A Peak Into Walmart's Sales: How To Maximize Business Profit

CS 5010

Submitted by

Arti Patel, Elizabeth Driskill, Mariah Hurt, and Sania Rasheed
Ap8qk, Ekd6bx, Mes3wv, Sr2xn

UNIVERSITY of VIRGINIA

Data Science Institute

UNIVERSITY OF VIRGINIA
DATA SCIENCE INSTITUTE

July 2019

Project Report: CS 5010

Introduction:

Our project involved analyzing a data set with historical sales data for 45 different Walmart stores located in various regions. We hoped to find out whether certain factors had a positive or negative effect on each store's weekly sales, and we also wanted to determine which specific stores were over performing or at risk of going out of business. We were able to visualize trends in weekly sales over time as well as trends of other variables over time to see if there was any correspondence between a spike in sales and an increase or decrease in a particular variable. From a marketing perspective, our goal was to extract useful information from this data set so that we could offer suggestions to Walmart executives about how they should move forward with their company and sales tactics in the future. Ultimately, we performed multiple queries on the data, and we used the results to come up with a final conclusion about which holidays have higher effects on sales and which stores seem to be lagging behind.

The Data:

We obtained our data set from kaggle.com, and it initially included five separate csv files, each with different information. After downloading the csv files and skimming through each one, we decided to focus on the features csv and the training data in order to perform our analysis. The features csv includes the following columns: Store (the store number), Date (the week in the format YYYY-MM-DD), Temperature, Fuel Price, MarkDown1-5 (promotional markdowns), CPI (consumer price index), Unemployment (the unemployment rate), and IsHoliday (whether or not the week is a special holiday). The training data includes the following columns: Store, Dept (department number), Date, Weekly_Sales (sales for the given department in the given store), and IsHoliday. We chose these two files because we were interested in analyzing how Walmart weekly sales are affected by certain predictor variables, and we also wanted to search for stores that had abnormal average sales.

The train csv contains weekly sales data for each store broken down by department, so there are many more rows of data in the train csv than there are in the features csv. Because each store had up to 98 different departments, and many stores did not all have the same number of departments, we chose to sum the weekly sales for each department using a groupby function to obtain an overall weekly sales value for each store as a whole. Additionally, we were more interested in general store trends rather than specific information within departments, so this further contributed to our

choice to reduce the data. The following code demonstrates how we reduced the train csv data:

```
train['datestore'] = list(zip(train.Date, train.Store))
sums = train['Weekly_Sales'].groupby([train['Date'],train['Store']]).sum()
```

In order to perform actual analysis of the data, we figured it would be most useful to merge the features csv and train csv files into one data frame so that weekly sales and the variables that could potentially influence weekly sales would all be grouped together. First, we made a new column in the original features data frame that was a tuple of Date and Store. This allowed us to create a unique identification that we could merge the two data frames based on:

```
features['datestore'] = list(zip(features.Date, features.Store))
```

Then, we had to convert the series of sums produced from the reduction of the train csv data above into a data frame and change the index of this data frame into an actual column that we would be able to merge on:

```
sumsdf = pd.DataFrame(sums)
sumsdf["datestore"] = sumsdf.index
```

Finally, we were able to merge these two data frames and produce one larger data frame, the first few rows of which can be seen below:

Index	Store	Date	Temperature	Fuel_Price
0	1	2010-02-05 00:00:00	42.31	2.572
1	1	2010-02-12 00:00:00	38.51	2.548
2	1	2010-02-19 00:00:00	39.93	2.514
3	1	2010-02-26 00:00:00	46.63	2.561
4	1	2010-03-05 00:00:00	46.5	2.625
5	1	2010-03-12 00:00:00	57.79	2.667

MarkDown1	MarkDown2	MarkDown3	MarkDown4	MarkDown5
nan	nan	nan	nan	nan
nan	nan	nan	nan	nan
nan	nan	nan	nan	nan
nan	nan	nan	nan	nan
nan	nan	nan	nan	nan
nan	nan	nan	nan	nan

CPI	Unemployment	IsHoliday	datestore	Weekly_Sales
211.096	8.106	False	('2010-02-0…	1.64369e+06
211.242	8.106	True	('2010-02-1…	1.64196e+06
211.289	8.106	False	('2010-02-1…	1.61197e+06
211.32	8.106	False	('2010-02-2…	1.40973e+06
211.35	8.106	False	('2010-03-0…	1.55481e+06
211.381	8.106	False	('2010-03-1…	1.43954e+06

We accomplished this by executing the following code:

```
merge = features.merge(sumsdf, left_on='datestore', right_on='datestore')
```

It should be noted that each of the MarkDown columns included many “NA” values. However, we did not use the information in these columns in our analysis, so these missing values did not affect our final conclusions, and we did not need to exclude these rows from our data frame. Our last step in cleaning our data was to convert the type of the Date column to be read as a date with the format Year-Month-Day. There is a function in pandas that allowed us to do this:

```
merge["Date"] = pd.to_datetime(merge["Date"])
```

Beyond the Original Specifications:

We thought that a useful application of our data analysis would be to create an interface that allows a user to input a specific store that they are interested in and compute the average weekly sales for that store. This provides the user with an opportunity to compare different stores to each other and potentially gain valuable

insight into how each store is performing. This could be especially useful to Walmart executives in determining which stores might need extra financial resources or which stores might be at risk for going out of business. The code for this extra-credit portion of the project can be seen here:

```
def averageSales(num):
    average=df_weekly['Weekly_Sales'].groupby(df_weekly['Store']).mean()
    print(average[num])
    return(average[num])]

#Get store number for user
num = int(input("Hello! Please input your store number."))

print ("For store number " + str(num) + " the average weekly sales in USD is: $ "
       + str(averagesales(num)))
```

Results:

Our first query involved examining scatterplots of different variables to see how they could potentially affect weekly sales. Four scatterplots of fuel price, temperature, unemployment, and CPI, all plotted against weekly sales, can be seen here:

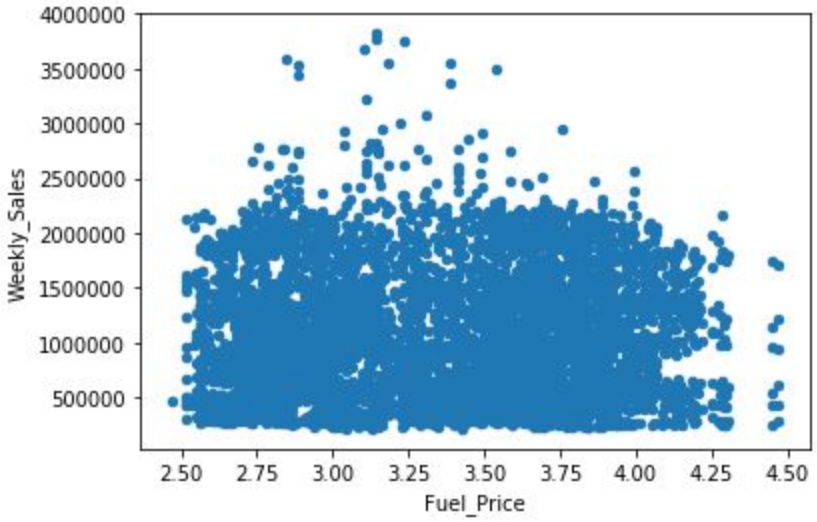

Figure 1: Scatterplot of Fuel Price vs. Weekly Sales. This scatterplot shows that as fuel prices increase, weekly sales decrease. There is an inverse relationship between the two variables.

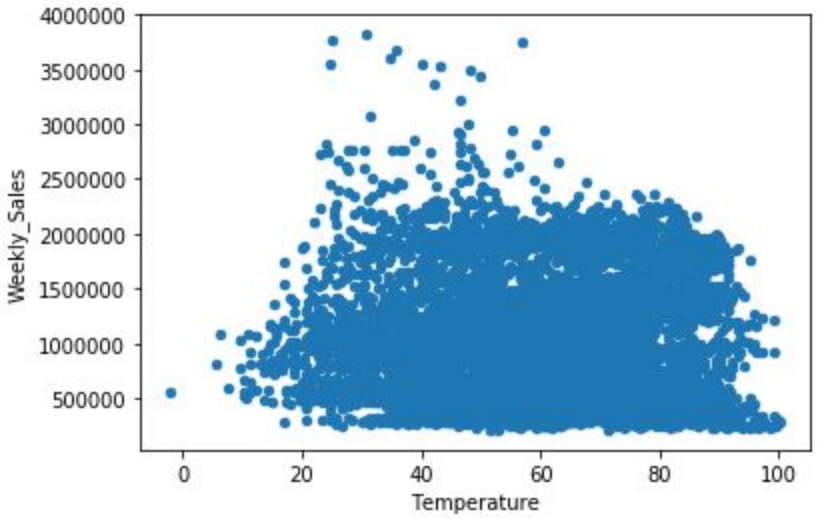

Figure 2: Scatterplot of Temperature vs. Weekly Sales. This scatterplot shows that temperature ranges at the extreme ends of the spectrum generate low weekly sales.

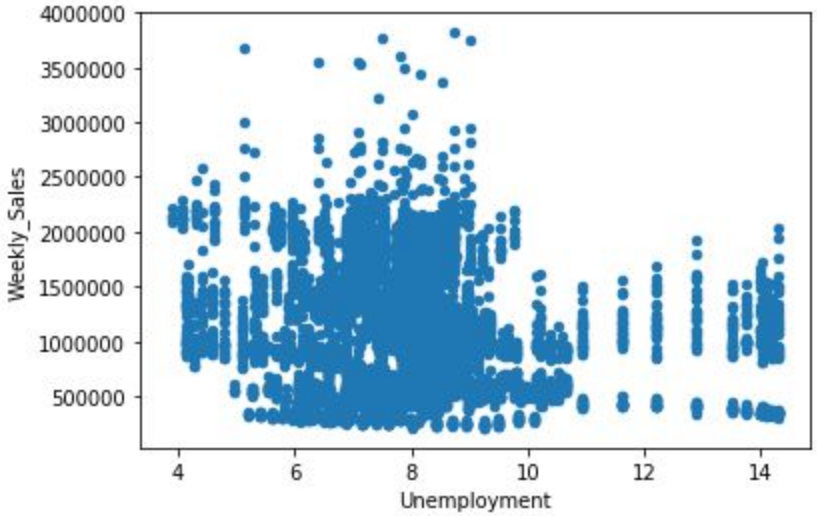

Figure 3: Scatterplot of Unemployment vs. Weekly Sales. This scatterplot shows that as unemployment rates increase, weekly sales decrease. There is an inverse relationship between the two variables.

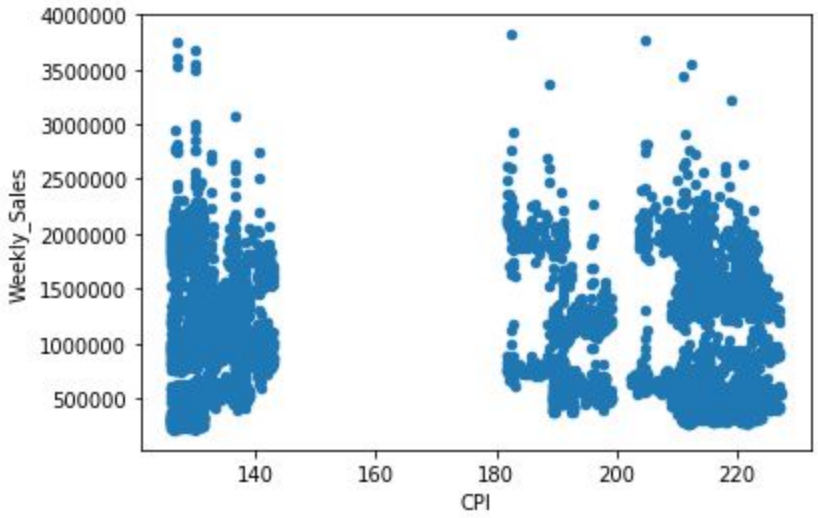

Figure 4: Scatterplot of CPI (Consumer Price Index) vs. Weekly Sales. This scatterplot shows that there is no correlation between CPI and Weekly Sales.

From these plots, we observed that the highest weekly sales tended to occur when fuel prices and unemployment rates were lower. Lower fuel prices and unemployment rates are factors that are typically associated with a good economy, so these trends are expected. Additionally, the highest weekly sales mostly occurred when the temperature was within the range of 20-40 degrees Fahrenheit. This temperature range indicates that sales are higher during the winter season. This could be attributed to the holiday season which includes Thanksgiving, Black Friday, Christmas, and after Christmas sales. The plot for CPI versus weekly sales, however, was inconclusive because there were similar trends across a wide range of values. This might be due to the fact that our data set only includes 2 calendar years, and CPI does not fluctuate very frequently.

Our next task was to visualize how weekly sales change over time for each store. Out of the 45 graphs that were produced, we chose three graphs that depict a store with steadily increasing sales, seasonally fluctuating sales (the most commonly observed pattern in our dataset), and steadily declining sales:

Figure 5: Weekly sales in US dollars plotted over time for store number 38. This plot shows a store with steadily increasing sales over time and no apparent seasonal sales spikes.

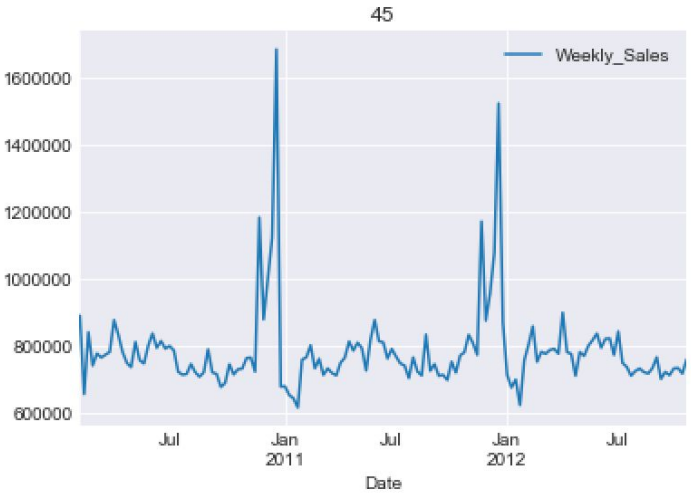

Figure 6: Weekly sales in US dollars plotted over time for store number 45. This plot shows large seasonal sales spikes around the winter holiday season in December. For the 45 Walmart stores analyzed, this was the most common sales pattern observed.

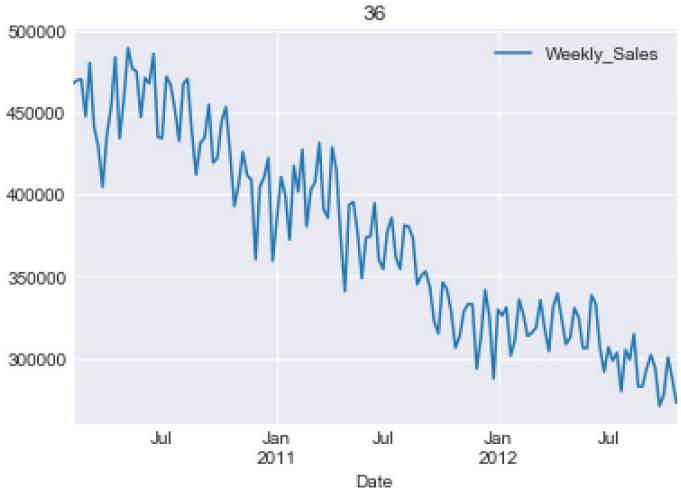

Figure 7: Weekly sales in US dollars plotted over time for store number 36. This plot shows a store with steadily declining sales over time. There are no seasonal sales spikes and the store's sales numbers steadily decrease.

As can be seen in the second graph that represents the sales trends for a typical Walmart store, sales tend to peak around the month of December. We hypothesized that this might be due to the holiday season, as people will likely purchase more things from Walmart during this time. The first graph shows a store that had exceptionally high weekly sales that increased over time. Walmart executives might be interested in doing more research on this particular store so that they can potentially find out what may be causing this store to do abnormally well and apply their findings to stores in other locations. The third graph shows a store with lower sales that also tended to remain low over time. This is another store that Walmart may want to examine further to decide how to improve their sales or whether or not they should shut down the store altogether.

It is important to keep in mind that there are many factors that play into sales for a store. The geographic attributes of the area and the expenditures and investment behind running the store may have effects on the sales of the store. However, we do not have data on these factors, so we are unable to include them in our analysis. We do have data on the size of the store, and we created a visual of one snapshot in time of the monthly sales for each store, where the shade of the bubble indicates the size of the store (the darker the shade, the bigger the store).

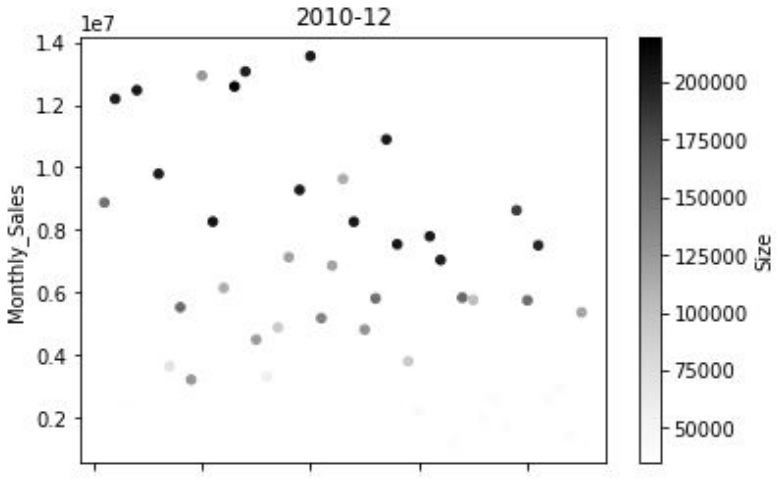

Figure 8: The scatter plot shows that most of the stores with highest sales tend to be of size greater than 125,000.

We also created a graph that depicts how fuel price changes over time for store number 45, which seemed to exhibit similar patterns to many other stores. From this graph, we noticed that fuel price increased significantly from 2011 to 2012. Additionally, fuel price tended to reach a local minimum around the winter holiday season. This could help to explain why sales often increase during the holiday season, as customers might be willing to spend more money at Walmart when fuel prices are low.

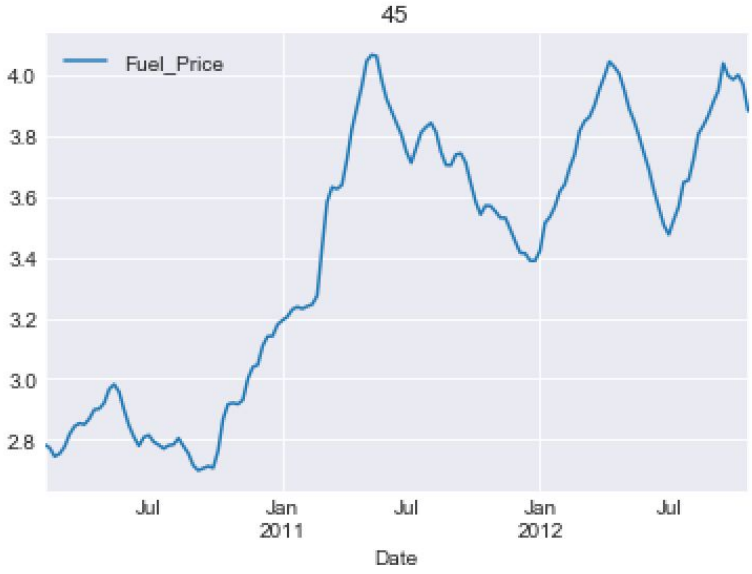

Figure 9: The pattern seen for fuel price over time was very similar for all stores examined. Fuel price fluctuates seasonally while at the same time increasing over time.

Our dataset also included data on the local unemployment rates for each store area over time. When plotted over time the unemployment rates decreased from 2010 through 2012. This pattern was observed over all 45 store locations. This indicated to us that unemployment may be more of a national phenomenon than a local one. When examining store number 36, a store that is in obvious decline, we thought perhaps we would find unusually high unemployment in the area. However, we found that unemployment decreased from 2010 to 2012 even for store 36 and so the declining sales cannot be attributed to local unemployment.

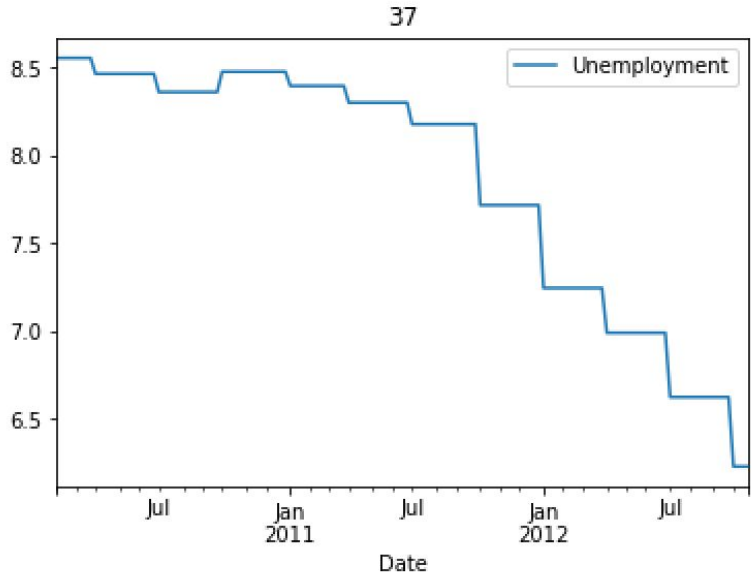

Figure 10: The pattern seen for unemployment over time shows a pattern that continually decreases. This figure depicts the data for store number 37, but it is characteristic of what we observed for all 45 stores.

For our third query, we wrote a function to compute the average weekly sales for each store. From this, we could then determine which stores had the highest and lowest averages. As mentioned earlier, this information might be useful to keep track of which stores are achieving high sales and which stores might be under performing. We also expanded this function to create a user interface, which was discussed in the extra credit section. A bar graph of the average weekly sales for each store can be seen here:

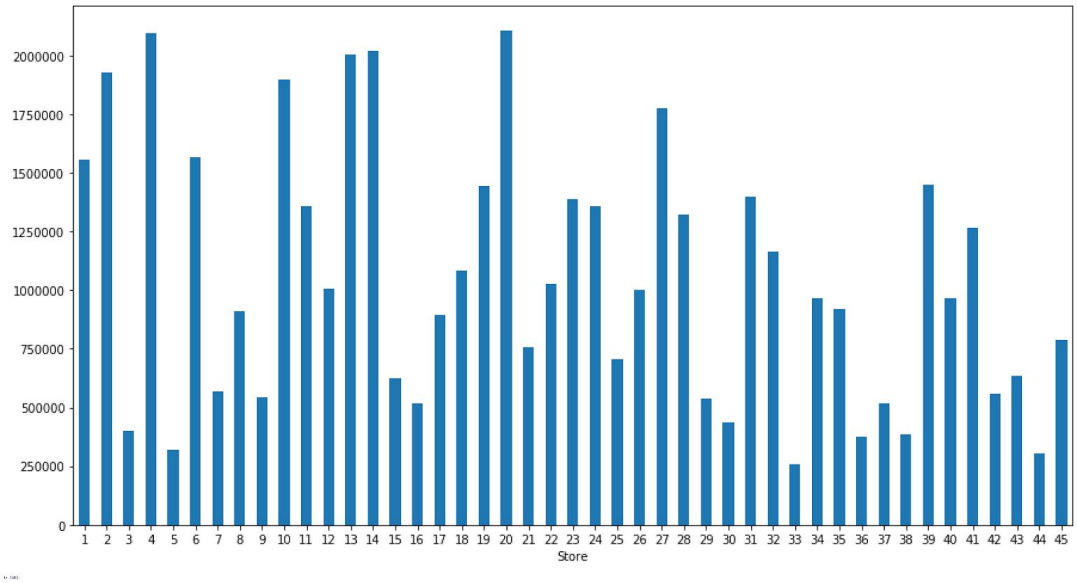

Figure 11: The average weekly sales in US dollars is shown above for each store.

Our final task involved creating a separate data frame that contained the median weekly sales and the IsHoliday column. We wanted to determine how sales changed during holiday weeks so that we could offer a better suggestion to the company about how they should approach their marketing and sales tactics during these times. A bar graph can be seen below that shows the median weekly sales for each store during holiday seasons (blue) vs. non-holiday seasons (orange). There were a few outliers in our data set; as can be seen in Figure 12, store numbers 16 and 17 had higher weekly sales in non-Holiday seasons than holiday seasons. This is not expected and should be investigated by Walmart executives. We also created a bar graph that has the average of the median weekly sales across all stores during holiday vs. non-holiday seasons. As expected, the average of the median weekly sale was higher for holiday seasons as compared to non-holiday seasons.

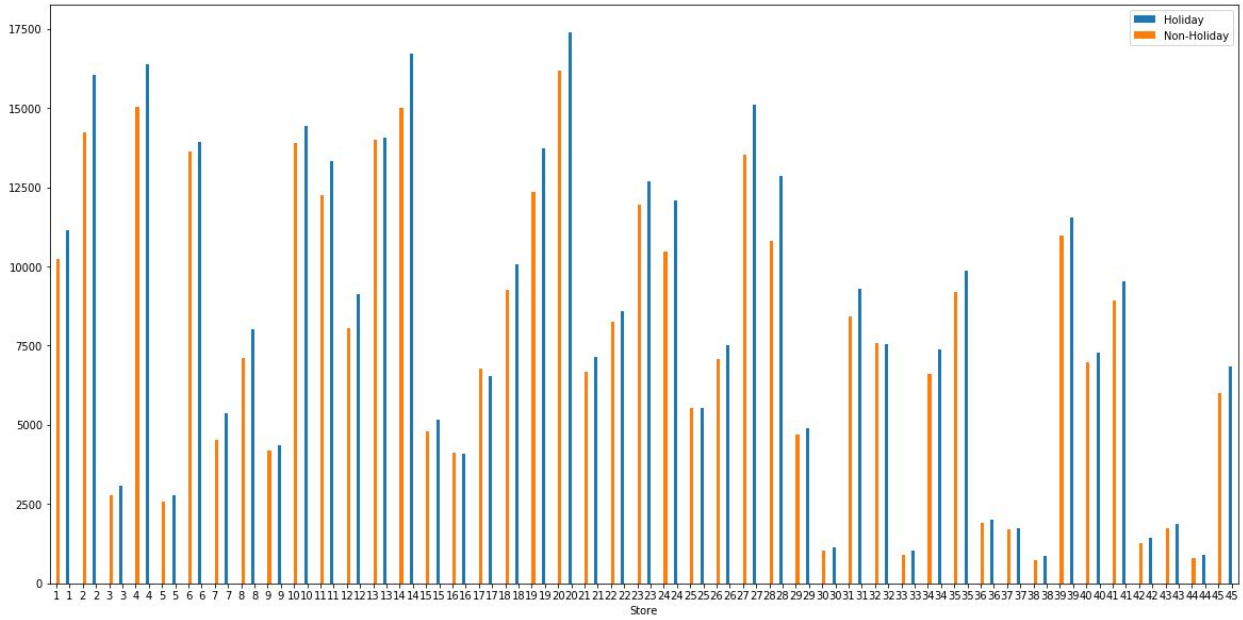

Figure 12: This is a bar graph that shows the difference in weekly sales between holiday and non-Holiday seasons for each store.

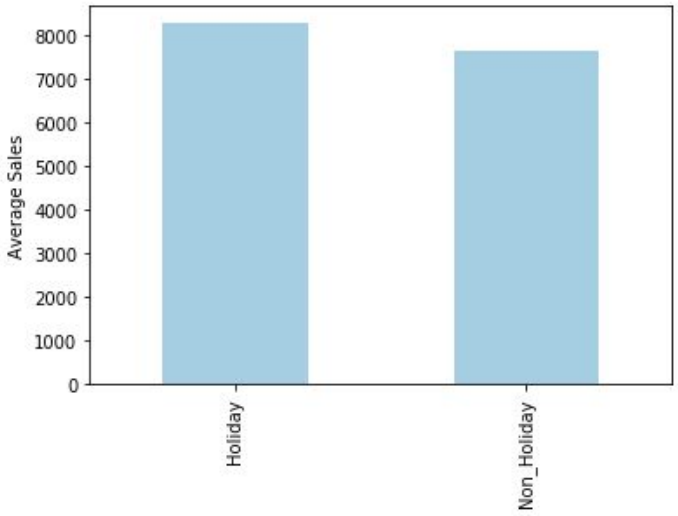

Figure 13: This bar graph shows the average difference in sales between holiday and non-Holiday seasons for all stores combined.

For the majority of stores, the median weekly sales increased during holiday seasons compared to non-holiday seasons. The overall average sales across all stores were also higher during holiday seasons than non-holiday seasons. Therefore, it might be beneficial for Walmart to have more markdowns or other recruiting events during holiday seasons in order to result in the largest profit.

Testing:

We conducted a few tests to verify that our code meets the specification and design requirements that we hoped to implement. First, when reading the data into Python, we used the head and tail functions to ensure that our data looked similar to the csv files. After merging the data, we also wanted to check that our new data frame had the correct number of rows because we would not want to proceed if we grouped the data incorrectly. The code for this portion of our test-driven development can be seen here:

```
features = pd.read_csv("features.csv")
train = pd.read_csv("train.csv")

print(features.head)
print(features.tail)

print(merge.head)
print(merge.tail)
print(merge.count) # examine the dimensions of the merged data
```

We conducted a unit test to verify that the average sales function was working correctly. This function is supposed to calculate the average sales for a user specified store location. For this we created a test case where the input was store number five and then used an "assertEqual" statement to compare the output of the test case function to the expected value of the function. We conducted a similar test for the function averageDecTemp, which calculates the average December temperature for a user specified store location. Both of these tests are examples of black box testing because they test the output based on a specific input.

```
----------------------------------------------------------------------
Ran 1 test in 0.009s

OK
..318011.8104895105
50.162
318011.8104895105

----------------------------------------------------------------------
Ran 2 tests in 0.013s

OK
```

Figure 14: This is output from the unit tests for the “averageDecTemp” function and the “averageSales” function. This output indicates that the code is passing these two tests and is working as intended.

Conclusions:

Overall, our program allowed us to visualize trends in weekly sales across 45 different Walmart stores over time and hypothesize about which variables, such as unemployment or fuel price, might be correlated with an increase or decrease in sales. Many stores showed an increase in sales around the winter holiday season as well as an increase in sales during other holiday weeks. There was a noticeable peak in weekly sales around the temperature range of 20-40 degrees Fahrenheit. Given this information, we would recommend to Walmart executives to allocate more funding towards advertising or other recruiting events during the winter holiday season, as this is typically the time of the year when they experience the greatest increase in sales. They could also look into applying the sales tactics that they incorporate during the winter holiday season to other holiday weeks, such as Halloween, Easter, and the 4th of July in an attempt to increase sales during these weeks as well.

From our analysis, we were also able to determine which stores showed abnormal trends in weekly sales. For example, store number 38 showed a steady increase in weekly sales, but it did not show spikes around the winter holiday season. This store could be examined further to see why it does not experience the same trends as many of the other stores. The store could potentially be located in a region of the country where the majority of the population does not celebrate Christmas or other holidays during this time, so the populace may not change their Walmart spending habits. Store number 38 might also be a helpful reference to use as an example for stores that are not performing as well since its weekly sales were exceptionally high. Another store that could be of interest is store number 36, which exhibited a steady decrease in weekly sales over time. Walmart may want to think about closing this store or incorporating new management and sales tactics in order to reverse the downward trend.

This project could be expanded further to include the location of the stores and examine whether or not the size of the store or the size of the city in which it is located has any effect on its weekly sales. Currently, the data does not include information on each store’s location, so we were not able to explore this option. Another application of our project would be to use regression models to predict the weekly sales for a certain store. This might be helpful in deciding where to open new Walmart stores, as Walmart management could predict how a store would perform given specifications about its location, size, and maybe sales information and trends from surrounding stores. Given

more time, we would have liked to analyze our data further, but we still were able to extract valuable information from the queries that we did perform.

Sources

https://pandas.pydata.org/pandas-docs/stable/user_guide/groupby.html
https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting/data
https://stackoverflow.com/questions/16031056/how-to-form-tuple-column-from-two-columns-in-pandas
https://www.geeksforgeeks.org/creating-a-dataframe-from-pandas-series/
http://jonathansoma.com/lede/algorithms-2017/classes/fuzziness-matplotlib/understand-df-plot-in-pandas/