Matthew Spitulnik

IST 718: Big Data Analytics

12/30/23

Iowa Alcohol Sales

**Introduction:**

For this project, publicly available Iowa liquor sales data from 2012 through 2023 was used to try to build an ARIMA model that could forecast out future alcohol sales. To do this, various smoothing techniques were explored to account for seasonality that was being observed in the timeseries data. The data being used was collected from here: <https://data.iowa.gov/Sales-Distribution/Iowa-Liquor-Sales/m3tr-qhgy/about_data>. The initial raw data contained hundreds of thousands of rows, so to trim it down to a level that was suitable to work with, only the sales of Tequila based products were examined. Data from the years 2012 through 2017 were utilized to create the initial models, then the data was expanded through 2023 to test the model’s accuracy.

**Initial Analysis:**

Before the modeling could begin, some additional manipulation of the data needed to be performed. The dataset was broken down by daily sales of individual stores, so the same daily time stamp appeared many times across multiple rows. To perform a proper time series analysis, the total bottles sold and revenue data for tequila across all the stores in Iowa needed to be combined to a per day level. Since we were only going to be looking at the total number of bottles sold and the revenue data during the modeling process, we could also perform the step of removing all other columns from the dataset. Figure 1 below displays the data before and after this transformation was performed:

**A comparison of a computer program

Description automatically generated with medium confidence**

**Figure 1: 2012-2016 Dataset Excerpts Before and After Model Transformation**

To confirm the accuracy and reliability of our eventual models, we located the data for the rest of 2016 and 2017. This data was manually collected, imported, and formatted to match our transformed 2012 through August 2016 data.

With the data now in the proper format, tests could be run to determine if it was stationary or if further manipulation would need to take place. This was accomplished using the Augmented Dickey-Fuller (ADF) test. The ADF test is based on the null hypothesis that a unit root is present in a time series, indicating non-stationarity. The alternative hypothesis is that the time series is stationary. As seen in figure 2, an initial “adfuller” test ran against “Bottles Sold” and “Sale (Dollars)” (AKA, revenue) returned ADF statistics of -3.70 and -4.17 which showed convincing evidence against the null hypothesis:

**A screenshot of a calculator

AI-generated content may be incorrect.**

**Figure 2: Initial “Adfuller” Test Results**

The output shows that the time series are stationary, with a constant mean, variance, and constant autocorrelation. Also, the ADF statistics lower than the 1%, 5%, and 10% intervals means they support the rejection of the null hypothesis, as well as a p-value below .05, which means we should be able to reject the null-hypothesis that the data is non-stationary and have constant mean, variance, and constant autocorrelation. When looking at plots of the data, however, some potential trends/seasonality can be seen similar points in time each year. Figures 3 through 8 show plots of the total bottles sold and revenue data for 2012 through 2016, as well as the individual 2014 and 2015 years:

**A blue line with black text

Description automatically generated with medium confidence**

**Figure 3: Total Bottles of Tequila Sold, 2012-2016**

**A blue lines on a white background

Description automatically generated**

**Figure 4: Total Bottles of Tequila Sold, 2014**

**A blue lines on a white background

Description automatically generated**

**Figure 5: Total Bottles of Tequila Sold, 2015**

**A blue line with black text

Description automatically generated with medium confidence**

**Figure 6: Total Revenue from Tequila, 2012-2016**

**A blue lines on a white background

Description automatically generated**

**Figure 7: Total Revenue from Tequila, 2014**

**A blue lines on a white background

Description automatically generated**

**Figure 8: Total Revenue from Tequila, 2015**

Across the years 2012 to 2016, a recurring pattern emerges in the sales of Tequila bottles, characterized by peaks in the summer, declines during winter, and renewed increases in subsequent years. In 2012, the pattern begins with a rise in January and peaks in summer, followed by a decline in winter and a subsequent rise in early 2013. This cycle repeats 2014 and continues into 2015-2016, with sales ranging between 2000 and 4000 at the lower end and 4000 to 6000 at the higher end. Even though there are variations, we can see that here is a consistent trend of seasonal fluctuations in bottle sales over these years. Figures 9 through 12 show the ACF and PACF plots for bottles sold and revenue:

**A graph with blue dots and numbers

Description automatically generated**

**Figure 9: Total Bottles of Tequila Sold, ACF**

**A graph with blue dots and numbers

Description automatically generated**

**Figure 10: Total Bottles of Tequila Sold, PACF**

**A graph with blue dots and numbers

Description automatically generated**

**Figure 11: Total Revenue from Tequila, ACF**

**A graph with blue dots

Description automatically generated**

**Figure 12: Total Revenue from Tequila, PACF**

Based on what was seen in these plots, some techniques for smoothing the data would later be attempted, but first the raw data would be used to see what kind of models could be produced.

**Modeling:**

To build the ARIMA models with the raw data, the p, d, and q values that should be used to produce the best possible results needed to be established. To accomplish this, one function was written that took an inputted dataset, separated out 30% of the data for testing, built an ARIMA model using the other 70% of the data and an inputted combination of p, d, and q values, made several predictions equal to the number of datapoints in the test data, used the test data to determine the accuracy of the model, then returned an RMSE score to reflect that accuracy. Another function was also written that took an inputted data frame, range of p values, range of d values, and range of q values, iterate through all the possible combinations of p, d, and q values, feed those values along with each column in the provided data frame (in this case, each column would be the two stats we were analyzing, total bottles sold and revenue) into the previously mentioned function, store the resulting RMSE value into a data frame, then return the data frame containing all the possible p, d, and q values for each stat along with the resulting RMSE scores from that respective model. This second function was then run to test p values of 0 to 8, d values of 0 to 3, and q values of 0 to 8.

Building the model and testing the scores for just one combination of p, d, and q values could take a while, so running the function with 243 potential combinations took 10+ hours to finish. To ensure the data was not lost if the function timed out before completion, a file was created and exported after the RMSE score was calculated for each p, d, and q combination and stat. This function did in fact need to be stopped and started multiple times, but the 243 individual files containing the required RMSE scores were still collected, allowing us to import them back in, then build one collective data frame displaying the best p, d, and q values. This resulting data frame can be seen in figure 13:

**A screenshot of a data

AI-generated content may be incorrect.**

**Figure 13: Excerpt of Resulting Dataframe from p, d, q Values Testing Function (Raw Data)**

Another function was now built that automatically went through this data frame for each stat and pulled out the respective p, d, and q combination that produced the best RMSE score, the results of which can be seen in figure 14:

**A close-up of a sign

AI-generated content may be incorrect.**

**Figure 14: Output of Best p, d, q Values Function (Raw Data)**

All the above collected data could now be used to build and test the models. The predetermined p, d, and q combinations were used with the 2012-2016 data to build and then export the best models for each stat so that they could easily be imported and used again. Another function was now created that took in training data, testing data, the best p, d, and q combinations for each stat, and whether the bias value should be calculated and included in the predictions. Two models were tested for each stat; one with the bias value calculated, and one with the bias value just set to zero. When the raw data was used to predict the remaining 2016 and entire 2017 data with the bias value calculated, the resulting RMSE for the total bottles sold was 1296.772 and for the revenue was 23445.465. Figures 15 and 16 display plots of these predictions along with the actual data:

**A graph showing a red and blue line

AI-generated content may be incorrect.**

**Figure 15: Total Bottles Sold, 2016/ 2017 Predictions (red) Vs Actual Totals (blue), Raw Data, Bias**

**A graph showing a graph of sales

AI-generated content may be incorrect.**

**Figure 16: Total Revenue, 2016/ 2017 Predictions (red) Vs Actual Totals (blue), Raw Data, Bias**

In both plots, the predictions follow the general trends in the actual data, but never reach the extremes of the highest and lowest values. When the same predictions were made using no bias, the RMSE for total bottles sold was 1297.307 and for revenue was 23462.086, so slightly higher than both values when bias was used. Figures 17 and 18 display these plots:

**A graph showing a red and blue line

AI-generated content may be incorrect.**

**Figure 17: Bottles Sold, 2016/ 2017 Predictions (red) Vs Actual Totals (blue), Raw Data, No Bias**

**A graph showing a graph of sales

AI-generated content may be incorrect.**

**Figure 18: Total Revenue, 2016/ 2017 Predictions (red) Vs Actual Totals (blue), Raw Data, No Bias**

Various smoothing processes were now applied to the data to see if accounting for the various ebbs and flows improved our ability to make predictions. The first method used was a differencing approach. In this approach, an integer was designated as an interval value, which was then used for 2 purposes. The interval number established what timestamp and corresponding stat values in our timeseries data would be the first to undergo the smoothing process. For example, if our timeseries data consisted of 10 straight days, and the interval was assigned the number 3, then the fourth day would be the first to have its’ values smoothed (by default in Python index’s, 0 is the first index, so 3 would be the fourth index). The values for that timestamp and each timestamp after it would then be adjusted by subtracting the values for the timestamp that came an interval’s worth of distance before. So once again, if we were working with a timeseries consisting of 10 straight days, and the interval was 3, then the values on the fourth day (index = 3) would have the values from the first day subtracted (index of 3 minus the interval of 3 = index of 0, which is the first day). The values on the fifth day (index = 4) would have the values from the second day subtracted (index of 4 minus the interval of 3 = index of 1, the second day). This pattern would continue for all the remaining values.

To determine which interval number should be used for each stat when applying the difference method, another function was created that would iterate through an inputted range of numbers, apply the difference function to the inputted data using each number from the inputted range as the interval, perform the adfuller test on the newly smoothed data, save the output from the adfuller test to a data frame, then return the data frame. The interval values that produced the lowest ADF Statistic would then be used when the difference function was applied to the data during the modeling process. For the total bottles sold stat, the interval chosen was 658, and for the total revenue, the interval chosen was 654.

Similar to the raw data, the best p, d, and q value combinations needed to be found, but now when testing the ARIMA models and looking for the best RMSE values, the difference function would need to be applied to the data first, followed by an inverse difference function that would revert each prediction back to its actual value, as opposed to its differenced value. Once again, this process took 10+ hours to complete, but the output was gathered intermittently, and the best p, d, q value combinations were able to be collected. These values were again used to build and test two models for each stat; one with the bias calculated and one with no bias. The model built for total bottles sold using the differenced data and the calculated bias resulted in an RMSE of 1702.026. This model for total revenue resulted in an RMSE of 28028.655. With no bias, the resulting RMSE for total bottles sold was 1706.505 and for total revenue was 28045.249. Figures 50 through 53 display the plots for these models:

**A graph showing a graph of sales

AI-generated content may be incorrect.**

**Figure 19: Bottles Sold, 2016/ 2017 Predictions (red) Vs Actual Totals (blue), Difference Data, Bias**

**A graph showing a graph of sales

AI-generated content may be incorrect.**

**Figure 20: Total Revenue, 2016/ 2017 Predictions (red) Vs Actual Totals (blue), Difference Data, Bias**

**A graph showing a graph of a number of bottles sold

AI-generated content may be incorrect.**

**Figure 21: Bottles Sold, 2016/ 2017 Predictions (red) Vs Actual Totals (blue), Diff Data, No Bias**

**A graph showing a graph of sales

AI-generated content may be incorrect.**

**Figure 22: Total Revenue, 2016/ 2017 Predictions (red) Vs Actual Totals (blue), Diff Data, No Bias**

The next smoothing method tested was the pandas.rolling approach. The pandas.rolling method smooths out data by establishing a moving window equal to a length. For each observation in the data being smoothed, several previous observations are collected equal to the length of the moving window, then all the values are either summed together or averaged out, depending on what is selected. That summed or averaged value is then applied to either the center most observation within the window, or the furthest right observation, once again depending on which option is chosen. Depending on the size of the window, a certain number of values at the beginning of the dataset may end up as “NaN” once the rolling method is applied, since there wouldn’t be any values before them to include in the calculation, but this can be avoided by choosing to require only a minimum of 1 value to complete each calculation, despite however long the moving window is.

Similar to the difference smoothing method, we created a function that used an input range of values as the moving window length, rolled the data using that window length, then calculated its adfuller statistics. This function also allowed us to test whether centering the data to be labeled was the most effective approach (which we would determine is not the case and did not use the center option for any of our modeling). Upon the initial testing of the window lengths, moving windows of length 66 for total bottles sold and 64 for revenue were observed to produce the best adfuller statistics. Further testing, however, would show that moving windows this large would not work for our purposes. As was done with the difference smoothing process, after the rolled data was put through the ARIMA model, the output predictions needed to be inversed back to non-rolled values to be interpreted. It appeared, however, that when the windows were too large, too many large outliers would be pulled into the window, so the reversed predictions were much larger than any of the previous data. This issue was still seen even after dropping the moving window sizes down to 23 and 25. Eventually it was determined that the best option was to use rolling windows of only size 2, for both the total bottles sold and the total revenue.

The lengthy process of collecting the best p, d, and q values was once again completed for the models built using the rolled data, and then the models were once again tested using bias and no bias to evaluate their ability to predict the remaining 2016 and entire 2017 data. The model for total bottles sold using rolled data and a bias resulted in an RMSE of 1517.702 For total revenue, this model resulted in an RMSE of 26833.603. When no bias was used, the resulting RMSE for total bottles sold was 1517.972, and for total revenue was 26834.944. The plots for these models can be seen in figures 22 through 25.

**A graph showing a blue and red line

AI-generated content may be incorrect.**

**Figure 22: Bottles Sold, 2016/ 2017 Predictions (red) Vs Actual Totals (blue), Rolled Data, Bias**

**A graph showing a graph of sales

AI-generated content may be incorrect.**

**Figure 23: Total Revenue, 2016/ 2017 Predictions (red) Vs Actual Totals (blue), Rolled Data, Bias**

**A graph showing a blue and red line

AI-generated content may be incorrect.**

**Figure 24: Bottles Sold, 2016/ 2017 Predictions (red) Vs Actual Totals (blue), Rolled Data, No Bias**

**A graph showing a graph of sales

AI-generated content may be incorrect.**

**Figure 25: Total Revenue, 2016/ 2017 Predictions (red) Vs Actual Totals (blue), Rolled Data, No Bias**

The final smoothing process tested was a savitzky-golay filter, using the savgol\_filter function from the scipy library. Kineticstoolkit.uqam.ca describes the Savitzky-Golay filter as “a generalization of the moving average. Instead of taking the mean of the n points of a moving window, the Savitzky-Golay filter fits a polynomial of a given order over each window.” So instead of calculating the average of a moving window (like we used in the previous pandas.rolling technique), the savgol\_filter applies a polynomial to the window, then is evaluated at the center of the window. A function would once again need to be created to test the best possible value for the length of the polynomial window, along with the best possible value for the “polyorder,” or the order of the polynomial used to fit the data. Based on the data inverting issue discovered previously with large moving windows and the rolling operation, a few different large and small values that were found to have good adfuller statistics were tested to ensure there would be no problems inverting the savgol\_filter data. For total bottles sold, the best window to use was discovered to have a length of 256 and a polynomial value of 5. For total revenue, the best window was determined to have a length of 208 and a polynomial value of 5.

As was done with the previous models, the best p, d, and q values were patiently tested and collected using data smoothed by the savgol\_filter, and then the resulting models were tested using bias and no bias to evaluate their accuracy and reliability. The model for total bottles sold using savgul data and a bias resulted in an RMSE of 1784.317. For total revenue, this model resulted in an RMSE of 32606.630. When no bias was used, the resulting RMSE for total bottles sold was 1784.303, and for total revenue was 32606.637. The plots for these models can be seen in figures 26 though 29.

**A graph showing a line of blue and red

AI-generated content may be incorrect.**

**Figure 26: Bottles Sold, 2016/ 2017 Predictions (red) Vs Actual Totals (blue), Savgul Data, Bias**

**A graph showing a graph of sales

AI-generated content may be incorrect.**

**Figure 27: Total Revenue, 2016/ 2017 Predictions (red) Vs Actual Totals (blue), Savgul Data, Bias**

**A graph showing a line of blue and red

AI-generated content may be incorrect.**

**Figure 28: Bottles Sold, 2016/ 2017 Predictions (red) Vs Actual Totals (blue), Savgul Data, No Bias**

**A graph showing a graph of sales

AI-generated content may be incorrect.**

**Figure 29: Total Revenue, 2016/ 2017 Predictions (red) Vs Actual Totals (blue), Savgul Data, No Bias**

**Results:**

Figure 30 shows the RMSE results for all the different models built using the 2012 through 2016 dataset:

**A table with numbers and symbols

AI-generated content may be incorrect.**

**Figure 30: All Models RMSE Scores for 2012-2016 Data**

As can be seen, the models built using the raw, unsmoothed data with the bias added in had the best RMSE scores. After this testing was completed, the alcohol sales data from 2018 through November 2023 was found. To further test our model, we imported in this new data and combined it with our already existing dataset. Since our previous tests showed the best model could be built using the raw data, our model built using the additional data from 2018 through 2023 also did not use any smoothing techniques. For one final time, we used the function to collect the best possible combination of p, d, and q values (the additional data stretched the runtime of this function to 4 days) to determine if they changed with more data included. It turned out however that the best p, d, and q values for the model with the additional data were the same as the values for the model with the original data.

The model's accuracy using the additional data was evaluated by separating the 2023 data for testing. The models were trained using raw data from 2012 through 2022, with bias, and then tested using the data from 2023 resulted in an RMSE of 2489.866 for total bottles sold and 59401.172 for total revenue. When no bias was used, the RMSE for total bottles sold was 2489.625 and for total revenue was 59396.70. Figures 31 through 34 display the plots for these models:

A graph showing a graph of a graph

AI-generated content may be incorrect.

**Figure 31: Bottles Sold, 2023 Predictions (red) Vs Actual Totals (blue), Raw Data, Bias**

A graph showing a graph of sales

AI-generated content may be incorrect.

**Figure 32: Total Revenue, 2023 Predictions (red) Vs Actual Totals (blue), Raw Data, Bias**

**A graph showing a graph of a graph

AI-generated content may be incorrect.**

**Figure 33: Total Bottles, 2023 Predictions (red) Vs Actual Totals (blue), Raw Data, No Bias**

**A graph showing a graph of sales

AI-generated content may be incorrect.**

**Figure 34: Total Revenue, 2023 Predictions (red) Vs Actual Totals (blue), Raw Data, No Bias**

These models could now be used to predict what the December 2023 tequila sales numbers would look like. If the 31 days (about 1 month) of December are forecasted out, it is predicted that ~161038 bottles of tequila will be sold at an average of ~5195 bottles per day. It is also predicted that the total revenue from these sales will total ~$3484176.76, equaling about $112392.80 per day. Figures 35 and 36 display the plots for these daily values:

**A graph with red lines

Description automatically generated**

**Figure 35: Bottles Sold, December 2023 Predictions**

**A graph with red lines

Description automatically generated**

**Figure 36: Total Revenue, December 2023 Predictions**

Figures 37 and 38 plot the December 2023 predictions for total bottles sold and total revenue against all the actual December values for 2012 through 2022:

**A graph of lines and numbers

Description automatically generated with medium confidence**

**Figure 37: Total Bottles, December 2023 Predictions vs. December 2012-2022**

**A graph of lines and numbers

Description automatically generated with medium confidence**

**Figure 38: Total Revenue, December 2023 Predictions vs. December 2012-2022**

Figures 39 and 40 plot the December 2023 predictions for total bottles sold and total revenue against the mean December values for the years 2012 through 2022:

**A graph of a graph showing the price of a stock market

Description automatically generated with medium confidence**

**Figure 39: Total Bottles, December 2023 Predictions vs. Mean December Numbers for 2012-2022**

**A graph of a price

Description automatically generated with medium confidence**

**Figure 40: Total Revenue, December 2023 Predictions vs. Mean December Numbers for 2012-2022**

**References:**

<https://data.iowa.gov/Sales-Distribution/Iowa-Liquor-Sales/m3tr-qhgy>

<https://www.kaggle.com/datasets/residentmario/iowa-liquor-sales>

https://kineticstoolkit.uqam.ca/doc/filters\_smooth\_savgol.html

<https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.savgol_filter.html>

<https://data.iowa.gov/Sales-Distribution/Iowa-Liquor-Sales/m3tr-qhgy/data>

<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.rolling.html>

<https://towardsdatascience.com/improve-your-data-science-workflow-with-rolling-functions-in-pandas-69c4b44e3f7e>