Portfolio Final Exam

December 7, 2023

```
[2]: import numpy as np
     from datascience import *
     import matplotlib.pyplot as plt
     plt.style.use("ggplot")
     %matplotlib inline
```

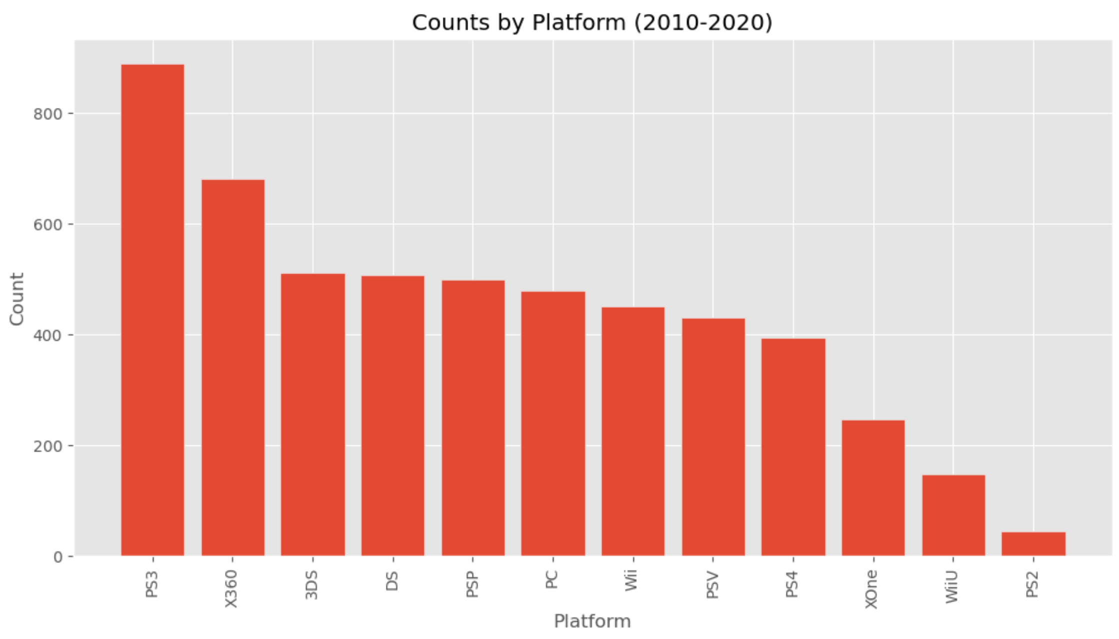
This is a small table that reads some data from the file.

```
[32]: data = Table.read_table('Video_game_data.csv')
      data.show(5)
```

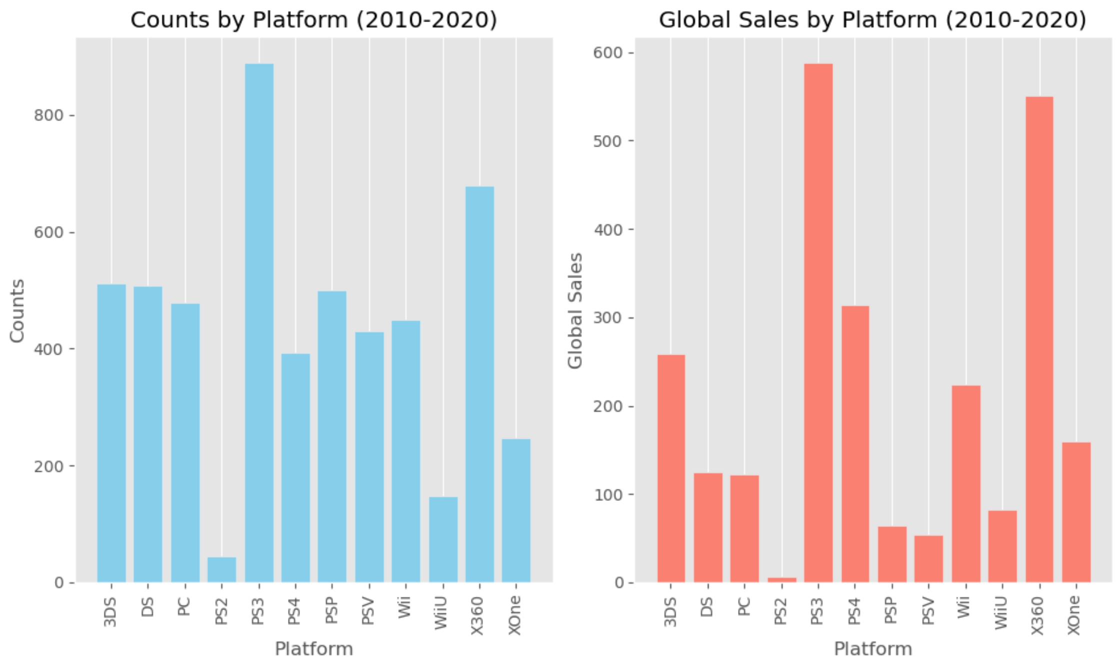
```
<IPython.core.display.HTML object>
```

This bar chart shows the amount of games per platform from 2010 to 2020.

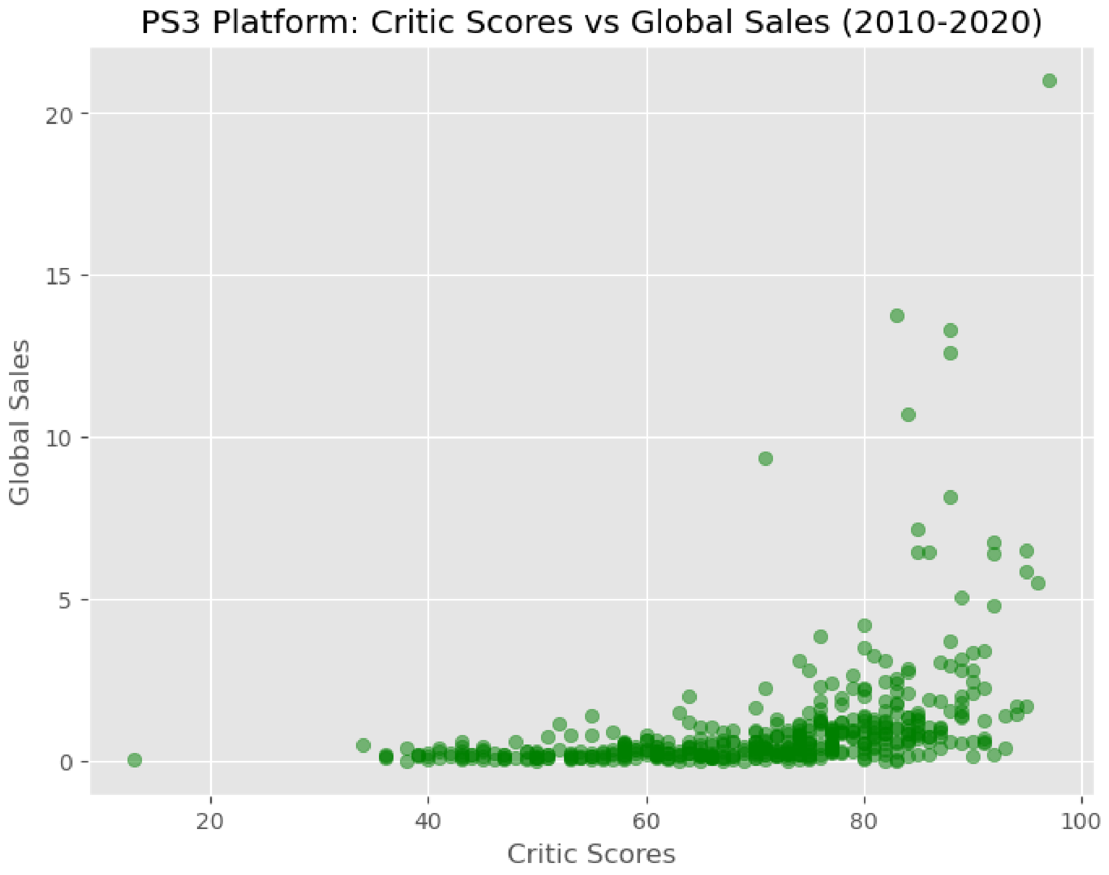
```
[32]: filtered_data = data.where('Year_of_Release', are.between_or_equal_to(2010,␣
       ↪2020))
      platform_counts = filtered_data.group('Platform').sort('count', descending=True)

      plt.figure(figsize=(12, 6))
      plt.bar(platform_counts['Platform'], platform_counts['count'])
      plt.title('Counts by Platform (2010-2020)')
      plt.xlabel('Platform')
      plt.ylabel('Count')
      plt.xticks(rotation=90)
      plt.show()
```



[47]:
```
filtered_data = data.where('Year_of_Release', are.between_or_equal_to(2010,␣
 ↪2020))


platform_sales_count = filtered_data.group('Platform')
platform_sales_sum = filtered_data.group('Platform', sum)


plt.figure(figsize=(10, 6))


plt.subplot(1, 2, 1)
plt.bar(platform_sales_count['Platform'], platform_sales_count['count'],␣
 ↪color='skyblue')
plt.title('Counts by Platform (2010-2020)')
plt.xlabel('Platform')
plt.ylabel('Counts')
plt.xticks(rotation=90)
plt.grid(axis='y')


plt.subplot(1, 2, 2)
plt.bar(platform_sales_sum['Platform'], platform_sales_sum['Global_Sales sum'],␣
 ↪color='salmon')
plt.title('Global Sales by Platform (2010-2020)')
plt.xlabel('Platform')
plt.ylabel('Global Sales')
plt.xticks(rotation=90)
plt.grid(axis='y')

plt.tight_layout()
plt.show()
```

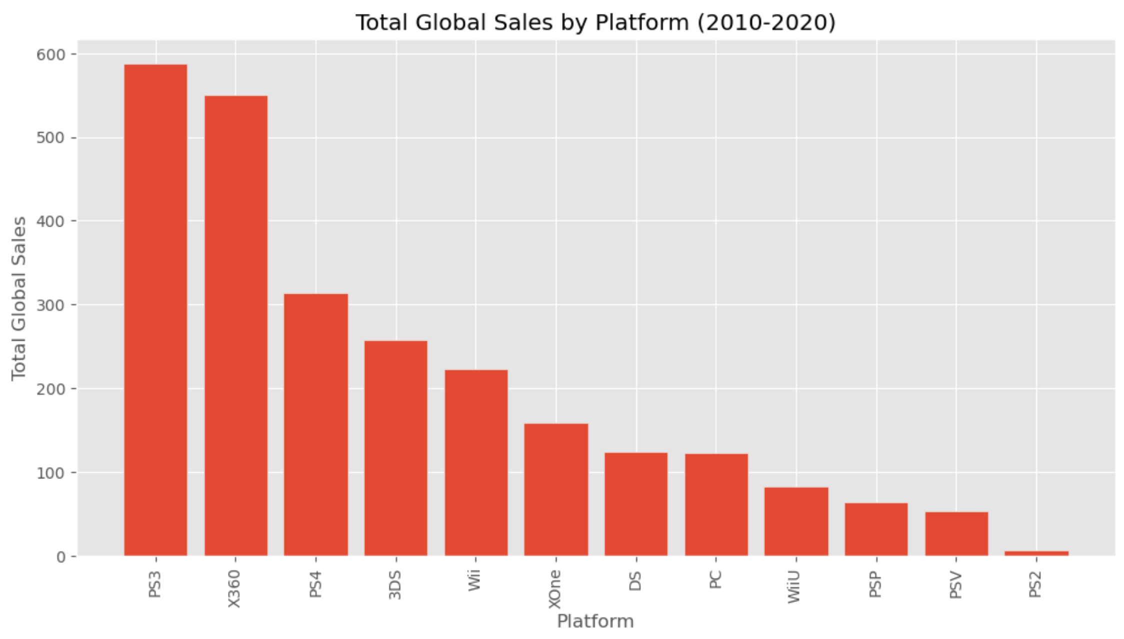


[94]:
```
ps3_data = data.where(
    np.logical_and(
        np.logical_and(data['Year_of_Release'] >= 2010, data['Year_of_Release']␣
 ↪<= 2020),
        data['Platform'] == 'PS3'
    )
)


ps3_critic_scores = ps3_data.column('Critic_Score')
ps3_global_sales = ps3_data.column('Global_Sales')


plt.figure(figsize=(8, 6))
plt.scatter(ps3_critic_scores, ps3_global_sales, color='green', alpha=0.5)
plt.title('PS3 Platform: Critic Scores vs Global Sales (2010-2020)')
plt.xlabel('Critic Scores')
plt.ylabel('Global Sales')
plt.grid(True)
plt.show()
```



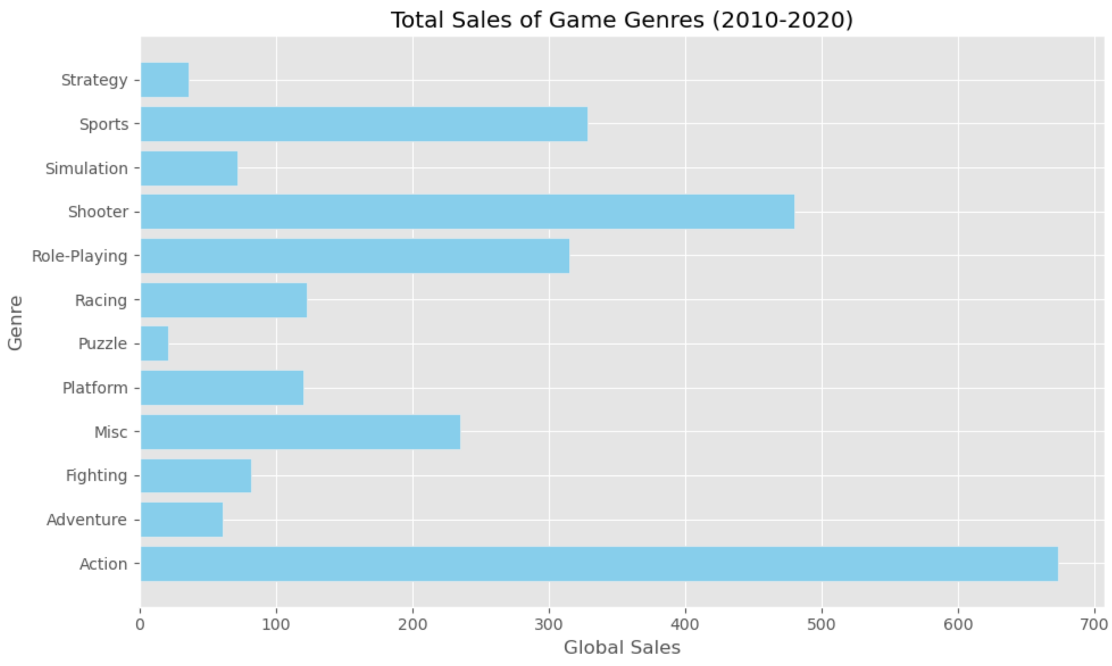
This chart shows the Global Sales of each platform from 2010-2020

```
[31]: filtered_data = data.where('Year_of_Release', are.between_or_equal_to(2010,␣
       ↪2020))
      platform_sales = filtered_data.group('Platform', sum)
      platform_sales = platform_sales.sort('Global_Sales sum', descending=True)

      plt.figure(figsize=(12, 6))
      plt.bar(platform_sales['Platform'], platform_sales['Global_Sales sum'])
      plt.title('Total Global Sales by Platform (2010-2020)')
      plt.xlabel('Platform')
      plt.ylabel('Total Global Sales')
      plt.xticks(rotation=90)
      plt.show()
```



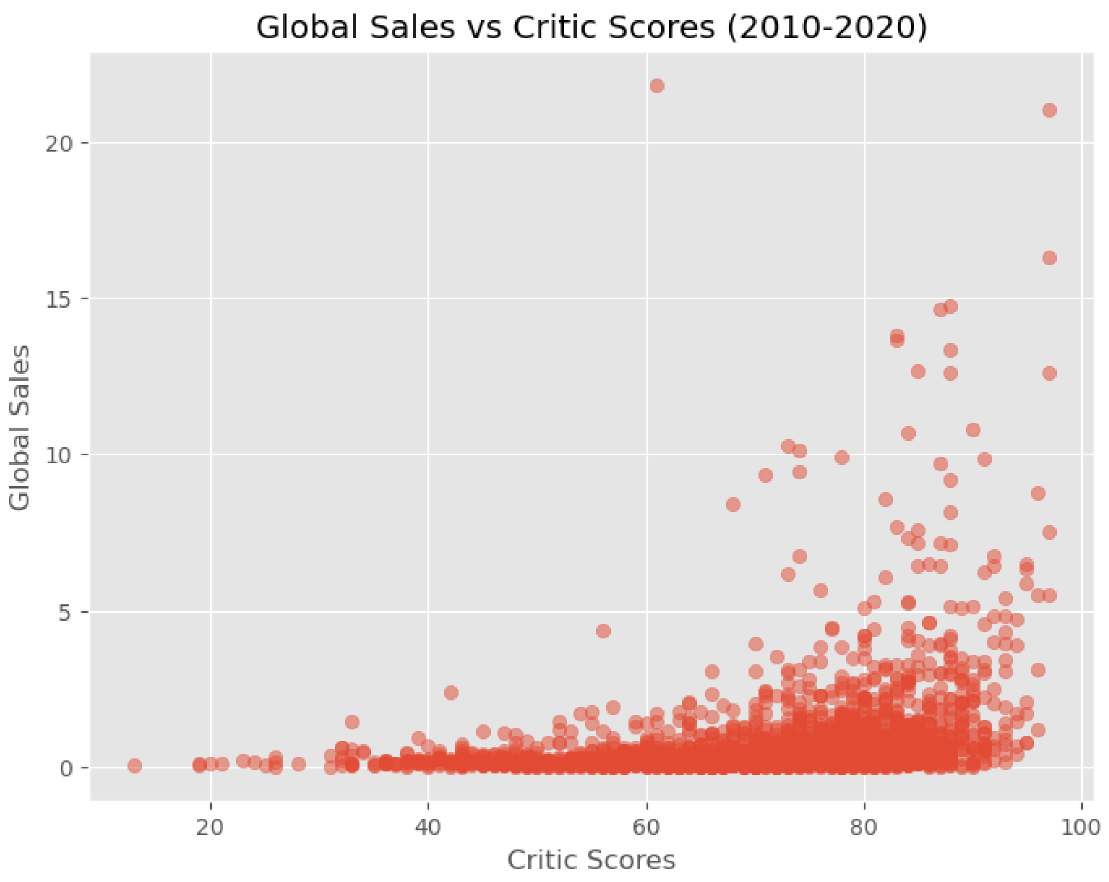
This code creates a horizontal bar chart displaying the total sales of different games from 2010 to 2010.

```
[46]: filtered_data = data.where('Year_of_Release', are.between_or_equal_to(2010,␣
        ↪2020))
      sales_by_genre = filtered_data.group('Genre', sum)
      genres = sales_by_genre['Genre']
      total_sales = sales_by_genre['Global_Sales sum']


      plt.figure(figsize=(10, 6))
      plt.barh(genres, total_sales, color='skyblue')
      plt.xlabel('Global Sales')
      plt.ylabel('Genre')
      plt.title('Total Sales of Game Genres (2010-2020)')
      plt.tight_layout()
      plt.show()
```

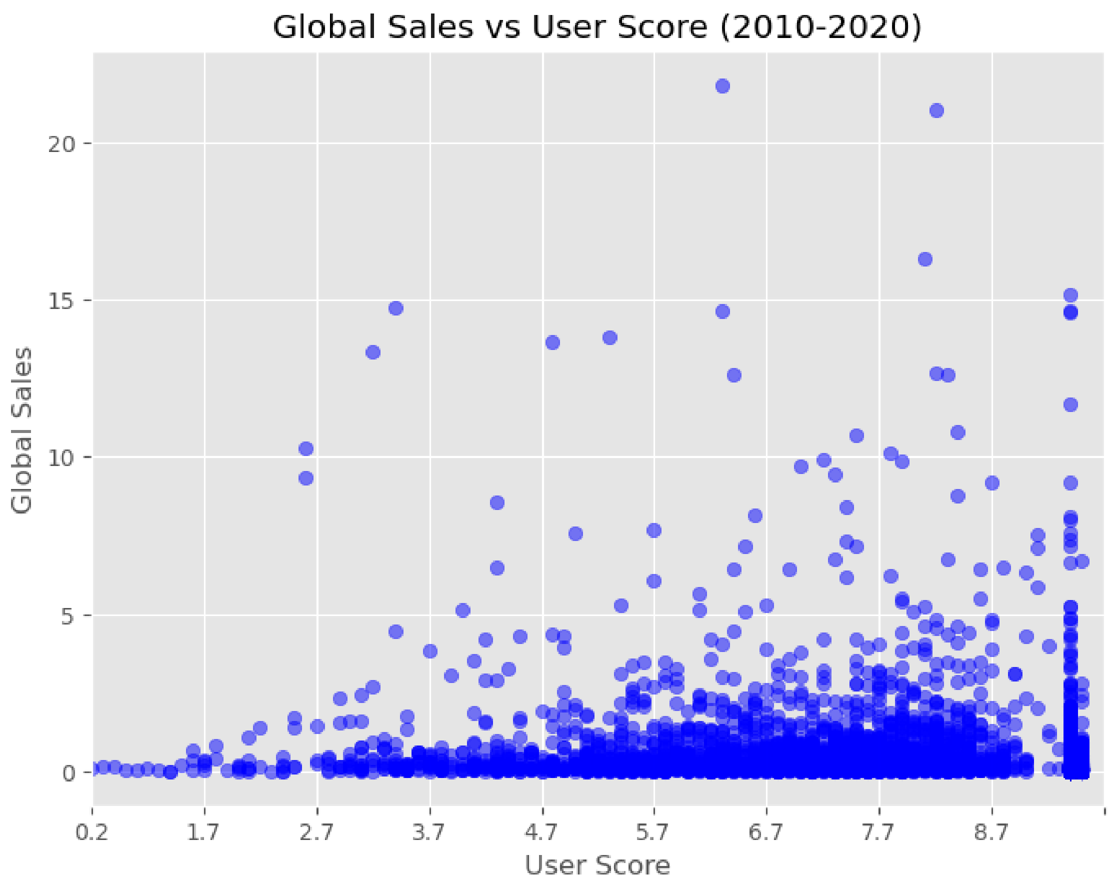


This code displays a scatter plot of Critic Scores against Global Sales.

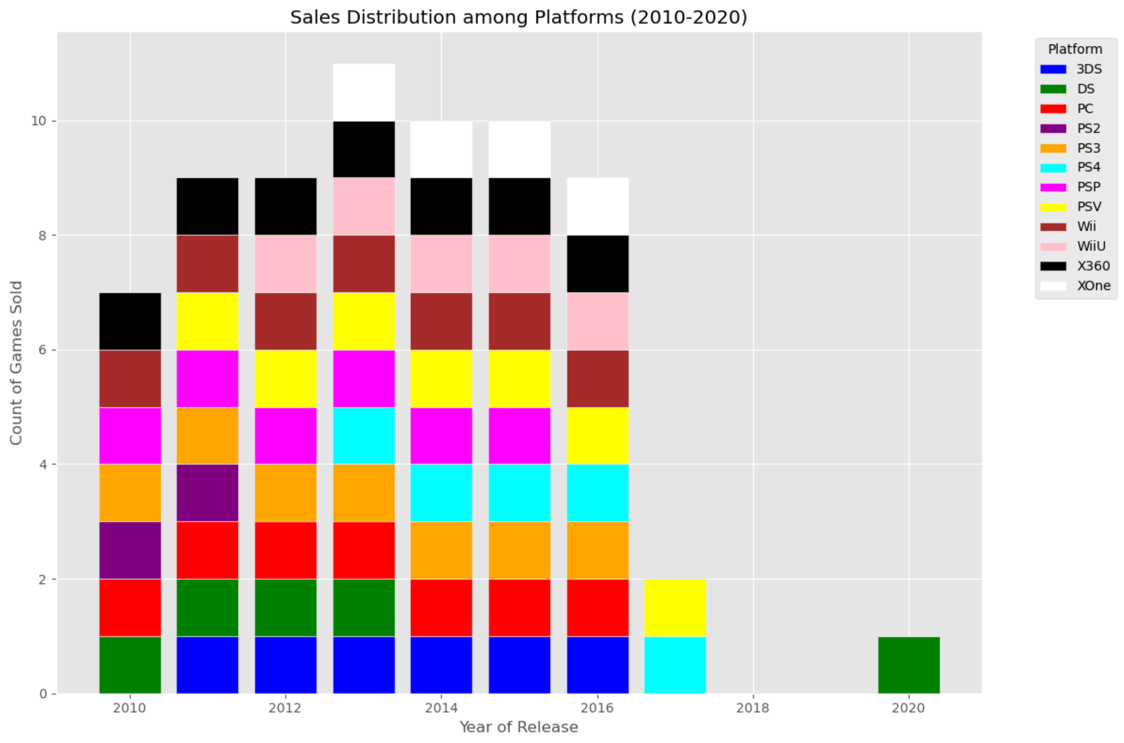
```
[122]: filtered_data = data.where('Year_of_Release', are.between_or_equal_to(2010,␣
         ↪2020))


       critic_scores_filtered = filtered_data.column('Critic_Score')
       global_sales_filtered = filtered_data.column('Global_Sales')


       plt.figure(figsize=(8, 6))
       plt.scatter(critic_scores_filtered, global_sales_filtered, alpha=0.5)
       plt.title('Global Sales vs Critic Scores (2010-2020)')
       plt.xlabel('Critic Scores')
       plt.ylabel('Global Sales')
       plt.show()
```



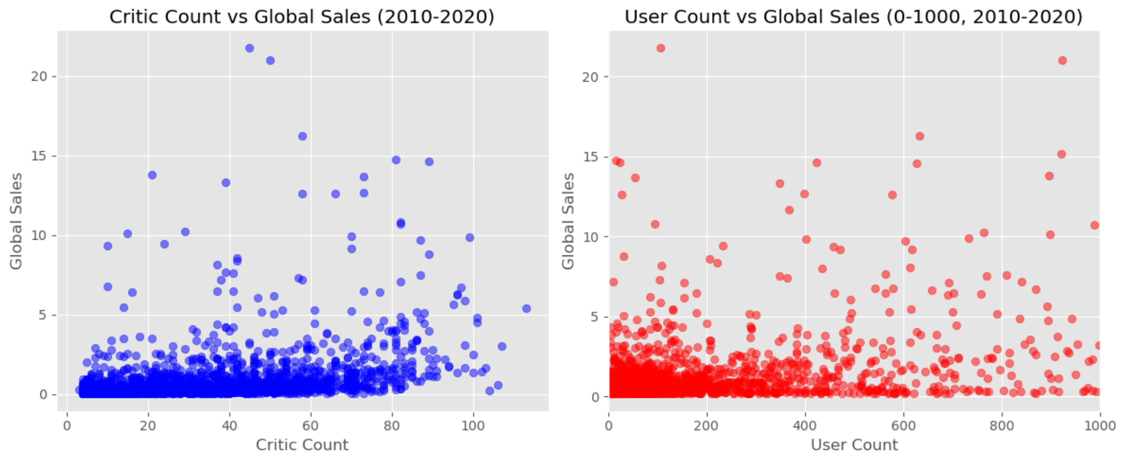
[64]:
```
filtered_data = data.where('Year_of_Release', are.between_or_equal_to(2010,␣
  ↪2020))


sorted_data = filtered_data.sort('User_Score')


global_sales = sorted_data.column('Global_Sales')
user_score = sorted_data.column('User_Score')

plt.figure(figsize=(8, 6))
plt.scatter(user_score, global_sales, alpha=0.5, color='blue')
plt.title('Global Sales vs User Score (2010-2020)')
plt.xlabel('User Score')
plt.ylabel('Global Sales')
plt.grid(True)


plt.xticks(np.arange(0, 100, 10))
plt.gca().set_xlim(0, 90)

plt.show()
```



[153]:
```
filtered_data = data.where('Year_of_Release', are.between_or_equal_to(2010,␣
↪2020))


grouped_data = filtered_data.group(['Year_of_Release', 'Platform'])


platforms = np.unique(filtered_data.column('Platform'))
years = np.unique(filtered_data.column('Year_of_Release'))


platform_counts = {platform: np.zeros(len(years)) for platform in platforms}


for platform in platforms:
    platform_sales = grouped_data.where('Platform', platform)
    for i, year in enumerate(years):
        platform_counts[platform][i] = platform_sales.where('Year_of_Release',␣
↪year).num_rows


colors = ['blue', 'green', 'red', 'purple', 'orange', 'cyan', 'magenta',␣
↪'yellow', 'brown', 'pink', 'black','white']


plt.figure(figsize=(12, 8))
bottom = np.zeros(len(years))

for i, platform in enumerate(platforms):
    plt.bar(years, platform_counts[platform], label=platform, bottom=bottom,␣
↪color=colors[i % len(colors)])
    bottom += platform_counts[platform]

plt.xlabel('Year of Release')
plt.ylabel('Count of Games Sold')
plt.title('Sales Distribution among Platforms (2010-2020)')
plt.legend(title='Platform', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```



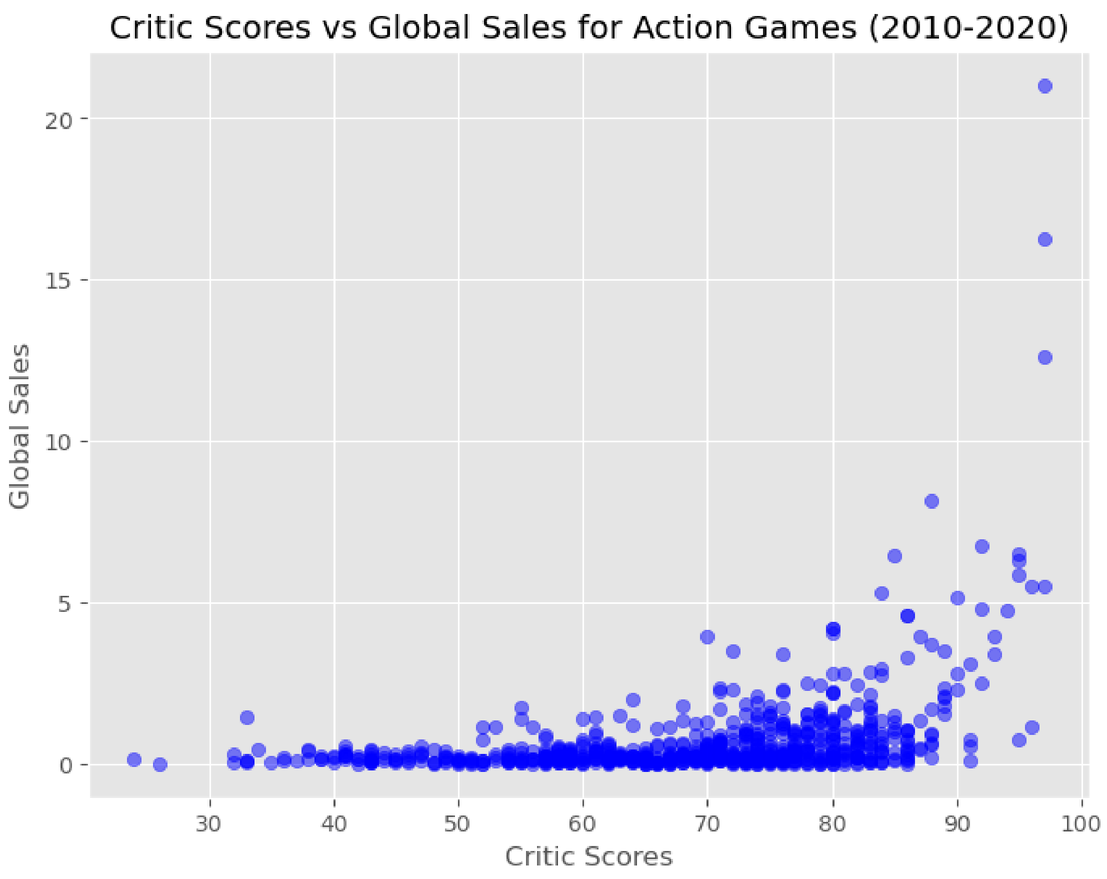
```
[65]: filtered_data = data.where('Year_of_Release', are.between_or_equal_to(2010,␣
        ↪2020))

      critic_count = filtered_data.column('Critic_Count')
      user_count = filtered_data.column('User_Count')
      global_sales = filtered_data.column('Global_Sales')


      filtered_user_count = [count for count in user_count if 0 <= count <= 1000]
      filtered_global_sales = global_sales[:len(filtered_user_count)]


      fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 5))


      ax1.scatter(critic_count, global_sales, color='blue', alpha=0.5)
      ax1.set_title('Critic Count vs Global Sales (2010-2020)')
      ax1.set_xlabel('Critic Count')
      ax1.set_ylabel('Global Sales')


      ax2.scatter(filtered_user_count, filtered_global_sales, color='red', alpha=0.5)
      ax2.set_title('User Count vs Global Sales (0-1000, 2010-2020)')
      ax2.set_xlabel('User Count')
      ax2.set_ylabel('Global Sales')
      ax2.set_xlim(0, 1000)

      plt.tight_layout()
      plt.show()
```



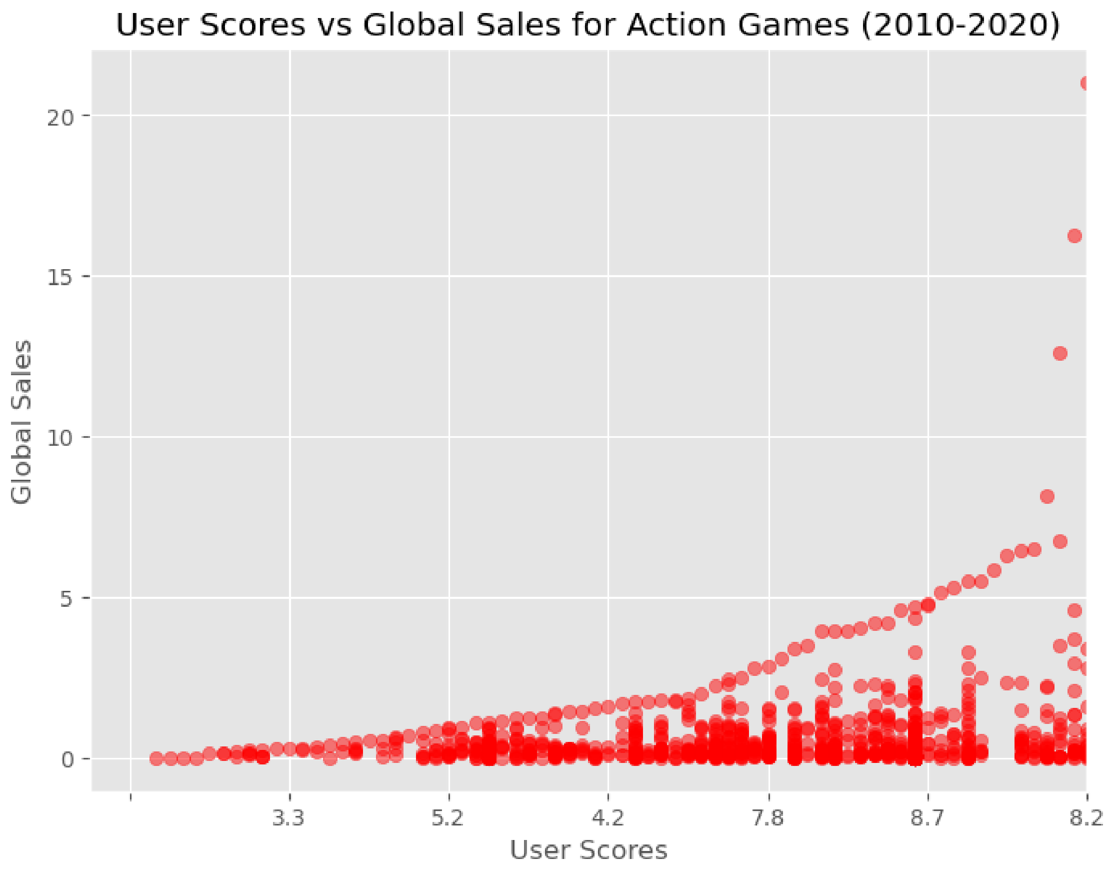
[67]:
```
filtered_data = data.where('Year_of_Release', are.between_or_equal_to(2010,␣
  ↪2020))


genre_scores = filtered_data.select('Genre', 'Critic_Score')


genre_critic_count = genre_scores.group('Genre', np.count_nonzero)
genre_critic_count.relabel('Critic_Score count_nonzero', 'Count')


sorted_genre_count = genre_critic_count.sort('Count', descending=True)
sorted_genre_count.show()
```

```
<IPython.core.display.HTML object>
```

[71]:
```
filtered_action_data = data.where(
    np.logical_and(data['Year_of_Release'] >= 2010, data['Year_of_Release'] <=␣
  ↪2020)
).where('Genre', are.equal_to('Action'))


action_critic_scores = filtered_action_data.column('Critic_Score')
action_global_sales = filtered_action_data.column('Global_Sales')


plt.figure(figsize=(8, 6))
plt.scatter(action_critic_scores, action_global_sales, color='blue', alpha=0.5)
plt.title('Critic Scores vs Global Sales for Action Games (2010-2020)')
plt.xlabel('Critic Scores')
plt.ylabel('Global Sales')
plt.grid(True)

plt.show()
```

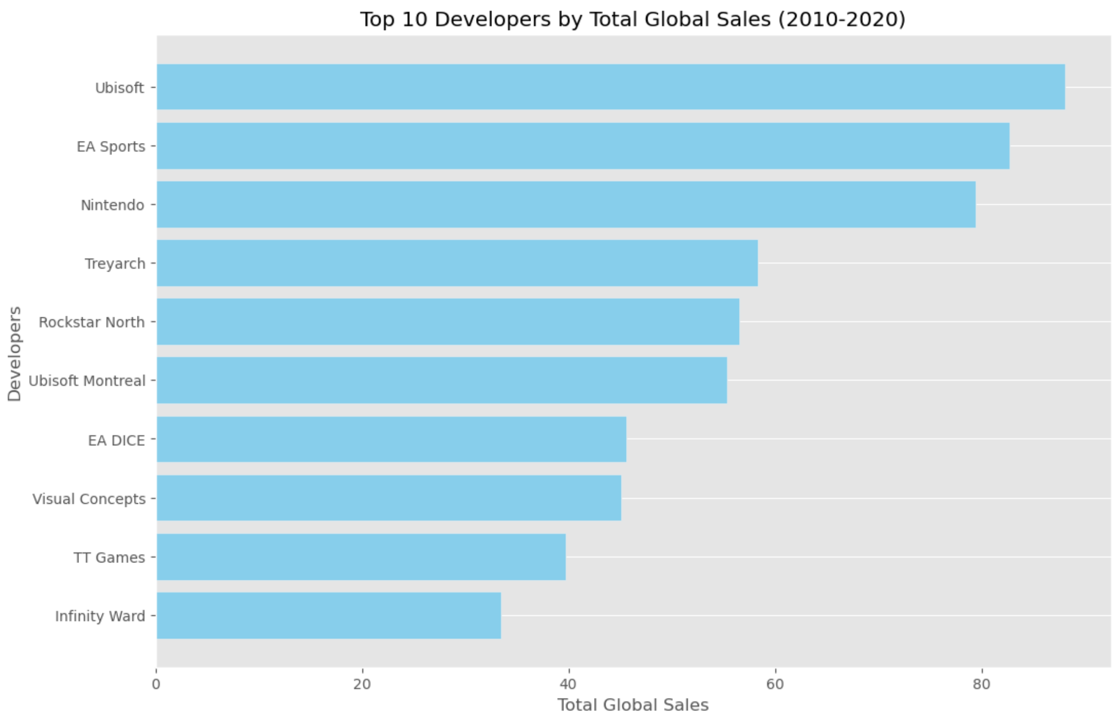


```
[91]: filtered_action_data = data.where(
          np.logical_and(data['Year_of_Release'] >= 2010, data['Year_of_Release'] <=␣
       ↪2020)
      ).where('Genre', are.equal_to('Action'))


      action_user_scores = filtered_action_data.column('User_Score')
      action_global_sales = filtered_action_data.column('Global_Sales')


      plt.figure(figsize=(8, 6))
      plt.scatter(action_user_scores, action_global_sales, color='red', alpha=0.5)
      plt.title('User Scores vs Global Sales for Action Games (2010-2020)')
      plt.xlabel('User Scores')
      plt.ylabel('Global Sales')
      plt.grid(True)
      plt.xticks(np.arange(0, 100, 12))
      plt.gca().set_xlim(0, 75)
      plt.gca().invert_xaxis()
plt.show()
```



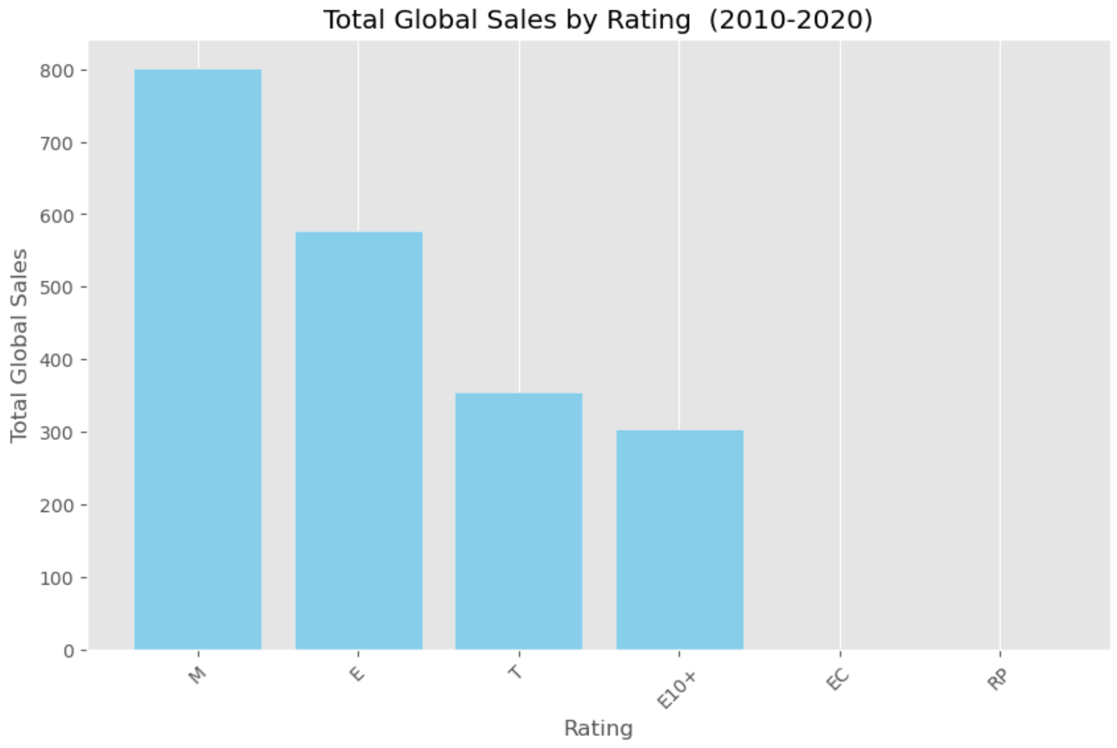
[117]:
```
filtered_data = data.where(
    np.logical_and(
        data['Year_of_Release'] >= 2010,
        data['Year_of_Release'] <= 2020
    )
)


nan_mask = filtered_data.apply(lambda x: x != 'nan', 'Developer')


filtered_data = filtered_data.where(nan_mask)


developer_sales = filtered_data.group('Developer', sum).sort('Global_Sales␣
↪sum', descending=True)
```

```
developers = developer_sales['Developer']
global_sales = developer_sales['Global_Sales sum']


plt.figure(figsize=(12, 8))
plt.barh(developers[:10], global_sales[:10], color='skyblue')
plt.title('Top 10 Developers by Total Global Sales (2010-2020)')
plt.xlabel('Total Global Sales')
plt.ylabel('Developers')
plt.gca().invert_yaxis()
plt.grid(axis='x')
plt.show()
```



[121]:
```
filtered_data = data.where(
    np.logical_and(
        data['Year_of_Release'] >= 2010,
        data['Year_of_Release'] <= 2020
    )
).where('Rating', are.not_containing('nan'))

rating_sales = filtered_data.group('Rating', sum).sort('Global_Sales sum',␣
 ↪descending=True)
```

```
rating_sales = rating_sales.where('Rating', are.not_equal_to('nan'))

ratings = rating_sales['Rating']
global_sales = rating_sales['Global_Sales sum']

plt.figure(figsize=(10, 6))
plt.bar(ratings, global_sales, color='skyblue')
plt.title('Total Global Sales by Rating  (2010-2020)')
plt.xlabel('Rating')
plt.ylabel('Total Global Sales')
plt.xticks(rotation=45)
plt.grid(axis='y')
plt.show()
```



Time Series Analysis of Sales: Create line charts or area charts to visualize the trend in sales across different platforms and genres over the years 2010 to 2020. You can aggregate sales data annually or monthly for each platform or genre to identify trends and fluctuations.

Genre-wise Sales Comparison: Bar charts or pie charts comparing total sales of different game genres within the specified time frame. This helps in understanding which genres were more popular in terms of sales.

Correlation between Ratings and Sales: Scatter plots or regression analysis to explore the relationship between critical acclaim (ratings, critic scores) and sales figures (global sales). This can provide insights into whether highly rated games tend to have higher sales.

Platform-wise Sales Distribution: Stacked bar charts or area charts displaying sales distribution among various platforms for each year within 2010-2020. This helps in visualizing which platforms dominated the sales in different years.

Heatmap of Correlations: A heatmap showing correlations between different variables like critic scores, user scores, and sales figures. This provides a comprehensive view of the relationships between these factors.

Boxplots for Sales and Ratings: Boxplots can display the distribution of sales and ratings across different platforms or genres, allowing for comparisons of their ranges, medians, and outliers.

Regression Analysis: Perform a regression analysis to quantitatively determine the relationship between critical acclaim (ratings, critic scores) and sales.