Further sales prediction Project
Final Report

Apr 30th 2020
To: Professor Yuan Xu

From: Group 4
Baiting Gai
Mengying Jiang
Xingyu Ye
Xixiang Chen
Yuyan Wang

1. Data Description

The project is based on a competition in Kaggle. The dataset consists of time series data and daily sales data is provided by 1C, one of the most famous software companies. Our goal is to build up a model that can describe the daily total sales for all stores from 2013 to 2015.

We combine two datasets provided by Kaggle for this competition: the daily historical sales data from January 2013 to October 2015 and the supplemental information about the items. Then, useful features are selected from the dataset. The final dataset has 2,935,849 observations with 7 variables: date, date_block_num (a consecutive number representing month for convenience. January 2013 is 0, February 2013 is 1, etc.), shop_id, item_id, item_price, item_cnt_day (number of products sold daily), item_category_id.

2. Exploratory Data Analysis

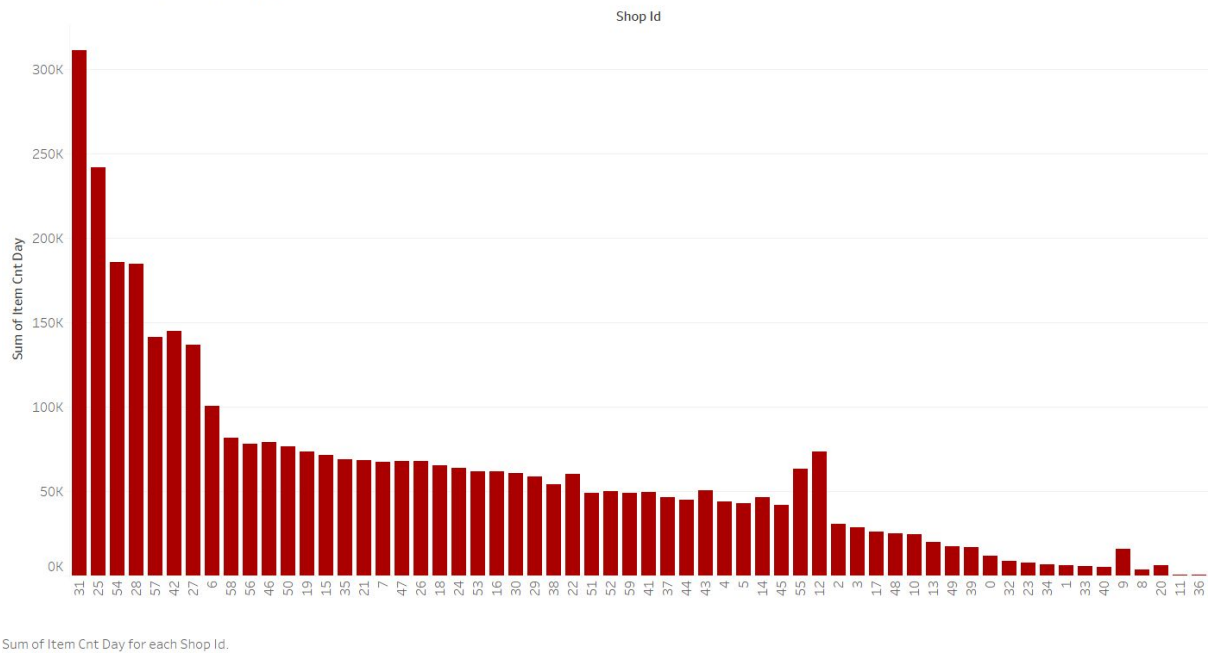
Since there are no missing values or null values, we only need to deal with the dates. The variable $date (year-month-day) is separated into new features ($year, $month, $date, $weekday) in order to give us more insights later in the EDA process.

We will answer the following questions in the exploratory data analysis part.

1. Which shop was the most popular shop?
2. Which shop had most items?
3. Which item sold the best in each shop?
4. Which shop had most categories?
5. Which category sold most?
6. Which category has the highest sales?

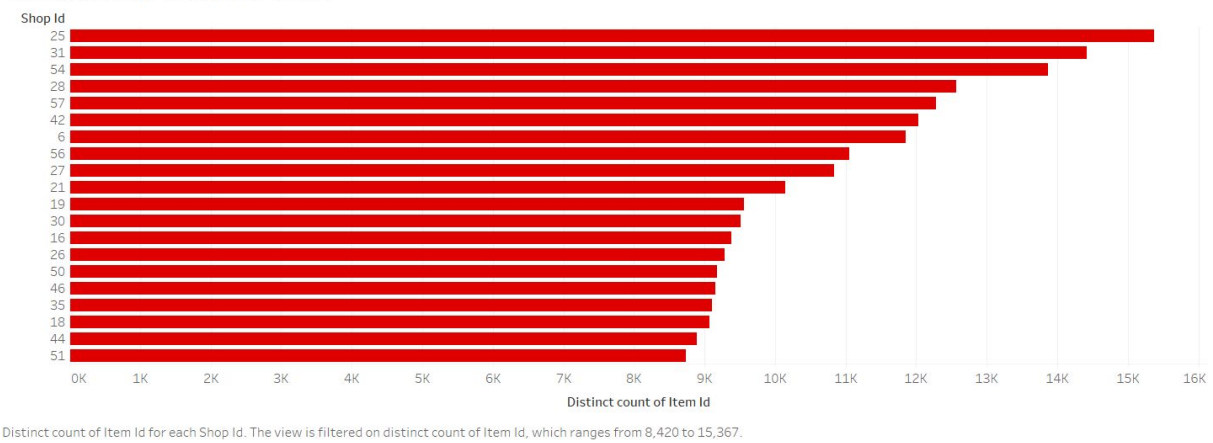
After that, we will take a look at the monthly total sales per year and the daily total sales per month.

Figure 2.1: Total Transaction Quantities of Each Shop



From Figure 2.1, we can see which shop is the most popular from 2013 to 2015. We calculate the popularity by summing up the number of items sold daily at each shop. According to the graph, shop id 31 was the most popular shop from 2013 to 2015 and sold 310,777 products. This shop is the Moscow Shopping Center "Semenovsky".

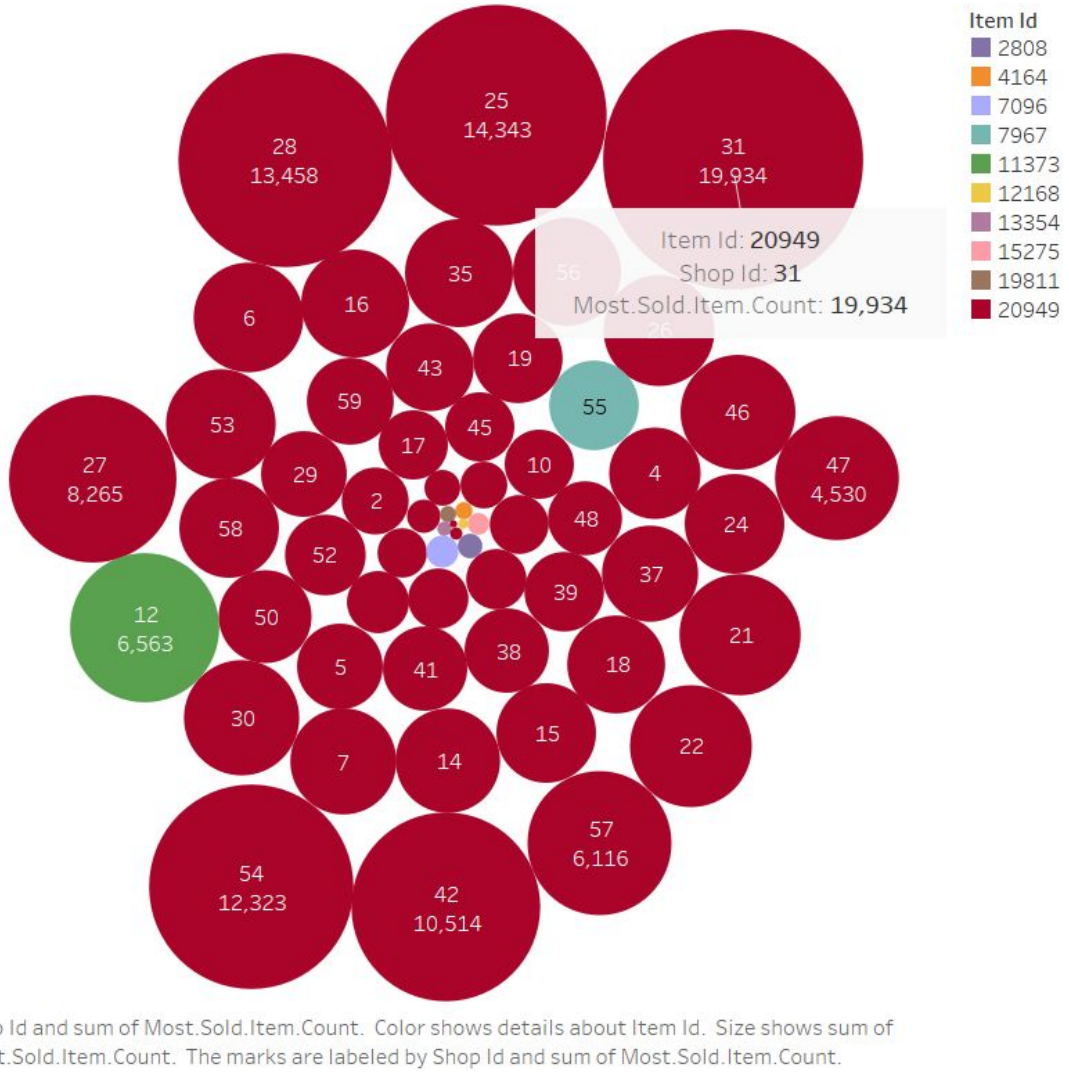
Figure 2.2: Count of Distinct Items of Each Shop



We count the distinct item id for each shop id and sort the counts from high to low. Figure 2.2 is filtered on the distinct count of item id, which ranges from 8,420 to 15,367. From Figure 2.2, we can see that the shop id 25, Moscow TRC "Atrium", had the most items, 15,367 products. Putting Figure 2.1 and Figure 2.2 together, it is obvious that the variety of

items contributes a lot to the store's popularity. The top five stores that have most items are also the top five most popular stores.

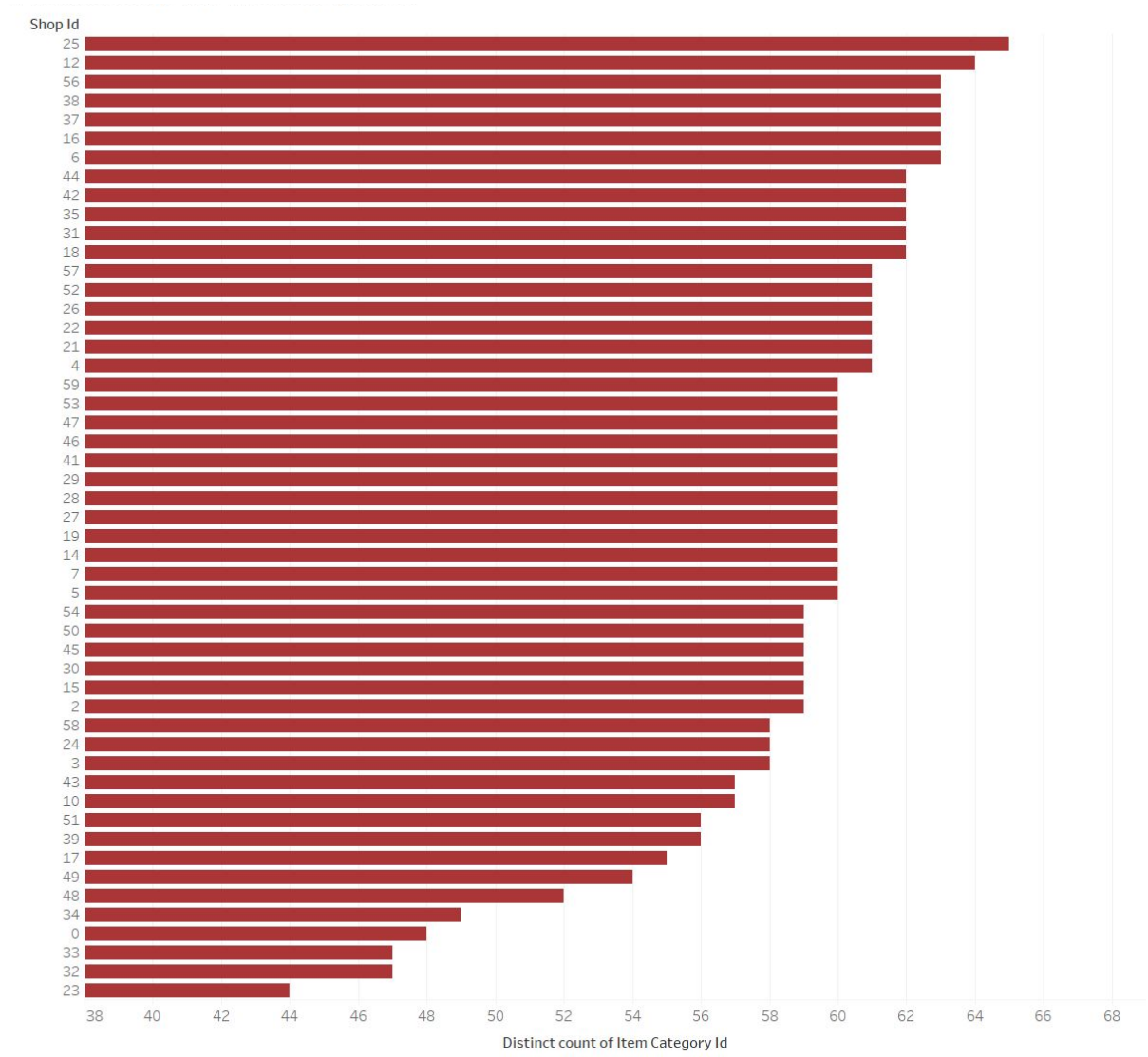
Figure 2.3: Best-Sold Item of Each Shop



Shop Id and sum of Most.Sold.Item.Count. Color shows details about Item Id. Size shows sum of Most.Sold.Item.Count. The marks are labeled by Shop Id and sum of Most.Sold.Item.Count.

From Figure 2.3, we can see which item is most popular/ sold from 2013-2015. The graph shows that item id 20949, the corporate package shirt 1C Interest white (34 * 42) 45 microns, was the best-sold item in 51 out of 60 shops. In shop id 31, the item id 20949 was sold 19,934 pieces and it was sold 185,567 pieces in the 51 shops.

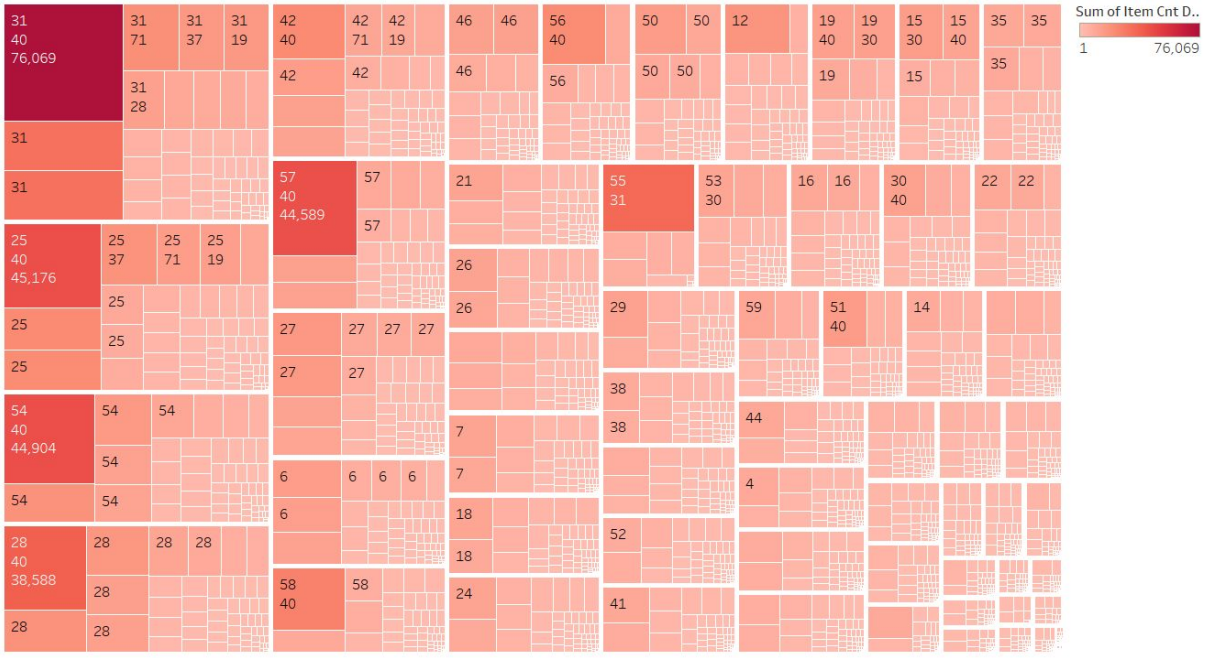
Figure 2.4: Count of Distinct Categories of Each Shop



Distinct count of Item Category Id for each Shop Id. The view is filtered on distinct count of Item Category Id, which ranges from 44 to 65.

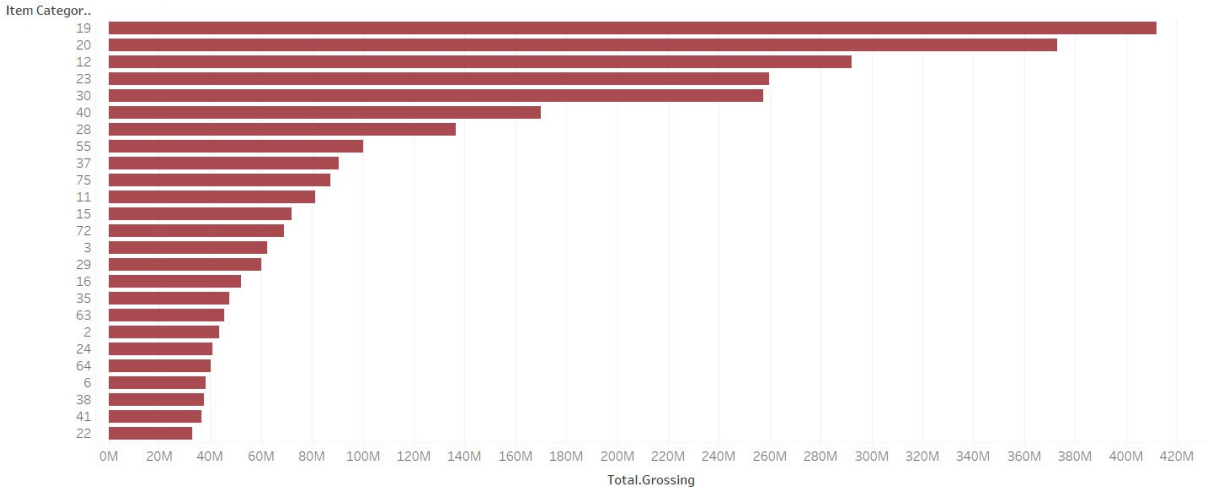
After that, we count the distinct item categories for each shop, to see which shop has the most categories. From Figure 2.4, we can see that there were 51 shops with more than 44 categories. The shop that had the most categories was shop id 25, with 65 categories.

Figure 2.5: Count of items sold of Each Category in Each Shop



We sum up the Item Cnt Day of each category in each shop, to find out the best-sold category in each shop. From Figure 2.5, we can see that the top best-sold category is the category id 40 in shop id 31, with 76,069 items sold. And from other blocks in the figure, we can see that the most popular category in most shops in category id 40, Cinema-DVD.

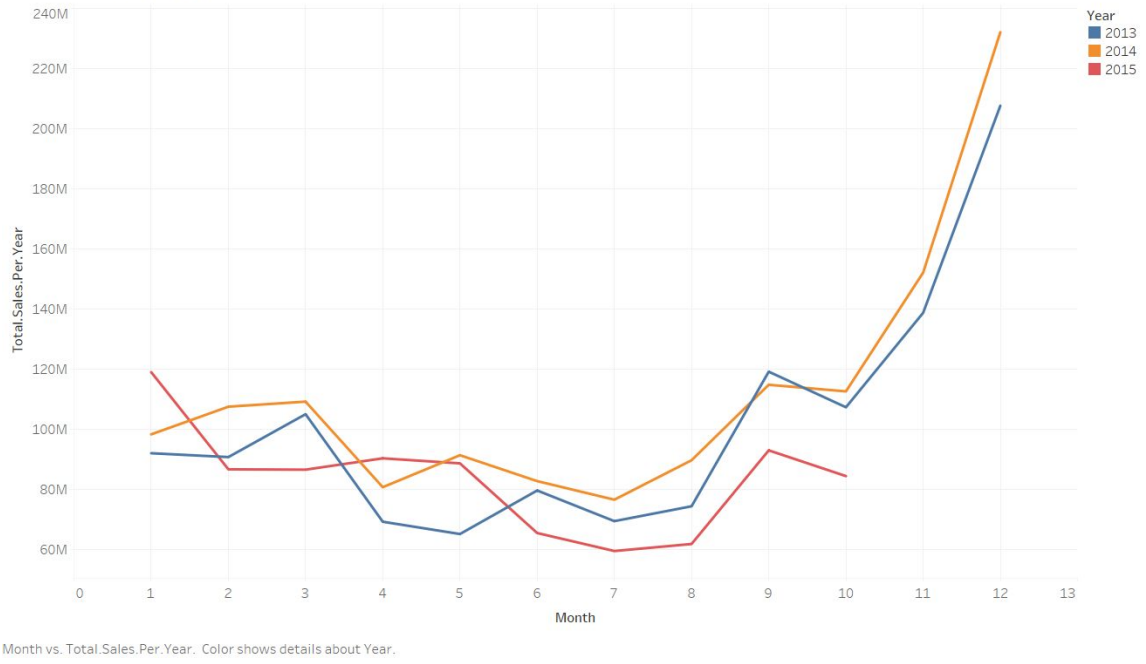
Figure 2.6: Sales of Each Category



We calculate the total sales of each category and sort the result from high to low as Figure 2.6. The category id 19, Games - PS3, had the highest sales, $412,050,281, with 254,887

items sold. However, category id 40, Cinema-DVD, which sold most items, only ranked the sixth, with the sales of $169,944,222.

Figure 2.7: Monthly Total Sales Per Year



Here we are looking at the monthly total sales trend per year. It is obvious that there is a seasonal trend from year to year. Starting from January, the sales gradually decrease until July. Then it starts to show gradual growth, with a steep rise from October to December. The pattern is useful for our further modeling.

Figure 2.8: Daily Sales Per Month

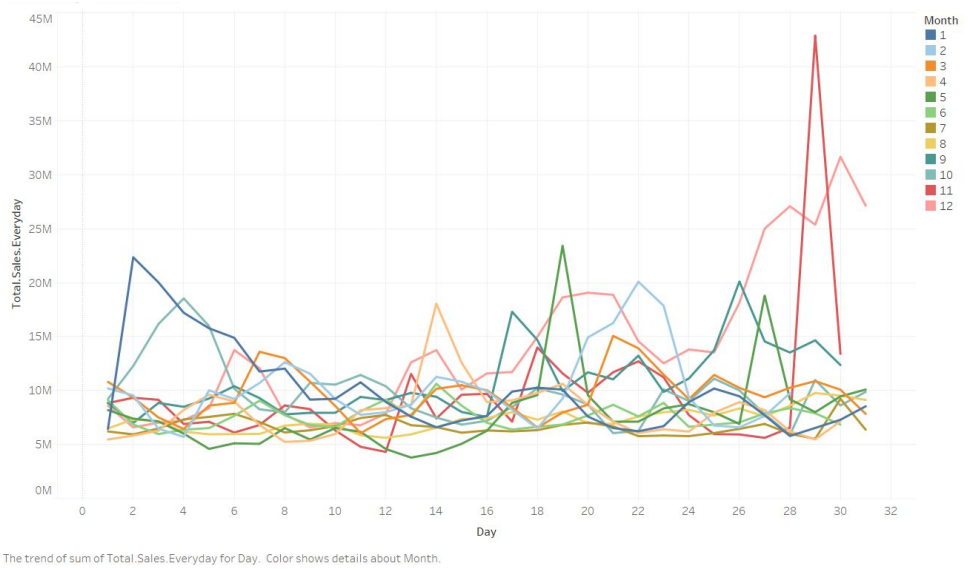


Figure 2.8 is about the daily sales of each month in three years. From Figure 2.7, we can see there is no significant trend in daily sales between months. However, there is a surge of sales in November and December.

Figure 2.9: Daily Sales in November and December

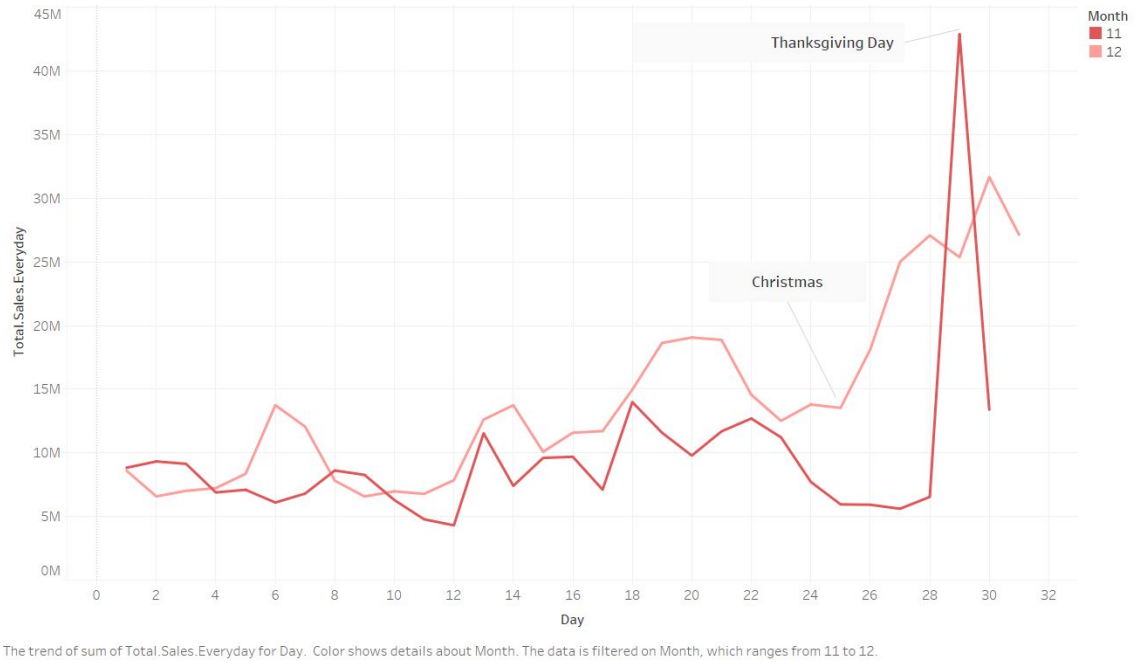


Figure 2.9 is about the trends of November and December from 2013 to 2015. There is a sudden increase in Daily sales at the end of November, during the Thanksgiving holiday. In December, the daily sales increase during the Christmas holiday. This is a pattern we can focus on.

3. ARIMA Model

ARIMA model is the combination of AutoRegressive model and Moving Average model. AutoRegressive refers to whether there is some relationship among the lags of differenced series. Moving Average refers to the lags of errors and I I is the number of difference used to make the time series stationary.

Before using the ARIMA model, we need to make sure the data is stationary and univariate. Firstly, stationary means that the properties of the series doesn't depend on the time when it is captured. A white noise series and series with cyclic behavior can also be considered as stationary series. To validate our dataset, we made Dickey–Fuller test. Figure 3.1 is the test result of the data. In the test, the null hypothesis test is the data is non-stationary. The p-value is smaller than 0.05, so we can reject the null hypothesis and the data is stationary. Secondly, the data is

univariate because it just has two variables, which are date and sales. It means that we assume the sales are just related to the sales of past value.

Figure 3.1: Dickey-Fuller Test

```
        Augmented Dickey-Fuller Test

data:  dailysales
Dickey-Fuller = -6.6675, Lag order = 10, p-value = 0.01
alternative hypothesis: stationary
```

After testing the assumptions of the ARIMA model, we did some exploratory analysis including autocorrelation analysis and trend estimation and decomposition.

Autocorrelation analysis: We used this to examine serial dependence. It can test whether the value in the past has a correlation with the current value so that we can provide the p,d,q estimate for ARIMA models. From Figure 3.2, we can see that a week (seven days) is a seasonal pattern. The value is most related to yesterday and the value before seven days. From Figure 3.3, we can see that there is a seasonal pattern.

Figure 3.2: ACF Plot

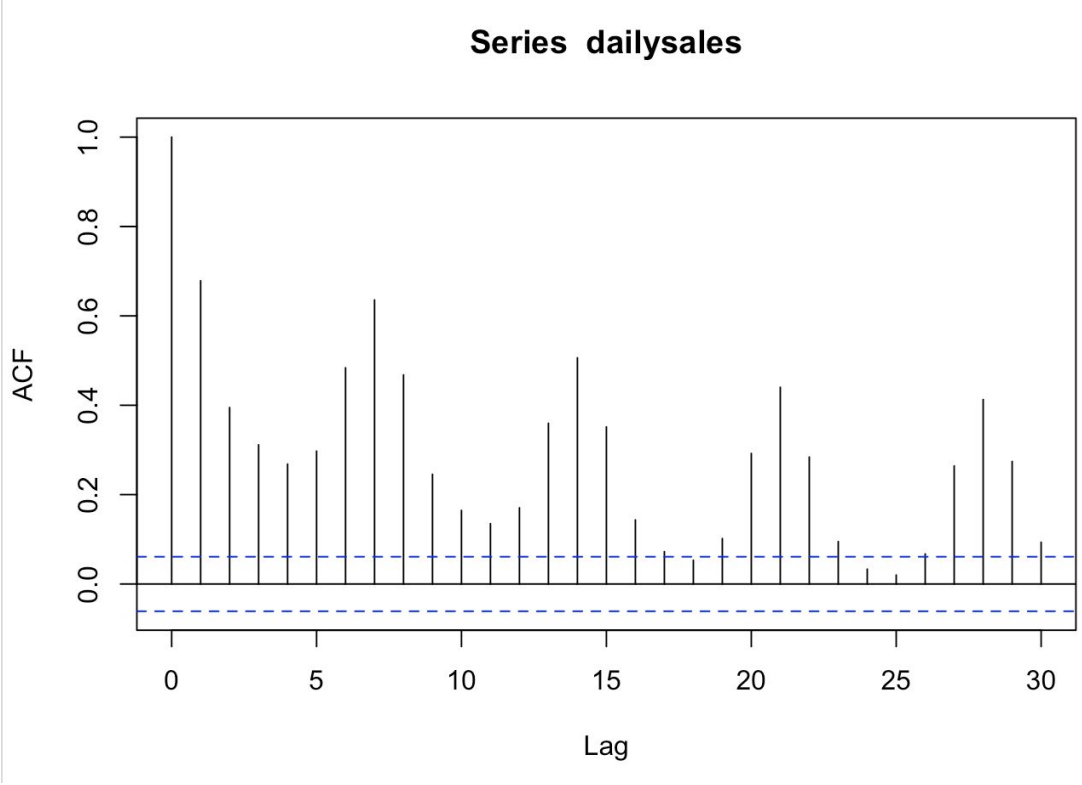
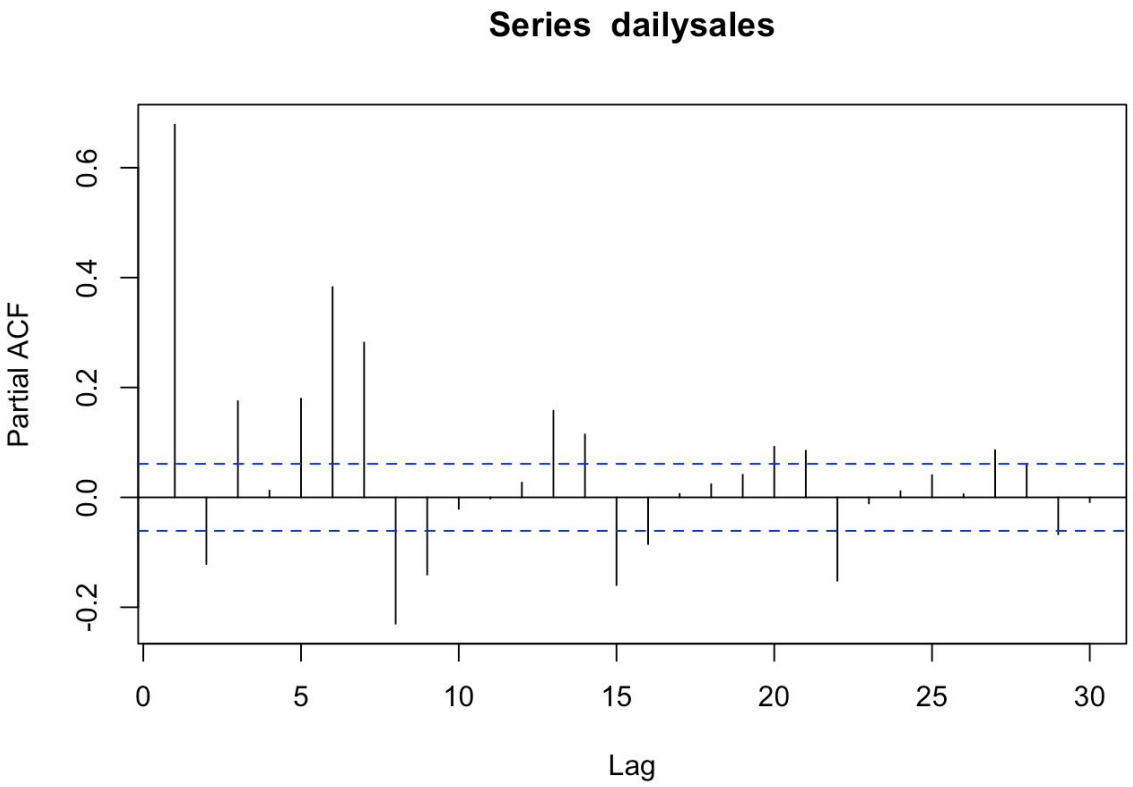


Figure 3.3: PACF Plot



Whether in ACF and PACF plot, there is no cut-off. Therefore, we need to use the ARIMA model rather than the AR or MA model.

Trend estimation and decomposition: This analysis helps us to analyze the time series components in the data. We used two methods, which are additive and multiplicative decomposition. In additive decomposition, x= Trend + Seasonal + Random and in multiplicative decomposition, x = Trend * Seasonal * Random. In two methods, we both see that the trend is decreasing and there is a seasonal pattern. The random component is stationary.

Figure 3.4: Decomposition of Additive time series

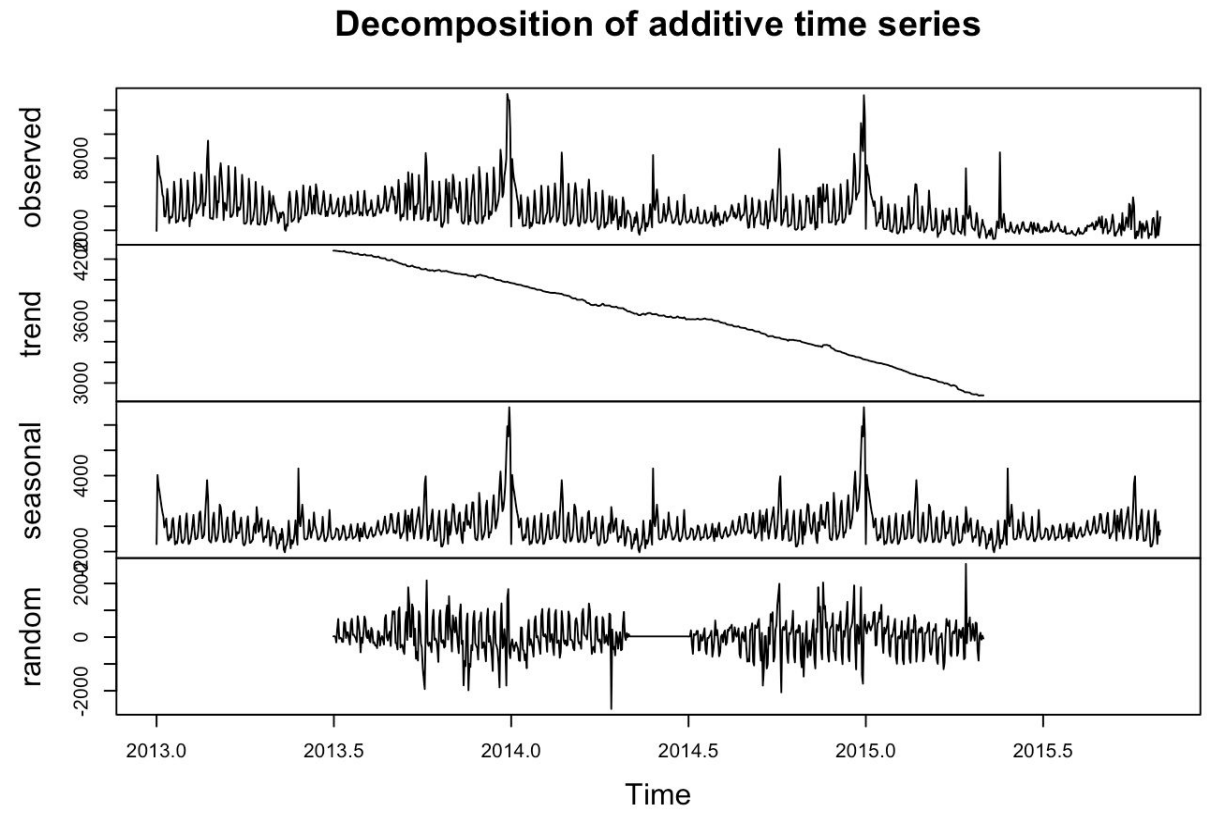
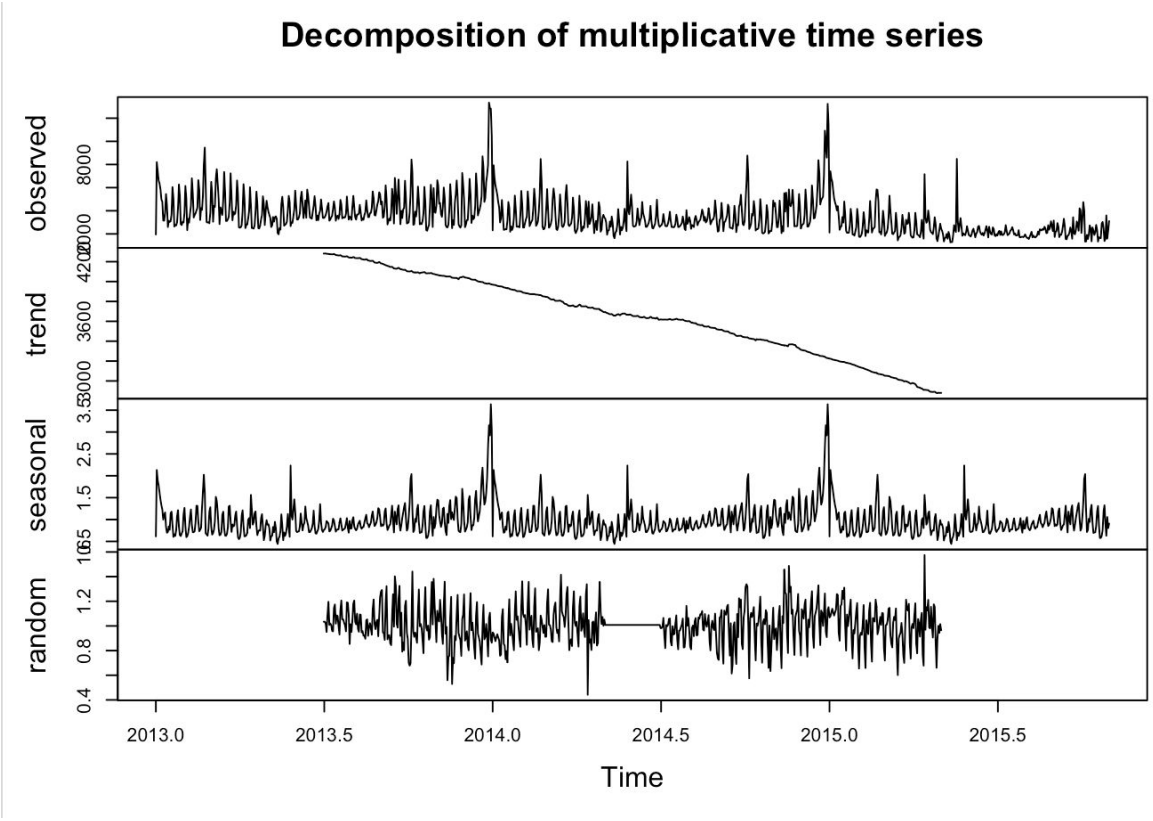


Figure 3.5: Decomposition of Multiplicative time series

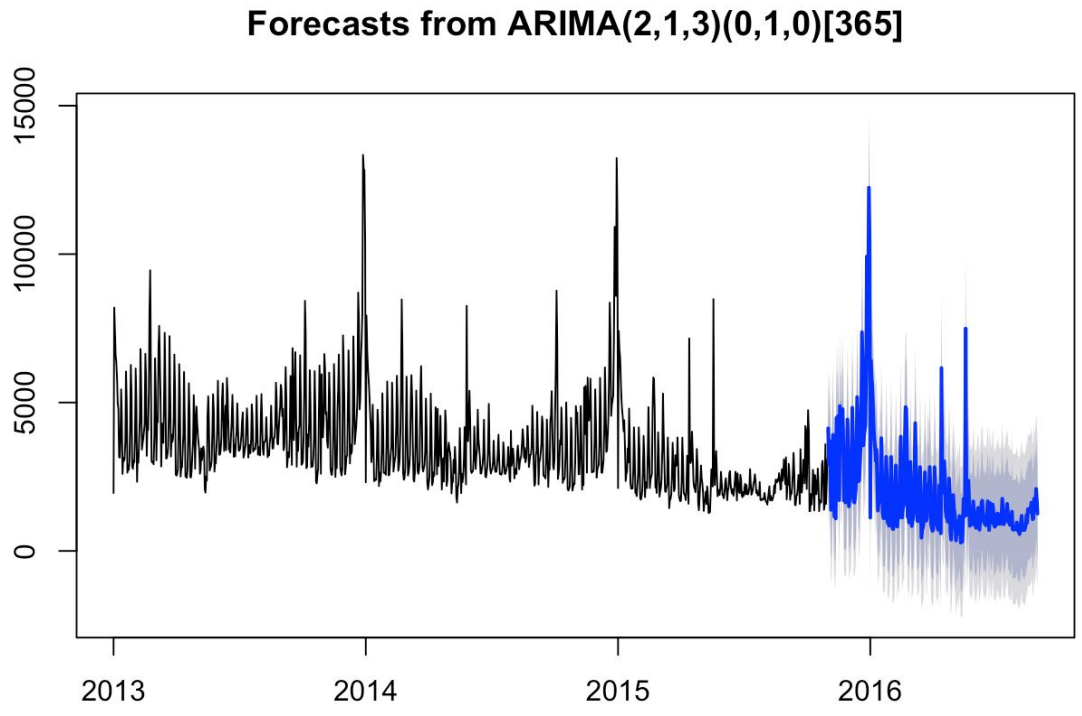


Using these analysis, we build an ARIMA model with the parameters (2,1,3)(0,1,0). Figure 3.6 shows the model result. The RMSE is 898.664 and MAPE is 17.3475% which means that the mean absolute error of the model is 17%. From the coefficients below, we can see that the model uses the first difference of Y. The estimated first difference is most related to the forecast error of Y before two days with a coefficient of 0.0555 and the yesterday forecast error of Y with a coefficient of 0.0456. It is also related to the Y two days ago.

Figure 3.6: ARIMA model result

```
Forecast method: ARIMA(2,1,3)(0,1,0)[365]

Model Information:
Series: dailysales
ARIMA(2,1,3)(0,1,0)[365]

Coefficients:
          ar1      ar2      ma1     ma2      ma3
      -0.4896  -0.8017  -0.2021  0.0072  -0.7731
s.e.   0.0354   0.0416   0.0456  0.0555   0.0335

sigma^2 estimated as 1259510:  log likelihood=-5642.45
AIC=11296.9   AICc=11297.03   BIC=11323.93

Error measures:
                    ME    RMSE      MAE       MPE    MAPE      MASE       ACF1
Training set -34.2975 898.664 517.1208 -3.481162 17.3475 0.4661364 0.09517542
```

Figure 3.7: ARIMA Forecast result



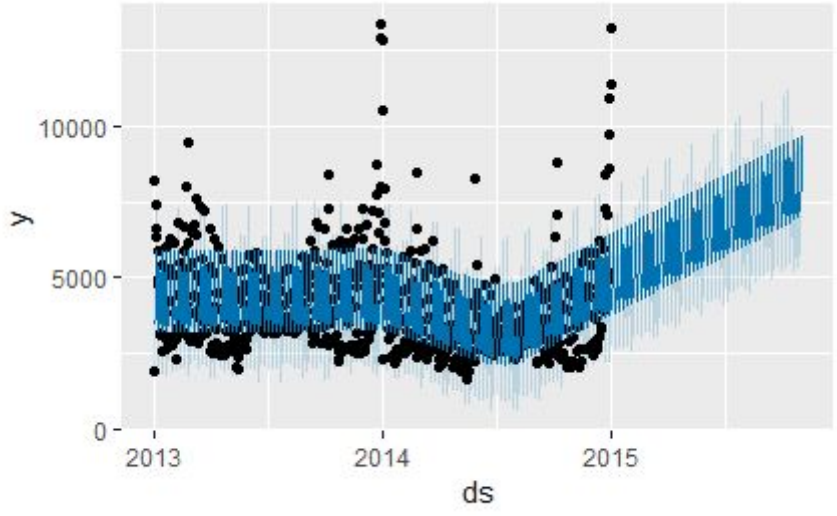
4. Prophet Model

Prophet is an open source software from Facebook and available on R. It is based on an additive model where nonlinear trends fit with yearly, weekly, and daily seasonality and works well with outliers. Our sales data has many outliers, especially at the end of the year so that prophet is good to use.

To predict the total sales of the next day, two models are set. One has the seasonality feature and the other one does not. The following are the results.

The RMSE of the non-seasonality model is 2586.42.

Figure 4.1 non-seasonality prophet model result



The RMSE of the seasonality model is 700.3344 and Mape is 0.2299826.

Figure 4.2 Seasonality prophet model result

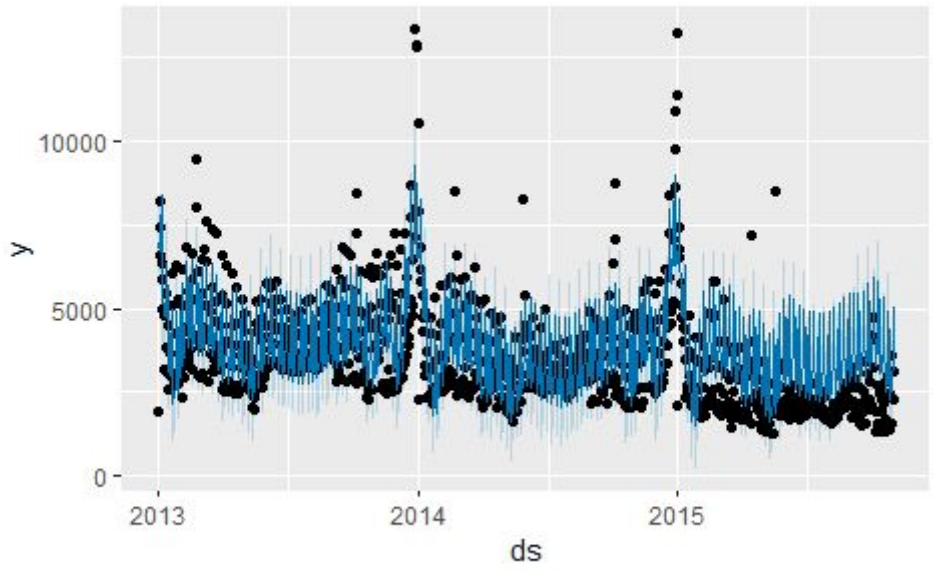


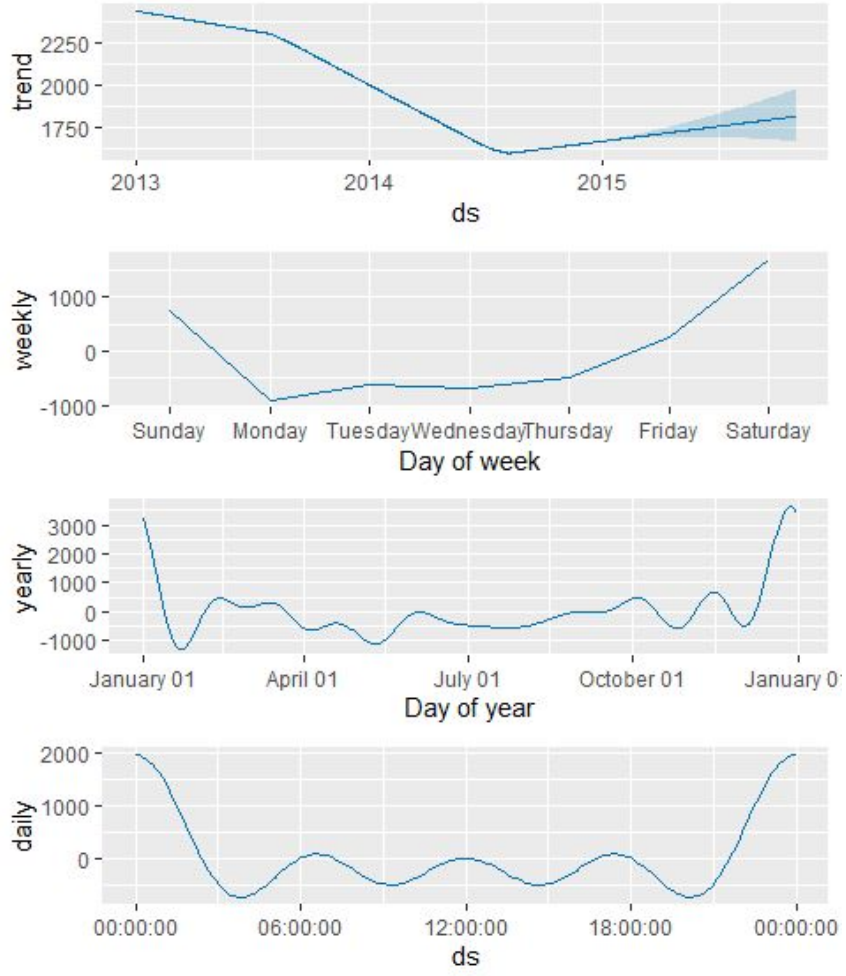
Figure 4.3 the evaluation of the seasonality prophet model

	horizon	mse	rmse	mae	mape	coverage
1	1 days	490468.3	700.3344	600.8230	0.2299826	0.9
2	2 days	405364.6	636.6825	481.7833	0.2070134	0.9
3	3 days	322756.3	568.1164	409.8286	0.2038862	1.0
4	4 days	486504.4	697.4987	561.3252	0.2426522	0.9
5	5 days	439360.4	662.8427	521.8571	0.2139592	1.0
6	6 days	561204.5	749.1358	550.1150	0.2224932	0.9

It turns out that the seasonality model works better.

The sales trend went down from 2013 to 2014 and grew in the middle of 2014. On Saturday the games sold the best and on Monday sold the worst. It also shows that big sales occurred at the beginning of each year. At last, the daily trend points out that most sales occurred at 12am.

Figure 4.4 the composition of the time series data



5. LSTM

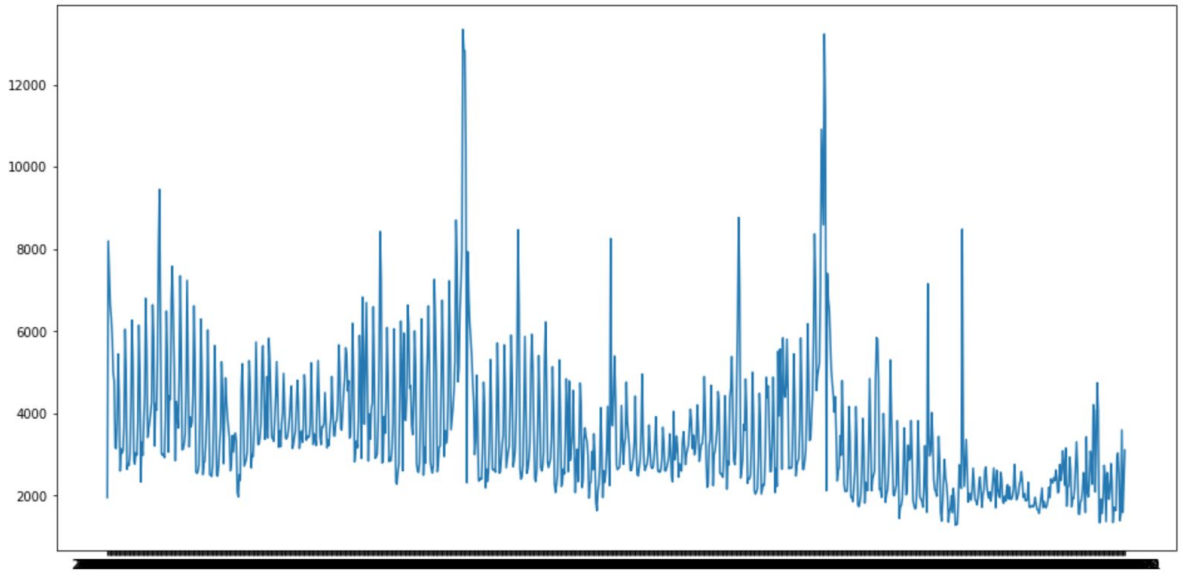
LSTM model is aimed at using recurrent neural network architecture to process the data. In this step, we intended to build the LSTM model and train the data. First of all, we summed up the number of daily sold items and the picture below shows that the number of sold items each day.

Figure 5.1 Daily Sold Items

date	item_cnt_day
2013-01-01	1951
2013-01-02	8198
2013-01-03	7422
2013-01-04	6617
2013-01-05	6346
...	...
2015-10-27	1551
2015-10-28	3593
2015-10-29	1589
2015-10-30	2274
2015-10-31	3104

Next, we planned to take a look at the time series plot. From the picture below, we can know that the daily sales changed significantly, ranging from 1000 to 13000. The date of our data resources started from January 2013, and ended at October 2015. The plot indicates that the annual sales were going to decrease each year, and in the mid of 2015, the daily sales were almost at the bottom of these three years.

Figure 5.2 Initial Plot of Time Series Data



In the next step, we built the LSTM model and trained the data. We set the first two years' data as the training data and data in 2015 as the validation data. Then we created and fitted the LSTM network: we set 60 epochs, 4 batch sizes and mean square error as the loss function. From the plot below, we can know that the MSE is ranging from 0.0033 to 0.0037. At the 60th epoch, we got the best mean square error, which was 0.0033. Then we ended training models.

Figure 5.3 Plot of Epochs

```
Epoch 52/60
 - 23s - loss: 0.0037
Epoch 53/60
 - 23s - loss: 0.0035
Epoch 54/60
 - 22s - loss: 0.0035
Epoch 55/60
 - 23s - loss: 0.0036
Epoch 56/60
 - 23s - loss: 0.0034
Epoch 57/60
 - 23s - loss: 0.0035
Epoch 58/60
 - 23s - loss: 0.0035
Epoch 59/60
 - 23s - loss: 0.0034
Epoch 60/60
 - 23s - loss: 0.0033
```

Next, we started to put the model on the validation data to test if it is overfitted and its performance is well. From the plot below, we can know that the predicted data can almostly represent the actual data. However, it still cannot represent the actual data precisely and maybe we can run it again in the future work. And the RMSE is about 322.09. Also, the MAPE of this model is about 0.14. The reason why the RMSE is relatively large is that we have very large daily sales figures. From the second picture below, we showed the predicted data at last.

Figure 5.4 LSTM Model Performance

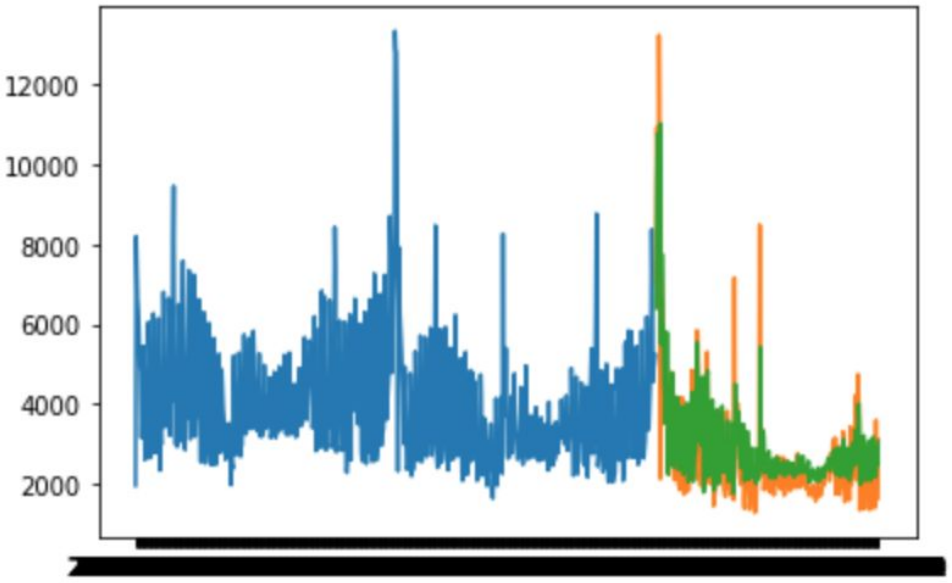


Figure 5.5 Output

```
date
2015-01-01    8604.247070
2015-01-02    3439.739014
2015-01-03    8528.908203
2015-01-04    8552.190430
2015-01-05    7338.582520
                  ...
2015-10-27    1983.369751
2015-10-28    2216.778320
2015-10-29    3424.277588
2015-10-30    2425.217285
2015-10-31    2605.038574
Name: Predictions, Length: 304, dtype: float32
```

6. Conclusion

The ARIMA model has some limitations. Firstly, the data must be stationary in the ARIMA model. In this example, although the Dickey-Fuller Test shows the data is stationary, the auto arima uses the differenced data to predict. Secondly, the model can just recognize the linear relationship. In this example, the relationship between sales and the sales in the past value is non-linear. Thirdly, there are 6 parameters in ARIMA and it is time-consuming to find the best ARIMA model. We need to try different combinations and compare the results.

There are several reasons.about why the prophet model doesn't work well. First, the daily component shows that most records are at 12 am, which is impossible. Most people at that time are sleeping and the game shop closed at that time. The time occurred just because of the system setting, such as the deadline for summing all sales in a day. Next, the trend does not predict correctly. In the prediction, the trend grew in 2015 while actually it went down. The grow in the end of 2014 misleads the model to predict the growth in 2015. Last and most importantly, the data itself has a large scope so that RMSE is quite large.

The performance of the long short-term memory model is the best among three models. It is probably because the LSTM model leverage recurrent neural network architecture to train the model and it has lags of unknown events in time series data. Thus, with forward and back-propagation processes, the training model can predict the future data more and more precisely. Thus, it is why it has the best performance.