Video Game Statistics Report

Jason Chandler

December 11, 2024

Probability & Applied Statistics

Section 001, MW 6:00-7:50

Abstract:

This report details obtaining and analyzing .csv data on 40 of the most popular games from each year 2024 to 2004. Data was obtained through running a python script utilizing the external requests library and RAWG API. This data was analyzed through visualization inspired by chapter 1's relative frequency histograms and PivotTables and through the creation of questions inspired by existing textbook problems.

<u>Introduction:</u>

Upon receiving this assignment, I was excited to work with a set of relevant data. I was inspired by the instruction example of focusing on video games, so I researched some data on specific games. I looked for something to analyze trends across multiple video games. In doing so, I found the RAWG API and decided to learn how to use it. I saw this gathering step as a beneficial knowledge builder, a potential extra credit opportunity, and a way to practice programming in a language I am not very familiar with.

It took a really long time figuring out the API's syntax, teaching myself the Python requests, and fine-tuning errors like messed up values, but finally getting the completed csv was super rewarding. With it, I got immediately to work with building textbook questions based on it.

<u>Data Retrieval and Storage:</u>

The data was retrieved by running the getDataPlease.py script I developed using Pycharm. It uses the external requests library, the standard csv library, and the RAWG API. Requests gets the information from the API, csv writes the information, and RAWG provides it.

The logic hinges on the getYearlyGames(year) function, as it accesses the information and stores it as a list. For a given year, it gets data from the server by using requests.get() on the API url with the year as the given parameter. It parses a usable list in Python and returns it. This is where and how data gets from the server to the python code.

The writing logic is handled in the 'with open' statement. With the file open, it first prints the header and then begins looping from the top year to the bottom year, decreasing by one each time. The loop calls getYearlyGame(year) on each year, loops through each individual game in the response, and prints the data in the relevant order using .get().

<u>Analysis Through Visualization:</u>

Data visualization is a key part of studying statistics. Creating and analyzing visualizations are nice important tools to have in professional environments. This section aims to strengthen those skills by generating visualizations using Excel PivotCharts and drawing conclusions from them.

Figure 1: Total Playtime Distribution

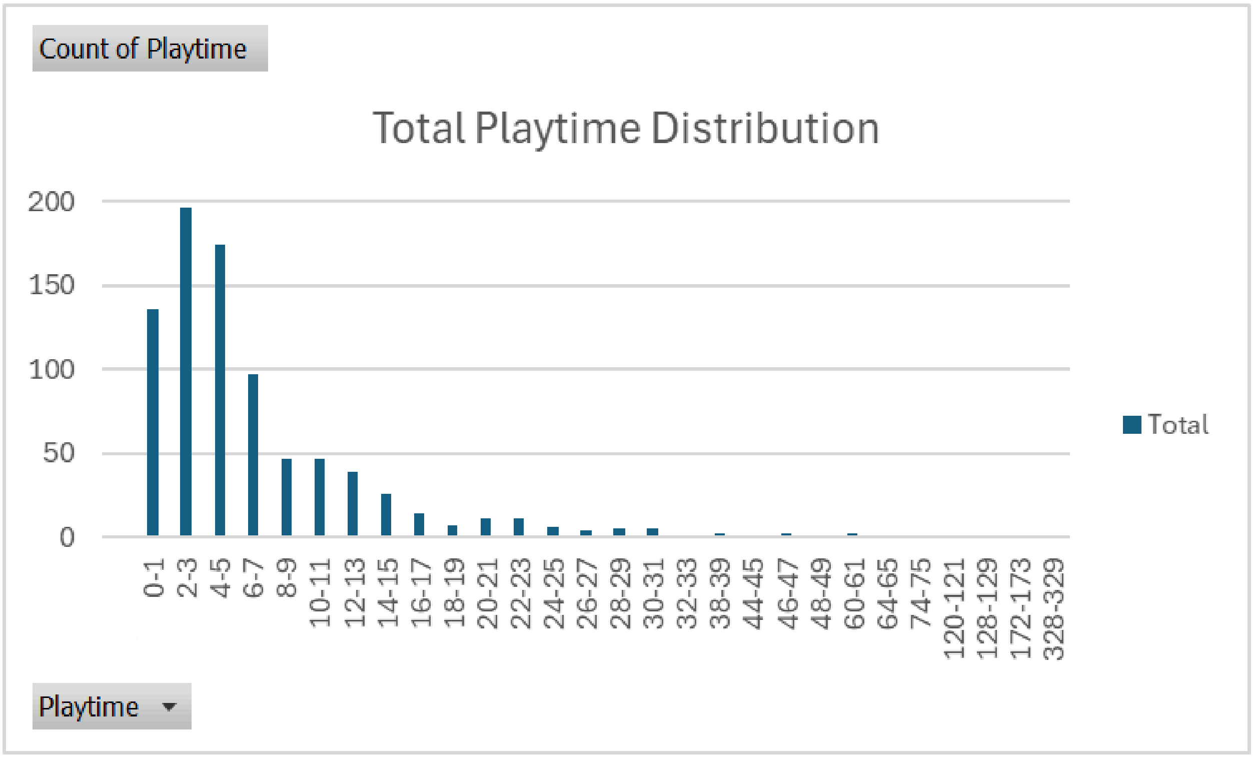


Figure 1 was created in excel after opening the csv as an excel document. It is a PivotTable PivotChart created as lectured. It clearly shows the distribution of the average video game play times. They are clearly skewed toward the front, meaning most games have an average play time of under 15 hours. Most games' play time is between 0 and 9, and play times drop off quickly thereafter.

Figure 2: Metacritic and Playtime Compared by Year

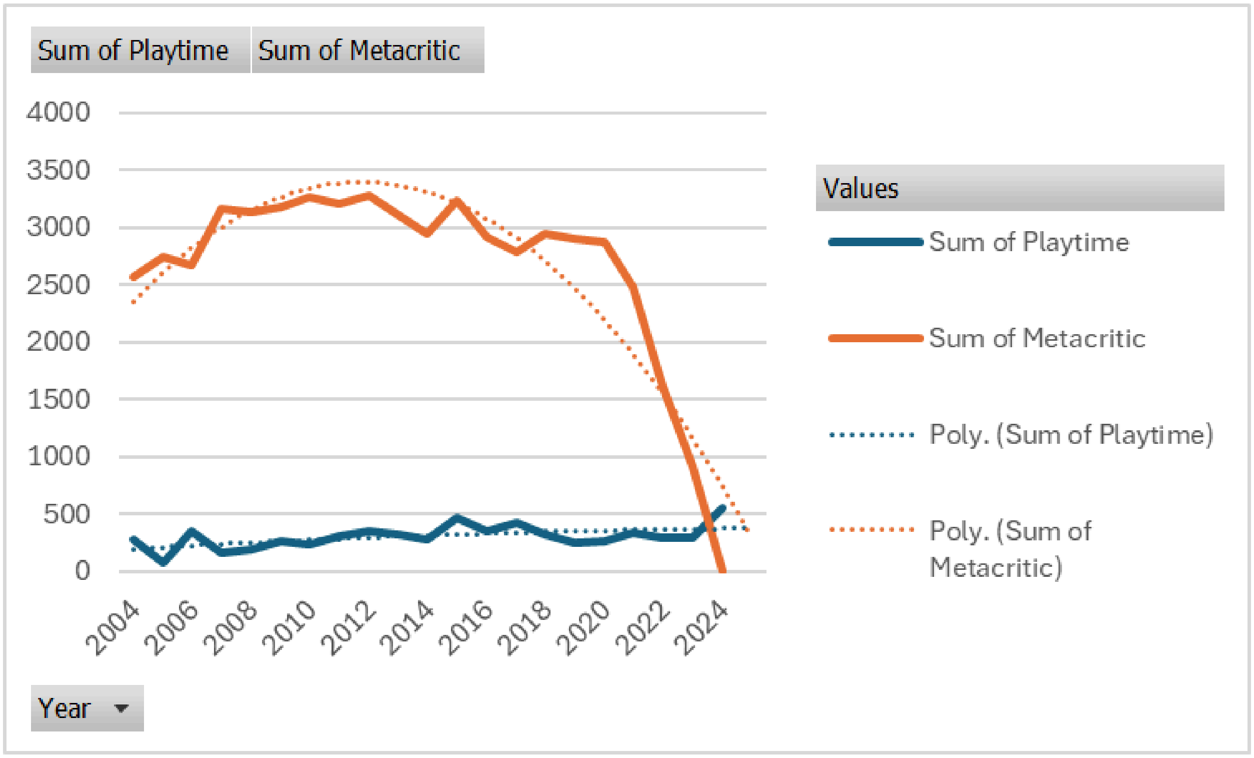


Figure 2 compares each year's sum of Metacritic scores and average playtime. The Metacritic sum hovers primarily between 2500 and 3500, whereas the playtime steadily rises to just below 500 hours. The Metacritic sum has some sort of problem with the data. Looking at it here, it starkly deviates from the average around 2020. This issue has something to do with the API or the accessing of information, as there is no correlation between the games that have 0 metacritic values aside from the fact that they are mostly from recent years.

Plotting each year's playtime and Metacritic sums against one another is both convenient and potentially interesting. It is convenient because both are stored as whole numbers. Furthermore, being able to see the lines against one another allows for easy comparison between trends — do they move distinctly in the same spot? There seems to be a correlation, especially between 2006 and 2018. At both of these endpoints, the sums jet upward, and between them, it stays fairly consistent. This implies that the two factors relate to each other in some way.

Figure 3: Metacritic ESRB Count Per Year

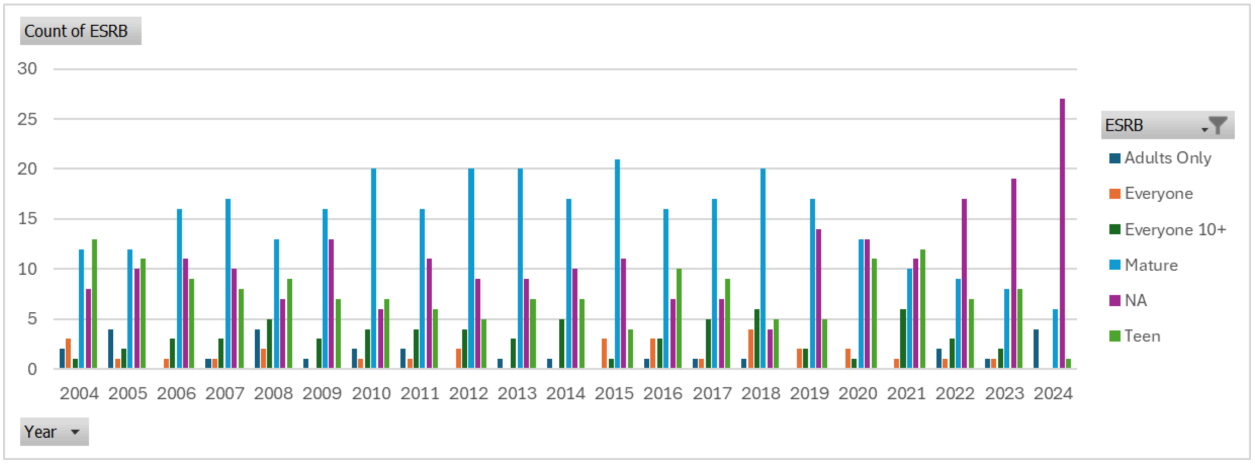


Figure 3 the sum value of each ESRB rating for the top 40 games of each year. Analyzing and visualizing this data gives relevant information about an important metric for labeling games. This information is relevant because comparing it to other variables can help in real world scenarios like an executive making an informed decision on whether or not to push for a different rating.

Figure 3 shows generally that Mature ratings are recently trending downward, whereas NA ratings are trending upward. Therefore, less games are being rated by the ESRB. Because this data set focuses on the 40 most popular games from each year, the missing ESRB ratings imply that the most popular games are not necessarily well-known enough to receive a rating.

Figure 4: Genre Count Per Year

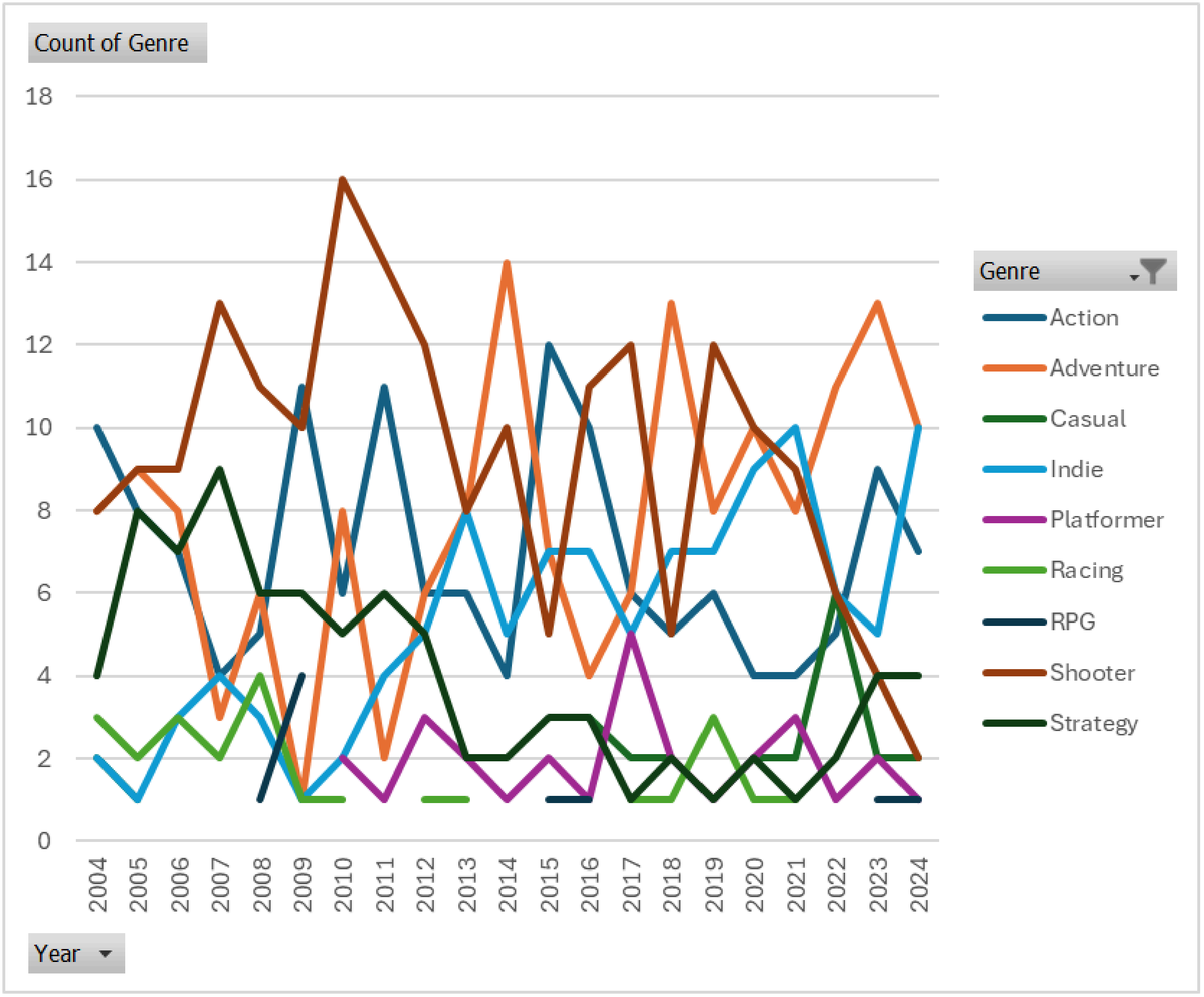


Figure 4 shows the counts of the most popular genres of games in the data set. Shooter and Adventure games are the clear leaders, steadily keeping above a count of 10. Many values are excluded here, as their constant zero values overlapped on the x-axis. Furthermore, indie games are rising, whereas strategy ones are declining.

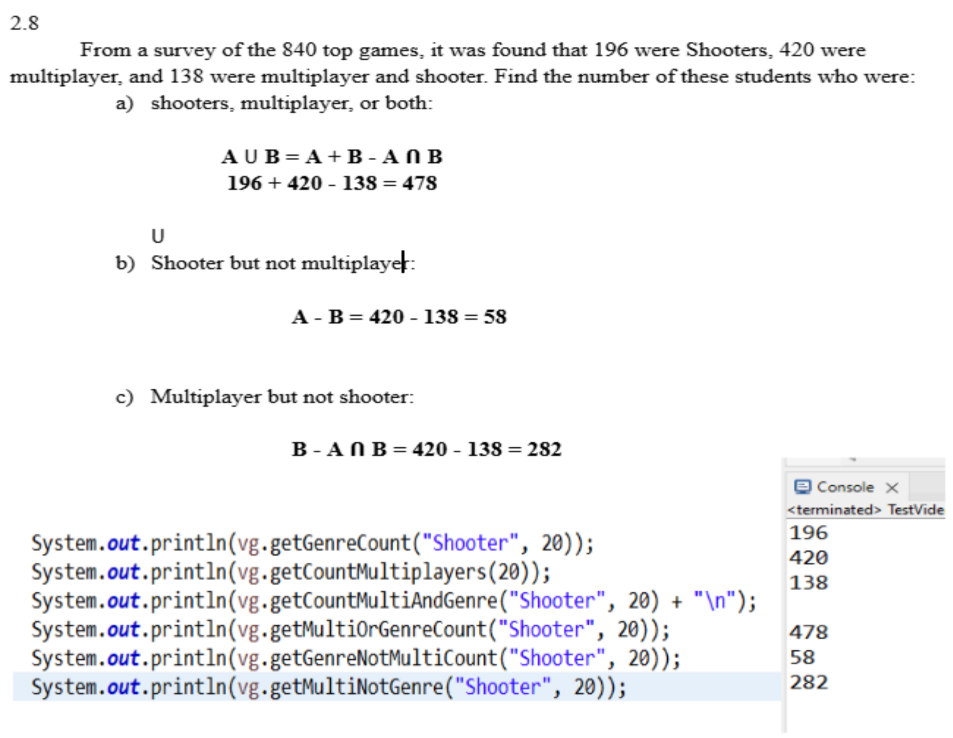
An interesting conclusion comes from comparing Figure 4 to Figure 2. Shooters were the top genre for much of the 2000s and 2010s until they began declining in 2020. At the same spot, Figure 2 shows the count of Metacritic scores plummeting. These two factors in conjunction imply that shooters typically have high Metacritic scores.

Analysis Through Creating Textbook Questions

Hypothetical textbook scenarios model real-life data sets. Although often idealized to teach students concepts effectively, they model real experiments that provide relevant information about their topics. Therefore, looking at the video game data and creating textbook questions based on it is an effective way of analyzing it.

Sample textbook questions were created for the textbook sections relevant to this course. Arbitrary numbers were replaced with numbers accurate to the data set. A Java program reads the data and stores it to use however the problems call for. The figures include the rewritten question, the applicable Java code, and a small screenshot of the output(s).

Figure 5: Textbook Question 2.8 Reimagined



2.8

From a survey of the 840 top games, it was found that 196 were Shooters, 420 were multiplayer, and 138 were multiplayer and shooter. Find the number of these students who were:

a) shooters, multiplayer, or both:

$$\mathbf{A \cup B = A + B - A \cap B}$$
$$\mathbf{196 + 420 - 138 = 478}$$

U

b) Shooter but not multiplayer:

$$\mathbf{A - B = 420 - 138 = 58}$$

c) Multiplayer but not shooter:

$$\mathbf{B - A \cap B = 420 - 138 = 282}$$

```java
System.out.println(vg.getGenreCount("Shooter", 20));
System.out.println(vg.getCountMultiplayers(20));
System.out.println(vg.getCountMultiAndGenre("Shooter", 20) + "\n");
System.out.println(vg.getMultiOrGenreCount("Shooter", 20));
System.out.println(vg.getGenreNotMultiCount("Shooter", 20));
System.out.println(vg.getMultiNotGenre("Shooter", 20));
```

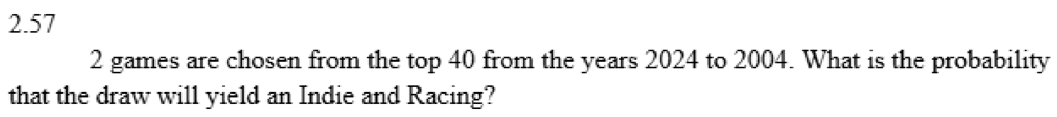
```
196
420
138

478
58
282
```

Figure 5 primarily proves that the textbook logic is correct. For peace and mind, to verify this, this figure calculated a set simple problem using the textbook's logical answer adapted with updated values and by manually counting them in Java.

Figure 5 also shows a correlation between shooter and multiplayer games. Because 138 / 196 shooter games are multiplayers and only 58 / 196 are not multiplayers, shooter games are mostly multiplayer. Furthermore, because 138 / 420 multiplayer games are shooters, multiplayer games are likely to be shooters.

Figure 6: Textbook Question 2.57 Reimagined


2.57

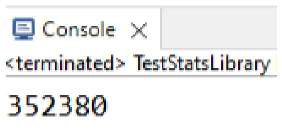
2 games are chosen from the top 40 from the years 2024 to 2004. What is the probability that the draw will yield an Indie and Racing?

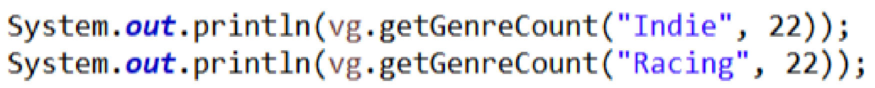
Ways choose 2 from 840: totalCombos = $\binom{840}{2}$

Ways choose 1 Indie: indieCombos = $\binom{111}{1}$

Ways choose 1 Racing: racingCombos = $\binom{28}{1}$

p(1 Indie and 1 Racing): $\frac{indieCombos * racingCombos}{totalCombos} = \frac{3108}{352380} = .0088 = .9\%$


```
Console
<terminated> TestStatsLibrary
352380
```


```
System.out.println(vg.getGenreCount("Indie", 22));
System.out.println(vg.getGenreCount("Racing", 22));
```

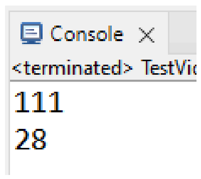
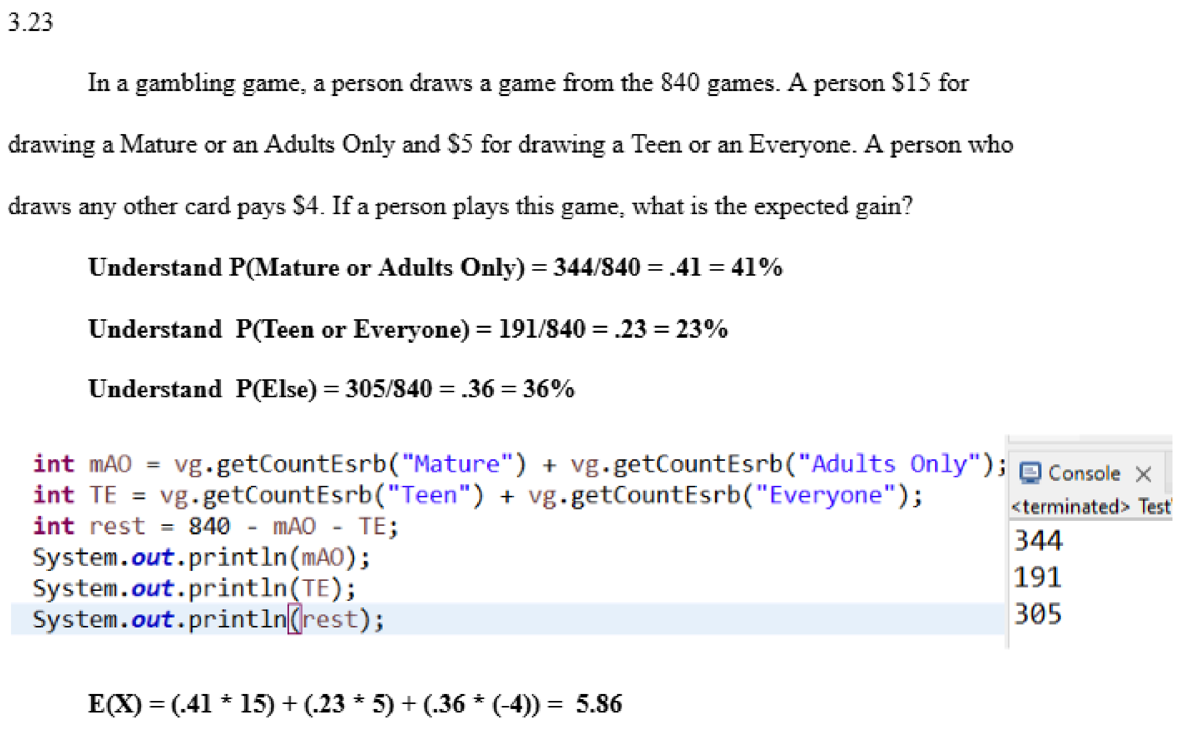

```
Console
<terminated> TestVi
111
28
```

Figure 6 is the first of many problems that test experiments using the data set. This calculates the probability of picking two games of set genres when selecting 2 random games.

The question asks for indie and racing, but it and the code can be modified by just swapping the swings for other genres. It clearly demonstrates that the chances of selecting just one item from each small set are very slim.

The problem inadvertently teaches about the distribution of the counts of genres by year. Creating this problem before the Analysis Through Visualization ones taught that some genres are less represented. Playing around with the Java methods to the counts of other genres further cemented this.

Figure 7: Textbook Question 3.23 Reimagined



3.23

In a gambling game, a person draws a game from the 840 games. A person $15 for drawing a Mature or an Adults Only and $5 for drawing a Teen or an Everyone. A person who draws any other card pays $4. If a person plays this game, what is the expected gain?

Understand P(Mature or Adults Only) = 344/840 = .41 = 41%

Understand P(Teen or Everyone) = 191/840 = .23 = 23%

Understand P(Else) = 305/840 = .36 = 36%

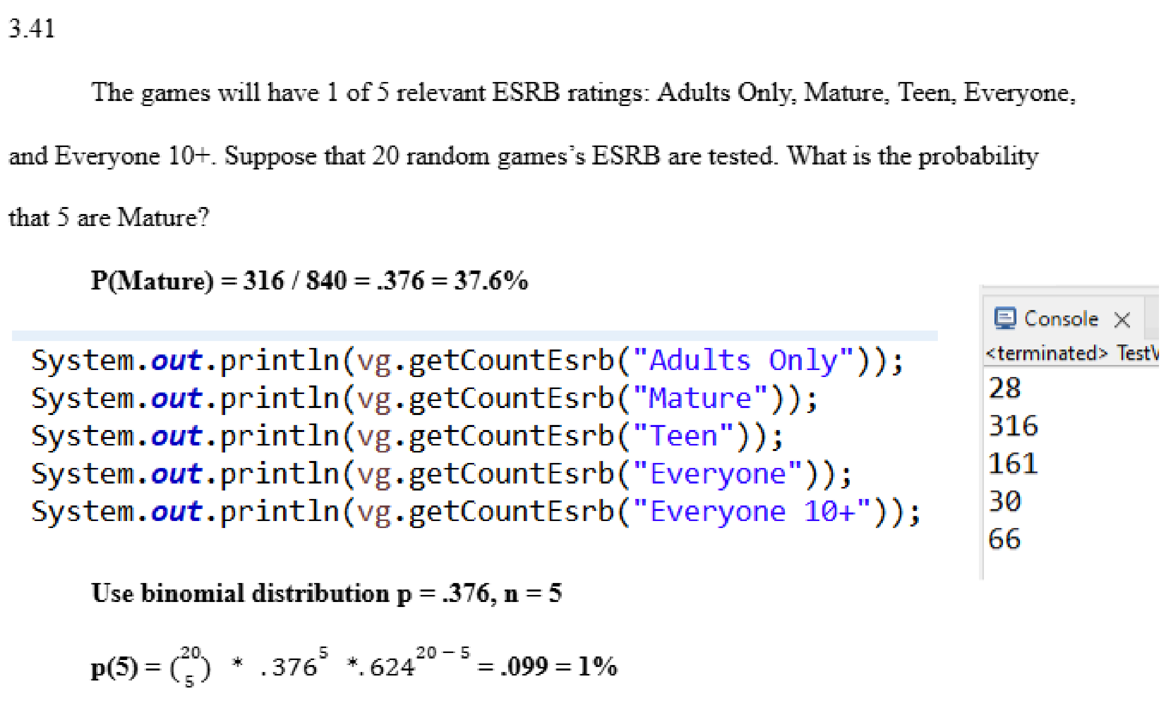
```
int mAO = vg.getCountEsrb("Mature") + vg.getCountEsrb("Adults Only");
int TE = vg.getCountEsrb("Teen") + vg.getCountEsrb("Everyone");
int rest = 840 - mAO - TE;
System.out.println(mAO);
System.out.println(TE);
System.out.println(rest);
```

```
344
191
305
```

E(X) = (.41 * 15) + (.23 * 5) + (.36 * (-4)) = 5.86

Figure 7 teaches about ESRB ratings. It uses the probabilities that games are of a particular rating to generate an example scenario. In doing so, it teaches about the data set's ESRB rating distribution. There are just as many occurrences as games with more mature themes as there are more family-friendly ones. This information is important, as the rating system has implications. For example, the rating may be a key factor in determining how widely a marketing team can push the game. Knowing that games with mature themes occupy most of the most popular games in this data set could help prove that ESRB ratings are not a significant factor in games.

Figure 8: Textbook Question 3.41 Reimagined



3.41

The games will have 1 of 5 relevant ESRB ratings: Adults Only, Mature, Teen, Everyone, and Everyone 10+. Suppose that 20 random games's ESRB are tested. What is the probability that 5 are Mature?

P(Mature) = 316 / 840 = .376 = 37.6%

```
System.out.println(vg.getCountEsrb("Adults Only"));
System.out.println(vg.getCountEsrb("Mature"));
System.out.println(vg.getCountEsrb("Teen"));
System.out.println(vg.getCountEsrb("Everyone"));
System.out.println(vg.getCountEsrb("Everyone 10+"));
```

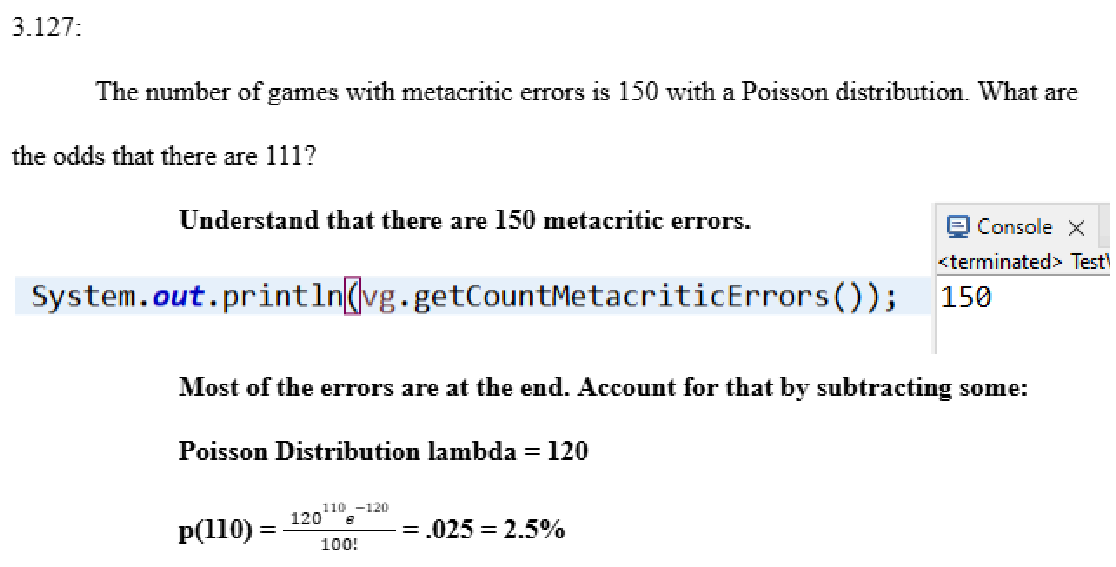
```
Console
<terminated> TestV
28
316
161
30
66
```

Use binomial distribution p = .376, n = 5

$$p(5) = \binom{20}{5} * .376^{5} * .624^{20-5} = .099 = 1\%$$

Figure 8 extends Figure 7's analysis of ESRB ratings. Here, the counts of each of the most relevant scores are listed to get the probability that games are a specific rating. The lesser-used, placeholder, and NA values are not included; they do not significantly impact the data. However, they still impact the total probability or denominator in calculations. Listing the counts of each rating proves that Mature games are the most popular, which is further emphasized by the problem solution that the probability of choosing 5 Mature games from 20 random games is 1%. Only a significant value would answer that high, as their probability for success is lower.

Figure 9: Textbook Question 3.127 Reimagined



3.127:

The number of games with metacritic errors is 150 with a Poisson distribution. What are the odds that there are 111?

Understand that there are 150 metacritic errors.

```
System.out.println(vg.getCountMetacriticErrors());
```

```
Console ×
<terminated> Test\
150
```

Most of the errors are at the end. Account for that by subtracting some:

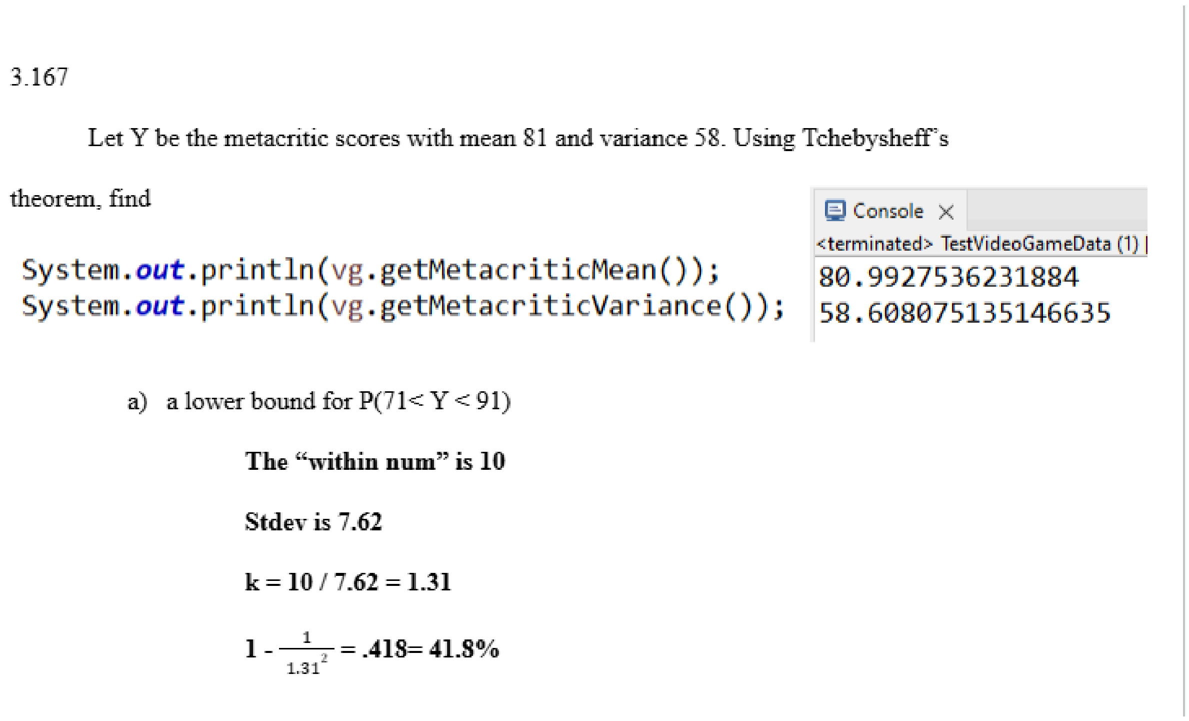
Poisson Distribution lambda = 120

$$p(110) = \frac{120^{110} e^{-120}}{100!} = .025 = 2.5\%$$

This original question focused on errors, inspiring this sample question to be based around an error in the csv file: the metacritic scores of 0. These scores are not 0 unless the game is unrated. There are not that many unrated popular games, and some of the ones with missing data's scores are on the internet. This implies that the error is from the API, as the Python script puts 0 if it does not exist.

Figure 9 demonstrated Poisson distribution with a very large average. The calculation was rough, as the exponential numerator increases much quicker than the factorial denominator. The main takeaway from this figure is that the data is not always perfect and may need manual guidance rather than a rigid problem to extract meaning. For example, here I altered the lambda to make it more fitting.

Figure 10: Textbook Question 3.167 Reimagined



3.167

Let Y be the metacritic scores with mean 81 and variance 58. Using Tchebysheff's theorem, find

```
System.out.println(vg.getMetacriticMean());
System.out.println(vg.getMetacriticVariance());
```

```
80.9927536231884
58.608075135146635
```

a) a lower bound for $P(71 < Y < 91)$

The "within num" is 10

Stdev is 7.62

k = 10 / 7.62 = 1.31

$$1 - \frac{1}{1.31^2} = .418 = 41.8\%$$

Figure 10 calculates the probability that data is within a range using Tchebysheff's theorem. The mean and variance are calculated via Java methods and used for the center number and calculating *k*.

The mean being 81 provides useful quantitative information about the average Metacritic scores. Comparing the final percentage to passing an exam, about half of the games receive passing scores. Given how many values are incorrect, as indicated by the high variance, that percentage is good and is likely higher in reality than this sample.

Conclusions

Overall, data on the top 40 games from each year from 2024 to 2004 has a lot of interesting information. Analyzing it led to many conclusions about both this particular data set and working with data in general.

The Analysis Through Visualization section uses Relative Frequency Histograms and PivotCharts to see trends across video games. Figures 1 and 2 describe the distribution of the games' average playtimes, the low per-year visualization aligning in Figure 2 with Figure 1's focus on just playtime. Figure 2 compares the total Metacritic and total playtime counts for each year; coinciding spikes in their graphs imply a correlation. Figure 3 displays the counts of each relevant ESRB rating as bars, showing that they are highly varied. Figure 4 visualizes the genre count per year as a line, showing that shooter and adventure games are the most popular, that indie games are steadily rising, and that strategy games are declining.

The Analysis Through Creating Textbook Questions section modifies textbook questions to analyze the data. Figure 5 verifies that the textbook logic matches the data's logic and teaches that most shooter games are multiplayer and that about ⅓ of multiplayer games are shooters. Figure 6 demonstrates how little some genres are represented in the data, showing skew. Figure 7 works with the ESRB rating to see what percentage of games may contain sensitive content; it teaches that more likely cases impact final averages. Figure 8 demonstrates the total probability distribution for the ESRB rating using a binomial experiment. Figure 9 highlights the potential errors in data and ways to handle them while exploring Poisson distribution. Figure 10 demonstrates Tchebysheff's theorem and how spread the data on average Metacritic score is.