Ann Arbor District Library: Summer Game Shop Economy

AADL Team: Group 15

Suhas Potluri, Steven Dobrovich, Matthew Fannin

SCHOOL OF INFORMATION
UNIVERSITY OF MICHIGAN

Ann Arbor District Library

Ann Arbor District Library

BSI SI 370 Final Project

SI 370

01 December 2020

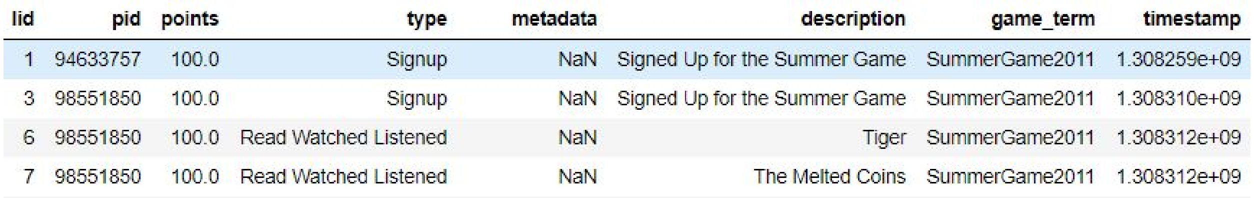
The Ann Arbor District Library (AADL) holds a Summer Game every year from June till August. During the event, participants complete various activities that allow them to score points. These points can be exchanged for prizes of varying degrees. The cost of prizes in the prize pool range from 2,000 to 25,000 points. Using a data set provided by the AADL we were tasked with providing insight into the point reward and prize pool effectiveness of the Summer Games.

The major questions we sought to answer were: what are the average points scored by players during the game, how the points scored compare to the prize pricing, and how players earn points? During this process, we gained insight into the structure of the games that will allow us to make recommendations on the future of AADL's Summer Games.

Data

We were granted access to a dataset that contained the transaction data of every addition of points and deduction of points. The raw data contained 12,806,211 rows and spanned from 2011-2020. This dataset contained 8 columns that can be seen below.

There is a "lid" and "pid" column that represent the player's library ID and player ID respectively. There is a "points" column which gives a numerical value of points being added or subtracted based on the type of event. The "type" column represents the type of event a player participated in. The "metadata" column is an informational column that gives specific information (if applicable) based on the "type" column. The "description" column specifies what a player did to earn points within the "type" column. The 'game_term' and 'timestamp' columns are columns giving more information on the time in which the transactions took place.



lid	pid	points	type	metadata	description	game_term	timestamp
1	94633757	100.0	Signup	NaN	Signed Up for the Summer Game	SummerGame2011	1.308259e+09
3	98551850	100.0	Signup	NaN	Signed Up for the Summer Game	SummerGame2011	1.308310e+09
6	98551850	100.0	Read Watched Listened	NaN	Tiger	SummerGame2011	1.308312e+09
7	98551850	100.0	Read Watched Listened	NaN	The Melted Coins	SummerGame2011	1.308312e+09

Methods

Our initial methodology for preparing the data was to convert it into smaller, workable datasets. Since the AADL performed a point economy rework starting in 2018, it was advised we focus on the data from the 2018-2020 games. We split the data frame using the "game_term" column into three different datasets for each year's games; 2018, 2019, and 2020. Once we had these datasets, we dropped the 'timestamp' and 'lid' columns to help improve performance when running analysis.

After analyzing the workable datasets, we made a discovery that there was a subtype under the 'type' column named 'Geekly Intervention' that appeared to be normal, but realized it acted as a pseudo miscellaneous column. This means some of the data contained outliers that could heavily skew our results, however some of the data was valuable to us. Using the “description” column we determined the descriptions that were most often responsible for outliers and using the Python pandas package, removed those outliers from the dataset.

From there we created two versions of each dataset for each year. One containing only additions of points and a dataframe with only the expenditures of points. Then using a “groupby” function we grouped the “pid” column allowing us to make calculations based on each player. The finished datasets contained players and their points accrued and a list of players and their points spent.

Analysis

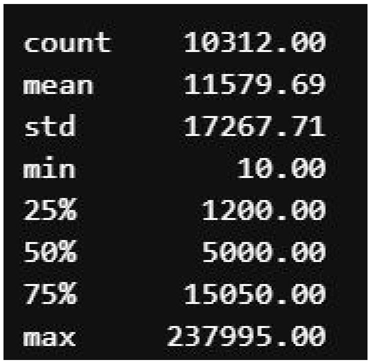
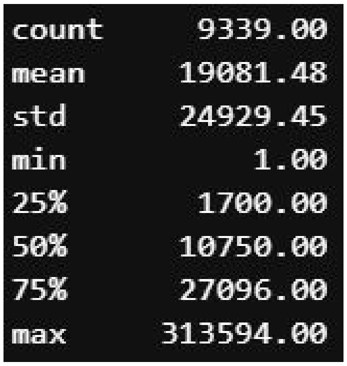
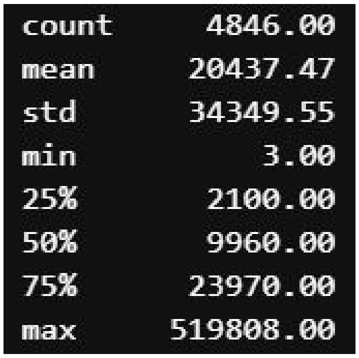
Once we established our workable datasets, we performed cross comparative analysis between each year to determine their similarities and differences. To begin average player scores and participants were obtained from the 2018, 2019, and 2020 games. This was done by using the describe function, and analyzing the results given. Each year was compared against one another to spot trends. From this, results for each year’s games were filtered, to contain values that were larger than 2,000 (smallest prize) and larger than 25,000 (largest prize).

Following a similar methodology, for each year's games, the number of spent and unspent points were calculated. To do this shop orders were stored into a separate data frame, and merged using an inner join with the clean dataframe for each summer to create a new data frame. Both data frames were grouped by the “pid” column. Averages for the spent and unspent points were calculated, along with the difference between the two values.

Lastly, to find the proportion of points that each game type gave out, the data frame was grouped by “type” and the columns were summed up. The total of each type was then divided by the length of the data set to create a proportion, expressed as a percentage. In addition, counting the occurrences of each game type was done in the same manner, counting the occurrences and dividing by the length of the data set.

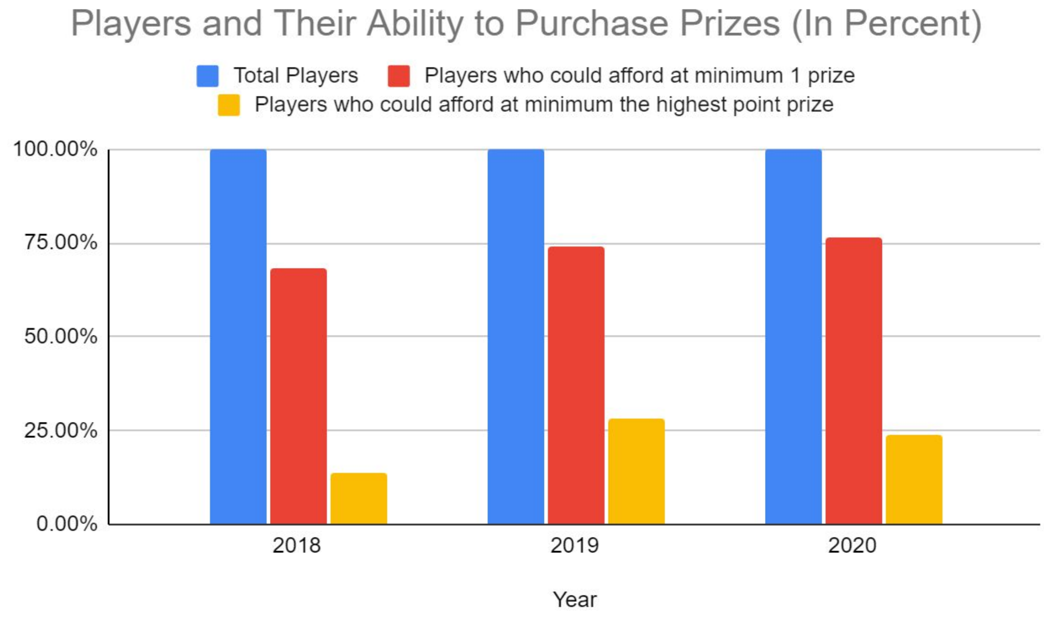
Results

The 2018 Summer Game had 10,312 participants, making it the most of the three years, however, produced the lowest average score of 11,579 points. On the other hand, 2020 had the lowest participation at 4,846 players, but had the highest average score of 20,437 points (keep in mind Covid-19). As the years progressed players tended to score more points. This can be highlighted by not only the maximum score increasing each year, but also the average score. These values can be referenced by year in the following table.

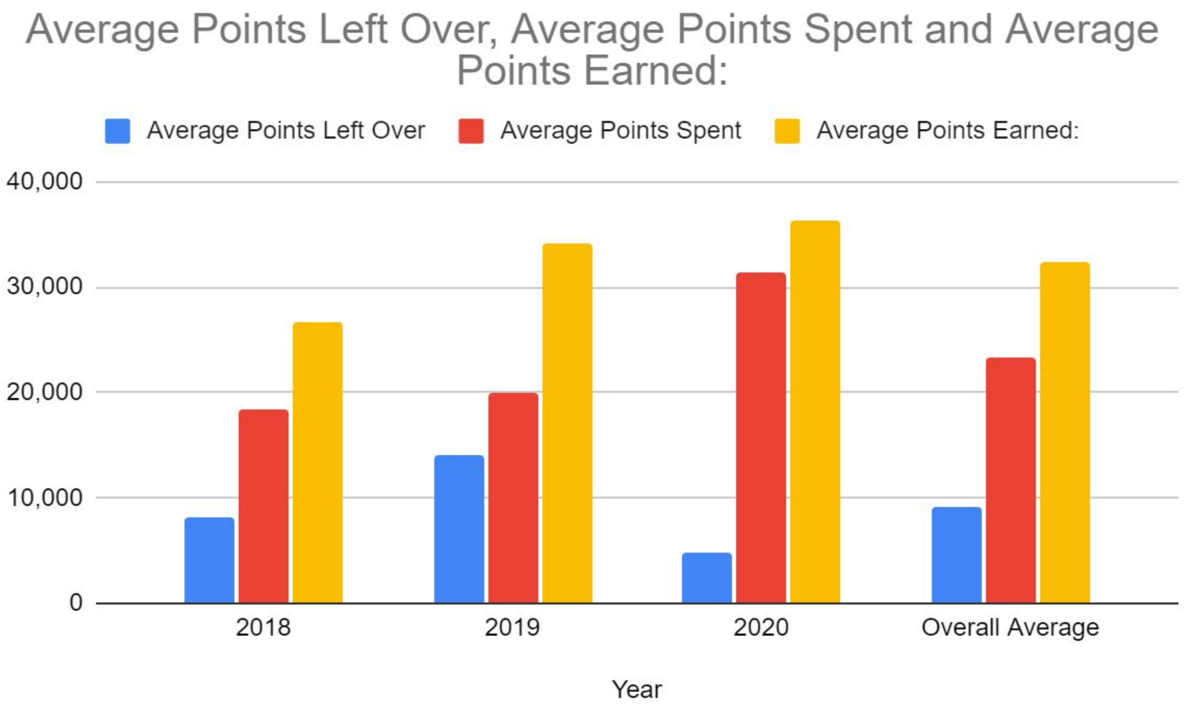
2018	2019	2020
 count 10312.00 mean 11579.69 std 17267.71 min 10.00 25% 1200.00 50% 5000.00 75% 15050.00 max 237995.00	 count 9339.00 mean 19081.48 std 24929.45 min 1.00 25% 1700.00 50% 10750.00 75% 27096.00 max 313594.00	 count 4846.00 mean 20437.47 std 34349.55 min 3.00 25% 2100.00 50% 9960.00 75% 23970.00 max 519808.00

The table above showcases how many players (count) were in each Summer Game, by year. The (mean) is the average number of points players scored in the game. In addition, (std) represents the standard deviation in points. The (min) represents the smallest value of points a player earned and (max) is the highest number of points earned by a player. Lastly, the range (25%, 50%, 75%) values represent the stated percentile value.

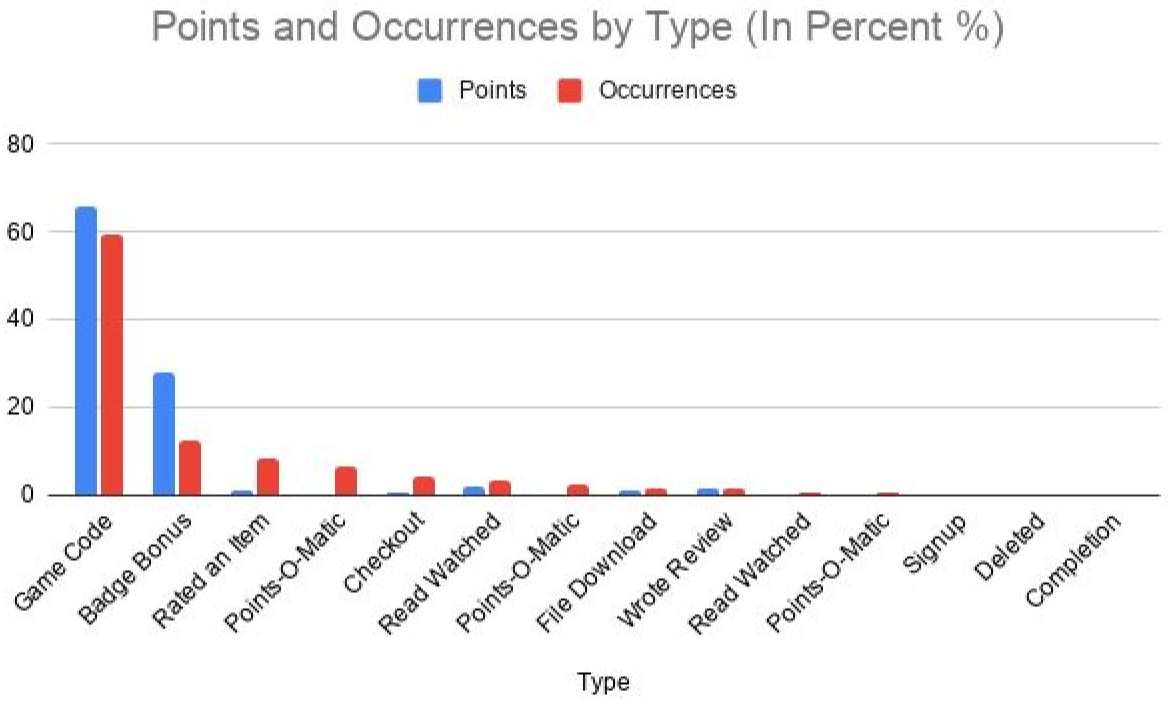
During each year, we analyzed how many players could afford the smallest prize (2,000 points) and the highest prize (25,000 points). The statistics for players that can purchase the smallest prize come from figure 1-A, and statistics for players that can purchase the largest prize can be found in figure 1-B, in the appendices. The graph below was created from the data in figure 1-C, which is a combination of figures 1-A and 1-B. We found that roughly 73% of players earn enough points to purchase the smallest prize, while about 22% of players can purchase the highest prize.



A trend emerges that the average points earned and the average points spent per year increase. In figure-2, the number of points earned in 2018 is 26,595 and jumps to 36,323 in 2020. In 2018, players on average spent 18,335 points and in 2020 spent on average 31,481 points. 2019 had, on average, the highest points left unspent at 14,162. The graph below shows how many points, on average, are left over, spent, and earned, with the average of the three years shown as "Overall Average."



When studying how points were earned, we analyzed what percent of points came from a given subtype in the "type" column and the frequency of that subtype occurring in a transaction. It became clear that points were earned from two major subtypes: "Game Code" and "Badge Bonus." Based on the table in figure-3, "Game Code" makes up 65.83% of the total points earned in the games, while only accounting for 59.25% of transactions in the dataset. "Badge Bonus" makes up 27.93% of points earned, and accounts for 12.13% of transactions. In all other subtypes in the "type" column, there was a higher proportion of occurrences than of points scored. The chart below illustrates the discrepancies in how points are earned.



We decided to dive deeper into the “Game Code” and “Badge Bonus” types. Game code occurs more than two million times, and produces 103 points on average. The smallest value of points earned for a “Game Code” is 1 and the largest is 50,000. There is a large variance for values in “Game Code” because the standard deviation is 237. The statistics for the “Game Code” can be found in figure 4 in the appendices. The ten most common game codes that occur in the dataset can be found in figure 4-B. “STAYCATION” occurs the most often, it appears 5,389 times.

Next, taking a look at “Badge Bonus”, this type occurred more than 500,000 times in the dataset. This type produces a larger score on average (213), than “Game Code.” Badge Bonuses have a smaller range of points than “Game Code”, ranging from 40 to 2,000. The statistics can be found in figure 4-C.

Challenges/Limitations

Although the goals and questions were well-defined, during our initial analysis of the dataset we noticed a specific category in the “type” column called “Geekly Intervention” which appeared to act as a pseudo miscellaneous column. Some of this data proved useful, but not all, which caused challenges in determining which areas of “Geekly Intervention” were skewing the data, and which areas were helpful to our understanding.

Another limitation we encountered was the lack of detail regarding prizes. We knew when points were being spent, but we didn’t know what prize was being purchased. This prevented us from attempting to analyze why there were unspent points at the end of each Summer Game.

Recommendations For Future Work

To begin, “Shop Refunds” should be added into the “Shop Order” subtype in the “type” column. Along with categorizing other functions such as “Point Transfers” into its own subtype, would allow for more accessible data. Adding the prize data into the dataset could allow for deeper analysis in the future on popularity of prizes. In addition, adding the price for each prize would also allow for analysis on the cost of an item versus its point cost. This could help understand why there is an overall average of 9,081 points unspent each year by adding the ability to gauge popularity of prizes.

It could also be valuable to add a difficulty review when someone earns points, to have an average of how difficult and how time consuming an activity was. A survey could be launched, asking for a 1-10 rating on the difficulty and time spent to complete a game.This would allow a closer analysis of points given out to difficulty, which could assist in tailoring a point strategy more effectively.

When looking specifically at how many players earn points, the general trend over the course of the 3 years is 73% of players will be able to earn at least a single prize, and roughly 22% of the participants could earn at least a single highest tier prize. To our team, this seemed like a fair spread of point distribution. It allows for accessibility into the Summer Game for everyone, while also showing the incentive program (points for prizes) for engagement is effective.

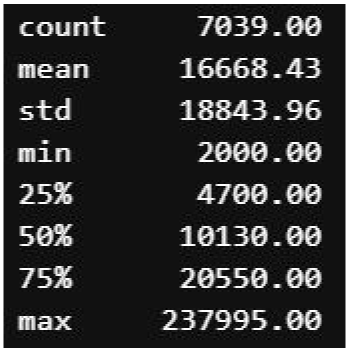
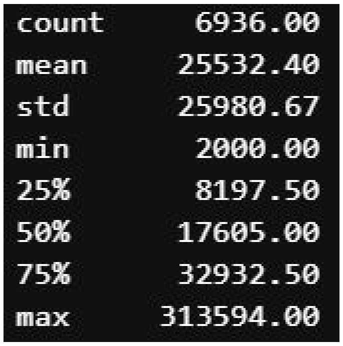
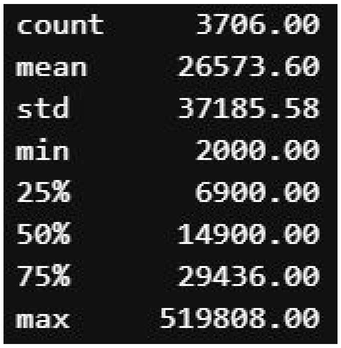
One major suggestion we have, is to slightly rework the “Game Code” subtype. The lowest reward from this subtype is 1 point, while the 25th percentile of rewards offers on average 25 points, and the 50th and 75th percentile of points given out averages 50 points. However, the maximum score of a ‘Game Code’ is 50,000 points. This immediately allows a player to purchase two of the highest point prizes and heavily skews the point distribution. It could be worth lowering that value to 25,000 to allow only a single highest point prize.

One final note is to be aware of the trend over the course of the 2018, 2019, and 2020 Summer Games, that players on average are earning more points each year than the year before. Although this isn’t negatively affecting the game, and due to Covid-19 the numbers could be skewed, this should be monitored going into the 2021 Summer Game to make sure players aren’t learning how to “game” the system.

Appendix

Figure 1-A

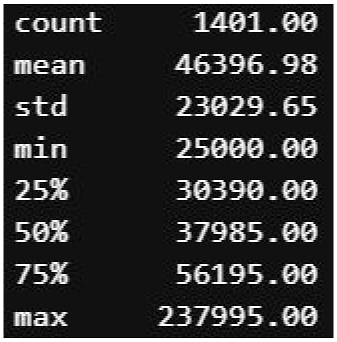
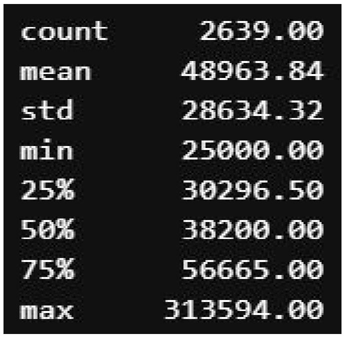
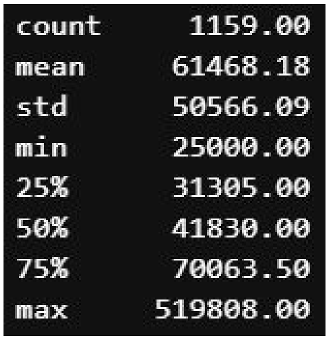
Players Earning Smallest Prize (2,000)- player scores are at least 2,000

2018	**2019**	**2020**
 count 7039.00 mean 16668.43 std 18843.96 min 2000.00 25% 4700.00 50% 10130.00 75% 20550.00 max 237995.00	 count 6936.00 mean 25532.40 std 25980.67 min 2000.00 25% 8197.50 50% 17605.00 75% 32932.50 max 313594.00	 count 3706.00 mean 26573.60 std 37185.58 min 2000.00 25% 6900.00 50% 14900.00 75% 29436.00 max 519808.00
7,039 / 10,312 = 68.26%	6,936 / 9,339 = 74.30%	3,706 / 4,846 = 76.48%
Total Average:	73.0%	

Count is the number of players that fell within the given threshold. Total Average is the overall average of 2018, 2019, and 2020

Figure 1-B

Players Earning Largest Prize (25,000) - player scores are at least 25,000

2018	**2019**	**2020**
 count 1401.00 mean 46396.98 std 23029.65 min 25000.00 25% 30390.00 50% 37985.00 75% 56195.00 max 237995.00	 count 2639.00 mean 48963.84 std 28634.32 min 25000.00 25% 30296.50 50% 38200.00 75% 56665.00 max 313594.00	 count 1159.00 mean 61468.18 std 50566.09 min 25000.00 25% 31305.00 50% 41830.00 75% 70063.50 max 519808.00
1,401 / 10,312 = 13.59%	2,639 / 9,339 = 28.26%	1,159 / 4,846 = 23.91%
Total Average:	21.92%	

Count is the number of players that fell within the given threshold. Total Average is the overall average of 2018, 2019, and 2020

Figure 1-C

Number of Players that could afford prizes per year

Year	**Total Players**	**Players who could afford at minimum 1 prize**	**Players who could afford at minimum the highest point prize**
2018	**10,312**	**7,039**	**1,401**
2019	**9,339**	**6,936**	**2,639**
2020	**4,846**	**3,706**	**1,159**

Total players column contains the total number of players in that year.

Figure-2

Year	Average Points Left Over	Average Points Spent	Average Points Earned:
2018	8,240	18,335	26,595
2019	14,162	19,965	34,127
2020	4,841	31,481	36,323
Overall Average	9,081	23,260	32,348

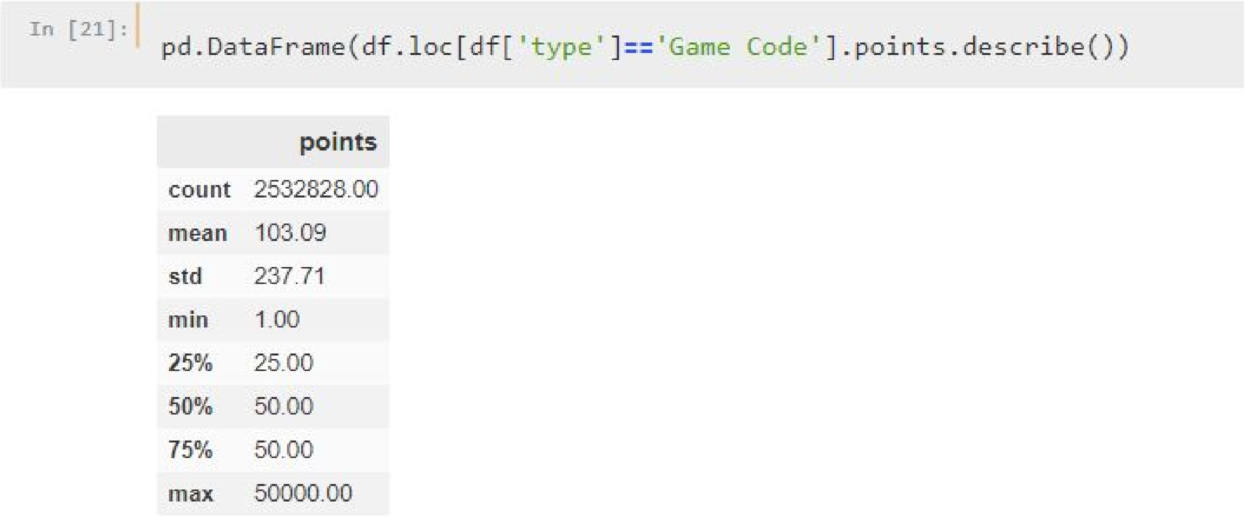
Average points left over is the average points left unspent for that year. Average points spent is the average points that players used to purchase prizes. Average Points Earned is the average amount of points a player earned in the given year.

Figure-3

Type of Activity	**Points (%)**	**Occurrences (%)**
Game Code	**65.83%**	**59.25%**
Badge Bonus	**27.93%**	**12.13%**
Rated an Item	**.88%**	**8.15%**
Points-O-Matic Review Reviewer Bonus	**.07%**	**6.56%**
Checkout History	**.70%**	**4.14%**
Read Watched Listened Daily Bonus	**1.69%**	**3.14%**
Points-O-Matic News Sprinter Bonus	**.16%**	**2.49%**
File Download	**.84%**	**1.57%**
Wrote Review	**1.56%**	**1.41%**
Read Watched Listened	**0.00%**	**.55%**
Points-O-Matic Super Serializer Bonus	**0.00%**	**.43%**
Signup	**0.13%**	**.12%**
Deleted Review	**0.00%**	**.02%**
Completion Bonus	**.20%**	**.02%**

The Points(%) column is the proportion of total points earned in that column over the 3 year time period. The Occurences(%) column is the proportion of total transactions over the 3 years time period. In addition, values that appear to 0, are created due to rounding to two decimal places.

Figure 4-A

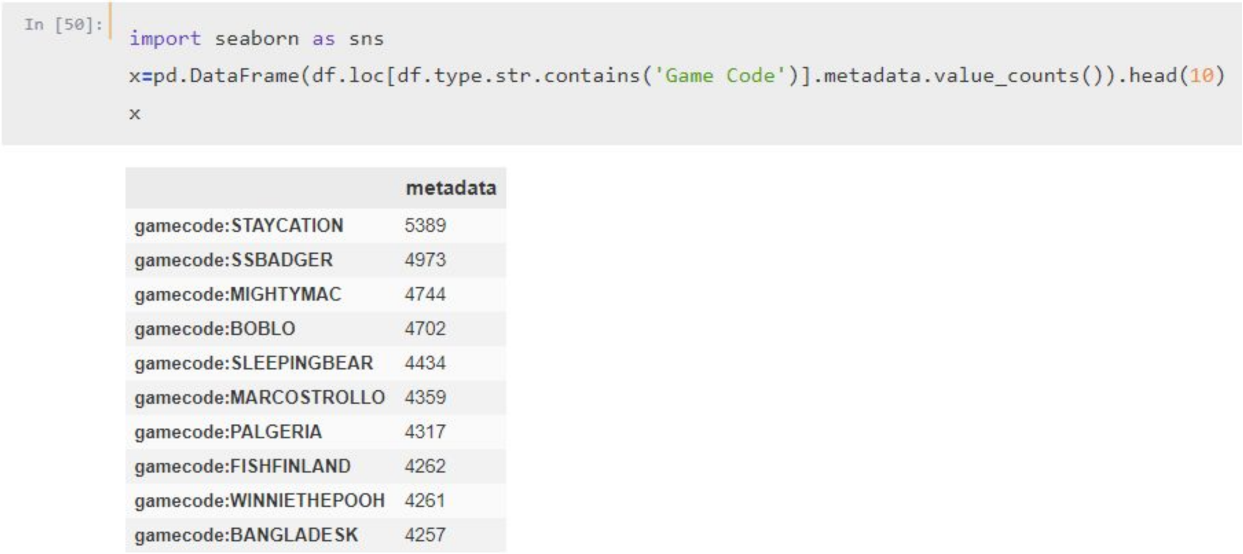


```
In [21]: pd.DataFrame(df.loc[df['type']=='Game Code'].points.describe())
```

	points
count	2532828.00
mean	103.09
std	237.71
min	1.00
25%	25.00
50%	50.00
75%	50.00
max	50000.00

Code segment for the "Game Code" type, (count) represents the number of occurrences of "Game Code" in the dataset. The (mean) is the average number of points game codes give. In addition, (std) represents the standard deviation in points. The (min) represents the smallest value of points a player received and (max) is the highest number of points received. Lastly, the range (25%, 50%, 75%) values represent the stated percentile value.

Figure 4-B

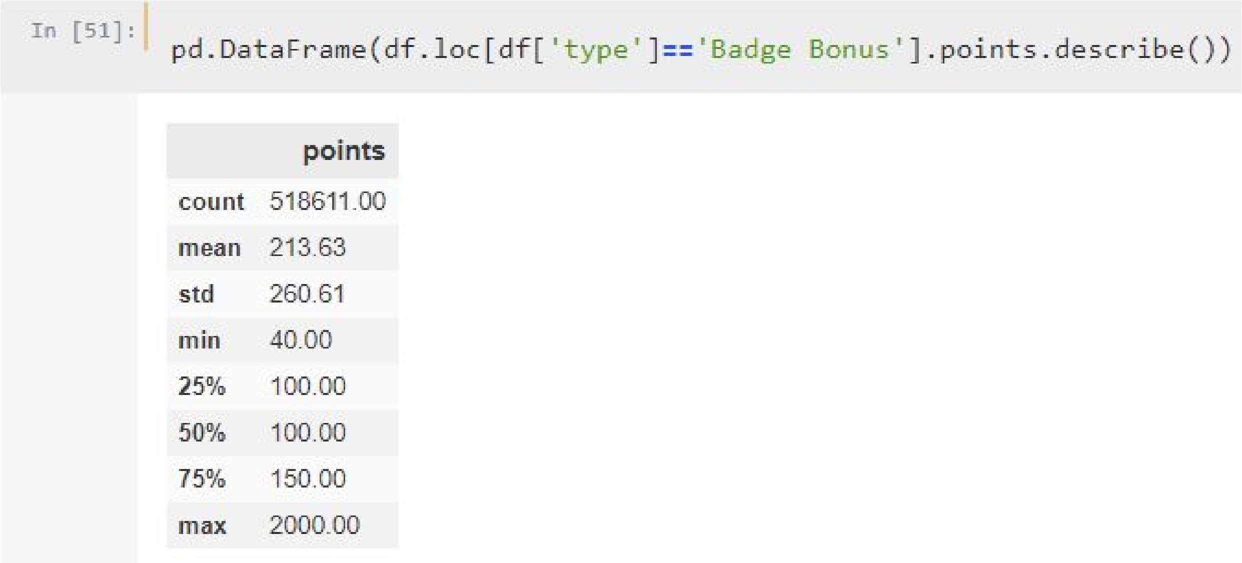


```
In [50]: import seaborn as sns
         x=pd.DataFrame(df.loc[df.type.str.contains('Game Code')].metadata.value_counts()).head(10)
         x
```

	metadata
gamecode:STAYCATION	5389
gamecode:SSBADGER	4973
gamecode:MIGHTYMAC	4744
gamecode:BOBLO	4702
gamecode:SLEEPINGBEAR	4434
gamecode:MARCOSTROLLO	4359
gamecode:PALGERIA	4317
gamecode:FISHFINLAND	4262
gamecode:WINNIETHEPOOH	4261
gamecode:BANGLADESK	4257

Figure contains the ten most frequently occurring game codes. "Metadata" refers to the count of occurrences in the dataset.

Figure 4-C



```
In [51]: pd.DataFrame(df.loc[df['type']=='Badge Bonus'].points.describe())
```

```
          points
count  518611.00
mean      213.63
std       260.61
min        40.00
25%       100.00
50%       100.00
75%       150.00
max      2000.00
```

Code segment for the “Badge Bonus” type, (count) represents the number of occurrences of “Badge Bonus” in the dataset. The (mean) is the average number of points a badge bonus gives. In addition, (std) represents the standard deviation in points. The (min) represents the smallest value of points a player can receive and (max) is the highest number of points received. Lastly, the range (25%, 50%, 75%) values represent the stated percentile value.