In [59]:
```
#Paul Galvez
#Date: 6/3/23
#Class: DSC 540
#Week 11 & 12 Final Project Milestone 5
```

In [60]:
```
import sqlite3
```

In [61]:
```
import pandas as pd
import numpy as np
import matplotlib
from matplotlib import pyplot as plt
import os
import requests
import json
from bs4 import BeautifulSoup
```

In [71]:
```
#Connectign and creating a SQL database - I am making a new database for my project because
#one does not exist

conn = sqlite3.connect('alldatabases.db')
```

In [72]:
```
#Loading and creating a new datatable for CSV file - the dataset from milestone 2

data_table_csv = pd.read_csv('vgsales2019.csv')
data_table_csv.to_sql('CSV', conn, if_exists='replace', index=False)
```

Out[72]: 55792

In [73]:
```
#Loading and creating a new datatable for Web Data - the dataset from milestone 3

data_table_web = pd.read_csv('Video_Games_Sales_as_at_22_Dec_2016.csv')
data_table_web.to_sql('Web', conn, if_exists='replace', index=False)
```

Out[73]: 16719

In [74]:
```
#Loading and creating a new datatable for API - the first dataset from milestone 4

data_table_api = pd.read_csv('deals.json')
data_table_api.to_sql('API', conn, if_exists='replace', index=False)
```

Out[74]: 0

In [75]:
```
#Committing the changes to the daatabses

conn.commit()
```

In [76]:
```
#closing the connection to the databases

conn.close()
```

In [77]:
```
conn = sqlite3.connect('alldatabases.db')
```

In [78]:
```
#combining the three tables created in SQL - all_tabels is the variable I'll be using

all_tables = pd.read_sql_query('SELECT name from sqlite_master WHERE type="table"', conn)
all_tables
```

Out[78]:

	name
0	CSV Table
1	Web Table
2	API Table
3	CSV
4	Web
5	API

In [85]:
```
#making sure the data is loaded for CSV table

query="select * from CSV"
csv_data=pd.read_sql_query(query, conn)
csv_data.head()
```

Out[85]:

	Rank	Name	basename	Genre	ESRB_Rating	Platform	Publisher	Developer	VGChartz_Score	Critic_Score	...	N
0	1	Wii Sports	wii-sports	Sports	E	Wii	Nintendo	Nintendo EAD	None	7.7	...	
1	2	Super Mario Bros.	super-mario-bros	Platform	None	NES	Nintendo	Nintendo EAD	None	10.0	...	
2	3	Mario Kart Wii	mario-kart-wii	Racing	E	Wii	Nintendo	Nintendo EAD	None	8.2	...	
3	4	PlayerUnknown's Battlegrounds	playerunknowns-battlegrounds	Shooter	None	PC	PUBG Corporation	PUBG Corporation	None	NaN	...	
4	5	Wii Sports Resort	wii-sports-resort	Sports	E	Wii	Nintendo	Nintendo EAD	None	8.0	...	

5 rows × 23 columns

In [86]:
```
#making sure the data is loaded for Web table

query="select * from Web"
web_data=pd.read_sql_query(query, conn)
web_data.head()
```

Out[86]:

	Name	Platform	Year_of_Release	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Critic_Score
0	Wii Sports	Wii	2006.0	Sports	Nintendo	41.36	28.96	3.77	8.45	82.53	76.0
1	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24	NaN
2	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.68	12.76	3.79	3.29	35.52	82.0
3	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.61	10.93	3.28	2.95	32.77	80.0
4	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	31.37	NaN

In [87]:
```
#making sure the data is loaded for API table

query="select * from API"
api_data=pd.read_sql_query(query, conn)
api_data.head()
```

Out[87]:

[{"internalName":"ONEPIECEBURNINGBLOODGOLDEDITION"	title:"ONE PIECE BURNING BLOOD GOLD EDITION"	metacriticLink:"\/game\/pc\/one-piece-burning-blood---gold-edition"	dealID:"Tn3N7Vw24RyW%2FroM7q

0 rows × 1145 columns

In [88]:
```
#reconecting to the databases - alldatabases.db

conn = sqlite3.connect('alldatabases.db')
```

In [116]:

```
#Joining the three databases using the join operation

query='''
    SELECT *
    FROM CSV
    JOIN WEB ON CSV.Name = WEB.Name
    '''
merge_data = pd.read_sql_query(query, conn)
merge_data
```

Out[116]:

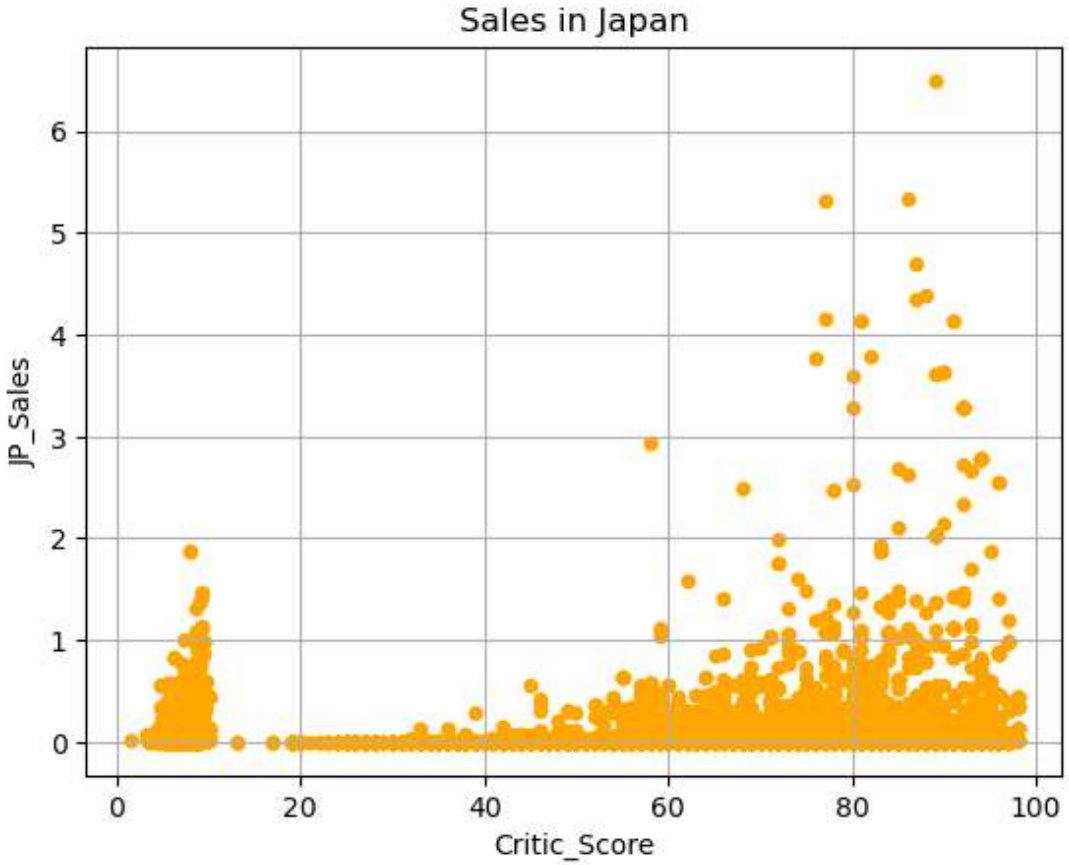
	Rank	Name	basename	Genre	ESRB_Rating	Platform	Publisher	Developer	VGChartz_Score	Critic_Score	...	EU_Sale
0	1	Wii Sports	wii-sports	Sports	E	Wii	Nintendo	Nintendo EAD	None	7.7	...	28.9
1	2	Super Mario Bros.	super-mario-bros	Platform	None	NES	Nintendo	Nintendo EAD	None	10.0	...	1.3
2	2	Super Mario Bros.	super-mario-bros	Platform	None	NES	Nintendo	Nintendo EAD	None	10.0	...	3.5
3	3	Mario Kart Wii	mario-kart-wii	Racing	E	Wii	Nintendo	Nintendo EAD	None	8.2	...	12.7
4	5	Wii Sports Resort	wii-sports-resort	Sports	E	Wii	Nintendo	Nintendo EAD	None	8.0	...	10.9
...	...	...	...	...	...	...	...	...	...	...	...	.
42600	55724	XCOM: Enemy Unknown	xcom-enemy-unknown	Strategy	None	Linux	Feral Interactive	Firaxis Games	None	NaN	...	0.2
42601	55724	XCOM: Enemy Unknown	xcom-enemy-unknown	Strategy	None	Linux	Feral Interactive	Firaxis Games	None	NaN	...	0.2
42602	55747	Yakuza 4	yakuza-4	Adventure	None	PS4	Sega	Sega	None	NaN	...	0.1
42603	55748	Yakuza 5	yakuza-5	Action-Adventure	None	PS4	Sega	Sega	None	NaN	...	0.0
42604	55763	Ys VIII: Lacrimosa of Dana	ys-viii-lacrimosa-of-dana	Role-Playing	None	PC	NIS America	Falcom	None	NaN	...	0.0

42605 rows × 39 columns

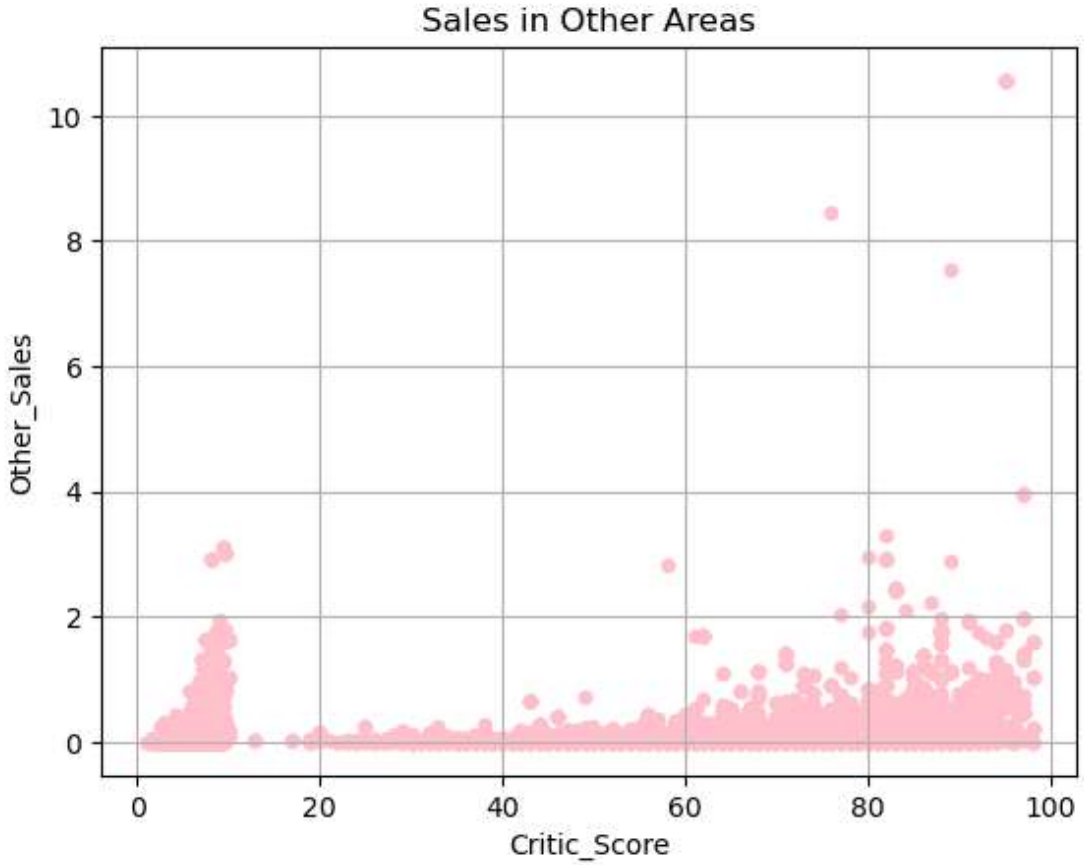
In [118]:
```
merge_data.shape
```

Out[118]: (42605, 39)

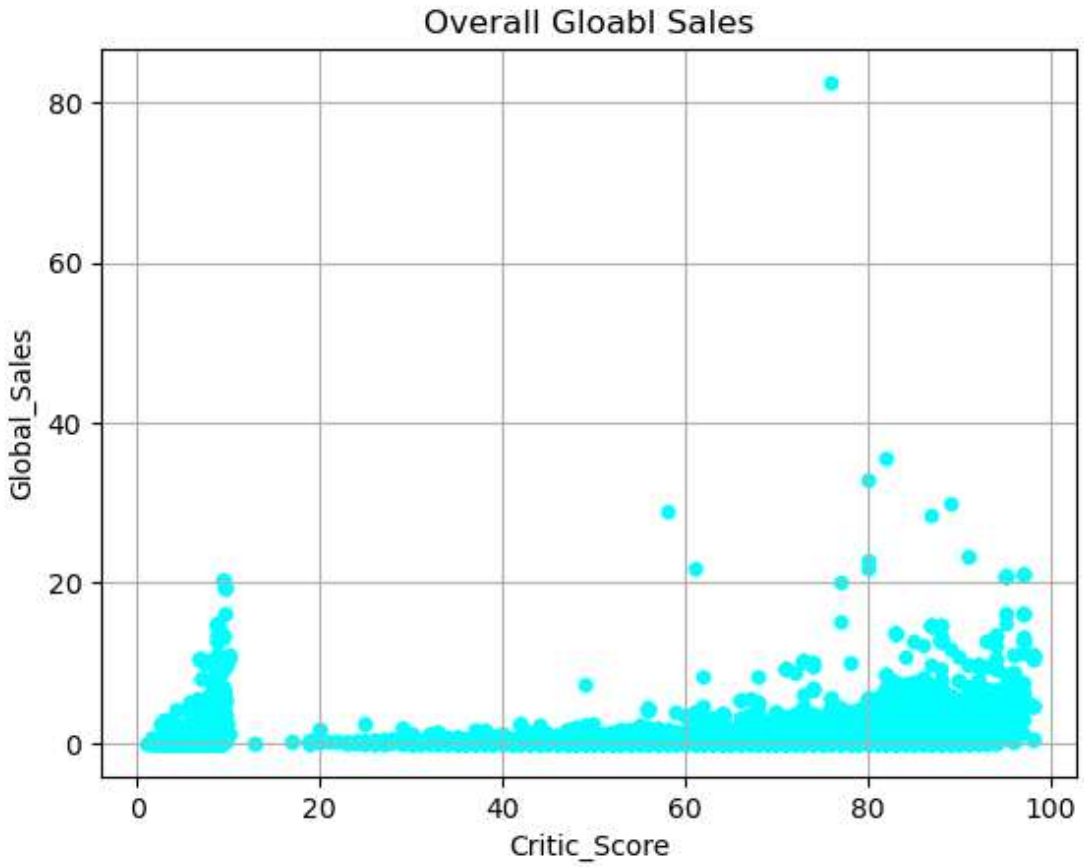
In [169]:
```
#Scatter plot where critic scores and sales for video games in Japan. We can see sales are not impacted negati
#most part when a game gets a low score. There are outliers where a high critic score correlates to high sales

merge_data.plot.scatter(x='Critic_Score', color='orange', y='JP_Sales')
plt.title('Sales in Japan')
plt.grid()
```



In [170]:
```
#We can see there is a similar visual result for sales in other areas outside of Japan. There is no correlatio
#sales and critic scores for video games

merge_data.plot.scatter(x='Critic_Score', color='pink', y='Other_Sales')
plt.title('Sales in Other Areas')
plt.grid()
```



In [171]:
```
#Again a simialr reslut for overall global sales

merge_data.plot.scatter(x='Critic_Score', color='cyan', y='Global_Sales')
plt.title('Overall Gloabl Sales')
plt.grid()
```



In [156]:
```
from scipy import stats
import seaborn as sns
```

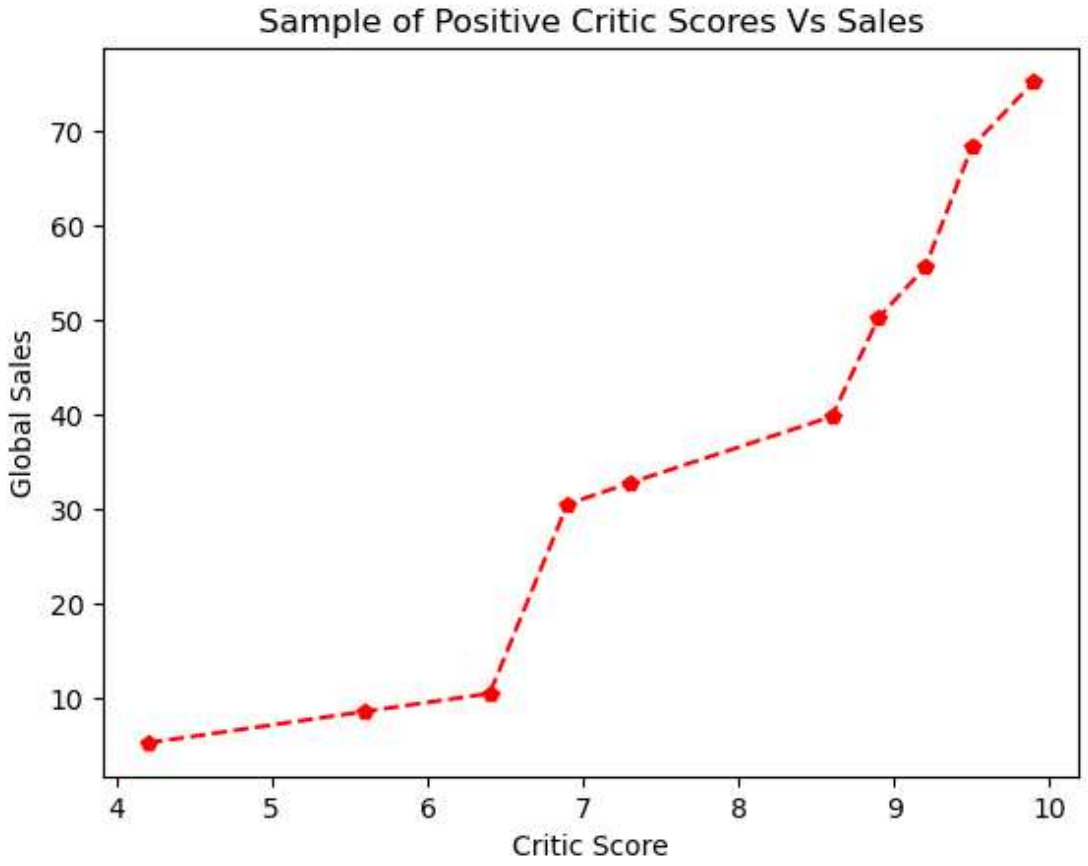
In [158]:
```
merge_data.head(10)
```

Out[158]:

Developer	VGChartz_Score	Critic_Score	...	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Critic_Score	Critic_Count	User_Score	Use
Nintendo EAD	None	7.7	...	28.96	3.77	8.45	82.53	76.0	51.0	8	
Nintendo EAD	None	10.0	...	1.30	0.15	0.22	5.07	NaN	NaN	None	
Nintendo EAD	None	10.0	...	3.58	6.81	0.77	40.24	NaN	NaN	None	
Nintendo EAD	None	8.2	...	12.76	3.79	3.29	35.52	82.0	73.0	8.3	
Nintendo EAD	None	8.0	...	10.93	3.28	2.95	32.77	80.0	73.0	8	
Nintendo EAD	None	9.1	...	9.14	6.50	2.88	29.80	89.0	65.0	8.5	
Bullet Proof Software	None	NaN	...	2.26	4.22	0.58	30.26	NaN	NaN	None	
Bullet Proof Software	None	NaN	...	0.69	1.81	0.11	5.58	NaN	NaN	None	
Nintendo EAD	None	8.6	...	6.94	4.70	2.24	28.32	87.0	80.0	8.4	
Mojang AB	None	10.0	...	2.37	0.00	0.87	5.26	NaN	NaN	None	

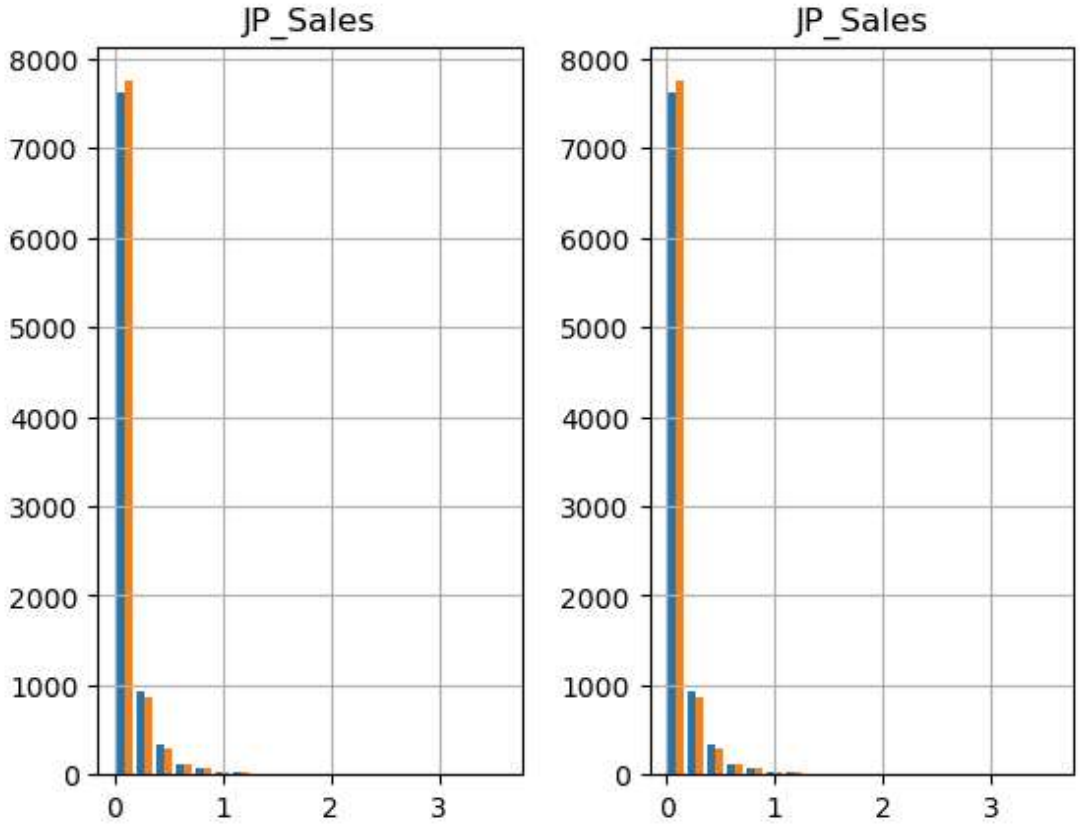
In [168]:
```
#I pulled some data score where positive critic score had a positive affect on sales from a global perspective

fig, ax = plt.subplots()
ax.plot([4.2, 5.6, 6.4, 6.9, 7.3, 8.6, 8.9, 9.2, 9.5, 9.9], [5.26, 8.6, 10.5, 30.52, 32.77, 39.80,
                                                              50.26, 55.58, 68.32, 75.26], '--pr')
plt.xlabel('Critic Score')
plt.ylabel('Global Sales')
plt.title('Sample of Positive Critic Scores Vs Sales')
```

Out[168]:
```
Text(0.5, 1.0, 'Sample of Positive Critic Scores Vs Sales')
```



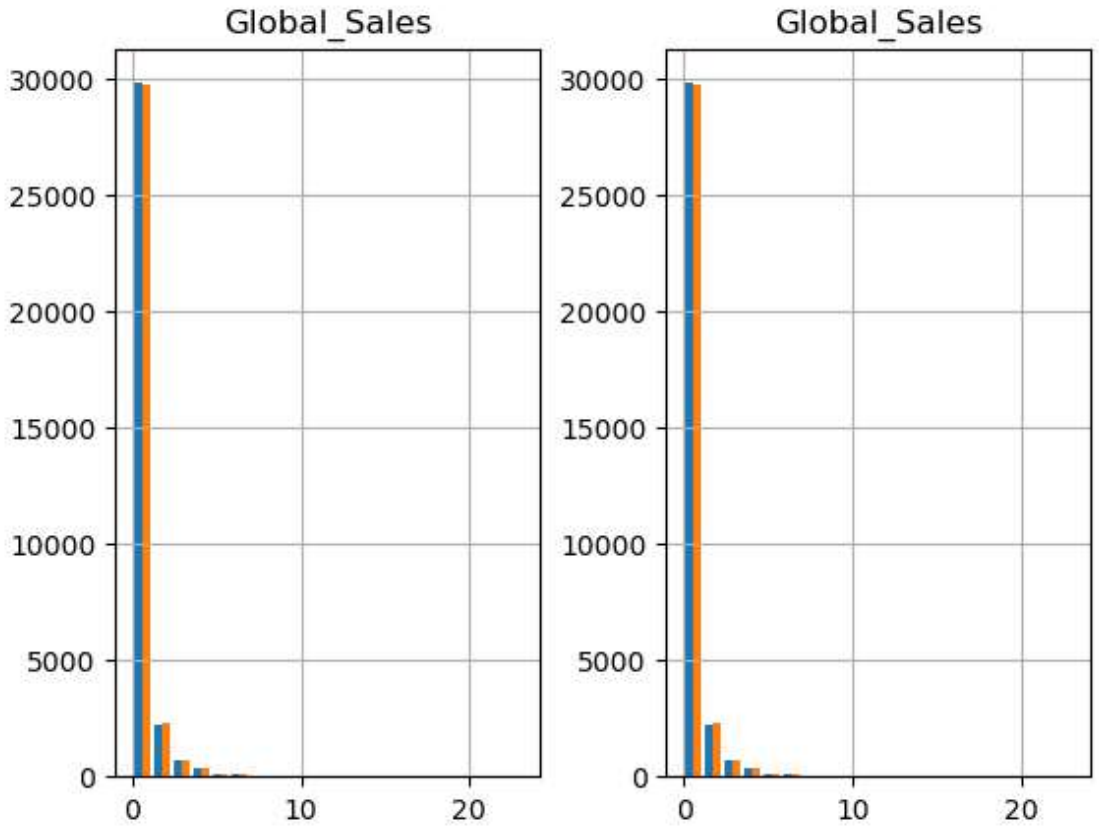
In [180]:
```
merge_data['JP_Sales'].hist(bins = 20)
```

Out[180]:
```
array([[<AxesSubplot:title={'center':'JP_Sales'}>,
        <AxesSubplot:title={'center':'JP_Sales'}>]], dtype=object)
```



In [184]:
```
merge_data['Global_Sales'].hist(bins = 20)
```

Out[184]:
```
array([[<AxesSubplot:title={'center':'Global_Sales'}>,
        <AxesSubplot:title={'center':'Global_Sales'}>]], dtype=object)
```



In [ ]:

```
#Summary -
#The project was challenging in a way that made me think about what I was doing in each milestone in a step by
#I found myself thinking about how each of the data prep steps would impact the final version of my dataset and
#process was critical to linking everyting together. Every time I would clean and prep the dataset, I found mys
#guessing if that particular process was necessary. Or asking myslef to justify why I was doing the data prep s
#taking. I thought the API section of the project was the most challenging because I never worked with API sour
#have heard a lot about them from playing video games with friends. However, the API source was initally confus
#with because there was some confusion on my part with the key and loading it into Python. I was happy when the
#was loaded and working and I was able to get it sorted out using similar processes I applied to the data clean
#other sources. The web scrapping was also an interesting week/milestone in the class and for the project. In t
#project, I was able to find a sales table through Wikipedia that was similar to the table from the assingment
#able to apply what we used in the assingment to my project. Overall I felt this was a steep learning curve for
#concepts we coverd. But I thought it was challengning in a fun way that frustrated me in a good way because I
#figure things out if I just kept trying.
```