games do well in different regions of the world. For example making a role-playing game in Japan, have higher sales than making a shooter. Not to say that you won't do well if you do develop a shooting game there may be chance that you won't sale as many. This is a indication of the general public interest of what type of games they may like. Making sports games in North America and Europe may produce higher sales. There will be many games that will succeed in sales even if that genre of the game is not as popular in that country. I do have to put in consideration of how fast a genre of games are being produce by companies.

1.6 Around 2005 - 2015 seem like the pinnacle of gaming sales.

1.7 No statistical work hasn't been implemented in my findings. There is many variables that could play part that is not in this dataset but this is a start to looking at other datasets.

```
In [ ]: from subprocess import call
        call(['python', '-m', 'nbconvert', 'Investigate_a_Dataset.ipynb'])
```