Video Game Sales Analytics

https://www.kaggle.com/gregorut/videogamesales
(https://www.kaggle.com/gregorut/videogamesales)

Video Game Sales Capstone Project

Our data, mesasured by Gregory Smith, includes all the video games recorded from 1980 to 2016. The categories include platforms, genres, publishers and variety of market sales. We will be analyzing the differences and correspondence of sales along side the years. In addition, we will be inspecting the dataset and cleaning the data by examing whether there are any NAN values. Once we had discovered that there was minimal NAN values, we decided to delete the rows, for the data will not be altered after removal. After we had reformated the dataset, we added some graphs to promote visualization for the audience, and further analyzed the video game sales through the different testing grounds.

Team:

1. Haowen Yin
2. Jacob Yim
3. Nicholas Chen

Data disclaimers:

Our data includes platforms for which games are no longer being released, and excludes data from more recent years, for which we were unable to obtain any data. 2016 is also a partial year, so our attempts to include it as part of any trends may be inaccurate. Most of our data also originates from publicly-available data, which large companies and publicly-traded companies tend to make readily available. Smaller companies may be excluded from this analysis.

Agenda

1. Cleaning the dataset (Remove NaN)
2. Displaying basic graphs for the dataset
3. An analysis for entering the industry with Hypothesis Testing
4. Further market inspection

Importing Libraries

In [1]:
```
#import the required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Importing the File

The file is provided by Kaggle.com. It is about the sales of video games. The data set is only for publically traded companies and games are sold by units. **Lets inspect the head**

Rank - Ranked by Global Sales

Name - name of the game

Platform - system the game is play on

Year - year of the sale

Genre - type of the game

Publisher - the companies that sold the game

NA_Sales - sales in North America

EU_Sales - sales in Europe

JP_Sales - sales in Japan

Other_Sales - sales in any other region

Global_Sales - sum of sales

In [2]:
```
df=pd.read_csv('vgsales.csv')
df1=pd.read_csv('vgsales.csv')
df.head()
```

Out[2]:

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

We will start by examining and cleaning the data so we have the correct things to work with

As we see from the output in the cell below, there are NaN values for **Year** and **Publisher**

In [3]:
```
df.isna().sum()
```

Out[3]:
```
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
```

Let us inspect those NaN values by sorting and looking at the tail

We now see the problem from this dataset.

Question: Do we remove the row from the dataset or not?

Answer: Yes, the sales has minimum effects on the entire dataset and with Year column unknown it is hard to see and graph trends.

In [4]:
```
df.sort_values(by='Year',ascending=False).tail(10)
```

Out[4]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	C
16191	16194	Homeworld Remastered Collection	PC	NaN	Strategy	NaN	0.00	0.01	0.00	
16194	16197	Shorts	DS	NaN	Platform	Unknown	0.01	0.00	0.00	
16198	16201	AKB1/48: Idol to Guam de Koishitara...	X360	NaN	Misc	NaN	0.00	0.00	0.01	
16229	16232	Brothers in Arms: Furious 4	X360	NaN	Shooter	NaN	0.01	0.00	0.00	
16246	16249	Agarest Senki: Re-appearance	PS3	NaN	Role-Playing	Idea Factory	0.00	0.00	0.01	
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 Kitan	PS3	NaN	Adventure	Unknown	0.01	0.00	0.00	
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	

Let us drop the NaN values and inspect it again

Everything looks good

In [5]:
```
df.dropna(inplace=True)
df.sort_values(by='Year',ascending=False).tail(10)
```

Out[5]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other
1556	1558	Atlantis	2600	1981.0	Shooter	Imagic	1.18	0.08	0.0	
544	545	Missile Command	2600	1980.0	Shooter	Atari	2.56	0.17	0.0	
4025	4027	Ice Hockey	2600	1980.0	Sports	Activision	0.46	0.03	0.0	
6896	6898	Checkers	2600	1980.0	Misc	Atari	0.22	0.01	0.0	
5366	5368	Freeway	2600	1980.0	Action	Activision	0.32	0.02	0.0	
2669	2671	Boxing	2600	1980.0	Fighting	Activision	0.72	0.04	0.0	
6317	6319	Bridge	2600	1980.0	Misc	Activision	0.25	0.02	0.0	
1766	1768	Kaboom!	2600	1980.0	Misc	Activision	1.07	0.07	0.0	
1969	1971	Defender	2600	1980.0	Misc	Atari	0.99	0.05	0.0	
258	259	Asteroids	2600	1980.0	Shooter	Atari	4.00	0.26	0.0	

In [6]:
```
df=df.sort_values(by='Year',ascending=False)
df.head()
```

Out[6]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	G
5957	5959	Imagine: Makeup Artist	DS	2020.0	Simulation	Ubisoft	0.27	0.0	0.00	
14390	14393	Phantasy Star Online 2 Episode 4: Deluxe Package	PS4	2017.0	Role-Playing	Sega	0.00	0.0	0.03	
16241	16244	Phantasy Star Online 2 Episode 4: Deluxe Package	PSV	2017.0	Role-Playing	Sega	0.00	0.0	0.01	
16438	16441	Brothers Conflict: Precious Baby	PSV	2017.0	Action	Idea Factory	0.00	0.0	0.01	
16220	16223	Dynasty Warriors: Eiketsuden	PS4	2016.0	Action	Tecmo Koei	0.00	0.0	0.01	

We know the data is from 3 years ago the 2020 year and 2017 data could be a mistake by gathering data.

Since there is only three rows it is save to remove them.

In [7]:
```
df=df[4:]
df.head(10)
```

Out[7]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales
16220	16223	Dynasty Warriors: Eiketsuden	PS4	2016.0	Action	Tecmo Koei	0.00	0.00	0.0
9993	9995	Dead Island Definitive Collection	PS4	2016.0	Action	Deep Silver	0.02	0.07	0.0
5621	5623	TrackMania Turbo	PS4	2016.0	Action	Ubisoft	0.03	0.24	0.0
10731	10733	Rise of the Tomb Raider	PC	2016.0	Adventure	Square Enix	0.00	0.09	0.0
14847	14850	Terraria	WiiU	2016.0	Action	505 Games	0.00	0.02	0.0
13994	13996	Destiny: The Collection	XOne	2016.0	Shooter	Activision	0.01	0.02	0.0
15281	15284	Hyakka Hyakurou: Sengoku Ninpoujou	PSV	2016.0	Adventure	D3Publisher	0.00	0.00	0.0
8221	8223	Jikkyou Powerful Pro Baseball 2016	PSV	2016.0	Sports	Konami Digital Entertainment	0.00	0.00	0.1
9538	9540	Dragon Quest Heroes II: Twin Kings and the Pro...	PS3	2016.0	Action	Square Enix	0.00	0.00	0.1
10733	10735	Persona 5	PS3	2016.0	Role-Playing	Unknown	0.00	0.00	0.1

Now we can inspect the data types

1.) Rank should be category because it for ranking

2.) Year can be category and still enjoy the benefit of sort and not be aggregated by numpy functions.

In [8]:
```
df.dtypes
```

Out[8]:
```
Rank             int64
Name            object
Platform        object
Year           float64
Genre           object
Publisher       object
NA_Sales       float64
EU_Sales       float64
JP_Sales       float64
Other_Sales    float64
Global_Sales   float64
dtype: object
```

So let's change the type for Year and Rank

In [9]:
```
df['Year']=df['Year'].astype('category')
df['Rank']=df['Rank'].astype('category')
df.dtypes
```

Out[9]:
```
Rank           category
Name             object
Platform         object
Year           category
Genre            object
Publisher        object
NA_Sales        float64
EU_Sales        float64
JP_Sales        float64
Other_Sales     float64
Global_Sales    float64
dtype: object
```

Basic Questions about the Game Industry:

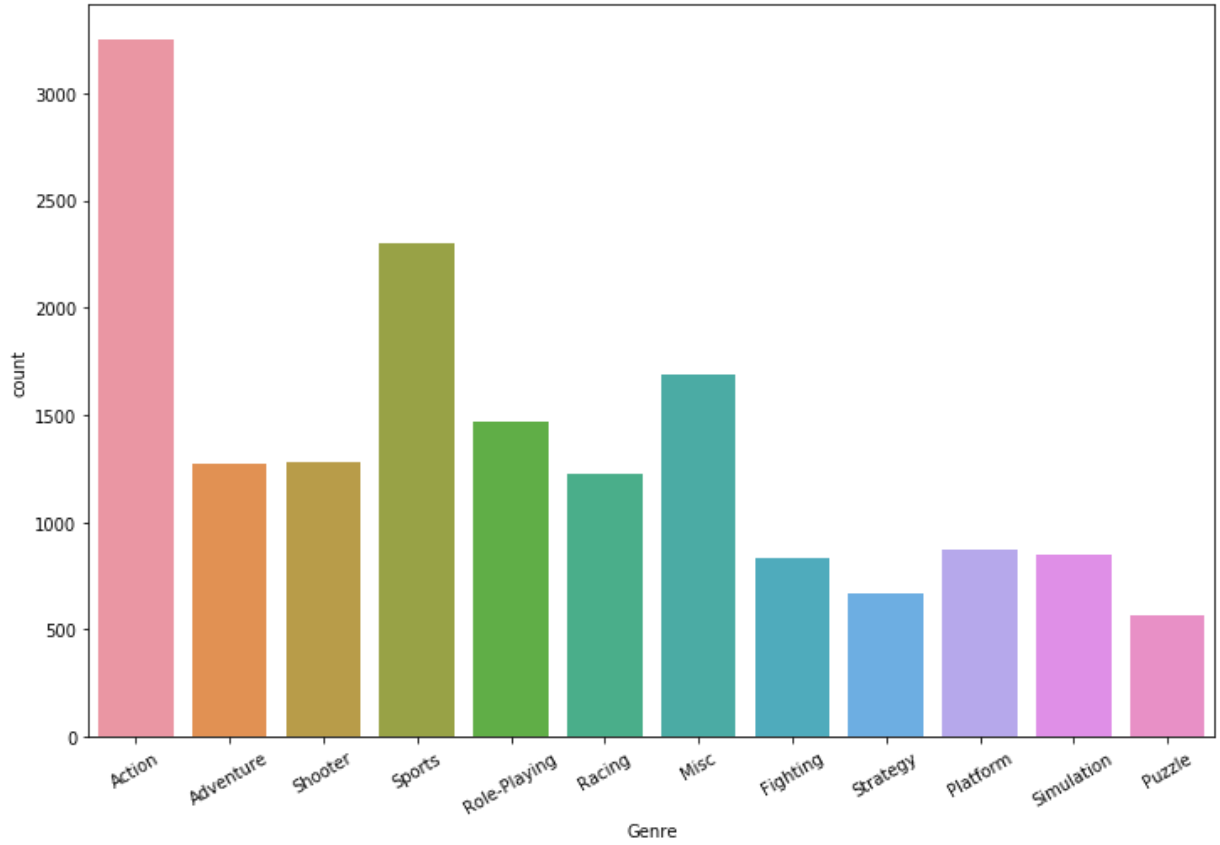
1. What are the most popular genre that developers like to make?
2. Who are the top 5 publishers in sales?
3. What are the top 5 game sales?
4. What platform has the most games?
5. What platform has the most game sales?
6. What year produced the most sale?
7. What genre generated the most sales?
8. What is the sales by year?

Question 1: What are the most popular genre that developers like to make?

Answer: The top 3 genres are: Action, Sports and Misc

In [10]:
```
plt.figure(figsize=(12,8))
plt.xticks(rotation=30)
sns.countplot(df['Genre'])
```

Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x20484cf85c0>



Question 2: Who are the top 5 publishers in sales?

Answer: Nintendo, Electronic Arts, Activision, Sony, and Ubisoft

In [11]:
```
pd.pivot_table(df, index='Publisher', aggfunc=np.sum).sort_values(by='Global_Sal
```

Out[11]:

Publisher	EU_Sales	Global_Sales	JP_Sales	NA_Sales	Other_Sales
Nintendo	418.30	1784.43	454.99	815.75	95.19
Electronic Arts	367.38	1093.39	13.98	584.22	127.63
Activision	213.72	721.41	6.54	426.01	74.79
Sony Computer Entertainment	187.55	607.28	74.10	265.22	80.40
Ubisoft	163.03	473.25	7.33	252.54	50.14

Question 3: What are the top 5 game sales?

Answer: Wii Sports, Grand Theft Auto V, Super Mario Bros., Tetris, Mario Kart Wii

In [12]:
```
top_five_sales = pd.pivot_table(df, index='Name', aggfunc=np.sum).sort_values(by=
top_five_sales
```

Out[12]:

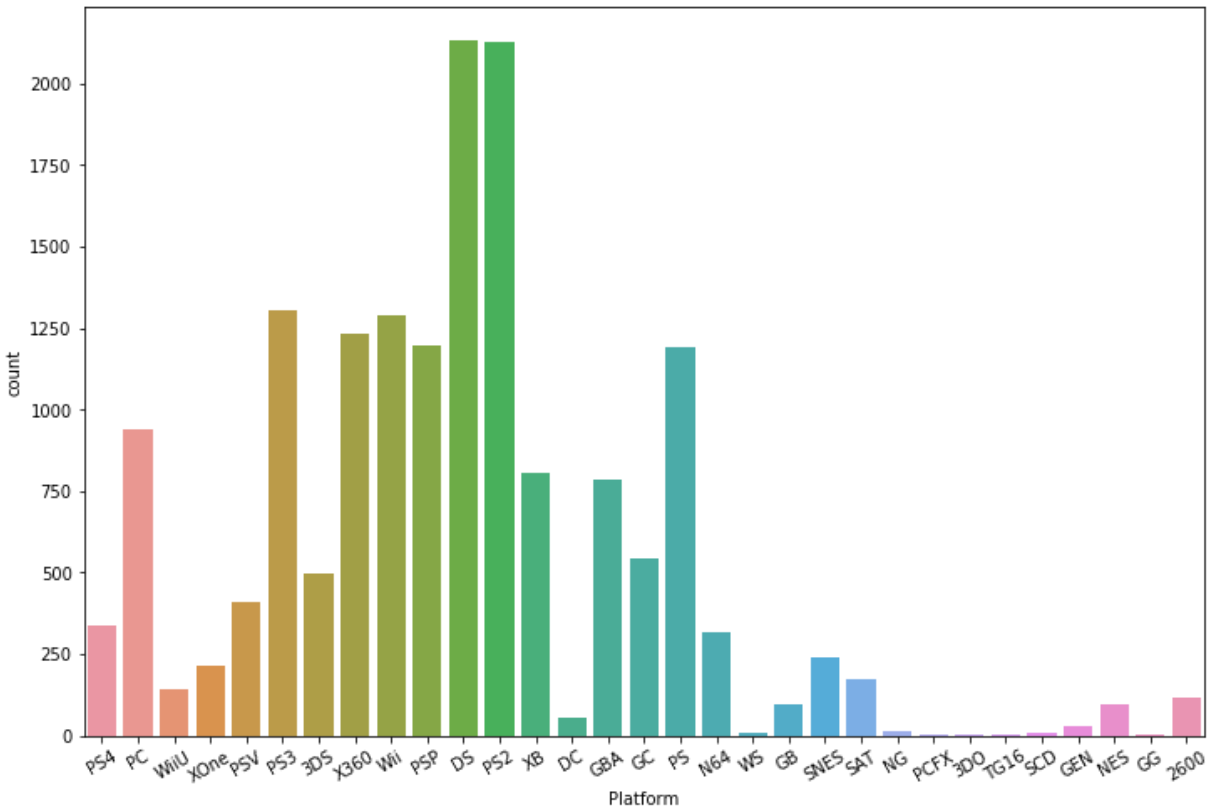
Name	EU_Sales	Global_Sales	JP_Sales	NA_Sales	Other_Sales
Wii Sports	29.02	82.74	3.77	41.49	8.46
Grand Theft Auto V	23.04	55.92	1.39	23.46	8.03
Super Mario Bros.	4.88	45.31	6.96	32.48	0.99
Tetris	2.95	35.84	6.03	26.17	0.69
Mario Kart Wii	12.88	35.82	3.79	15.85	3.31

Question 4: What platform has the most games?

Answer: DS has most games and PS2 is a close second

In [13]:
```
plt.figure(figsize=(12,8))
plt.xticks(rotation=30)
sns.countplot(df['Platform'])
```

Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x20485244a20>



Question 5: What platform has the most game sales?

Answer: PS2

In [14]:
```
pd.pivot_table(df, index='Platform', aggfunc=np.sum).sort_values(by='Global_Sale
```

Out[14]:

Platform	EU_Sales	Global_Sales	JP_Sales	NA_Sales	Other_Sales
PS2	332.63	1233.46	137.54	572.92	190.47
X360	278.00	969.60	12.30	594.33	84.67
PS3	340.47	949.35	79.21	388.90	140.81
Wii	264.35	909.81	68.28	497.37	79.20
DS	194.05	818.62	175.02	388.26	60.27
PS	212.38	727.39	139.78	334.71	40.69
GBA	72.49	305.62	46.56	178.43	7.51
PSP	67.16	291.71	75.89	107.09	41.52
PS4	123.70	278.07	14.27	96.80	43.36
PC	137.35	254.70	0.17	92.04	24.33

Question 6: What year produced the most sale?

Answer: 2008

In [15]:
```
pd.pivot_table(df, index='Year', aggfunc=np.sum).sort_values(by='Global_Sales', a
```

Out[15]:

Year	EU_Sales	Global_Sales	JP_Sales	NA_Sales	Other_Sales
2008.0	184.40	678.90	60.26	351.44	82.39
2009.0	191.59	667.30	61.89	338.85	74.77
2007.0	160.18	609.92	60.29	311.18	77.58
2010.0	176.57	600.29	59.49	304.24	59.90
2006.0	129.24	521.04	73.73	263.12	54.43

Question 7: What genre generated the most sales?

Answer: Action

In [16]:
```
pd.pivot_table(df, index='Genre', aggfunc=np.sum).sort_values(by='Global_Sales',
```

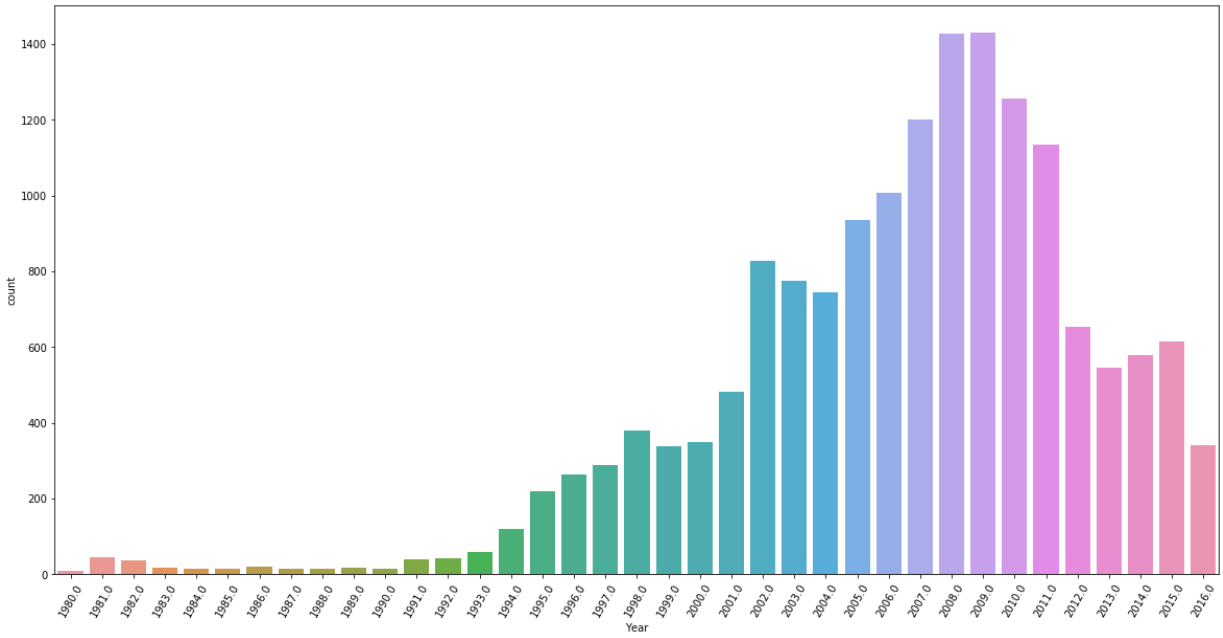
Out[16]:

Genre	EU_Sales	Global_Sales	JP_Sales	NA_Sales	Other_Sales
Action	516.48	1722.83	158.64	861.77	184.92
Sports	371.34	1309.24	134.76	670.09	132.65
Shooter	310.45	1026.20	38.18	575.16	101.90
Role-Playing	187.57	923.79	350.25	326.50	59.38
Platform	200.65	829.13	130.65	445.99	51.51

Question 8: What is the sales by year?

In [17]:
```
plt.figure(figsize=(20,10))
plt.xticks(rotation='60')
sns.countplot(x=df['Year'])
```

Out[17]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x204850fc5f8>
```



Deep Dive Section

In [18]:
```
df=df.sort_values(by='Rank')
df.head(10)
```

Out[18]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77	
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79	
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3.28	
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	
5	6	Tetris	GB	1989.0	Puzzle	Nintendo	23.20	2.26	4.22	
6	7	New Super Mario Bros.	DS	2006.0	Platform	Nintendo	11.38	9.23	6.50	
7	8	Wii Play	Wii	2006.0	Misc	Nintendo	14.03	9.20	2.93	
8	9	New Super Mario Bros. Wii	Wii	2009.0	Platform	Nintendo	14.59	7.06	4.70	
9	10	Duck Hunt	NES	1984.0	Shooter	Nintendo	26.93	0.63	0.28	

Statement: A CEO walks in and tell us he wants to enter the video game industry. After he look at the data set, across the world he believe he will be making average of 0.541 million dollars if he enters the industry. He also told us he initially will start in North America.

Null Hypothesis

H0: μ= 0.541

Alternative Hypothesis

Ha: μ!=0.541

In [19]:
```
np.mean(df['Global_Sales'])
```

Out[19]:
```
0.5410222877141792
```

Let us use the z test because our data is well above 30 samples.

In [20]:
```
from statsmodels.stats.weightstats import ztest
import scipy.stats as stats
```

In [21]:
```
(test_statistic, p_value) = ztest(df['Global_Sales'], value=0.541, alternative='
```

In [22]:
```
print("The test statistic is: ", round(test_statistic,5))
print("The p-value is: ", round(p_value,5))
if p_value<0.05:
    print("Reject null hypothesis")
else:
    print("Accept null hypothesis")
```

```
The test statistic is:  0.00181
The p-value is:  0.99855
Accept null hypothesis
```

Conclusion: The since the p-value is 0.933 we accept the H0 according to the data he is statistically possible to make 0.541 million if he enters the industry

---

Statement:The CEO is starting his company in North America he will be able to make at least 0.541 million.

H0: μ= 0.541

Ha: μ!= 0.541

Use the same z test except changing it to NA_Sales

In [23]:
```
(test_statistic, p_value) = ztest(df['NA_Sales'], value=0.541, alternative='two-
print("The test statistic is: ", round(test_statistic,5))
print("The p-value is: ", round(p_value,5))
if p_value<0.05:
    print("Reject null hypothesis")
else:
    print("Accept null hypothesis")
```

```
The test statistic is:  -42.71544
The p-value is:  0.0
Reject null hypothesis
```

Conclusion: Null hypothesis rejected, CEO will not be able to make 0.541 million if he only operates in North America.

CEO says okay and accept that he might make 0.265 million as he decided to start in North America

In [24]:
```
np.mean(df['NA_Sales'])
```

Out[24]:
```
0.26569533984161153
```

Statement: The CEO says since he is okay making mid-range sales, the genre of game should not matter he can make any type of game and still make about NA's sales mean.

H0: μ_Sports = μ_Platform = μ_Racing...

Ha: μ_Sports != μ_Platform != μ_Racing... Note: at least one group does not have the same mean

In [25]:
```
print(pd.unique(df['Genre'].values))
```

```
['Sports' 'Platform' 'Racing' 'Role-Playing' 'Puzzle' 'Misc' 'Shooter'
 'Simulation' 'Action' 'Fighting' 'Adventure' 'Strategy']
```

Let us use ANOVA(F test) because there are more than 3 categorical data

In [26]:
```
df[['Genre','NA_Sales']].head()
```

Out[26]:

	Genre	NA_Sales
0	Sports	41.49
1	Platform	29.08
2	Racing	15.85
3	Sports	15.75
4	Role-Playing	11.27

In [27]:
```
#find the unique group values assign it to grps
grps=pd.unique(df['Genre'].values)
#display them
print(grps)
```

```
['Sports' 'Platform' 'Racing' 'Role-Playing' 'Puzzle' 'Misc' 'Shooter'
 'Simulation' 'Action' 'Fighting' 'Adventure' 'Strategy']
```

In [28]:
```
#group weight by group
df_data= {grp:df['NA_Sales'][df.Genre == grp] for grp in grps}
```

In [29]:
```
#find the statistic F and P value calling the stats.f_oneway method from scipy
F,p= stats.f_oneway(df_data['Action'],df_data['Sports'],df_data['Shooter'],df_da
```

In [30]:
```
#print the p-value
print("p-value for significance is: ", p)
#check whether to reject or accept the null hypothesis
if p < 0.05:
    print("Reject null hypothesis: at least one group does not have the same mea
else:
    print("Accept null hypothesis: all the groups have the same mean")
```

```
p-value for significance is:  1.8246845336371894e-30
Reject null hypothesis: at least one group does not have the same mean
```

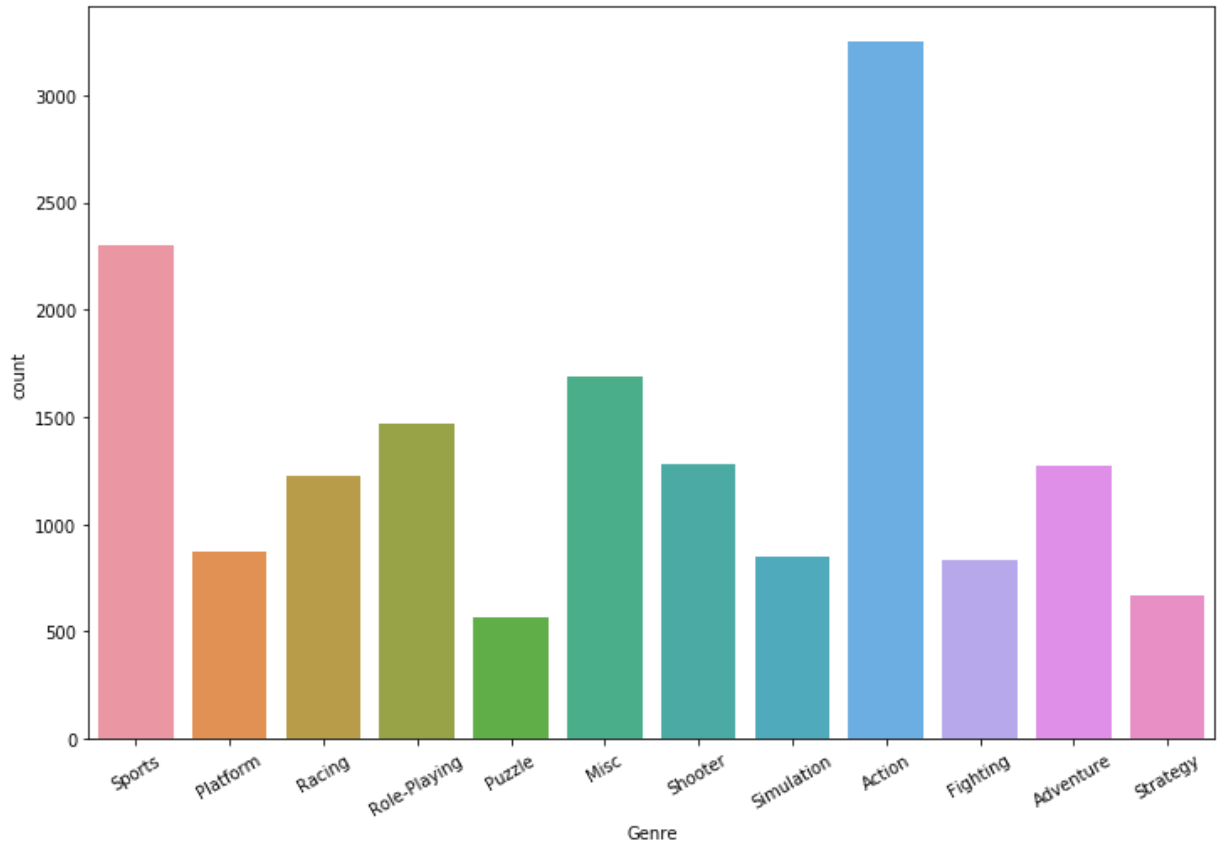
Conclusion: The p-value shows that there are significant difference between the different genre. The CEO can't not pick any genre and make the sales' mean

---

The CEO understands and looks at the genre count plot

In [31]:
```
plt.figure(figsize=(12,8))
plt.xticks(rotation=30)
sns.countplot(df['Genre'])
```

Out[31]: <matplotlib.axes._subplots.AxesSubplot at 0x20485446da0>



Statement: CEO says okay, people are producing massive amount of Action and Sports game. I should be able to make sales mean if I produce either Action or Sports

H0: μ_Action = μ_Sports

Ha: μ_Action != μ_Sports

We can use the ANOVA test again

In [32]:
```
F,p= stats.f_oneway(df_data['Action'],df_data['Sports'])
#print the p-value
print("p-value for significance is: ", p)
#check whether to reject or accept the null hypothesis
if p < 0.05:
    print("Reject null hypothesis: at least one group does not have the same mean
else:
    print("Accept null hypothesis: all the groups have the same mean")
```

```
p-value for significance is:  0.24044484879138
Accept null hypothesis: all the groups have the same mean
```

Conclusion: We accept the Null Hypothesis. Thus, the CEO is correct; he can produce either Action game or Sports game and make the same sales mean for the two genres.

The CEO knows what type of game he need to produce and now he is looking at the platform for those games

In [33]:
```
df.groupby('Platform').sum().sort_values(by='Global_Sales',ascending=False).head
```

Out[33]:

Platform	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
PS2	572.92	332.63	137.54	190.47	1233.46
X360	594.33	278.00	12.30	84.67	969.60
PS3	388.90	340.47	79.21	140.81	949.35
Wii	497.37	264.35	68.28	79.20	909.81
DS	388.26	194.05	175.02	60.27	818.62
PS	334.71	212.38	139.78	40.69	727.39
GBA	178.43	72.49	46.56	7.51	305.62
PSP	107.09	67.16	75.89	41.52	291.71
PS4	96.80	123.70	14.27	43.36	278.07
PC	92.04	137.35	0.17	24.33	254.70

He learned his lesson so now he will be picking from the top 3 platforms. He want to scale his sales to global sales so he want to know which platform correlate with global sales the best.

Statement: He believes PS2 would be the platform because it's the top selling platform

H0: μ_PS2 = μ_Global_Sales

Ha: μ_PS2 != μ_Global_Sales

Let us use Z test to see if it's significant towards global sales

In [34]:
```
(test_statistic, p_value) = ztest(df[df['Platform']=='PS2']['Global_Sales'], valu
print("The test statistic is: ", round(test_statistic,5))
print("The p-value is: ", round(p_value,5))
if p_value<0.05:
    print("Reject null hypothesis")
else:
    print("Accept null hypothesis")
```

```
The test statistic is:  1.57506
The p-value is:  0.11524
Accept null hypothesis
```

Let's also check X360

H0: μ_X360 = μ_Global_Sales

Ha: μ_X360 != μ_Global_Sales

In [35]:
```
(test_statistic, p_value) = ztest(df[df['Platform']=='X360']['Global_Sales'], val
print("The test statistic is: ", round(test_statistic,5))
print("The p-value is: ", round(p_value,5))
if p_value<0.05:
    print("Reject null hypothesis")
else:
    print("Accept null hypothesis")
```

```
The test statistic is:  5.25501
The p-value is:  0.0
Reject null hypothesis
```

Conclusion: Accept Null Hypothesis. The CEO is right. PS2 sales mean is same as the global sales. Thus, it scales better than other platforms like X360.

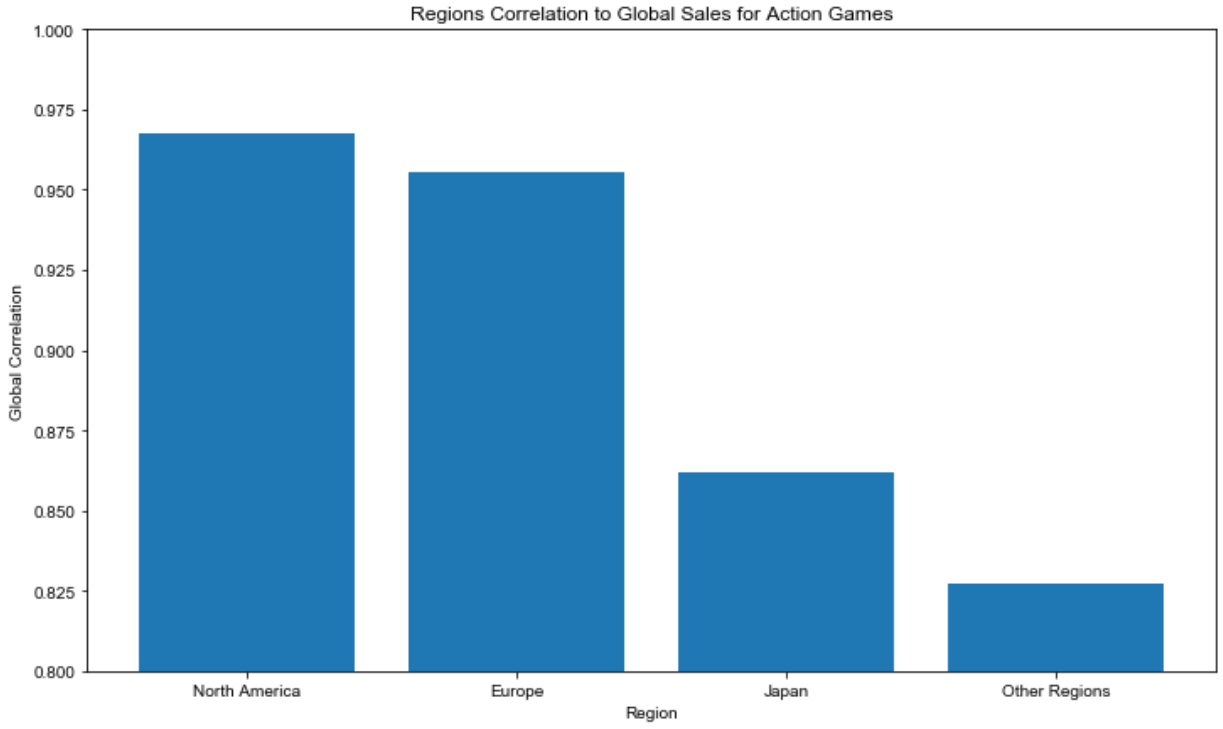
The CEO now knows the type of game, the platform and now he is looking to see which region he should prioritize marketing.

In [36]:
```
df_action=df.loc[(df['Genre']=='Action')&(df['Publisher']=='Nintendo')].sort_valu
#Pearson's correlation Matrix all regions in relation to Global Sales.
corr_matrix_NA=np.corrcoef(df_action['Global_Sales'],df_action['NA_Sales'])[0,1]
corr_matrix_EU=np.corrcoef(df_action['Global_Sales'],df_action['EU_Sales'])[0,1]
corr_matrix_JP=np.corrcoef(df_action['Global_Sales'],df_action['JP_Sales'])[0,1]
corr_matrix_Other=np.corrcoef(df_action['Global_Sales'],df_action['Other_Sales']
```

In [37]:
```
#Outputing the all correlation matrix

corr_mat_dict={
    'North America':corr_matrix_NA,
    'Europe':corr_matrix_EU,
    'Japan':corr_matrix_JP,
    'Other Regions':corr_matrix_Other
}
```

In [38]:
```
#Plotting the x and y values into bar chart
plt.figure(figsize=(12,7))

plt.bar(range(len(corr_mat_dict)), list(corr_mat_dict.values()), align='center')
plt.xticks(range(len(corr_mat_dict)), list(corr_mat_dict.keys()))


#Labeling
plt.ylim(0.8,1.0)
plt.title('Regions Correlation to Global Sales for Action Games')
plt.ylabel('Global Correlation')
plt.xlabel('Region')

sns.set()

plt.show
corr_mat_dict
```

Out[38]:
```
{'North America': 0.967369180975589,
 'Europe': 0.9556990452421376,
 'Japan': 0.8618070420831633,
 'Other Regions': 0.8272027624651398}
```



Conclusion: Since North America has the highest correlation (0.967), the CEO should prioritize marketing in North America. The second highest is Europe so that is the second priority.

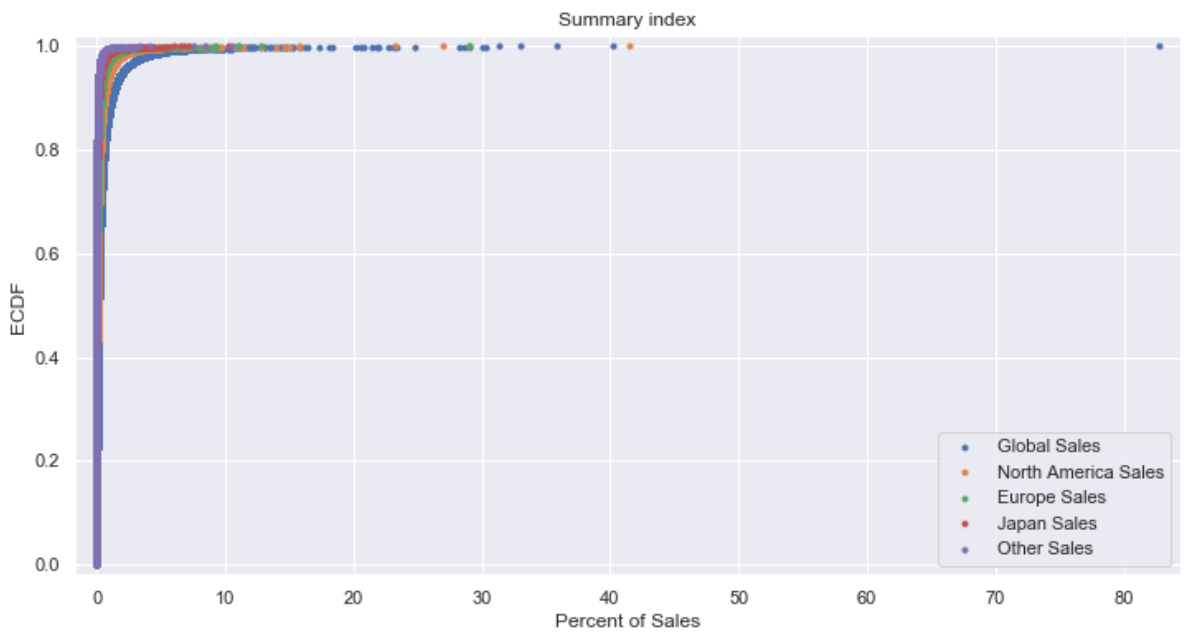
The CEO is prepared to enter the game industry on the basic level

Let's inspect the market more

ECDF of the all the regions including Global Sales

In [39]:
```
#define ECDF function: ecdf
def ecdf(data):
    n=len(data)
    x=np.sort(data)
    y=np.arange(1,n+1)/n

    #return x and y
    return x,y


#Create Variables for X and Y for new data
x_GS,y_GS=ecdf(df['Global_Sales'])
x_NA,y_NA=ecdf(df['NA_Sales'])
x_EU,y_EU=ecdf(df['EU_Sales'])
x_JP,y_JP=ecdf(df['JP_Sales'])
x_Other,y_Other=ecdf(df['Other_Sales'])


plt.figure(figsize=(12,6))

#plot each variable
plt.plot(x_GS,y_GS,marker='.',ls='none')
plt.plot(x_NA,y_NA,marker='.',ls='none')
plt.plot(x_EU,y_EU,marker='.',ls='none')
plt.plot(x_JP,y_JP,marker='.',ls='none')
plt.plot(x_Other,y_Other,marker='.',ls='none')

#set margins for 2%
plt.margins(.02)

#create legend
plt.legend(['Global Sales','North America Sales','Europe Sales','Japan Sales','O

#labeling
plt.xlabel('Percent of Sales')
plt.ylabel('ECDF')
plt.title('Summary index')
```

Out[39]:
```
Text(0.5, 1.0, 'Summary index')
```



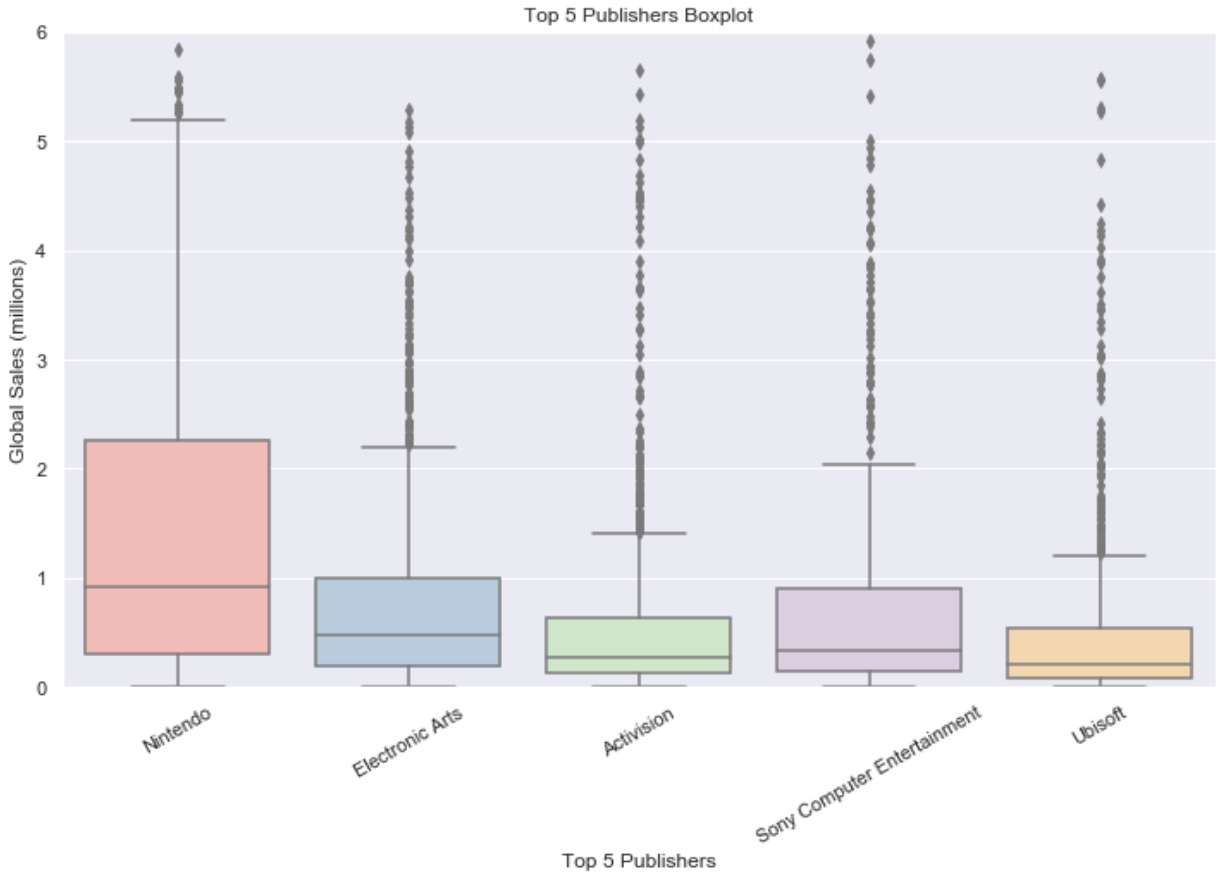
Top 5 Publishers Box Plot

In [40]:
```
#Creating dataframes for the top 5 publishers
Genre_Nintendo=df[df['Publisher']=='Nintendo']
Genre_EA=df[df['Publisher']=='Electronic Arts']
Genre_Activision=df[df['Publisher']=='Activision']
Genre_Sony=df[df['Publisher']=='Sony Computer Entertainment']
Genre_Ubisoft=df[df['Publisher']=='Ubisoft']
```

In [41]:
```
#concated to top five publishers into one dataframe
top5=[Genre_Nintendo,Genre_EA,Genre_Activision,Genre_Sony,Genre_Ubisoft]
top5_result=pd.concat(top5)
```

In [42]:
```
#Create figure size
plt.figure(figsize=(12,7))

#plot data into boxplot
sns.boxplot(x=top5_result['Publisher'],y=top5_result['Global_Sales'],palette='Pas

#label graph
plt.xticks(rotation=30)
plt.ylim(0,6)
plt.title('Top 5 Publishers Boxplot')
plt.xlabel('Top 5 Publishers')
plt.ylabel('Global Sales (millions)')
```

Out[42]:
```
Text(0, 0.5, 'Global Sales (millions)')
```



The box plot above represents the top 5 publishers in regards to the top global sales. We can see that games published by the top five do not break one million; the only exception is Nintendo, which dominate the video game industry from 1980 - 2016. In addition, the boxplots all begin around 0; it is impossible to make negative in sales.

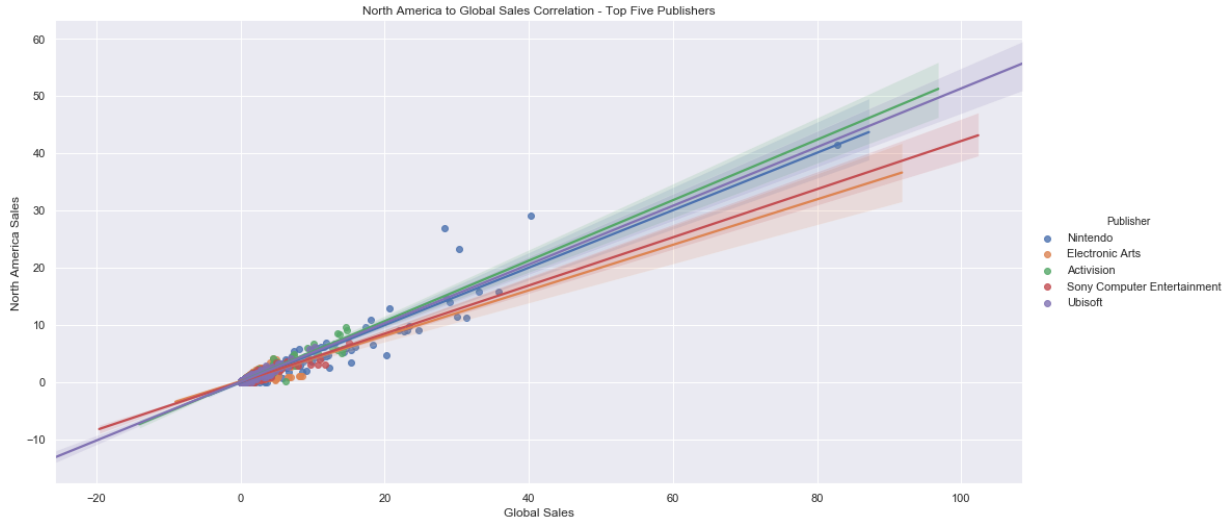
Percentiles of the top 5 Publishers.

In [43]:
```
percentiles=[25,50,75]
percent_nintendo=np.percentile(Genre_Nintendo['Global_Sales'],percentiles)
percent_ea=np.percentile(Genre_EA['Global_Sales'],percentiles)
percent_activision=np.percentile(Genre_Activision['Global_Sales'],percentiles)
percent_sony=np.percentile(Genre_Sony['Global_Sales'],percentiles)
percent_ubisoft=np.percentile(Genre_Ubisoft['Global_Sales'],percentiles)

print('Percentiles of the top 5 Publishers.')
print('\n')
print('Nintendo Percentiles:.....................',percent_nintendo)
print('Electronic Arts Percentiles:.............',percent_ea)
print('Activision Percentiles:...................',percent_activision)
print('Sony Computer Entertainment Percentiles:..',percent_sony)
print('Ubisoft Percentiles:......................',percent_ubisoft)
```

```
Percentiles of the top 5 Publishers.


Nintendo Percentiles:..................... [0.3    0.92   2.2625]
Electronic Arts Percentiles:.............. [0.19 0.47 1.  ]
Activision Percentiles:................... [0.13 0.28 0.64]
Sony Computer Entertainment Percentiles:.. [0.14 0.34 0.9 ]
Ubisoft Percentiles:...................... [0.09 0.21 0.54]
```

North America to Global Sales Correlation - Top 5 Publishers

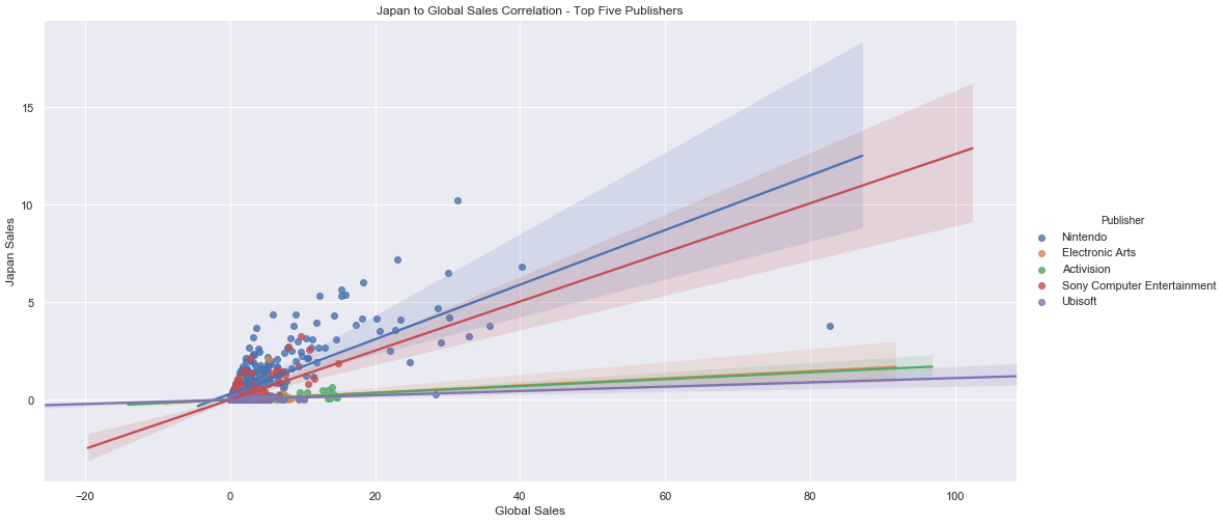
In [44]:
```
#concat all the dataframe to isolate the top 5 publishers
top5_Publisher=[Genre_Nintendo,Genre_EA,Genre_Activision,Genre_Sony,Genre_Ubisof
top5_Publisher_result=pd.concat(top5_Publisher)

#Create figure size
sns.lmplot(x='Global_Sales',y='NA_Sales',data=top5_Publisher_result,hue='Publish
plt.xlabel('Global Sales')
plt.ylabel('North America Sales')
plt.title('North America to Global Sales Correlation - Top Five Publishers')

plt.show()
```



Even though Nintendo has the highest global sales, we can see the Activision has the highest corrleation to global sales. This shows that not one producer can dominate one region, even if they dominate the global market sales.

Japan to Global Sales Correlation - Top 5 Publishers

In [45]:
```
#concat all the dataframe to isolate the top 5 publishers
top5_Publisher=[Genre_Nintendo,Genre_EA,Genre_Activision,Genre_Sony,Genre_Ubisof
top5_Publisher_result=pd.concat(top5_Publisher)

#Create figure size
sns.lmplot(x='Global_Sales',y='JP_Sales',data=top5_Publisher_result,hue='Publishe
plt.xlabel('Global Sales')
plt.ylabel('Japan Sales')
plt.title('Japan to Global Sales Correlation - Top Five Publishers')

plt.show()
```

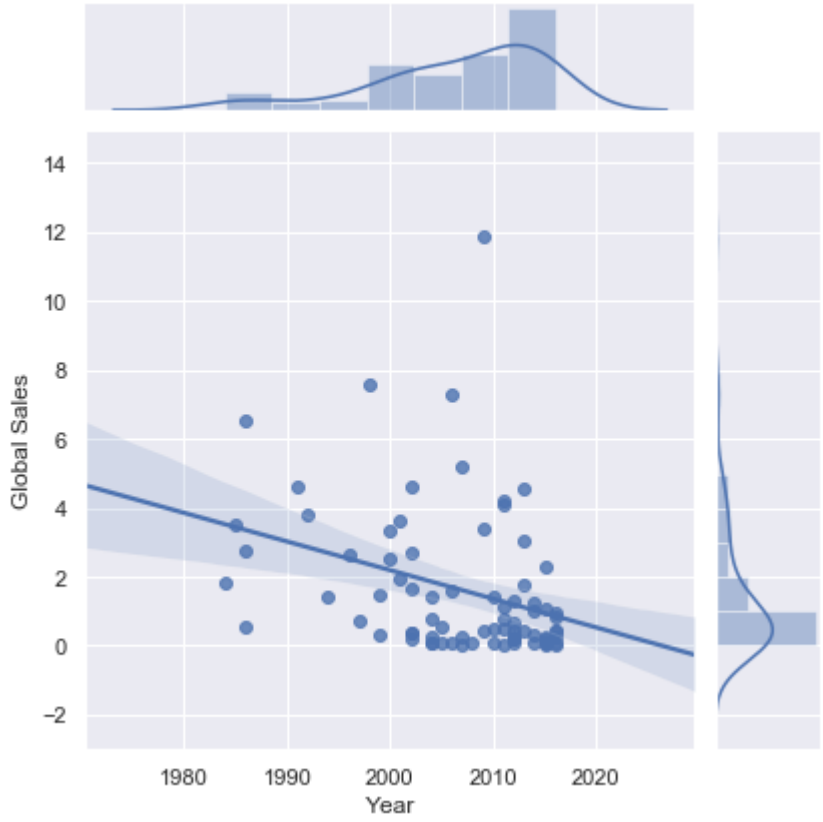


As you can see, Nintendo and Sony's Japanese sales are the most strongly correlated to their global sales compared to other similarly-performing publishers.

Year vs Global Sales Trend

In [46]:
```
#plotting the year vs global sales trend
sns.jointplot(x='Year',y='Global_Sales',data=df_action,kind='reg')

#label
plt.ylabel('Global Sales')
plt.show
```

Out[46]:
```
<function matplotlib.pyplot.show(*args, **kw)>
```

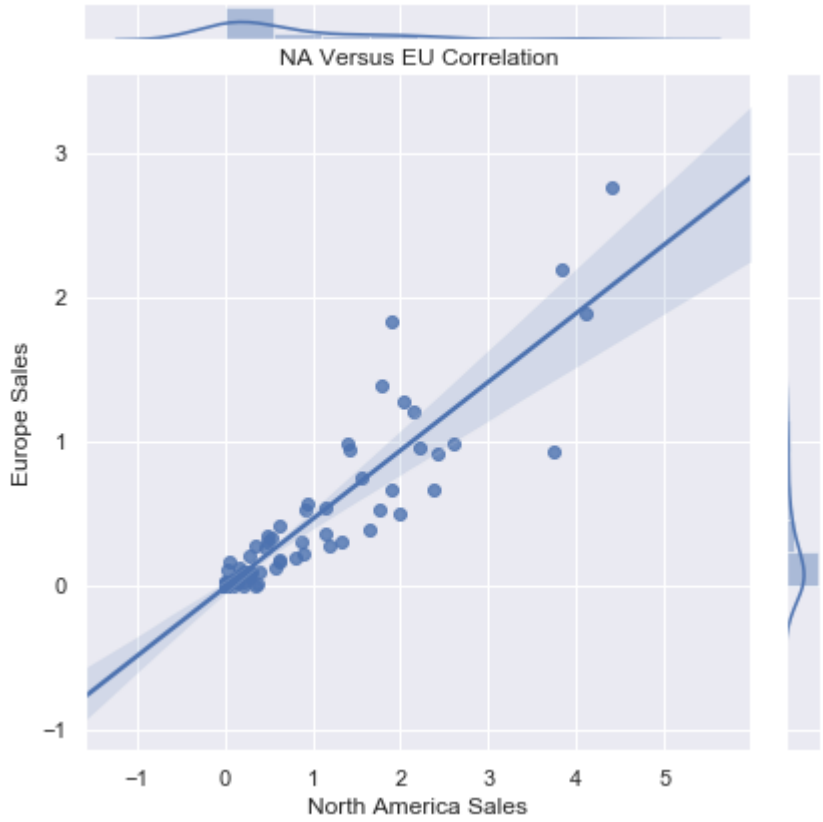


We originally assumed that inflation was going to become a factor within the sale of games and the year that it was produced. However, upon further inspection of the regression line, we can see that inflation did not play a role with the correlation of the year that the game was produced and global revenue it generated. In addition, our hypothesis shows that there was a very weak correlation and no trend with action games being produced alongside the year. We orginally thought that there

may be some pattern such that action games were produced with the high amount of sales each game was generating. However, in light of the scatterplot, we can see that there was weak correlation.

North America vs Europe Correlation

In [47]:
```
#plot NA and EU Sales to find the regression line that represents the high correl
sns.jointplot(x='NA_Sales',y='EU_Sales',data=df_action,kind='reg',ratio=10,space=

#Label
plt.xlabel('North America Sales')
plt.ylabel('Europe Sales')
plt.title('NA Versus EU Correlation')
plt.show()
```



North America and Europe have high correlation to each other. This can be due to many things such as the influence of the western audience.

Genre and Sales Correlations

Null:ρ ≈ 0

Sales of a particular genre and the number of titles released for that genre in following years are very weakly or not at all correlated.

Alternate: ρ ≠ 0

Sales of a particular genre and the number of titles released for that genre in following years have at least some nonzero correlation.

In [48]:
```
#examining the relationship between number of titles released for a particular ge
#as a function of time. each data point by year.

action = df.loc[df['Genre'] == 'Action']

titles_count = action.groupby('Year').describe()['Global_Sales']['count']
global_sales_by_year = action.groupby('Year').sum()['Global_Sales']

print(np.corrcoef(global_sales_by_year, titles_count)[0,1])
```

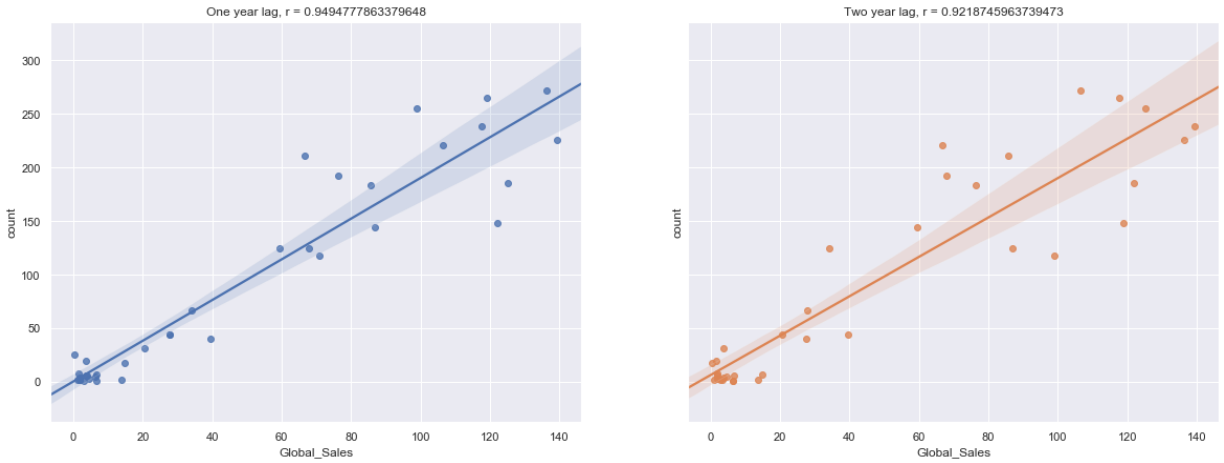
```
0.9320617164266952
```

Time lag

Correlation between number of titles and sales tracks fairly well, i.e.,it appears that publishers try to release more titles in a genre if it does well in sales, but this data is synchronized by year. It doesn't make sense that a publisher can decide which games to release in the same year that sales figures are released, so let's explore a time lag effect whereby one year's sales affect the number of titles released one/two/three years later.

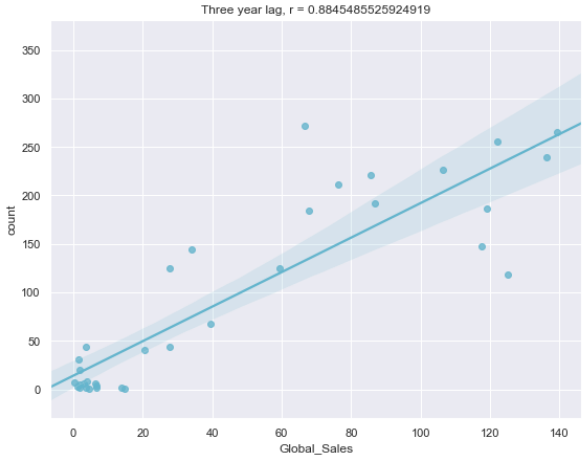
In [49]:
```
#we take values off from the beginning of the titles series and at the end of the
#correlate the sales of the year 2008 with the number of titles in 2006, thus try
fig, (ax1, ax2) = plt.subplots(ncols=2, sharey=True, figsize=(20, 7))
sns.regplot(x=global_sales_by_year[:-1],  y=titles_count[1:], ax=ax1)
ax1.set_title(f'One year lag, r = {np.corrcoef(global_sales_by_year[:-1], titles_
sns.regplot(x=global_sales_by_year[:-2],  y=titles_count[2:], ax=ax2)
ax2.set_title(f'Two year lag, r = {np.corrcoef(global_sales_by_year[:-2], titles_
```

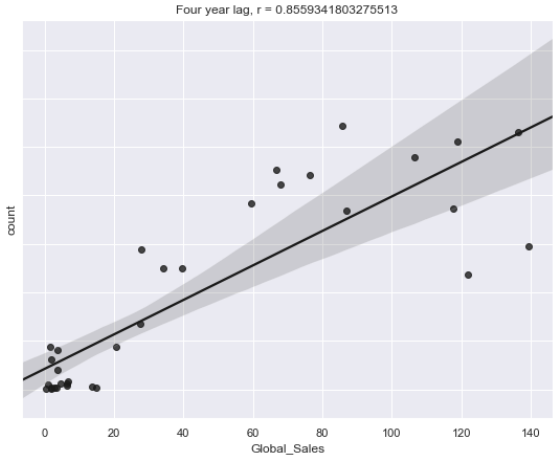
Out[49]:
```
Text(0.5, 1.0, 'Two year lag, r = 0.9218745963739473')
```



In [50]:
```
fig, (ax1, ax2) = plt.subplots(ncols=2, sharey=True, figsize=(20, 7))
sns.regplot(x=global_sales_by_year[:-3],  y=titles_count[3:], ax=ax1, color='c')
ax1.set_title(f'Three year lag, r = {np.corrcoef(global_sales_by_year[:-3], titl
sns.regplot(x=global_sales_by_year[:-4],  y=titles_count[4:], ax=ax2, color='k')
ax2.set_title(f'Four year lag, r = {np.corrcoef(global_sales_by_year[:-4], titles
```

Out[50]:
```
Text(0.5, 1.0, 'Four year lag, r = 0.8559341803275513')
```





In [51]:
```
#let's explore the effect in other genres

sports = df.loc[df['Genre'] == 'Sports']

titles_count = sports.groupby('Year').describe()['Global_Sales']['count']
global_sales_by_year = action.groupby('Year').sum()['Global_Sales']

print(np.corrcoef(global_sales_by_year[:-1], titles_count[1:])[0,1])
print(np.corrcoef(global_sales_by_year[:-2], titles_count[2:])[0,1])
print(np.corrcoef(global_sales_by_year[:-3], titles_count[3:])[0,1])
print(np.corrcoef(global_sales_by_year[:-4], titles_count[4:])[0,1])
```

```
0.7316196307338689
0.6080695034095316
0.5003197454101629
0.4353885802733632
```

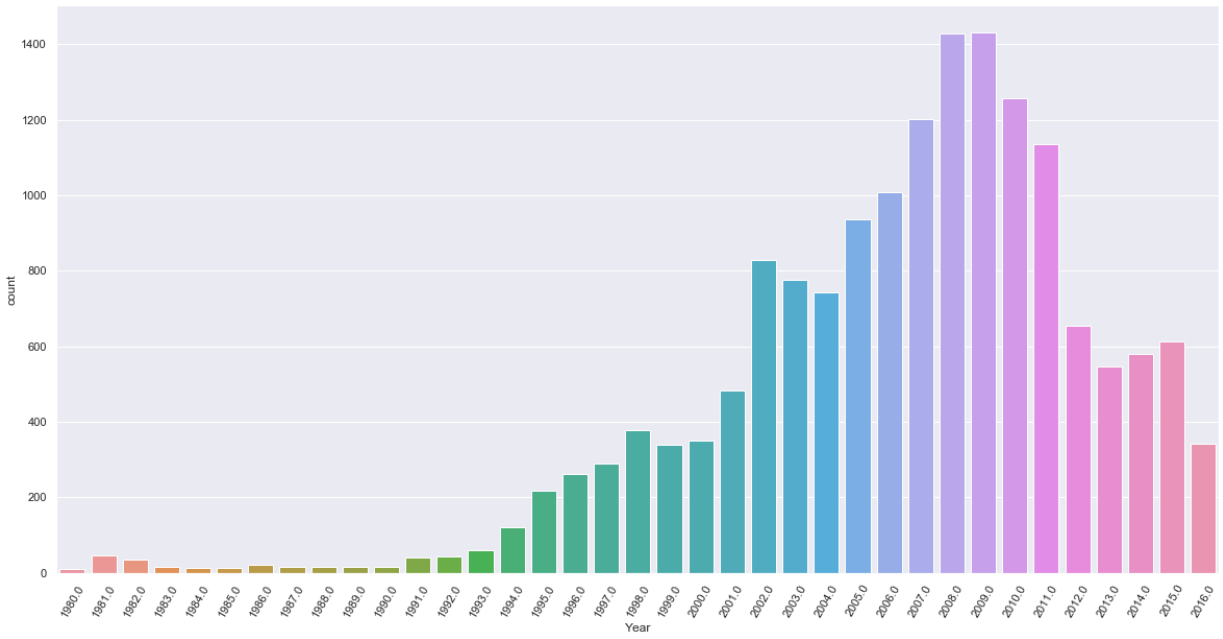
Our data generally tracks less accurately over time, but even four years out, a moderately-strong correlation can be demonstrated, which should properly encompass the primary development period for most triple A games.

In [52]:
```
shooter = df.loc[df['Genre'] == 'Shooter']

titles_count = shooter.groupby('Year').describe()['Global_Sales']['count']
global_sales_by_year = shooter.groupby('Year').sum()['Global_Sales']

print(np.corrcoef(global_sales_by_year[:-1], titles_count[1:])[0,1])
print(np.corrcoef(global_sales_by_year[:-2], titles_count[2:])[0,1])
print(np.corrcoef(global_sales_by_year[:-3], titles_count[3:])[0,1])
print(np.corrcoef(global_sales_by_year[:-4], titles_count[4:])[0,1])
```

```
0.7185857521546588
0.6443454237760932
0.5650149133145304
0.4617813470479386
```

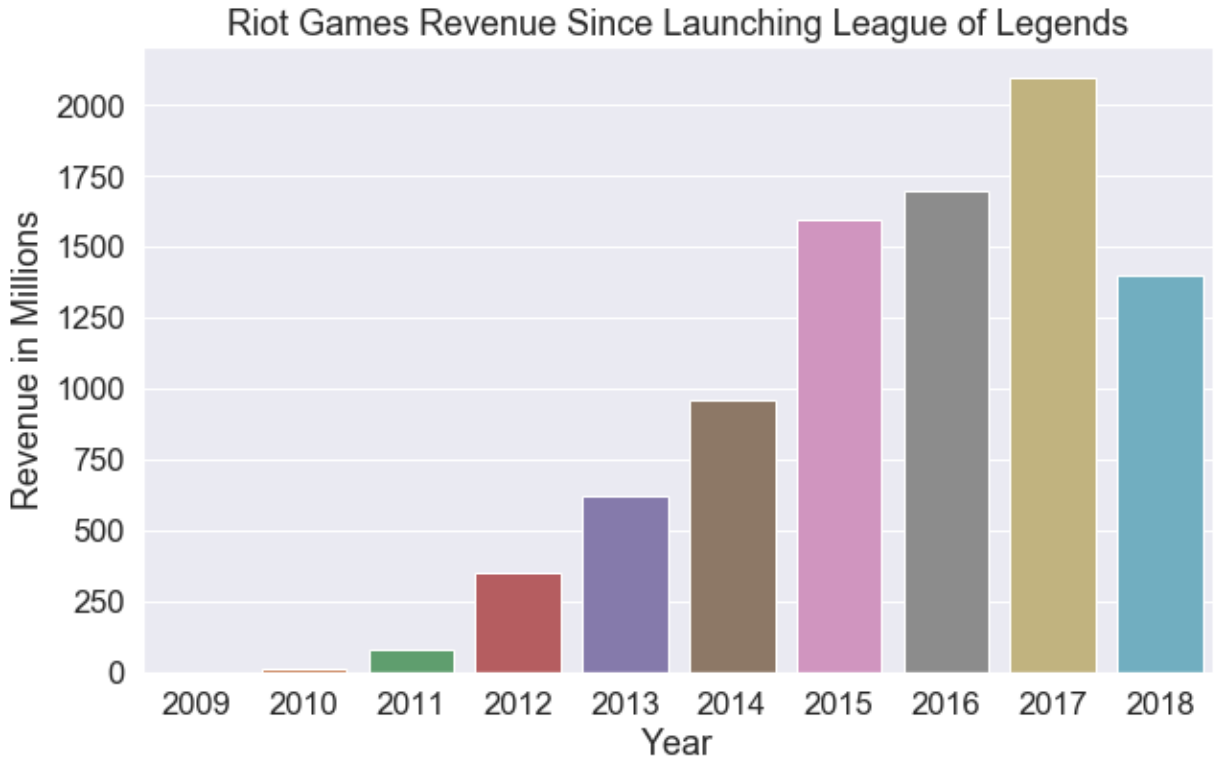
In [53]:
```
plt.figure(figsize=(20,10))
plt.xticks(rotation='60')
sns.countplot(x=df['Year'])
```

Out[53]: <matplotlib.axes._subplots.AxesSubplot at 0x2048761eda0>



In [54]:
```
revenue = [1.29, 17.25, 85, 354.5, 624, 964, 1600, 1700, 2100, 1400]
year = [2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018]

plt.figure(figsize=(10, 6))
sns.set(font_scale=1.5)
plot = sns.barplot(x = year, y = revenue)
plot.set(xlabel='Year', ylabel='Revenue in Millions', title='Riot Games Revenue S

#Note: 2012 revenue data was not available, figures were extrapolated according t
```

Out[54]:
```
[Text(0, 0.5, 'Revenue in Millions'),
 Text(0.5, 0, 'Year'),
 Text(0.5, 1.0, 'Riot Games Revenue Since Launching League of Legends')]
```



While not an exhaustive analysis by any means, the success of Riot Games compared to the industry at large may demonstrate changing trends in consumer preferences, e.g., in long-term games that are free-to-play with in-game monetization aspects.

Conclusion:

In this dataset, we inspected the market and understood the top publishers, genres, and sales amongst the years. If one is entering the video game market, this information would help that individual to analyze and understand the industry at the time of 2016. Utilizing the various hypothesis testing techniques such as z and ANOVA test, we were able to analyze and reformulate the thinking for the CEO. We used graphs such as the ECDF and combined the information given from the descriptive statistic and boxplot. We also attempted to determine whether publishers seemed to prioritize releasing titles for genres that sold well, and through our correlation testing, the data seemed to suggest that publishers were indeed trying to publish games under well-performing genres.