Understanding the Demographics of Slot Machine Play in Las Vegas and Across the Country

"What is life if not a gamble?"

F.E. Higgins

Harvard University

Abstract

Harvard University School of Engineering and Applied Sciences
Department of Applied Mathematics

by Jack Heavey

In this paper, I look at the demographics of slot machine play across 31 different casinos in order to try to work out the relationships between location, age, frequency of play, and overall win to the casino. This analysis looks at all of these variables in tandem in order to try to understand the shifting demographics of slot machine play and why slot machine revenue has been declining as a fraction of total revenue within Las Vegas. Using the different preferences in different regions, I make recommendations on areas of improvement in order to capture the younger demographic, a key subsection that has seen declining slot play over time.

Acknowledgements

I would like to thank Caesars Entertainment for their generosity in providing data for this analysis. Chief Analytics Officer Gene Lee, Vice President of Gaming Analytics Joshua Frank, and Senior Data Scientist Thompson Nguyen have been invaluable in providing both the data for this project and answering miscellaneous questions throughout the writing of this paper.

Additionally, I would like to thank Professor Pavlos Protopapas for overseeing the writing of this paper and steering me in the right direction on both the analyses and the writing itself.

Contents

Chapter 1

Introduction

The new generation of visitors to Las Vegas is changing greatly, with individuals coming for unique and fun experiences rather than to simply gamble [1]. And while this presents new and exciting opportunities for diversification within the entertainment industry, it also presents challenges. Capturing the gaming segment of this new community and building brand loyalty is going to determine the success of casinos not only in Las Vegas but across the country.

Despite the visitor profile in Las Vegas being notably younger and more diverse than other casinos across the country [2], the resulting demographics found in slot machines are much more similar to what is seen across the country, with the average slots player being much older and more likely to be white than the overall visiting populations. These differences are a problem that need to be reconciled moving in to the future.

1.1 Background Information

Gaming markets are broken down between local markets, which constitute small casinos and gambling halls, regional markets, that include larger resorts and hotels/casino combinations, and destination markets, which people will travel to from a large distance away. Casinos within these markets will share similar visitor profiles and data breakdowns. The University of Nevada, Las Vegas (UNLV) Center for Gaming Research compiles governmental data for gaming in Nevada as well as other states with legalized gambling or Indian Casinos that report to the state governments. Over the time period that our data covers, slot machine revenue accounts for approximately 50% of all gaming revenues on the Las Vegas Strip[3], a much lower percentage of revenue than we see in other markets across the country. Even Atlantic City, another destination market, has slot machines make up almost 75% of gaming revenue [4], with local and regional markets having a similar slot breakdown. These differences are something all casino owners are trying to understand, and if the revenue breakdown of slots can be increased without affecting other revenue measures, casinos stand to profit greatly.

Caesars Entertainment is an international gaming corporation based out of Las Vegas, Nevada, that owns more than 40 properties across the United States and the world. They are the fifth largest gaming company by market capitalization based out of the United States and operate more than 40 companies in the United States and abroad. Historically, a vast majority of their gaming revenue has come from slot machines, and while this is still true, the share has decreased greatly. While some volatility can be attributed to the random nature of slot machines, the consistent decline is a major worry to both the gaming industry and the communities built around these industries, such as Las Vegas and Atlantic City. The principal culprit of this decrease is the changing demographics of slot machine play; the average user's age has increased almost linearly year over year, an unsustainable trend in the long run.

Caesars keeps track of slot machine play, and other activity within their properties using their rewards program, Total Rewards. In exchange for increasing levels of rewards, Caesars gets data in order to analyze the micro- and macro-trends within their customer base, similar to other rewards programs. Using this rewards program, they have accumulated an extensive database of slot machine play from their customers. The data set involving all slot machine play from rewards members from May 2, 2017, to January 31, 2018, contains 142,880,287 unique entries to look at. Within this data set, I hope to identify factors that increase levels of slot play from populations that have historically had much lower levels, namely those under the age of 50.

Within this work I will present both methods of analysis and the findings that these methods produced. This includes looking at the proportional population size of different age brackets and the relative bet size attached to these individuals, strategies that different age groups participate in while playing, and the differences within groups across different regions and types of markets.

This chapter will cover all necessary background information, using both industry information and information within the data set used. Chapter two will begin to explore the findings within the data set, and chapter three will contain conclusions and future avenues to explore.

1.1.1 The Data

Caesars has provided slot machine information about each individual play session at a machine and demographic information about the players. The parts of the data set used are described below:

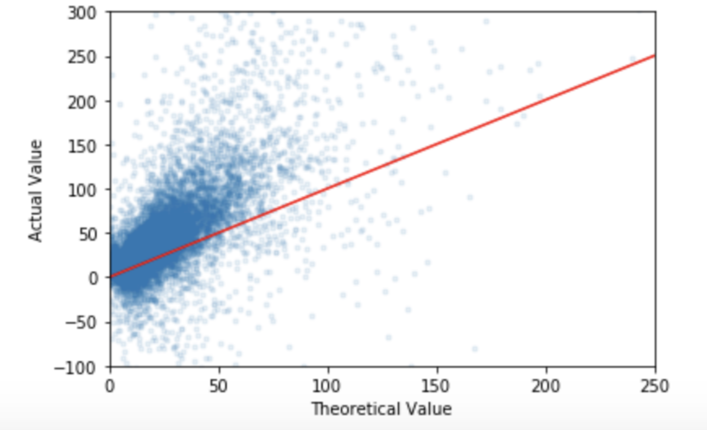
- i_dmid — a unique, scrambled identifier to denote individuals.
- d_rating_date — the date at which the slot machine session occurs.
- c_prop_cd — the property where the play occurs. There are 36 different properties in the data set
- f_coin_out — amount of money distributed by the machine at the end of a session
- f_coin_in — amount of money put in to the machine at the start of a session
- f_theo_win — theoretical value to the casino of a session of play. Explained further below.
- f_play_time — the number of minutes that the session of play lasted.
- f_winloss — the difference between coin-in and coin-out, representing the results of the play session.
- c_game_pref — a string denoting the individual's most played type of casino game. E.g. table games, video poker, slot machines, etc.
- c_frequency — how often an individual plays. The possible values are weekly, monthly, or infrequent.
- i_age — the age of the individual playing

The theoretical value is the general metric used to measure profit, rather than win/loss. This metric measures the expected value, taking the starting fee to play and subtracting the sum of all ending states multiplied by the payout of said states. This constant can be represented formally as the following mathematical formula for every machine:

$$C - \sum_{i \in Result} p_i x_i$$

This theoretical value attempts to remove the randomness from slot machine play within the slot machines, as all of the probabilities and payouts are fixed. The actual income value can be found by taking the difference between the coin-in and coin-out values.

Caesars operates with the assumption that the total actual value is expected to converge to the total theoretical value. However, due to the randomness from the machines as well as differences in individual preferences, very rarely does theoretical value properly indicate the true amount of money made from an individual. Usually, theoretical value actually understates the true amount of revenue from an individual, as theoretical value is taking into account low probability/high payout events. People will often lose before these events occur, meaning actual win/loss is much lower than the theoretical value that takes these unlikely events into account.



We can see within the above plot that the relationship between theoretical value, plotted along the x coordinates, and actual value, plotted along the y coordinates, shows that actual win/loss value is higher than theoretical value, although there is a high degree of randomness within the entire data set. Throughout the paper, we will refer to the theoretical value as theo.

1.1.2 Cleaning the data

The full data set was much too large to sift through on a single machine and run code on due to limited computing resources. As such, I broke down the full data set in to smaller data subsets that would then be used for their relevant analyses. I utilized the Google Cloud platform, specifically the cloud storage for moving files between machines and Google BigQuery in order to filter data in to smaller, workable data sets. Using SQL-like commands, I was able to extract the relevant data from the full data set, looking at entries by property, by region, etc.

1.1.3 Missingness and Incomplete Data

The data provided by Caesars Entertainment is extremely extensive and complete, but it does not tell the whole story. From their own internal estimates, Caesars estimates that approximately $\frac{2}{3}$ of all play comes from Total Rewards members, although this fraction is not evenly distributed across all different locations and markets. The Las Vegas region, for example, only contains 45,785 unique players compared to more than 300,000 in Atlantic City over the same time period. Actual visitation rates, however, show that Las Vegas had much higher rates of visitation over the time period of interest. This shows the stark difference between irregular tourists, who are less likely to be Total Rewards members, and regulars, who are rewards members.

The data from Total Rewards also trends older than data according to the Las Vegas visitor profile, with the average age of slot machine players being 55.5 in the Caesars database compared to 44.0 in the visitor profile [2]. This is much more in line with what we would expect from regional or local markets. For example, the average age of casino visitors in the west pod region is 58.61.

Unfortunately, there is no way to reconcile the differences between the fraction of play by Total Rewards members and the fraction of play from people not within the rewards database. This is because the statistics for these players cannot be tracked from person to person. Without a more complete database, it will be hard to draw conclusions for people outside of the Total Rewards program, as all conclusions are based on them. And while this is slightly disappointing, the Rewards members are a much higher and more important fraction of income, as they are the people considered regular customers who the casinos want to keep and develop a relationship with.

With these caveats in mind, let's jump in to the data.

Chapter 2

Findings

Within the data set, there are a multitude of possible conclusions to draw about individual actor's actions and characteristics, based on things such as location, age, theoretical value, etc. This chapter will section off conclusions based on the main independent variable that is being used to draw conclusions.

2.0.1 Breaking down properties by region

We can divide the full data set into groups of properties, with each group having similar demographics, and regional preferences. It can be useful to look for conclusions within these smaller subsets in order to try to understand differences in the customer base across the country. Looking at region instead of by individual casino can also help to remove some differences between individual properties caused by random variation, especially in the destination markets of Las Vegas and Atlantic City. In 2016, the mean number of different casinos that an individual gambled at in Las Vegas was 2.0, and they visited 6.3 different casinos on their trip [2]. Thus, looking holistically at an entire region makes sense. The different regions that we will look at throughout this paper are as follows:

1. Las Vegas — nine different properties located on or very near to the Las Vegas Strip. Considered a destination market.
2. West Pod — four different casinos located in Nevada but outside of Las Vegas.
3. Atlantic City — three different properties located in Atlantic City, New Jersey. Considered a destination market.
4. Cut Pod — four different properties in the middle-east coast. Two properties are in inner-state Mississippi and two in Pennsylvania.

5. Heartland — three properties operated in the US Heartland, near the great plains. These properties are in Iowa and Missouri.

6. North Pod — four properties found in Indiana and Illinois.

7. South Pod — four properties found along the Gulf Coast, in Louisiana and Mississippi.

Caesars breaks up their revenue reporting into these groups due to the correlation that casinos within the same group share. For regional breakdowns, we will refer to these terms.

2.1 Frequency

Caesars breaks up their customers by frequency in to three different categories: weekly customers, monthly customers, or infrequent customers, with these categories being based on the previous year of play and the approximate number of days a customer has played. If we take these categories at face value, we would expect weekly customers to play approximately once a week on average, monthly customers once a month, and infrequent customers to be less than this. These categories refer to trips instead of days, however, which can last multiple days. On average through the nine months that we are looking at, weekly customers had 24.04 days where they played, monthly customers had 12.30 days with casino play, and infrequent customers had 3.55 days of play.

These overall averages are not consistent across different regions, however. The destination markets of Atlantic City and Las Vegas see these numbers drop precipitously, with the average weekly customer in Las Vegas playing only 2.44 days over our time period and the average monthly customer playing 3.83 days. This supports the idea that Las Vegas is not necessarily built on "regulars," rather on tourists who are visiting for a special occasion. Visitation statistics to Las Vegas say that, in 2016, 53% of repeat visitors made only one visit to Las Vegas that year, with the mean value being two visits to Las Vegas [2], and so individuals are not visiting nearly enough to build up a 'Weekly' or 'Monthly' frequency in Las Vegas alone.

If we pick out people who have either the "Weekly" or "Monthly" frequencies and have been to at least one Las Vegas property, we see that the average number of properties that person has visited is 1.48. This number, despite being lower than the average visitation statistics across the whole city [2], is higher than similar values for other regions, which vary from 1.13 to 1.31. This supports the idea that many of the people that have "Weekly" or "Monthly" frequencies are not locals to Las Vegas, but rather visiting Las Vegas from another region and are simply playing while on vacation. It is worth noting that other regions save Atlantic City do not have multiple casinos close enough to each other for it to be very practical to play at multiple casinos regularly.

Looking more closely at the different proportions of frequencies across different markets, we see that the idea of people in smaller markets being "regulars" holds true, with a higher proportion of individuals being "Weekly" or "Monthly" customers than we see in Las Vegas.

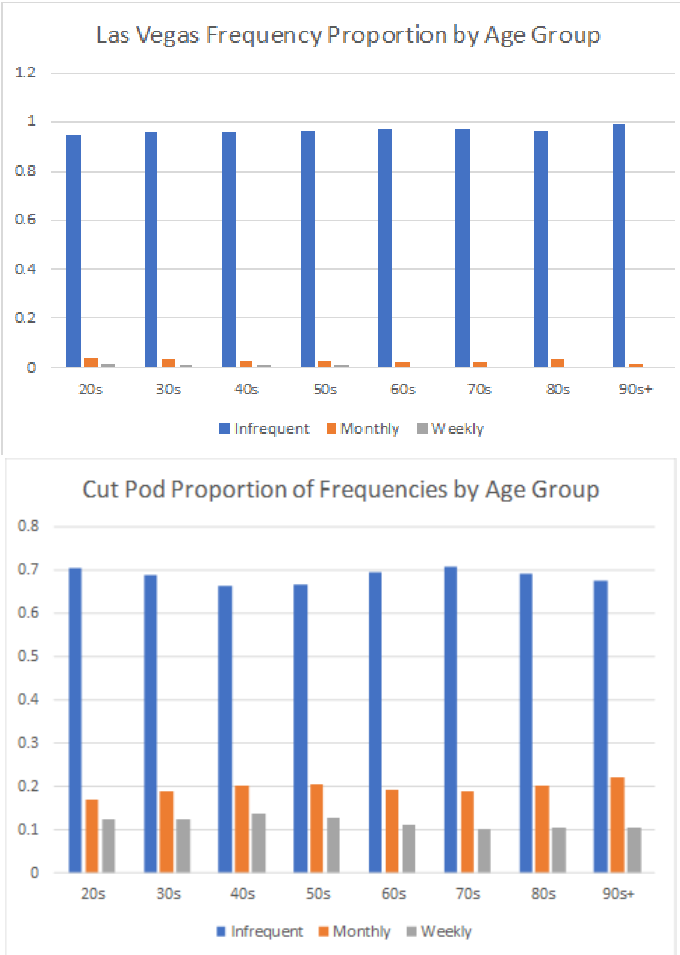


FIGURE 2.1: Graphs displaying the proportion of each frequency by age bin

The figure above shows what we would expect, with a much higher proportion across all age bins being "regular" players within the Cut Pod.

The most interesting results from this regional breakdown actually comes from the West Pod, which is the only region that contains groups where the "Weekly" proportion is higher than the "Monthly" proportion.

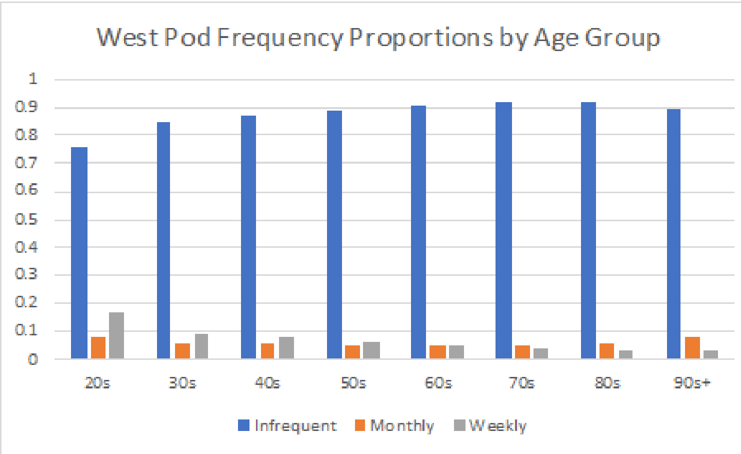


FIGURE 2.2: West Pod's interesting ratio of Weekly Players

We can see here that the younger age brackets have a much higher proportion of "Weekly" players as compared to both "Monthly" players and the other regions. The West Pod includes properties in Laughlin, Lake Tahoe, and Reno, Nevada. These properties are considered regional properties, not being only local markets such as properties in the Cut Pod, and drawing customers from surrounding areas such as the Bay Area in California. This shows that attracting young customers may not be a matter of bringing them in from a different part of the country, but rather building up regional loyalty from the younger generation.

We can also break down the average theo by frequency metric over the entirety of the time period that we are looking at. When looking at this, we would expect the proportion of theo coming from "Weekly" players to be proportionally higher based on how much more often they play.

In both Las Vegas and the West pod, we actually see that the average theo for a monthly player is significantly higher than that of of a weekly player. This supports the ideas that "Monthly" players are the real tourists in Las Vegas. Similarly for the regional parts of the West Pod, the tourist regions are where a majority of the money can be found. This also shows that within the West Pod, despite weekly players making up a majority of young players, they still do not account for a significant amount of theoretical value.

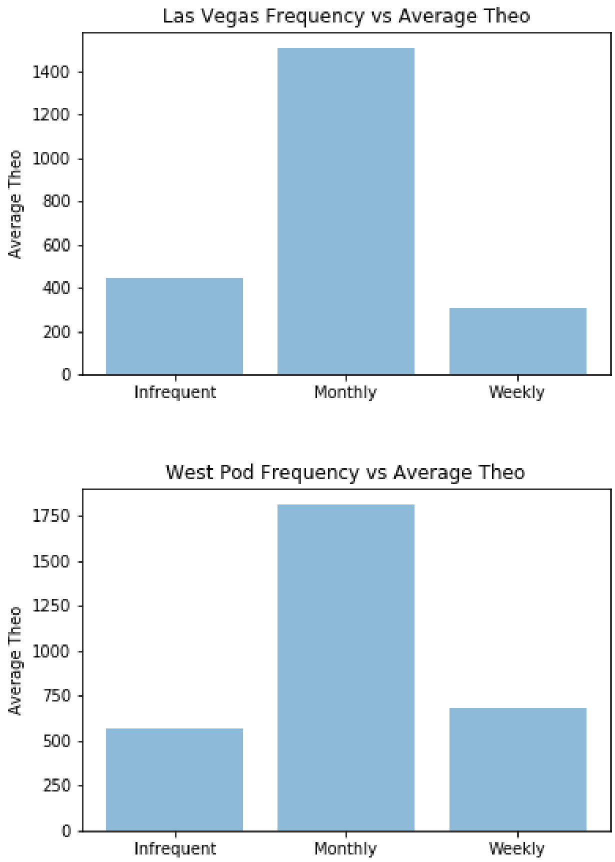


FIGURE 2.3: The greatest disparity from expectations came in the West Pod and Las Vegas

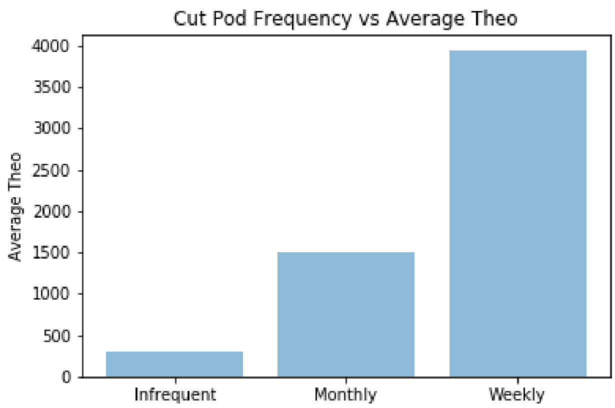


FIGURE 2.4: The other regional markets looked how we would expect

It is worth noting that the average theo of monthly players in Atlantic City, another destination/regional market, is higher than the average theo of weekly players, although the disparity is not as great as seen in Las Vegas/the West Pod.

2.2 Age

Currently, Caesars runs a prediction model in order to determine individual customers' preferences in terms of game they play. And with an extremely high level of accuracy, they can predict that a person prefers table games, such as blackjack, craps, etc., to slot machines using only an individual's age. Researchers and industry members alike have recognized this trend for years, although nobody has figured out the recipe for increasing the preferences of slot machines for the new generation of customers [5].

When looking for a direct relationship between a person's theo over the relevant time period and their age, no clear relationship appears. The distribution is slightly skewed left due to the older general tilt of slot machine players, with a steep drop off beyond age 70.

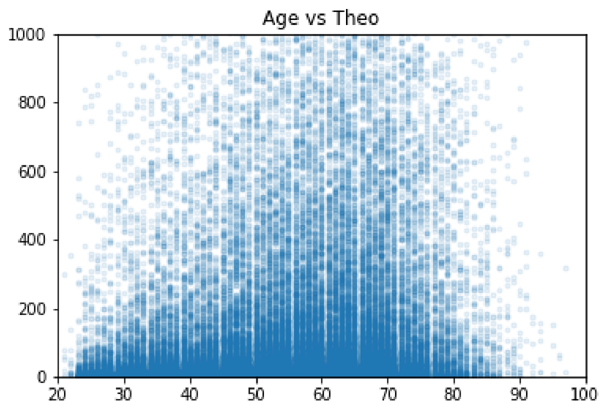


FIGURE 2.5: Example plot of Age vs Theo from Las Vegas

In the other regions, the relationship looks similar, with no obvious trends beyond the general distribution of theo, and play overall. However, when we start to break down the relationship in to age brackets, we can start to see the relationships that arise.

If we start by looking initially at the proportion of individuals who come from each age bracket that we have defined, nothing significant arises; the modal age group is either 50s or 60s depending on the region, the distribution is slightly skewed left, and everything looks in place. We can do the same thing for proportion of theo from each age bracket and the same trends seem to hold within region.

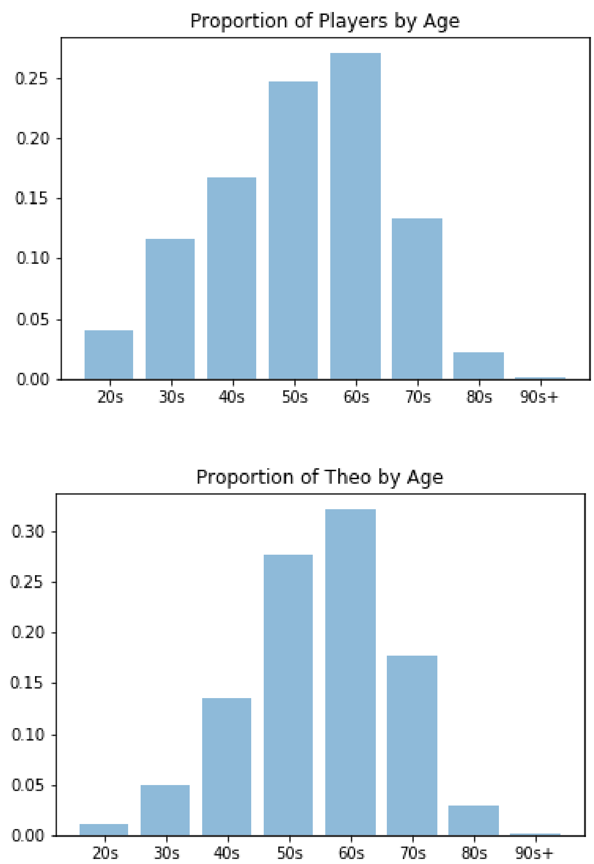


FIGURE 2.6: Example Plots Looking at Proportion of Players and Proportion of Theo

However, when we start to compare the proportion of summed theo to the proportion of total players from an age bracket, we can see that not all age brackets are the same. This difference between the proportion of total players coming from an age bracket and the proportion of total theo coming from an age bracket shows that some age brackets are playing more than you would expect, even when controlling for size of the age bracket. This relationship holds, to a certain degree, across all regions and markets.

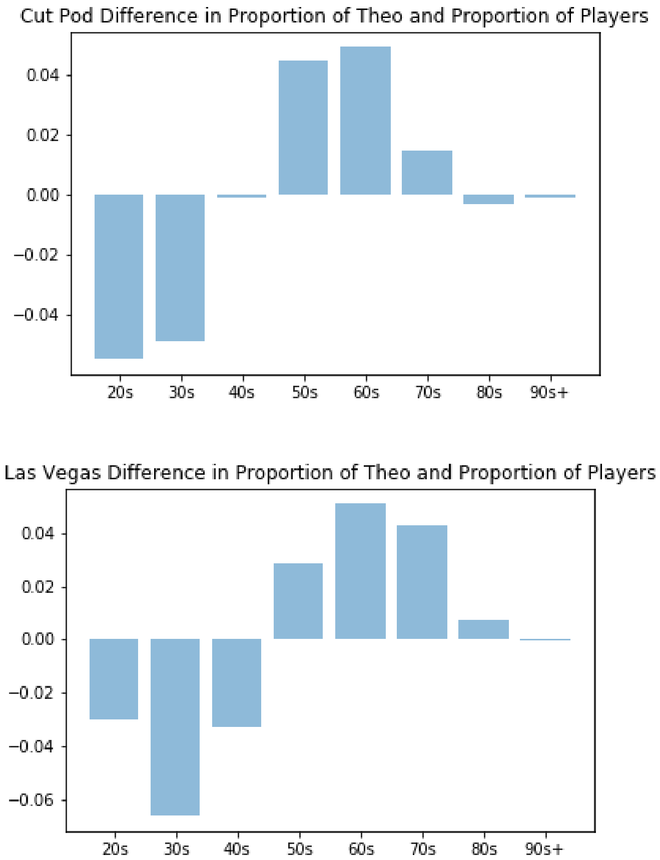


FIGURE 2.7: Plots showing the differences between Theo and Age Proportion

In all of the regions, the 60s and 70s age brackets have the highest over-representation while the 20s and 30s have the lowest under-representation. This means that not only are not as many younger people playing, but even when they do sit down at a slot machine, they play less. However, we are not able to break down whether the increase in play is because of increased playtime or higher than average bet size. And while we are not able to look at the individual bets that an individual makes during their play, we can look at the total average play time across age brackets over the entire time period to try to draw out the relationship:

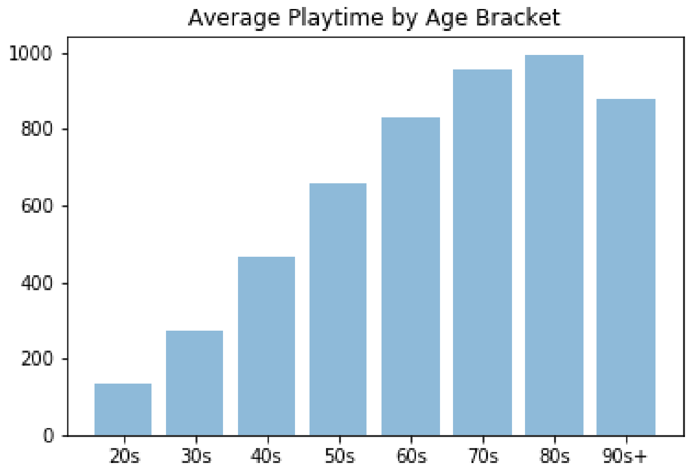


FIGURE 2.8: This figure shows the average playtime by age bracket

Surprisingly, the age bracket with the highest playtime is people in their 80s. This is despite the 80s age bracket having less than their share of theo in a number of regions, and never having the highest difference between theo and population in any bracket. We see that people in their 20s and 30s have the lowest average playtime of any bracket, less than half of what people in their 70s and 80s play. This average playtime graph shows that casinos are struggling to keep younger people playing slot machines in the way that the older generation is.

We can think of total theo as the following equation:

$$\sum theo = \alpha_m * b * p \rightarrow b = \frac{\sum theo}{\alpha_m * b}$$

where b is the average bet size, an unknown, p is the total playtime, and α_m is the expected percentage hold to the casino for a given machine m, a constant value that is not publicly available. Using this equation, we can solve for the average bet size across age brackets at each property.

The average bet sizes vary greatly across different regions, with the highest average bet size being found in the North Pod and the lowest average bet being found in Las Vegas.

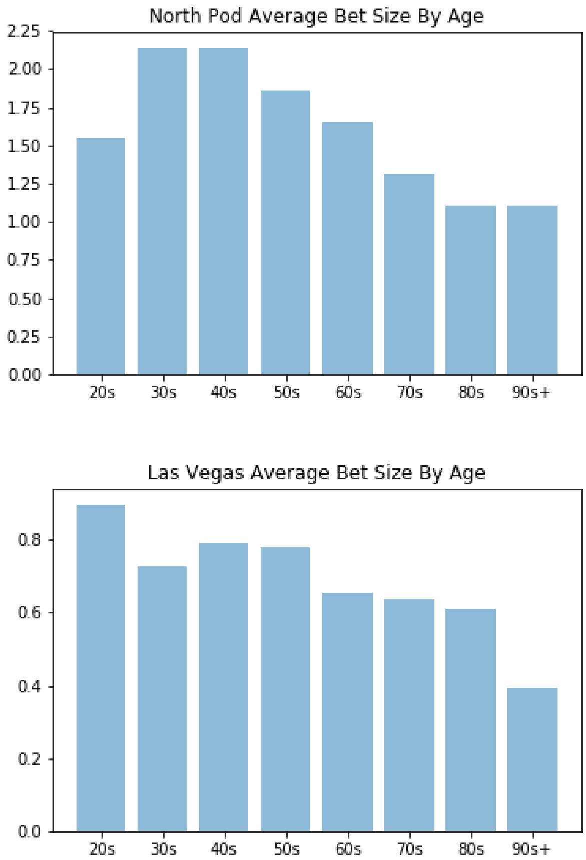


FIGURE 2.9: Caption

Surprisingly, in Las Vegas, the younger a player is, the higher their general bet size is going to be. This matches the image that people have of young adults coming to Las Vegas as tourists and playing big and playing short, both of the trends that we see, but is not what we would expect when looking at the total amount of income that comes from people in their 20s and 30s. The trend of younger people betting bigger tentatively holds across all regions, with no region having an age bracket older than '50s' have the highest average bet. This is contrary average play time, where the older you are, the longer you are expected to play. This could be because smaller bet size causes the same amount of money to go further, as spending \$10 per spin and not winning is going to run through an individual's gambling budget much more quickly than spending \$1 assuming that both people win the same amount. Although with more stable sources of income, we might expect the average older adults to bet bigger than regular young adults, usually in the infancy of careers.

Running a one way ANOVA test for means, we obtain p-values showing a statistically significant difference in means, indicating that these bet sizes are different across age groups. Additionally, we see that the means across regions for the same age group are also statistically different by the same test.

One important caveat to this analysis is that these numbers are all over α_m. That means, to get the true average bet size, we must multiply each of these values by α_m, a constant that is not public. There is also no guarantee that $\alpha_m = \alpha_n$ for two different machines m and n. Therefore, there is going to be a high degree of variance within the true bet sizes, even within age brackets.

2.2.1 Are Older Players More Strategic?

In slot machines, there is not too much strategy to play. It is the purest form of gambling, with minimal control on the outcome on most machines. The only serious bits of strategy would be where to start and when to stop. Many gamblers will play until they either win big or lose it all, and so to try and attempt to look at the playing habits of groups by age, we can plot the ratio of coin-in to coin-out against a person's age. The ratio of coin-in to coin-out can tell us how much a person won or lost before they left the machine.

In figure 2.10 below, we can see the Coin-in/Coin-out ratios get less scattered as age increases, converging on a value slightly less than 1. While the modal value of this function is 0 across all age brackets, we can see that outlying values start to decrease in frequency as an individual gets older than 70, with a high frequency occurring between .25 and 1.5. This difference cannot be explained by the number of individuals alone, as there are more than double the number of customers in their 70s as there are in their 20s despite the 20s having a much larger spread of end results. As we have shown earlier, older people are generally much more conservative with their bet sizes, and so it makes sense that they would also be more conservative with their betting strategy, cashing out well before they have run out of credits to play with.

2.3 Can We Use This Info to Predict?

If we look at both the frequency and age variables, we can try to use these to build a linear model for prediction of the total amount of theo that an individual will have. For this, we cast the frequency as the following values:

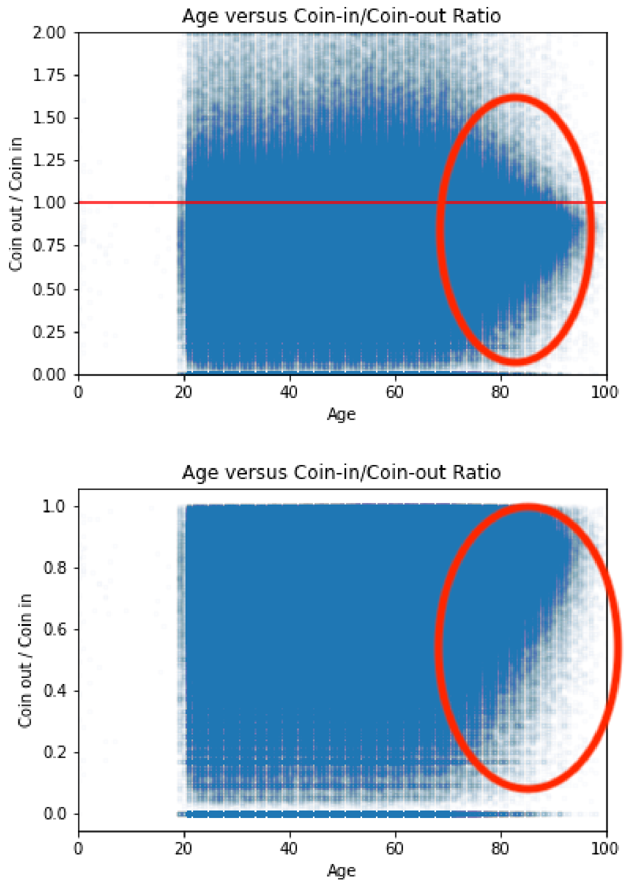


FIGURE 2.10: The first figure highlights differences in betting strategy, with a break even line plotted at 1.The second figure zooms in on coin in/coin out ratios less than 1, highlighting the decrease in loss with age.

$$T(n) = \begin{cases} 2 & \text{if n='Infrequent'} \\ 12 & \text{if n='Monthly'} \\ 52 & \text{if n='Weekly'} \end{cases}$$

We choose to cast "Infrequent" as 2, as the median millennial has visited Las Vegas twice in the past 12 months, and the mean number of visits in the past 12 months according to the visitor profile is 2.0 [1][2].

Creating the model for each region, we can see that, although the variables themselves hold statistically significant p-values, the overall model is not very good, having R^2 values ranging from 0.059 to .128. This shows that the age and frequency variables will predict some level of the variance, but only around 6-13%. This means that there are either other predictive variables that we are not capturing or simply too much random noise to make accurate predictions. It is also possible that a linear model is the incorrect shape. Taking expected transformations, such as log transformations, marginally changes the

R^2 values for each model, although they are not all improvements.

All 7 model summaries can be found in Appendix A.

Chapter 3

Conclusions

We have shown that there are large differences in the preferences of customers across a multitude of variables. While initially just looking at age as a predictive variable, we have seen that differences also arise from location and frequency of play.

In regional markets, we see that the total amount of theo that a person has tracks well with frequency, with more frequent players contributing more theo total. In regional and destination markets, however, we see that we cannot use frequency as a naive predictor, as monthly customers often contribute more theo than weekly customers.

Additionally, we see that older individuals are more frequent players as well across every region except for the West Pod. This tracks well with our other conclusions about age, with older customers contributing to more theo and having generally much longer play times. Older customers are also more conservative with their bet sizes and the losses that they are willing to take before they cash out.

The frequency of play from customers seems like it would be a very strong indicator of play itself due to self selection; people who play more frequently clearly enjoy playing slot machines more and would therefore be more likely to bet big, but this breaks down if we look exclusively at Las Vegas, or the West Pod. And the proportions of different frequencies by age group, while seeming intuitive, was actually not at all; why the West Pod has more 25 year olds that are weekly players rather than monthly players while other, similar markets such as the Cut Pod do not, is not at all obvious.

I initially believed that age trends would hold across the country, but we can see that, while trends definitely do still exist, they begin to vary substantially from region to region. And even after breaking down the markets by age, due to the high amount of noise in the data, we needed to look at age in in buckets instead. Within these buckets, while clear trends do exist, large differences in regional preferences arise, hurting the conventional wisdom. While young people betting big in Las Vegas fits the conventional wisdom, people in their 40s in the North Pod bet the biggest across the country. Average bet sizes in Las Vegas are relatively tame by comparison.

Future work could refine the analyses done within this paper even further, breaking it down on a machine by machine level. Across the country, common slot machine brands such as Wheel of Fortune and Buffalo Bill can be found in many different casinos. And not all of these machines will attract the same audience. Bringing the analysis down this level could make sense of some of the regional disparities based on what machines can be found where.

Additionally, survey data, especially in destination markets such as Las Vegas and Atlantic City, could help to capture the data that is missing from Total Rewards. Because this data set is missing a large enough chunk that the median age is off by 10 years in Las Vegas, we are clearly missing the most data from the audience that we are trying to explore the most.

Overall, however, this paper brings forth conclusions on the relationship between age, frequency, location, and theoretical value and bet size all in to one, showing clear areas for improvement in order to capture the customer group that is going to help the gaming industry continue to grow well into the future.

Appendix A

Summaries of Linear Regressions

The summaries of the seven linear regression models created can be found below.

OLS Regression Results

Dep. Variable:	f_theo_win	**R-squared:**	0.068
Model:	OLS	**Adj. R-squared:**	0.068
Method:	Least Squares	**F-statistic:**	1.485e+04
Date:	Fri, 30 Mar 2018	**Prob (F-statistic):**	0.00
Time:	01:10:42	**Log-Likelihood:**	-3.8302e+06
No. Observations:	405908	**AIC:**	7.660e+06
Df Residuals:	405906	**BIC:**	7.660e+06
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
i_age	8.8830	0.091	97.338	0.000	8.704	9.062
trips_per_year	33.5636	0.399	84.105	0.000	32.781	34.346

Omnibus:	1011616.812	**Durbin-Watson:**	1.599
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	47663639606.755
Skew:	26.727	**Prob(JB):**	0.00
Kurtosis:	1680.896	**Cond. No.**	4.94

Figure A.1: Summary of regression in the Atlantic City Region

OLS Regression Results

Dep. Variable:	f_theo_win	**R-squared:**	0.105
Model:	OLS	**Adj. R-squared:**	0.105
Method:	Least Squares	**F-statistic:**	2.095e+04
Date:	Fri, 30 Mar 2018	**Prob (F-statistic):**	0.00
Time:	01:11:53	**Log-Likelihood:**	-3.4494e+06
No. Observations:	357997	**AIC:**	6.899e+06
Df Residuals:	357995	**BIC:**	6.899e+06
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
i_age	4.6636	0.126	37.098	0.000	4.417	4.910
trips_per_year	56.8394	0.371	153.034	0.000	56.111	57.567

Omnibus:	1004159.653	**Durbin-Watson:**	1.611
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	149870393785.315
Skew:	36.164	**Prob(JB):**	0.00
Kurtosis:	3171.916	**Cond. No.**	3.57

FIGURE A.2: Summary of regression in the Cut Pod Region

OLS Regression Results

Dep. Variable:	f_theo_win	**R-squared:**	0.128
Model:	OLS	**Adj. R-squared:**	0.128
Method:	Least Squares	**F-statistic:**	1.392e+04
Date:	Fri, 30 Mar 2018	**Prob (F-statistic):**	0.00
Time:	01:15:50	**Log-Likelihood:**	-1.7732e+06
No. Observations:	190237	**AIC:**	3.546e+06
Df Residuals:	190235	**BIC:**	3.546e+06
Df Model:	2		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
i_age	5.5909	0.124	45.144	0.000	5.348	5.834
trips_per_year	41.1785	0.367	112.230	0.000	40.459	41.898

Omnibus:	421939.654	**Durbin-Watson:**	1.688
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	5367747772.734
Skew:	20.352	**Prob(JB):**	0.00
Kurtosis:	824.906	**Cond. No.**	3.62

FIGURE A.3: Summary of regression in the Heartland Pod Region

OLS Regression Results

Dep. Variable:	f_theo_win	**R-squared:**	0.064
Model:	OLS	**Adj. R-squared:**	0.064
Method:	Least Squares	**F-statistic:**	1570.
Date:	Fri, 30 Mar 2018	**Prob (F-statistic):**	0.00
Time:	01:10:21	**Log-Likelihood:**	-4.1090e+05
No. Observations:	46012	**AIC:**	8.218e+05
Df Residuals:	46010	**BIC:**	8.218e+05
Df Model:	2		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
i_age	7.8825	0.170	46.319	0.000	7.549	8.216
trips_per_year	9.8859	1.871	5.285	0.000	6.219	13.553

Omnibus:	113374.785	**Durbin-Watson:**	1.818
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	3016913407.226
Skew:	26.189	**Prob(JB):**	0.00
Kurtosis:	1256.351	**Cond. No.**	12.6

FIGURE A.4: Summary of regression on the Las Vegas Stirp

OLS Regression Results

Dep. Variable:	f_theo_win	**R-squared:**	0.099
Model:	OLS	**Adj. R-squared:**	0.099
Method:	Least Squares	**F-statistic:**	1.983e+04
Date:	Fri, 30 Mar 2018	**Prob (F-statistic):**	0.00
Time:	01:12:56	**Log-Likelihood:**	-3.5257e+06
No. Observations:	361018	**AIC:**	7.051e+06
Df Residuals:	361016	**BIC:**	7.051e+06
Df Model:	2		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
i_age	6.6370	0.139	47.623	0.000	6.364	6.910
trips_per_year	62.3518	0.447	139.398	0.000	61.475	63.228

Omnibus:	1296621.107	**Durbin-Watson:**	1.571
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	2453854774023.780
Skew:	72.815	**Prob(JB):**	0.00
Kurtosis:	12774.367	**Cond. No.**	3.88

Figure A.5: Summary of regression in the North Pod Region

OLS Regression Results

Dep. Variable:	f_theo_win	**R-squared:**	0.059
Model:	OLS	**Adj. R-squared:**	0.059
Method:	Least Squares	**F-statistic:**	9565.
Date:	Fri, 30 Mar 2018	**Prob (F-statistic):**	0.00
Time:	01:14:13	**Log-Likelihood:**	-2.8744e+06
No. Observations:	303553	**AIC:**	5.749e+06
Df Residuals:	303551	**BIC:**	5.749e+06
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
i_age	9.6559	0.113	85.500	0.000	9.435	9.877
trips_per_year	23.1328	0.387	59.807	0.000	22.375	23.891

Omnibus:	811225.008	**Durbin-Watson:**	1.664
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	53703618504.668
Skew:	32.060	**Prob(JB):**	0.00
Kurtosis:	2062.585	**Cond. No.**	3.88

Figure A.6: Summary of regression in the South Pod Region

OLS Regression Results

Dep. Variable:	f_theo_win	**R-squared:**	0.068
Model:	OLS	**Adj. R-squared:**	0.068
Method:	Least Squares	**F-statistic:**	7162.
Date:	Fri, 30 Mar 2018	**Prob (F-statistic):**	0.00
Time:	01:15:02	**Log-Likelihood:**	-1.8018e+06
No. Observations:	197213	**AIC:**	3.604e+06
Df Residuals:	197211	**BIC:**	3.604e+06
Df Model:	2		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
i_age	9.4654	0.091	104.172	0.000	9.287	9.643
trips_per_year	6.1458	0.431	14.272	0.000	5.302	6.990

Omnibus:	516649.477	**Durbin-Watson:**	1.741
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	33300166279.692
Skew:	30.374	**Prob(JB):**	0.00
Kurtosis:	2015.163	**Cond. No.**	5.17

FIGURE A.7: Summary of regression in the West Pod Region

Appendix B

Python Code

This code was run in a Jupyter Notebook in order to perform all analyses. Some slight changes have been made to facilitate it's placement here.

```
# coding: utf-8

# In[1]:


import pandas as pd
import sklearn
from matplotlib import pyplot as plt
import numpy as np
#%matplotlib inline


# In[2]:


las = pd.read_csv('data_by_prop/data_useful_las_vegas.csv')
atl_cit = pd.read_csv('data_by_prop/data_useful_atlan_city.csv')
atl_cit = atl_cit[atl_cit['i_dmid'] != 'i_dmid']
cut = pd.read_csv('data_by_prop/data_useful_cut_pod.csv')
cut = cut[cut['i_dmid'] != 'i_dmid']
heart = pd.read_csv('data_by_prop/data_useful_heartland.csv')
heart = heart[heart['i_dmid'] != 'i_dmid']
```

```
north = pd.read_csv('data_by_prop/data_useful_north_pod.csv')
north = north[north['i_dmid'] != 'i_dmid']
south = pd.read_csv('data_by_prop/data_useful_south_pod.csv')
south = south[south['i_dmid'] != 'i_dmid']
west = pd.read_csv('data_by_prop/data_useful_west_pod.csv')


# In[3]:


def age_bracket(age):
    if age < 30:
        return '20s'
    elif age < 40:
        return '30s'
    elif age < 50:
        return '40s'
    elif age < 60:
        return '50s'
    elif age < 70:
        return '60s'
    elif age < 80:
        return '70s'
    elif age < 90:
        return '80s'
    else:
        return '90s+'


# In[4]:


atl_cit[['i_dmid', 'f_coin_in', 'f_winloss',
    'f_theo_win', 'f_worth', 'i_age']] =
    atl_cit[['i_dmid', 'f_coin_in', 'f_winloss',
    'f_theo_win', 'f_worth', 'i_age']].astype(float)

cut[['i_dmid', 'f_coin_in', 'f_winloss',
    'f_theo_win', 'f_worth', 'i_age']] =
```

```
    cut[['i_dmid', 'f_coin_in', 'f_winloss',
    'f_theo_win', 'f_worth', 'i_age']].astype(float)

heart[['i_dmid', 'f_coin_in', 'f_winloss',
    'f_theo_win', 'f_worth', 'i_age']] =
    heart[['i_dmid', 'f_coin_in', 'f_winloss',
    'f_theo_win', 'f_worth', 'i_age']].astype(float)

north[['i_dmid', 'f_coin_in', 'f_winloss',
    'f_theo_win', 'f_worth', 'i_age']] =
    north[['i_dmid', 'f_coin_in', 'f_winloss',
    'f_theo_win', 'f_worth', 'i_age']].astype(float)

south[['i_dmid', 'f_coin_in', 'f_winloss',
    'f_theo_win', 'f_worth', 'i_age']] =
    south[['i_dmid', 'f_coin_in', 'f_winloss',
    'f_theo_win', 'f_worth', 'i_age']].astype(float)


# In[5]:


las['age_bin'] = las.apply(lambda row:
                           age_bracket(row.i_age), axis = 1)
total_theo_las = las['f_theo_win'].sum()


# In[6]:


atl_cit['age_bin'] = atl_cit.apply(lambda row:
                                   age_bracket(row.i_age), axis = 1)
total_theo_atl = atl_cit['f_theo_win'].sum()


# In[7]:


heart['age_bin'] = heart.apply(lambda row:
```

```
                              age_bracket(row.i_age), axis = 1)
total_theo_heart = heart['f_theo_win'].sum()


# In[8]:


north['age_bin'] = north.apply(lambda row:
                              age_bracket(row.i_age), axis = 1)
total_theo_north = north['f_theo_win'].sum()


# In[9]:


cut['age_bin'] = cut.apply(lambda row:
                          age_bracket(row.i_age), axis = 1)
total_theo_cut = cut['f_theo_win'].sum()


# In[10]:


south['age_bin'] = south.apply(lambda row:
                              age_bracket(row.i_age), axis = 1)
total_theo_south = south['f_theo_win'].sum()


# In[11]:


west['age_bin'] = west.apply(lambda row: age_bracket(row.i_age), axis = 1)
total_theo_west = west['f_theo_win'].sum()


# In[12]:


#sizes:
```

```
#Las Vegas: 9 casinos, 902,412 entries
#Atlantic City: 3 casinos, 16,291,614 entries
#Cut: 4 casinos, 20,086,889 entries
#Heartland: 3 casinos, 10,734,750 entries
#North Pod: 4 casinos, 23,167,299 entries
#south pod: 4 casinos, 10,659,479 entries
#west pod: 4 casinos, 6,231,952 entries


# In[13]:


las_freqgroup = las.groupby(['age_bin', 'c_frequency', 'i_dmid'],
                            as_index=False)[['f_theo_win']].sum()
las_freq_final = las_freqgroup.groupby(['age_bin', 'c_frequency'],
                                       as_index=False)[['i_dmid']].count()
las_freq_final = las_freq_final.append(
    {'age_bin': '90s+', 'c_frequency': 'Weekly',
     'i_dmid': 0}, ignore_index=True)
sums = las_freq_final.groupby(['age_bin'],
                              as_index=False)[['i_dmid']].sum()
las_freq_final['age_bins_totals'] =
    [val for val in sums.i_dmid for _ in (0, 1, 2)]
las_freq_final['proportion'] = las_freq_final.apply(lambda row:
                      row.i_dmid / row.age_bins_totals, axis = 1)
las_freq_final


# In[14]:


atl_freqgroup = atl_cit.groupby(['age_bin', 'c_frequency', 'i_dmid'],
                                as_index=False)[['f_theo_win']].sum()
atl_freq_final = atl_freqgroup.groupby(['age_bin', 'c_frequency'],
                        as_index=False)[['i_dmid']].count()
sums = atl_freq_final.groupby(['age_bin'],
                              as_index=False)[['i_dmid']].sum()
atl_freq_final['age_bins_totals'] =
    [val for val in sums.i_dmid for _ in (0, 1, 2)]
```

```
atl_freq_final['proportion'] = atl_freq_final.apply(lambda row:
                 row.i_dmid / row.age_bins_totals, axis = 1)
atl_freq_final


# In[15]:


heart_freqgroup = heart.groupby(['age_bin', 'c_frequency', 'i_dmid'],
                 as_index=False)[['f_theo_win']].sum()
heart_freq_final = heart_freqgroup.groupby(['age_bin', 'c_frequency'],
                       as_index=False)[['i_dmid']].count()
sums = heart_freq_final.groupby(['age_bin'],
                               as_index=False)[['i_dmid']].sum()
heart_freq_final['age_bins_totals'] =
    [val for val in sums.i_dmid for _ in (0, 1, 2)]
heart_freq_final['proportion'] = heart_freq_final.apply(lambda row:
                   row.i_dmid / row.age_bins_totals, axis = 1)
heart_freq_final


# In[16]:


north_freqgroup = north.groupby(['age_bin', 'c_frequency', 'i_dmid'],
                               as_index=False)[['f_theo_win']].sum()
north_freq_final = north_freqgroup.groupby(['age_bin', 'c_frequency'],
                     as_index=False)[['i_dmid']].count()
sums = north_freq_final.groupby(['age_bin'],
                               as_index=False)[['i_dmid']].sum()
north_freq_final['age_bins_totals'] =
    [val for val in sums.i_dmid for _ in (0, 1, 2)]
north_freq_final['proportion'] = north_freq_final.apply(lambda row:
                      row.i_dmid / row.age_bins_totals, axis = 1)
north_freq_final


# In[17]:
```

```
south_freqgroup = south.groupby(['age_bin', 'c_frequency', 'i_dmid'],
                                as_index=False)[['f_theo_win']].sum()
south_freq_final = south_freqgroup.groupby(['age_bin', 'c_frequency'],
                           as_index=False)[['i_dmid']].count()
sums = south_freq_final.groupby(['age_bin'],
                     as_index=False)[['i_dmid']].sum()
south_freq_final['age_bins_totals'] =
    [val for val in sums.i_dmid for _ in (0, 1, 2)]
south_freq_final['proportion'] = south_freq_final.apply(lambda row:
                     row.i_dmid / row.age_bins_totals, axis = 1)
south_freq_final


# In[18]:


west_freqgroup = west.groupby(['age_bin', 'c_frequency', 'i_dmid'],
                              as_index=False)[['f_theo_win']].sum()
west_freq_final = west_freqgroup.groupby(['age_bin', 'c_frequency'],
                         as_index=False)[['i_dmid']].count()
sums = west_freq_final.groupby(['age_bin'],
                               as_index=False)[['i_dmid']].sum()
west_freq_final['age_bins_totals'] =
    [val for val in sums.i_dmid for _ in (0, 1, 2)]
west_freq_final['proportion'] = west_freq_final.apply(lambda row:
                        row.i_dmid / row.age_bins_totals, axis = 1)
west_freq_final


# In[21]:


cut_freqgroup = cut.groupby(['age_bin', 'c_frequency', 'i_dmid'],
                            as_index=False)[['f_theo_win']].sum()
cut_freq_final = cut_freqgroup.groupby(['age_bin', 'c_frequency'],
                          as_index=False)[['i_dmid']].count()
sums = cut_freq_final.groupby(['age_bin'],
                              as_index=False)[['i_dmid']].sum()
```

```
cut_freq_final['age_bins_totals'] =
    [val for val in sums.i_dmid for _ in (0, 1, 2)]
cut_freq_final['proportion'] = cut_freq_final.apply(lambda row:
                    row.i_dmid / row.age_bins_totals, axis = 1)
cut_freq_final


# In[22]:


#start here when looking at theo_win vs frequency


# In[23]:


las_freq_theo = las_freqgroup.groupby(['c_frequency'],
                     as_index=False)[['f_theo_win']].mean()
las_freq_theo.columns = ['c_frequency', 'avg_theo_per_person']
plt.bar(las_freq_theo.c_frequency,
        las_freq_theo.avg_theo_per_person, align='center', alpha=0.5)
plt.ylabel('Average Theo')
plt.title('Las Vegas Frequency vs Average Theo')
plt.savefig('las_freq_theo.png')


# In[24]:


atl_freq_theo = atl_freqgroup.groupby(['c_frequency'],
                        as_index=False)[['f_theo_win']].mean()
atl_freq_theo.columns = ['c_frequency', 'avg_theo_per_person']
plt.bar(atl_freq_theo.c_frequency,
        atl_freq_theo.avg_theo_per_person, align='center', alpha=0.5)
plt.ylabel('Average Theo')
plt.title('Atlantic City Frequency vs Average Theo')
plt.savefig('atl_cit_freq_theo.png')
```

```
# In[25]:



heart_freq_theo = heart_freqgroup.groupby(['c_frequency'],
                            as_index=False)[['f_theo_win']].mean()
heart_freq_theo.columns = ['c_frequency', 'avg_theo_per_person']
plt.bar(heart_freq_theo.c_frequency,
        heart_freq_theo.avg_theo_per_person, align='center', alpha=0.5)
plt.ylabel('Average Theo')
plt.title('Heartland Frequency vs Average Theo')
plt.savefig('heart_freq_theo.png')


# In[26]:



north_freq_theo = north_freqgroup.groupby(['c_frequency'],
                           as_index=False)[['f_theo_win']].mean()
north_freq_theo.columns = ['c_frequency', 'avg_theo_per_person']
plt.bar(north_freq_theo.c_frequency,
        north_freq_theo.avg_theo_per_person, align='center', alpha=0.5)
plt.ylabel('Average Theo')
plt.title('North Pod Frequency vs Average Theo')
plt.savefig('north_freq_theo.png')


# In[27]:



south_freq_theo = south_freqgroup.groupby(['c_frequency'],
                          as_index=False)[['f_theo_win']].mean()
south_freq_theo.columns = ['c_frequency', 'avg_theo_per_person']
plt.bar(south_freq_theo.c_frequency,
        south_freq_theo.avg_theo_per_person, align='center', alpha=0.5)
plt.ylabel('Average Theo')
plt.title('South Pod Frequency vs Average Theo')
plt.savefig('south_freq_theo.png')
```

```
# In[28]:


west_freq_theo = west_freqgroup.groupby(['c_frequency'],
                          as_index=False)[['f_theo_win']].mean()
west_freq_theo.columns = ['c_frequency', 'avg_theo_per_person']
plt.bar(west_freq_theo.c_frequency,
        west_freq_theo.avg_theo_per_person, align='center', alpha=0.5)
plt.ylabel('Average Theo')
plt.title('West Pod Frequency vs Average Theo')
plt.savefig('west_freq_theo.png')


# In[29]:


cut_freq_theo = cut_freqgroup.groupby(['c_frequency'],
                         as_index=False)[['f_theo_win']].mean()
cut_freq_theo.columns = ['c_frequency', 'avg_theo_per_person']
plt.bar(cut_freq_theo.c_frequency,
        cut_freq_theo.avg_theo_per_person, align='center', alpha=0.5)
plt.ylabel('Average Theo')
plt.title('Cut Pod Frequency vs Average Theo')
plt.savefig('cut_freq_theo.png')


# In[30]:


#start here when looking at theo_win vs age across properties.


# In[31]:


las_group1 = las.groupby(['i_dmid', 'i_age', 'age_bin'],
                         as_index=False)[['f_theo_win']].sum()
las_group1.columns = ['i_dmid', 'i_age',
                      'age_bin', 'sum_theo_per_player']
```

```
# In[32]:


atl_group1 = atl_cit.groupby(['i_dmid', 'i_age', 'age_bin'],
                             as_index=False)[['f_theo_win']].sum()
atl_group1.columns = ['i_dmid', 'i_age',
                      'age_bin', 'sum_theo_per_player']


# In[33]:


heart_group1 = heart.groupby(['i_dmid', 'i_age', 'age_bin'],
                             as_index=False)[['f_theo_win']].sum()
heart_group1.columns = ['i_dmid', 'i_age',
                        'age_bin', 'sum_theo_per_player']


# In[34]:


north_group1 = north.groupby(['i_dmid', 'i_age', 'age_bin'],
                             as_index=False)[['f_theo_win']].sum()
north_group1.columns = ['i_dmid', 'i_age',
                        'age_bin', 'sum_theo_per_player']


# In[35]:


south_group1 = south.groupby(['i_dmid', 'i_age', 'age_bin'],
                             as_index=False)[['f_theo_win']].sum()
south_group1.columns = ['i_dmid', 'i_age',
                        'age_bin', 'sum_theo_per_player']


# In[36]:
```

```
west_group1 = west.groupby(['i_dmid', 'i_age', 'age_bin'],
                           as_index=False)[['f_theo_win']].sum()
west_group1.columns = ['i_dmid', 'i_age',
                       'age_bin', 'sum_theo_per_player']


# In[37]:


cut_group1 = cut.groupby(['i_dmid', 'i_age', 'age_bin'],
                         as_index=False)[['f_theo_win']].sum()
cut_group1.columns = ['i_dmid', 'i_age',
                      'age_bin', 'sum_theo_per_player']


# In[38]:


las_group_final = las_group1.groupby(['age_bin'],
                       as_index=False)[['sum_theo_per_player']].sum()
las_group_ages =  las_group1.groupby(['age_bin'],
                      as_index=False)[['i_dmid']].count()
las_group_final.columns = ['age_bin', 'prop_theo_per_bracket']
las_group_ages.columns = ['age_bin', 'players']
total_players_las = las_group_ages.players.sum()
las_group_ages['prop_age_bin'] = las_group_ages.apply(lambda row:
                              row.players / total_players_las, axis = 1)
las_group_final['prop_theo_per_bracket'] = las_group_final.apply(lambda row:
                       row.prop_theo_per_bracket / total_theo_las, axis = 1)


# In[39]:


atl_group_final = atl_group1.groupby(['age_bin'],
                    as_index=False)[['sum_theo_per_player']].sum()
atl_group_ages =  atl_group1.groupby(['age_bin'],
```

```
                    as_index=False)[['i_dmid']].count()
atl_group_final.columns = ['age_bin', 'prop_theo_per_bracket']
atl_group_ages.columns = ['age_bin', 'players']
total_players_atl = atl_group_ages.players.sum()
atl_group_ages['prop_age_bin'] = atl_group_ages.apply(lambda row:
                           row.players / total_players_atl, axis = 1)
atl_group_final['prop_theo_per_bracket'] = atl_group_final.apply(lambda row:
                row.prop_theo_per_bracket / total_theo_atl, axis = 1)


# In[40]:



heart_group_final = heart_group1.groupby(['age_bin'],
               as_index=False)[['sum_theo_per_player']].sum()
heart_group_ages =  heart_group1.groupby(['age_bin'],
                     as_index=False)[['i_dmid']].count()
heart_group_final.columns = ['age_bin', 'prop_theo_per_bracket']
heart_group_ages.columns = ['age_bin', 'players']
total_players_heart = heart_group_ages.players.sum()
heart_group_ages['prop_age_bin'] = heart_group_ages.apply(lambda row:
                              row.players / total_players_heart, axis = 1)
heart_group_final['prop_theo_per_bracket'] = heart_group_final.apply(lambda row:
                     row.prop_theo_per_bracket / total_theo_heart, axis = 1)


# In[41]:



north_group_final = north_group1.groupby(['age_bin'],
             as_index=False)[['sum_theo_per_player']].sum()
north_group_ages =  north_group1.groupby(['age_bin'],
             as_index=False)[['i_dmid']].count()
north_group_final.columns = ['age_bin', 'prop_theo_per_bracket']
north_group_ages.columns = ['age_bin', 'players']
total_players_north = north_group_ages.players.sum()
north_group_ages['prop_age_bin'] = north_group_ages.apply(lambda row:
                        row.players / total_players_north, axis = 1)
north_group_final['prop_theo_per_bracket'] = north_group_final.apply(lambda row:
```

```
                row.prop_theo_per_bracket / total_theo_north, axis = 1)


# In[42]:


south_group_final = south_group1.groupby(['age_bin'],
                 as_index=False)[['sum_theo_per_player']].sum()
south_group_ages =  south_group1.groupby(['age_bin'],
                    as_index=False)[['i_dmid']].count()
south_group_final.columns = ['age_bin', 'prop_theo_per_bracket']
south_group_ages.columns = ['age_bin', 'players']
total_players_south = south_group_ages.players.sum()
south_group_ages['prop_age_bin'] = south_group_ages.apply(lambda row:
                      row.players / total_players_south, axis = 1)
south_group_final['prop_theo_per_bracket'] = south_group_final.apply(lambda row:
                row.prop_theo_per_bracket / total_theo_south, axis = 1)


# In[43]:


west_group_final = west_group1.groupby(['age_bin'],
                  as_index=False)[['sum_theo_per_player']].sum()
west_group_ages =  west_group1.groupby(['age_bin'],
                as_index=False)[['i_dmid']].count()
west_group_final.columns = ['age_bin', 'prop_theo_per_bracket']
west_group_ages.columns = ['age_bin', 'players']
total_players_west = west_group_ages.players.sum()
west_group_ages['prop_age_bin'] = west_group_ages.apply(lambda row:
                             row.players / total_players_west, axis = 1)
west_group_final['prop_theo_per_bracket'] = west_group_final.apply(lambda row:
                 row.prop_theo_per_bracket / total_theo_west, axis = 1)


# In[44]:


cut_group_final = cut_group1.groupby(['age_bin'],
```

```
            as_index=False)[['sum_theo_per_player']].sum()
cut_group_ages =  cut_group1.groupby(['age_bin'],
                   as_index=False)[['i_dmid']].count()
cut_group_final.columns = ['age_bin', 'prop_theo_per_bracket']
cut_group_ages.columns = ['age_bin', 'players']
total_players_cut = cut_group_ages.players.sum()
cut_group_ages['prop_age_bin'] = cut_group_ages.apply(lambda row:
            row.players / total_players_cut, axis = 1)
cut_group_final['prop_theo_per_bracket'] = cut_group_final.apply(lambda row:
              row.prop_theo_per_bracket / total_theo_cut, axis = 1)


# In[45]:


las_vegas_data = pd.concat([las_group_final.reset_index(drop=True),
                 las_group_ages], axis=1)
las_vegas_data['diff'] = las_vegas_data.apply(lambda row:
             row.prop_theo_per_bracket - row.prop_age_bin, axis = 1)
las_vegas_data['total_players'] = total_players_las
las_vegas_data


# In[46]:


plt.bar(las_vegas_data.iloc[:,0], las_vegas_data.prop_age_bin,
        align='center', alpha=0.5)
plt.title('Proportion of Players by Age')
plt.savefig('las_prop_players_age.png')


# In[47]:


plt.bar(las_vegas_data.iloc[:,0],
        las_vegas_data.prop_theo_per_bracket,
        align='center', alpha=0.5)
plt.title('Proportion of Theo by Age')
```

```
plt.savefig('las_prop_theo_age.png')


# In[48]:


atlantic_city_data = pd.concat([atl_group_final.reset_index(drop=True)
             , atl_group_ages], axis=1)
atlantic_city_data['diff'] = atlantic_city_data.apply(lambda row:
         row.prop_theo_per_bracket - row.prop_age_bin, axis = 1)
atlantic_city_data['total_players'] = total_players_atl
atlantic_city_data


# In[49]:


heartland_pod_data = pd.concat([heart_group_final.reset_index(drop=True),
          heart_group_ages], axis=1)
heartland_pod_data['diff'] = heartland_pod_data.apply(lambda row:
          row.prop_theo_per_bracket - row.prop_age_bin, axis = 1)
heartland_pod_data['total_players'] = total_players_heart
heartland_pod_data


# In[50]:


west_pod_data = pd.concat([west_group_final.reset_index(drop=True),
           west_group_ages], axis=1)
west_pod_data['diff'] = west_pod_data.apply(lambda row:
     row.prop_theo_per_bracket - row.prop_age_bin, axis = 1)
west_pod_data['total_players'] = total_players_west
west_pod_data


# In[51]:
```

```
south_pod_data = pd.concat([south_group_final.reset_index(drop=True),
           south_group_ages], axis=1)
south_pod_data['diff'] = south_pod_data.apply(lambda row:
      row.prop_theo_per_bracket - row.prop_age_bin, axis = 1)
south_pod_data['total_players'] = total_players_south
south_pod_data


# In[52]:


north_pod_data = pd.concat([north_group_final.reset_index(drop=True),
          north_group_ages], axis=1)
north_pod_data['diff'] = north_pod_data.apply(lambda row:
         row.prop_theo_per_bracket - row.prop_age_bin, axis = 1)
north_pod_data['total_players'] = total_players_north
north_pod_data


# In[53]:


cut_pod_data = pd.concat([cut_group_final.reset_index(drop=True),
          cut_group_ages], axis=1)
cut_pod_data['diff'] = cut_pod_data.apply(lambda row:
       row.prop_theo_per_bracket - row.prop_age_bin, axis = 1)
cut_pod_data['total_players'] = total_players_cut
cut_pod_data


# In[55]:


ages = las_vegas_data.iloc[:,0]
diff = las_vegas_data.iloc[:,5]

plt.bar(ages, diff, align='center', alpha=0.5)
plt.title('Las Vegas Difference in \          Proportion of Theo and Proportion of Pl
plt.savefig('diff_theo_las.png')
```

```
# In[56]:


ages = atlantic_city_data.iloc[:,0]
diff = atlantic_city_data.iloc[:,5]


plt.bar(ages, diff, align='center', alpha=0.5)
plt.title('Atlantic City Difference in \         Proportion of Theo and Proportion o
plt.savefig('diff_theo_ac.png')


# In[57]:


ages = heartland_pod_data.iloc[:,0]
diff = heartland_pod_data.iloc[:,5]


plt.bar(ages, diff, align='center', alpha=0.5)
plt.title('Heartland Difference in \         Proportion of Theo and Proportion of Pl
plt.savefig('diff_theo_heart.png')


# In[58]:


ages = north_pod_data.iloc[:,0]
diff = north_pod_data.iloc[:,5]


plt.bar(ages, diff, align='center', alpha=0.5)
plt.title('North Pod Difference in \         Proportion of Theo and Proportion of Pl
plt.savefig('diff_theo_north.png')


# In[59]:
```

```
ages = south_pod_data.iloc[:,0]
diff = south_pod_data.iloc[:,5]


plt.bar(ages, diff, align='center', alpha=0.5)
plt.title('South Pod Difference in \\
    Proportion of Theo and Proportion of Players')
plt.savefig('diff_theo_south.png')


# In[60]:


ages = west_pod_data.iloc[:,0]
diff = west_pod_data.iloc[:,5]


plt.bar(ages, diff, align='center', alpha=0.5)
plt.title('West PodDifference in \\
    Proportion of Theo and Proportion of Players')
plt.savefig('diff_theo_west.png')


# In[61]:


ages = cut_pod_data.iloc[:,0]
diff = cut_pod_data.iloc[:,5]


plt.bar(ages, diff, align='center', alpha=0.5)
plt.title('Cut Pod Difference in \\
    Proportion of Theo and Proportion of Players')
plt.savefig('diff_theo_cut.png')


# In[62]:
```

```
playtime = pd.read_csv('data_by_prop/total_play_time_idmid.csv')


# In[63]:


playtime['age_bin'] = playtime.apply(lambda
                        row: age_bracket(row.i_age), axis = 1)


# In[64]:


avg_playtime = playtime.groupby(['age_bin'],
                        as_index=False)[['total_play_time']].mean()


# In[65]:


avg_playtime.head()


# In[66]:


plt.bar(avg_playtime.age_bin, avg_playtime.total_play_time,
        align='center', alpha=.5)
plt.title('Average Playtime by Age Bracket')
plt.savefig('avg_playtime.png')


# In[67]:


#start here for finding average bet size
```

```
# In[68]:


las_theo_age = las_group1.groupby(['age_bin'],
             as_index=False)[['sum_theo_per_player']].mean()
las_theo_age['total_play_time'] = avg_playtime.total_play_time
las_theo_age['avg_bet_over_alpha'] = las_theo_age.apply(lambda row:
           row.sum_theo_per_player / row.total_play_time, axis = 1)
las_theo_age


# In[69]:


atl_theo_age = atl_group1.groupby(['age_bin']
               , as_index=False)[['sum_theo_per_player']].mean()
atl_theo_age['total_play_time'] = avg_playtime.total_play_time
atl_theo_age['avg_bet_over_alpha'] = atl_theo_age.apply(lambda row:
          row.sum_theo_per_player / row.total_play_time, axis = 1)
atl_theo_age


# In[70]:


north_theo_age = north_group1.groupby(['age_bin'],
                 as_index=False)[['sum_theo_per_player']].mean()
north_theo_age['total_play_time'] = avg_playtime.total_play_time
north_theo_age['avg_bet_over_alpha'] = north_theo_age.apply(lambda row:
        row.sum_theo_per_player / row.total_play_time, axis = 1)
north_theo_age


# In[71]:


south_theo_age = south_group1.groupby(['age_bin']
               , as_index=False)[['sum_theo_per_player']].mean()
```

```
south_theo_age['total_play_time'] = avg_playtime.total_play_time
south_theo_age['avg_bet_over_alpha'] = south_theo_age.apply(lambda row:
         row.sum_theo_per_player / row.total_play_time, axis = 1)
south_theo_age


# In[72]:


west_theo_age = west_group1.groupby(['age_bin'],
                as_index=False)[['sum_theo_per_player']].mean()
west_theo_age['total_play_time'] = avg_playtime.total_play_time
west_theo_age['avg_bet_over_alpha'] = west_theo_age.apply(lambda row:
         row.sum_theo_per_player / row.total_play_time, axis = 1)
west_theo_age


# In[73]:


heart_theo_age = heart_group1.groupby(['age_bin'],
                as_index=False)[['sum_theo_per_player']].mean()
heart_theo_age['total_play_time'] = avg_playtime.total_play_time
heart_theo_age['avg_bet_over_alpha'] = heart_theo_age.apply(lambda row:
        row.sum_theo_per_player / row.total_play_time, axis = 1)
heart_theo_age


# In[74]:


cut_theo_age = las_group1.groupby(['age_bin'],
           as_index=False)[['sum_theo_per_player']].mean()
cut_theo_age['total_play_time'] = avg_playtime.total_play_time
cut_theo_age['avg_bet_over_alpha'] = cut_theo_age.apply(lambda row:
          row.sum_theo_per_player / row.total_play_time, axis = 1)
cut_theo_age
```

```
# In[75]:


plt.bar(las_theo_age.age_bin, las_theo_age.avg_bet_over_alpha
        , align='center', alpha=.5)
plt.title('Las Vegas Average Bet Size By Age')
plt.savefig('las_bet_size.png')


# In[76]:


plt.bar(atl_theo_age.age_bin, atl_theo_age.avg_bet_over_alpha
        , align='center', alpha=.5)
plt.title('Atlantic City Average Bet Size By Age')
plt.savefig('atl_bet_size.png')


# In[77]:


plt.bar(north_theo_age.age_bin, north_theo_age.avg_bet_over_alpha
        , align='center', alpha=.5)
plt.title('North Pod Average Bet Size By Age')
plt.savefig('north_bet_size.png')


# In[78]:


plt.bar(south_theo_age.age_bin, south_theo_age.avg_bet_over_alpha
        , align='center', alpha=.5)
plt.title('South Pod Average Bet Size By Age')
plt.savefig('south_bet_size.png')


# In[79]:
```

```
plt.bar(west_theo_age.age_bin, west_theo_age.avg_bet_over_alpha
        , align='center', alpha=.5)
plt.title('West Pod Average Bet Size By Age')
plt.savefig('west_bet_size.png')


# In[80]:


plt.bar(heart_theo_age.age_bin, heart_theo_age.avg_bet_over_alpha
        , align='center', alpha=.5)
plt.title('Heartland Average Bet Size By Age')
plt.savefig('heart_bet_size.png')


# In[81]:


plt.bar(cut_theo_age.age_bin, cut_theo_age.avg_bet_over_alpha
        , align='center', alpha=.5)
plt.title('Cut Pod Average Bet Size By Age')
plt.savefig('cut_bet_size.png')


# In[ ]:


#start here for general theo plot


# In[82]:


x_l = las_group1['i_age']
y_l = las_group1['sum_theo_per_player']


plt.scatter(x_l, y_l, marker = '.', alpha = .1)
plt.ylim([0, 1000])
```

```
plt.xlim([20, 100])
plt.title('Age vs Theo')
plt.savefig('las_theo_age')


# In[83]:


x_l.mean()


# In[ ]:


#start here for histogram by age


# In[84]:


atl_group1 = atl_cit.groupby(['i_dmid', 'i_age'],
                    as_index=False)[['f_theo_win']].mean()
atl_group1.columns = ['i_dmid', 'i_age', 'avg_theo']

cut_group1 = cut.groupby(['i_dmid', 'i_age'],
                as_index=False)[['f_theo_win']].mean()
cut_group1.columns = ['i_dmid', 'i_age', 'avg_theo']

heart_group1 = heart.groupby(['i_dmid', 'i_age'],
                    as_index=False)[['f_theo_win']].mean()
heart_group1.columns = ['i_dmid', 'i_age', 'avg_theo']

north_group1 = north.groupby(['i_dmid', 'i_age'],
                    as_index=False)[['f_theo_win']].mean()
north_group1.columns = ['i_dmid', 'i_age', 'avg_theo']

south_group1 = south.groupby(['i_dmid', 'i_age'],
                    as_index=False)[['f_theo_win']].mean()
south_group1.columns = ['i_dmid', 'i_age', 'avg_theo']
```

```
west_group1 = west.groupby(['i_dmid', 'i_age'],
                as_index=False)[['f_theo_win']].mean()
west_group1.columns = ['i_dmid', 'i_age', 'avg_theo']


# In[86]:


bins = list(range(20, 110, 10))


# In[87]:


plt.hist(las.i_age, bins = bins, normed = True)


# In[88]:


plt.hist(atl_cit.i_age, bins = bins, normed = True)


# In[89]:


plt.hist(cut.i_age, bins = bins, normed = True)


# In[90]:


plt.hist(heart.i_age, bins = bins, normed = True)


# In[91]:
```

```
plt.hist(north.i_age, bins = bins, normed = True)


# In[92]:


plt.hist(south.i_age, bins = bins, normed = True)


# In[93]:


plt.hist(west.i_age, bins = bins, normed = True)
```

Bibliography

[1] Oliver Lovat. Elvis who? understanding, attracting and retaining the next generation of las vegas customers. *Center for Gaming Research Occasional Paper Series: Paper 35*, 35:1–19, June 2016. URL https://digitalscholarship.unlv.edu/occ_papers/34.

[2] 2016 las vegas visitor profile study, 2016. URL http://www.lvcva.com/includes/content/images/media/docs/2016-Las-Vegas-Visitor-Profile.pdf.

[3] Nevada gaming statistics: The last six months, 2018. URL http://gaming.unlv.edu/reports/6_month_NV.pdf.

[4] Atlantic city february casino summary, 2018. URL http://gaming.unlv.edu/reports/2018_02_AC.pdf.

[5] Seyhmus Baloglu Michelle Millar. The relationship of demographics to gaming preferences and behavior. 4, 2008. URL https://repository.usfca.edu/cgi/viewcontent.cgi?article=1008&context=hosp.