Summary of VGChart Graphs

Gabriel Grieco

Google Collab Project Referenced: https://colab.research.google.com/drive/1GLzScFejPew0Z2JpU7fdc71-ScZriras?usp=sharing

For this project, my goal for analyzing the data from VGChart is to get a better understanding of what’s popular, what has historically sold well, and which publishers/genres are the best for getting sales. The data that is shown can be helpful to anyone looking to make and sell games, or to invest in companies.

The dataset I’m using was retrieved from <https://www.kaggle.com/datasets/gregorut/videogamesales>, and describes the sales of all games from 1980 to 2016. Each game on the list is accompanied by its name, rank, publisher, genre, year, global sales, North American sales, Japanese sales, European sales, and other sales. According to the page, the data was gathered from scraping vgchartz.com. I chose this source because it has everything I need to reach my goals stated above.

At least one other person has done an analysis similar to mine, as seen at <https://www.kaggle.com/code/artyomvp1/exploratory-analysis-video-game-sales/notebook>. They also took a similar approach by using both pandas and matplotlib. However, they only used graphs to show the global data, whereas I use separate charts for each region for the majority of graphs.

For my code, I first imported the csv data into a pandas dataframe. I then calculated the mean, median, mode, max, and/or min for the relevant rows, doing checks on the column datatypes to determine which ones to do and how to format them. After that, I created box plots to show the distribution of years and each region’s sales. I filtered out any games with sales of less than 10,000 (0.00m) but the quartiles are still near the bottom of the charts, even with this sanity check. I also cleaned out records without a year for the Year’s boxplot.

Finally, I went through each of the seven graph sets, preparing the data in the correct formats so that they would show correctly, then displaying them using the appropriate plot type, formatting them for the best readability. For graphs involving years, I do not go beyond 2016, as that was when the source dataset was created.

First, I use a pie chart to display total video game sales by genre for each region. Given this is a high level showing of each genre’s popularity, I felt the pie chart was best at displaying each one’s popularity relative to the others.

For the second graph set, I use a plot graph to go into more detail of showing the total sales of each genre by year for each region. This should allow the viewer to see how genres of games have trended over time.

For the third, fourth, and fifth graph sets, I limited the number of publishers to the top 20 and 10 globally, respectively, as there are far more than would fit on a single bar graph, and having a plot for each publisher would make any graph illegible from a glance. For the third, I use bars to show the 20 publishers that have had the most games sold. With the fourth and fifth, I use plots to chart number of games sales published per year by publishers, and total number of sales per year by publisher, respectively. This information could help a would-be investor see which companies are most successful, and thus safest (generally) to invest in.

For the final two graphs, both use bar graphs to show the total number of sales on each platform, and the average number of sales for the top 100 games of each platform. The former shows the total general game sales, while the latter shows how successful the best-selling games are on each system. A low selling system with high average sales could indicate that there were less games made for that platform in general.

Overall, I believe my analyses were successful, as it allowed me to show the data in many ways in a visual format that would be easy to read/understand, and also be useful for a multitude of purposes.

Sources:

<https://www.kaggle.com/datasets/gregorut/videogamesales>

<https://www.kaggle.com/code/artyomvp1/exploratory-analysis-video-game-sales/notebook>