Divvy Bike Forecasting Capstone

Presented by: Tina Carr

Purpose

Divvy is Chicago’s bike sharing system which is owned by the Chicago Department of Transportation with the aim to “promote economic recovery, reduce traffic congestion and improve air quality”.

With over 6,000 bikes and 580 docking stations in the Chicagoland / Evanston area that are available to the public 24/7 the higher the accuracy rate Divvy’s forecasting models can achieve translates into more happy customers looking for a sustainable and fun way to get around the city.

Therefore, this capstone project will use forecasting models with the aim of creating a daily forecast for the the Lake Shore Dr & Monroe St Divvy station.

Dataset

The source of the data can be found here and was provided by the City of Chicago

This data set has a total of 13,821,994 observations and 22 columns including:

Trip ID	Community areas	To Longitude	From Latitude	From Longitude	From Location
From Station ID	Start Time	Wards	To Location	Boundaries	Zip Codes
Gender	From Station Name	Stop Time	Bike ID	Trip Duration	To Latitude
Birth Year	To Station ID	To Station Name	User Type		

For forecasting purposes there are 2 key columns for every station: Bikes brought TO and FROM the station.

I have chosen the Lake Shore Dr & Monroe St Divvy station to forecast since it is the station the most bikes are taken from and the 2nd busiest station customers return bikes to.

Customer Insights

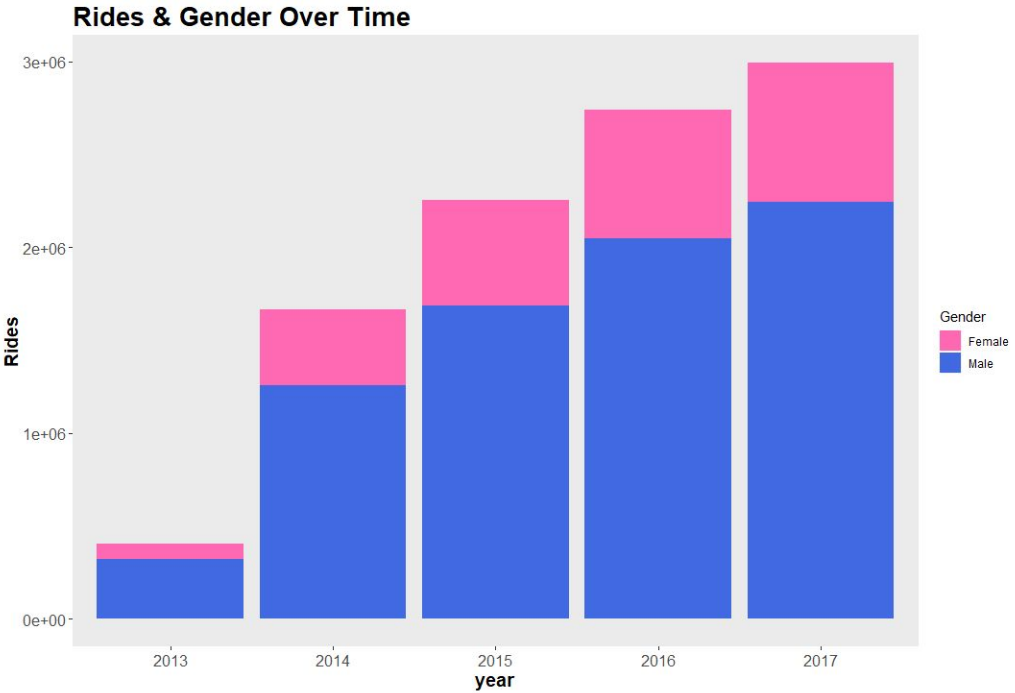
DIVVY

Overview

In addition to the time series modelling in this capstone I ran an analysis on the complete Divvy data set to gain insights on the Divvy brand and customer. In order to do this I concentrated on the following categories:

- Gender
- Age
- Trip Duration

Gender



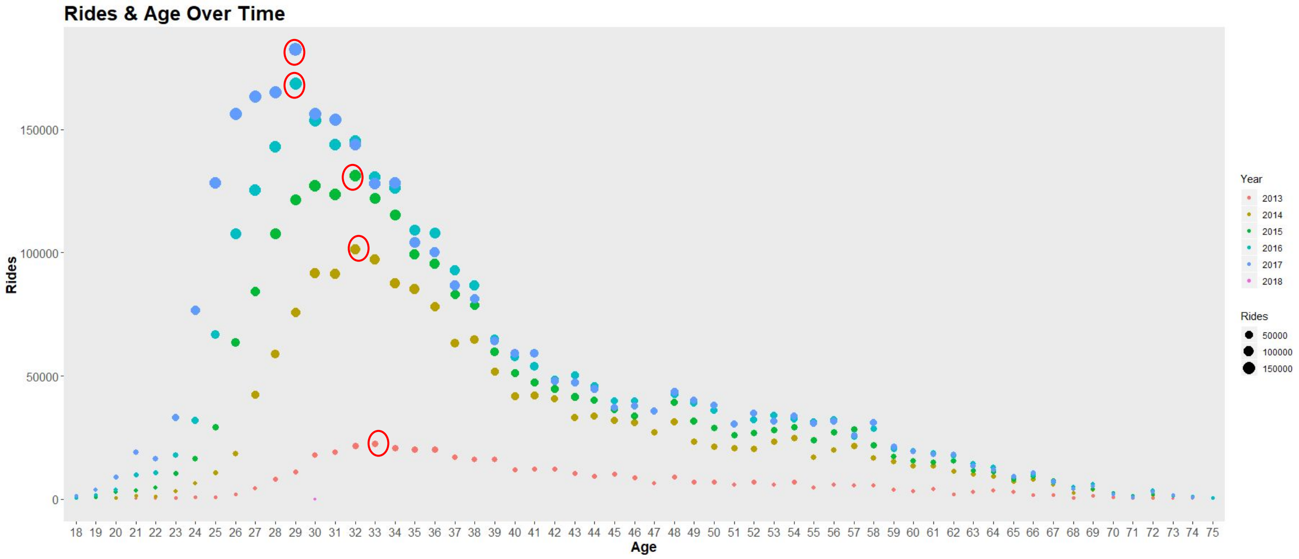
Male / Female Trip Ratio: 7,552,990 / 2,495,777*

Insights

- Divvy was able to increase its female riders trips from only 21% in 2013 to 25% in 2017
- However, the US Census reports Chicago's female population at about 51%
- Customer research could be conducted on marketing Divvy to women since this segment is massively underrepresented in their customer base compared to the overall population of Chicago

*not all customers revealed their gender

Age



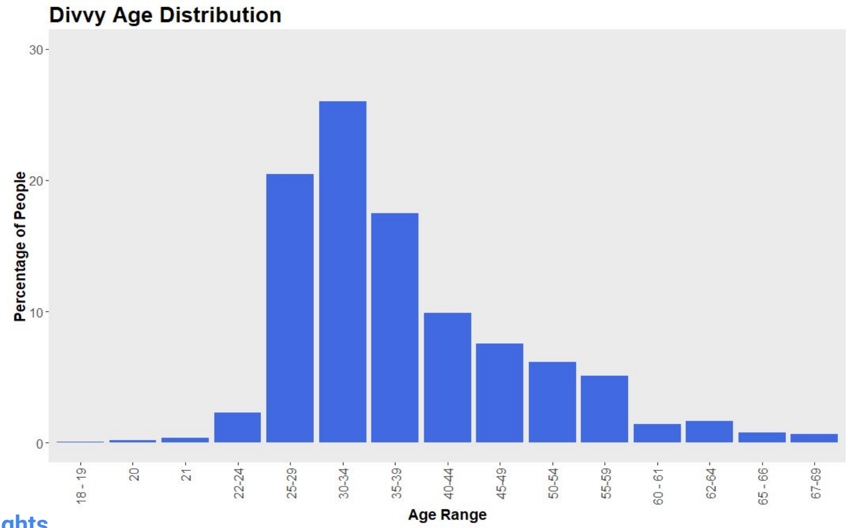
*Data was cleaned to represent the 18-75 age range only

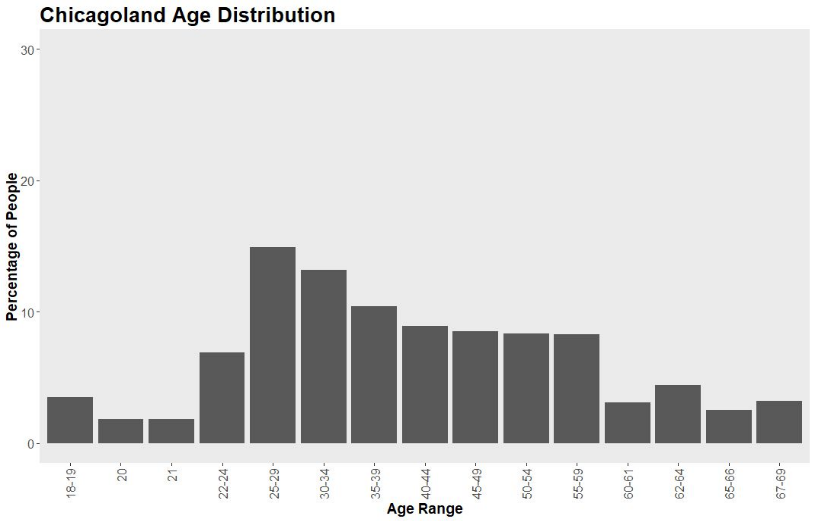
Insights

- Over time the “peak” age for riders has shifted from 33 in 2013 to around 28 in 2018 and

- Divvy seems to capture the late 20s / early 30’s demographic quite well however there seems to be an untapped segment in the early 20s and late 30s

Divvy Age vs. Chicago Age Distribution

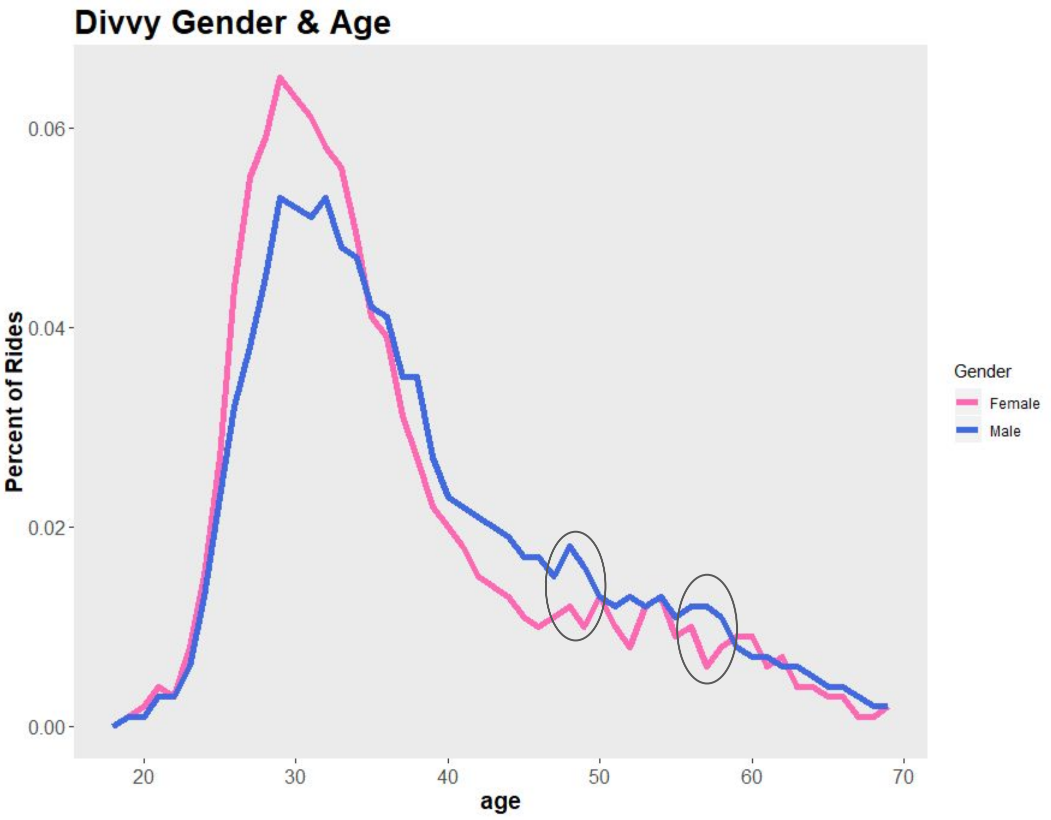




Insights

- The Divvy age distribution is skewed to the left compared to the overall population in Chicago which demonstrates there is untapped potential in the mid-age market segment
- As seen in the previous slide there is also untapped potential in the 18-24 market for Divvy compared to the overall Chicago population
- Divvy age standard deviation = 10.4, Chicago age standard deviation = 14.2. The wider spread of the Chicago standard deviation could in part be explained by the fact older people, which are considered part of the Chicago population, are physically unable to bike ride thus preventing Divvy bike from capturing this segment.
- In any case taking standard deviation into consideration if Divvy targets new age segments which are underrepresented in their customer base compared to the Chicago population the Divvy standard deviation will naturally increase towards the Chicago standard deviation

Gender & Age

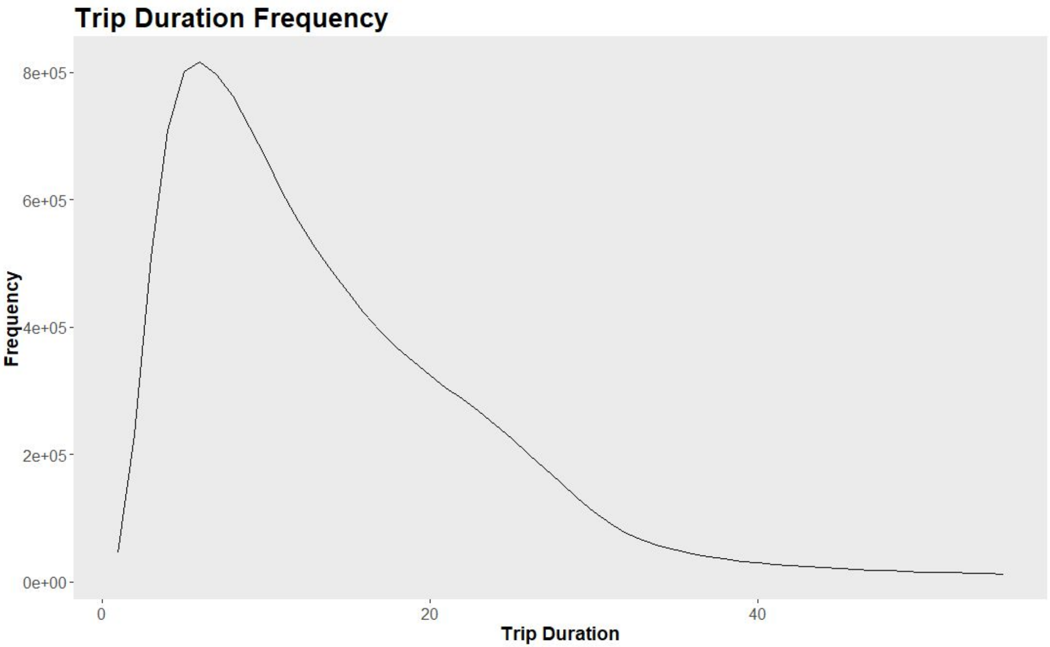


Insights

- The male and female data experiences similar peaks and dips with some exceptions
- A peak in male an female riders happens around the late 20's - however males experience another peak in the early 30's
- There is a peak in male riders in their late 40's while female riders experience a dip in that age - a similar instance happens in the late 50's

Overall Divvy male riders come from a more diverse age segment pool compared to females which peak in their late 20s/early 30s and decrease significantly after that. If Divvy is able to attract more female riders in the future it will be interesting to see if the female segment develops a wider spread as well and is not so dependant on the late 20s/early 30s female demographic.

Trip Duration



Insights

- Steep drop in trips over 30 minutes

This trip length visualization makes sense when taking Divvy's price structure into consideration, keeping in mind that this data set's last observation was in 2017. Divvy was receiving complaints about their 30 minute ride pricing-structure and in 2018 they decided to change their pricing structure in order to encourage rides longer than 30 minutes.

Since the price structure which divvy had set up for the daily and annual offering encouraged rides of 30 minutes or less prior to 2018, naturally that was also reflected in the trip duration visualization since customers wanted to avoid any extra cost for exceeding the standard trip duration.

TO and FROM Data

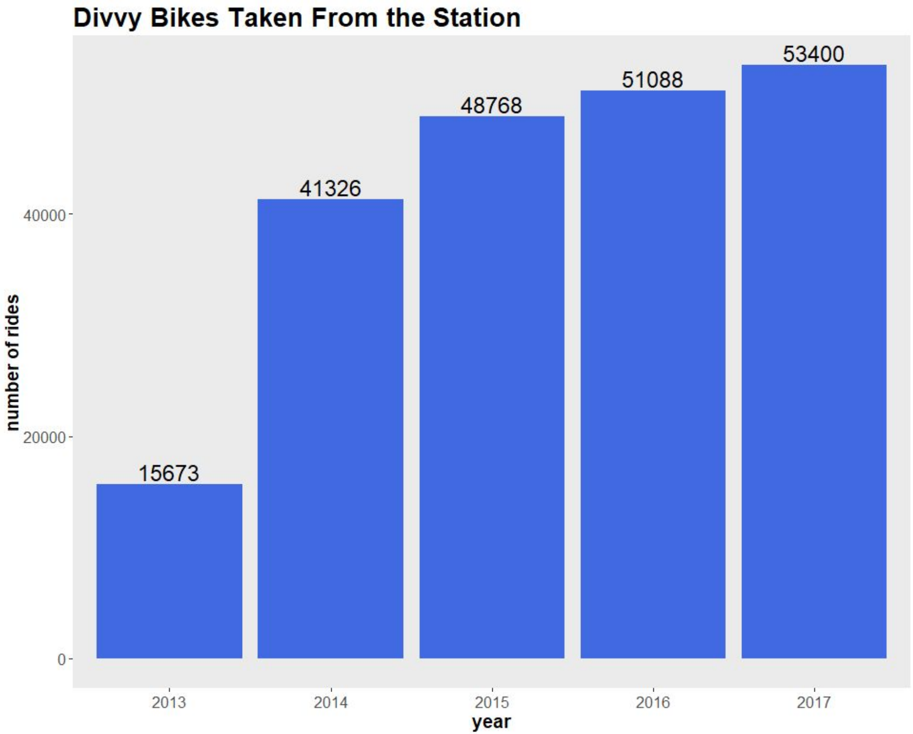
DIVVY

FROM Data

The graph to the right is an overview of the number of bikes taken from the Lake Shore Dr. & Monroe Divvy Station

Observation Dates: Jun 27 2013 - December 28 2017
Total Observations: 210,255

- Divvy Bike began operations in June 2013
- The significant increase from 2013 - 2014 can in part be explained by the fact the 2013 only covered 6 months of the year while 2014 - 2017 data was recorded for a full 12 month calendar year
- There is a positive trend of rides as each year passes which is natural for a brand just starting out as more and more people become aware of the brand

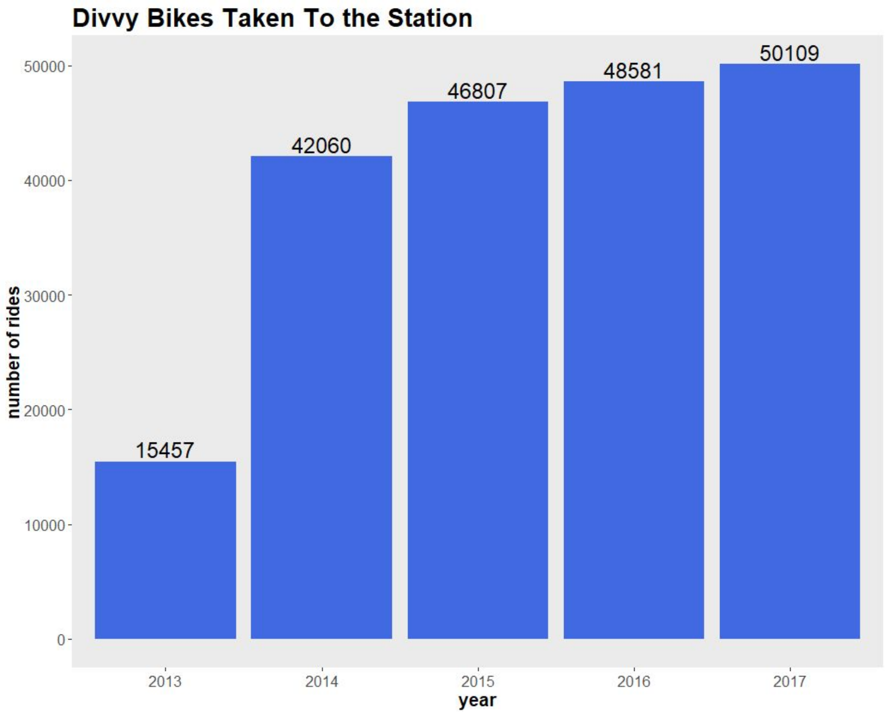


TO Data

The graph to the right is an overview of the number of bikes taken to the Lake Shore Dr. & Monroe Divvy Station

Observation Dates: Jun 27 2013 - December 31 2017
Total Observations: 203,014

- 7,241 fewer bikes were taken to the station compared taken from the station from 2013-2017
- The significant increase from 2013 - 2014 can in part be explained by the fact the 2013 only covered 6 months of the year while 2014 - 2017 data was recorded for a full 12 month calendar year
- There is a positive trend of rides as each year passes which is natural for a brand just starting out as more and more people become aware



Time Series (ts) Modelling with Arima

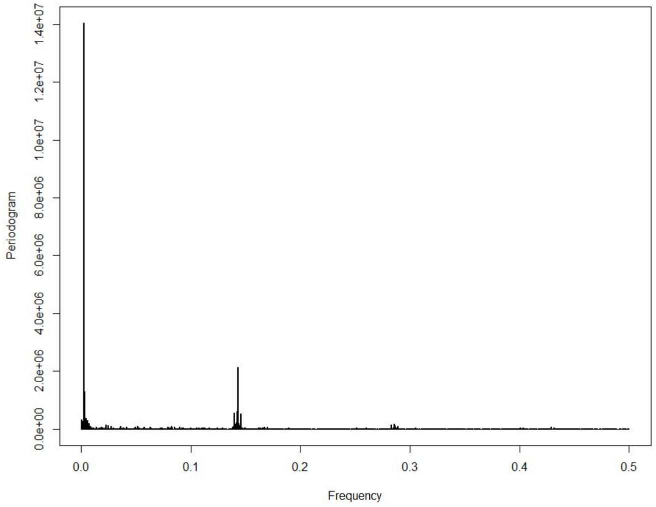
Approach

1. Data Wrangling
   a. In order to properly prep this data for the time series I first used tidyr to organize my data by “date” and “daily_observations”. While examining this data I noticed there were days missing in areas as well as a leap year. I removed the leap year from the data in order to prevent any issues with Arima later.
2. Created test & train data sets
   a. Since my data covers approx. 3.5 years I decided to leave my train data set larger and forecasted 4 months out.
3. Ran a periodogram to find the optimal seasonality
4. Created the ts function
5. “Baseline model” with Arima (1,1,1)
6. Auto.Arima model with seasonality
7. Models based on acf & pacf estimations
8. Additional model exploration

FROM ts Modelling

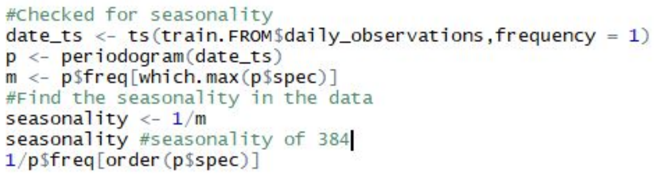
DIVVY

Seasonality



Creating the ts model & frequency

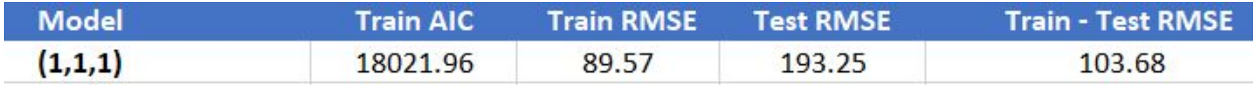
Before running the ts model I used the following code to find the seasonality in R:



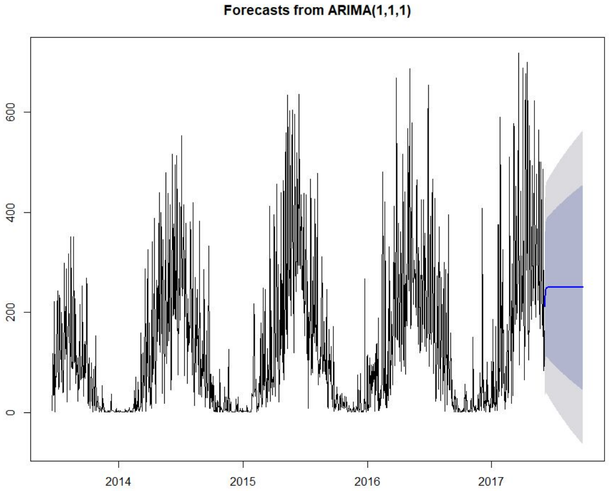
```
#Checked for seasonality
date_ts <- ts(train.FROM$daily_observations,frequency = 1)
p <- periodogram(date_ts)
m <- p$freq[which.max(p$spec)]
#Find the seasonality in the data
seasonality <- 1/m
seasonality #seasonality of 384
1/p$freq[order(p$spec)]
```

The result was an optimum seasonality of 384 for which I also accounted for while creating the ts functions

Baseline Model



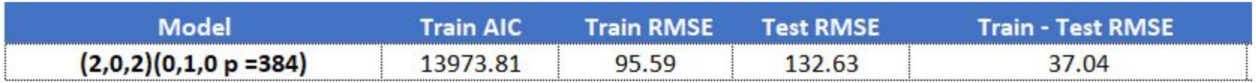
Model	Train AIC	Train RMSE	Test RMSE	Train - Test RMSE
(1,1,1)	18021.96	89.57	193.25	103.68



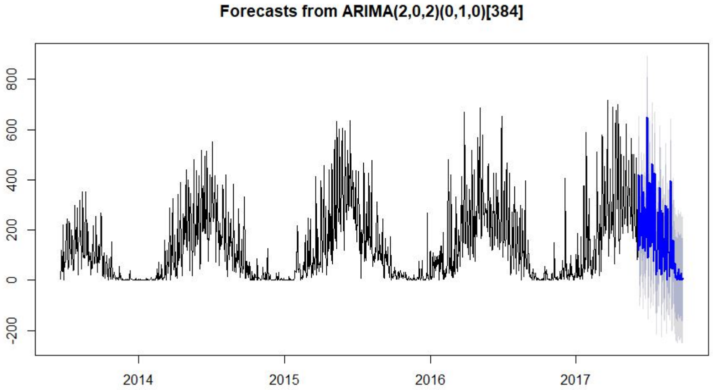
Forecasts from ARIMA(1,1,1)

Ideally the baseline model should be improved by adding seasonality, lowering the test RMSE and improving the train AIC

Auto.Arima Model with Seasonality



Model	Train AIC	Train RMSE	Test RMSE	Train - Test RMSE
(2,0,2)(0,1,0 p =384)	13973.81	95.59	132.63	37.04

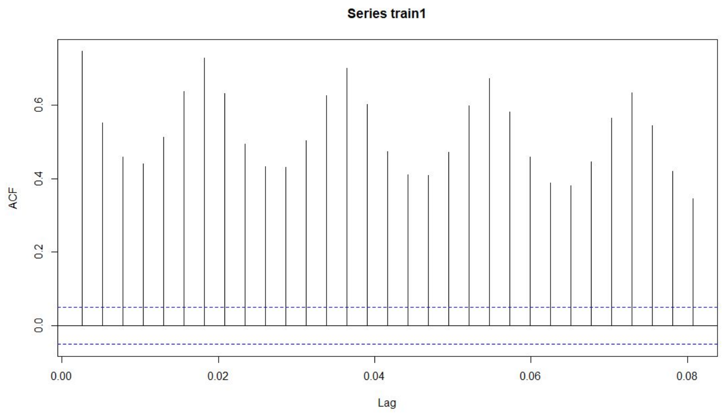


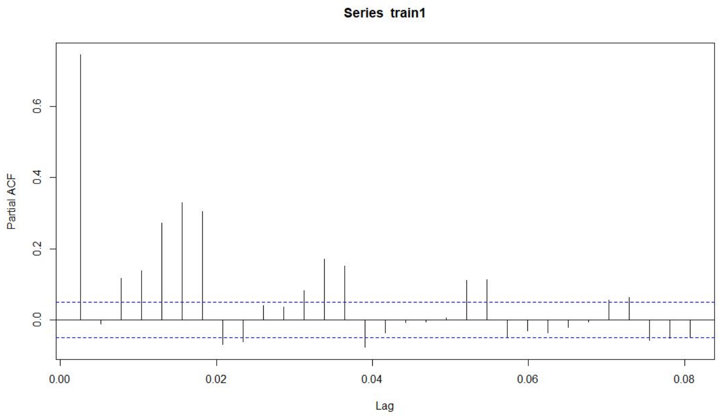
Forecasts from ARIMA(2,0,2)(0,1,0)[384]

This model's performance is a considerable improvement over the baseline model. The train AIC and test RMSE decreased considerably. Additionally the forecast, highlighted in blue, was able to pick up on seasonal trends.

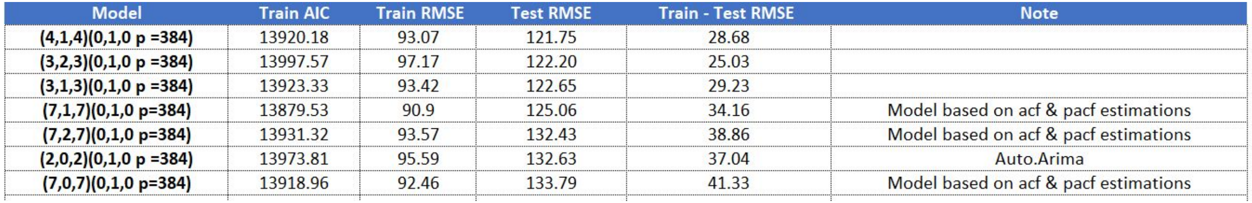
ACF & PACF Models

Models were also ran which which were based off of the observations for the ACF and PACF graphs below with a p and q value of 7. The results will be included in the final comparison table.





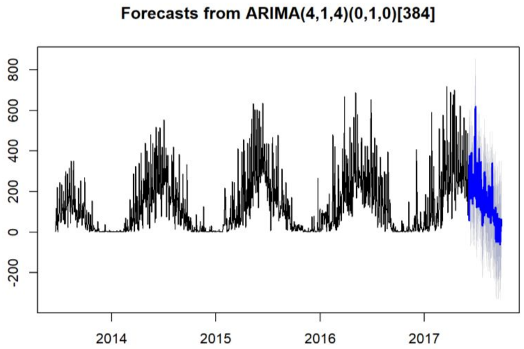
FROM Final Comparison Table



Model	Train AIC	Train RMSE	Test RMSE	Train - Test RMSE	Note
(4,1,4)(0,1,0 p =384)	13920.18	93.07	121.75	28.68	
(3,2,3)(0,1,0 p =384)	13997.57	97.17	122.20	25.03	
(3,1,3)(0,1,0 p =384)	13923.33	93.42	122.65	29.23	
(7,1,7)(0,1,0 p=384)	13879.53	90.9	125.06	34.16	Model based on acf & pacf estimations
(7,2,7)(0,1,0 p=384)	13931.32	93.57	132.43	38.86	Model based on acf & pacf estimations
(2,0,2)(0,1,0 p =384)	13973.81	95.59	132.63	37.04	Auto.Arima
(7,0,7)(0,1,0 p=384)	13918.96	92.46	133.79	41.33	Model based on acf & pacf estimations

The table above summarizes the various models I used in forecasting observations from this Divvy bike station. The various model types are ordered lowest to highest based on the test RMSE. The table listed above only includes the top 3 performing models compared to the models which were mentioned in the previous sections, for a full list of the models which were used, please see the appendix.

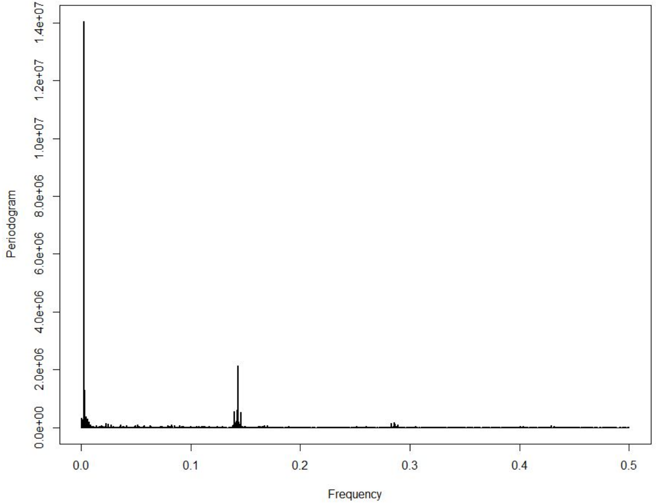
The models based on the acf and pcf values did perform better than the auto.arima models in most cases but in the end the **(4,1,4)(0,1,0 p=384)** model performed the best when it came to test accuracy and is therefore the best fit model out of the models listed above for the bikes taken FROM the Divvy station.



T0 ts Modelling

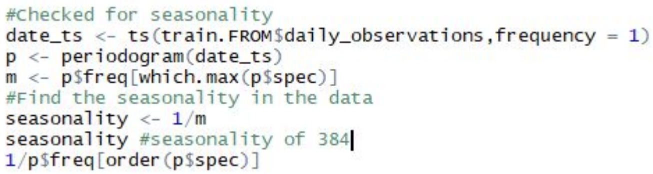
DIVVY

Seasonality



Creating the ts model & frequency

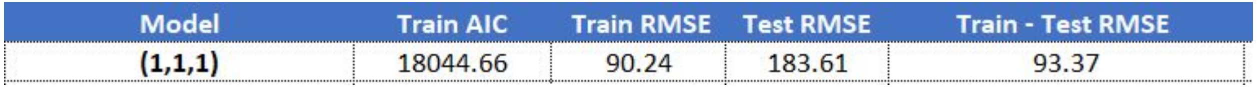
Before running the ts model I used the following code to find the seasonality in R:



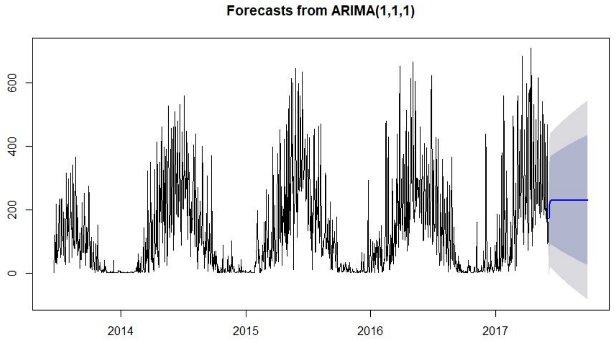
```
#Checked for seasonality
date_ts <- ts(train.FROM$daily_observations,frequency = 1)
p <- periodogram(date_ts)
m <- p$freq[which.max(p$spec)]
#Find the seasonality in the data
seasonality <- 1/m
seasonality #seasonality of 384
1/p$freq[order(p$spec)]
```

The result was an optimum seasonality of 384 for which I also accounted for while creating the ts functions, this was the same seasonality as the FROM data set.

Baseline Model



Model	Train AIC	Train RMSE	Test RMSE	Train - Test RMSE
(1,1,1)	18044.66	90.24	183.61	93.37

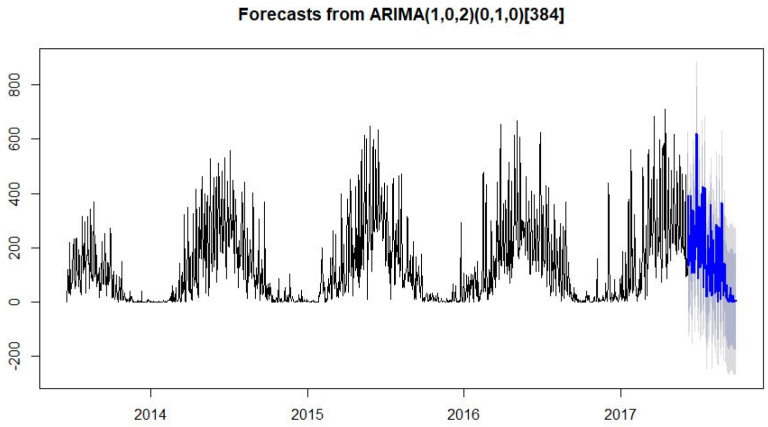


Forecasts from ARIMA(1,1,1)

Ideally the baseline model should be improved by adding seasonality, lowering the test RMSE and improving the train AIC

Auto.Arima Model with Seasonality

Model	Train AIC	Train RMSE	Test RMSE	Train - Test RMSE
(1,0,2) (0,1,0 p = 384)	14148.26	103.29	130.51	27.22

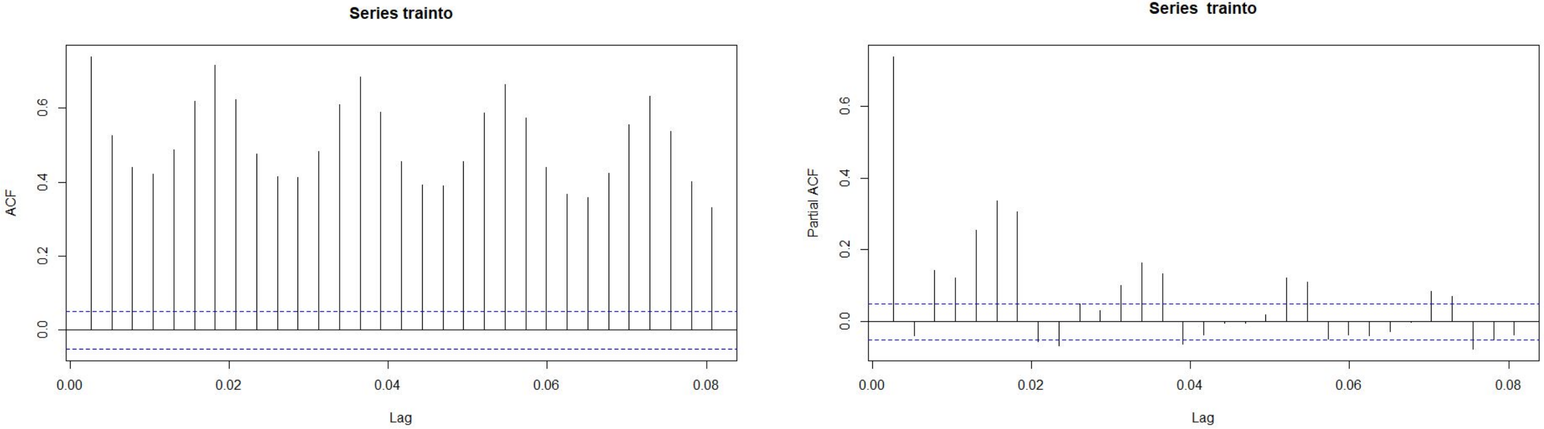


This model’s performance is a considerable improvement over the baseline model. The train AIC and test RMSE decreased considerably. Additionally the forecast, highlighted in blue, was able to pick up on seasonal trends.

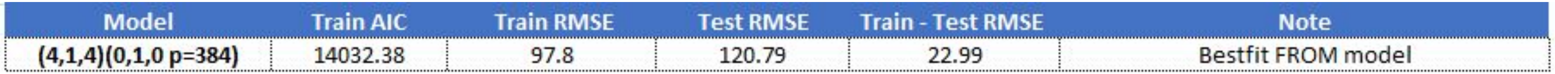
Although the FROM and TO data sets appeared quite similar when it came to seasonality the TO dataset recommendation from auto.arima has a slightly lower p value than the FROM data.

ACF & PACF Models

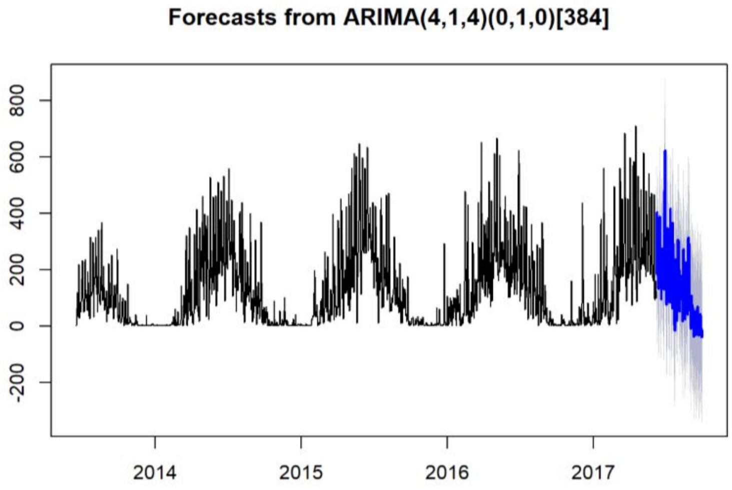
Models were also ran which which were based off of the observations for the ACF and PACF graphs below with a p and q value of 7. The results will be included in the final table.



Best Fit FROM Model



Model	Train AIC	Train RMSE	Test RMSE	Train - Test RMSE	Note
(4,1,4)(0,1,0 p=384)	14032.38	97.8	120.79	22.99	Bestfit FROM model

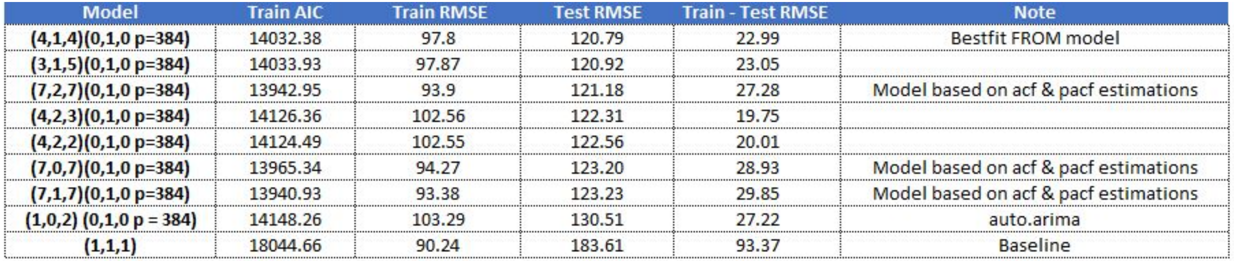


Forecasts from ARIMA(4,1,4)(0,1,0)[384]

In addition to the baseline and auto.arima model, since the TO and FROM data sets shared a common seasonality I used the best fit from model (4,1,4)(0,1,0 p= 384) .

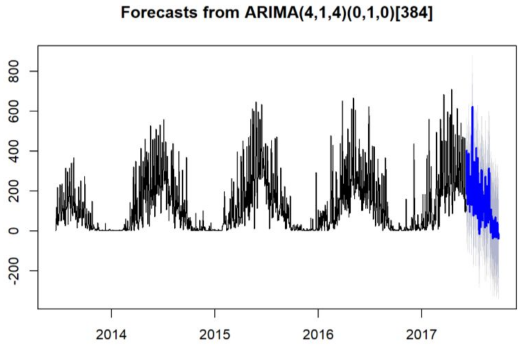
This model has a lower test RMSE and therefore was a better fit model to use compared to auto.arima.

TO Final Comparison Table



Model	Train AIC	Train RMSE	Test RMSE	Train - Test RMSE	Note
(4,1,4)(0,1,0 p=384)	14032.38	97.8	120.79	22.99	Bestfit FROM model
(3,1,5)(0,1,0 p=384)	14033.93	97.87	120.92	23.05	
(7,2,7)(0,1,0 p=384)	13942.95	93.9	121.18	27.28	Model based on acf & pacf estimations
(4,2,3)(0,1,0 p=384)	14126.36	102.56	122.31	19.75	
(4,2,2)(0,1,0 p=384)	14124.49	102.55	122.56	20.01	
(7,0,7)(0,1,0 p=384)	13965.34	94.27	123.20	28.93	Model based on acf & pacf estimations
(7,1,7)(0,1,0 p=384)	13940.93	93.38	123.23	29.85	Model based on acf & pacf estimations
(1,0,2) (0,1,0 p = 384)	14148.26	103.29	130.51	27.22	auto.arima
(1,1,1)	18044.66	90.24	183.61	93.37	Baseline

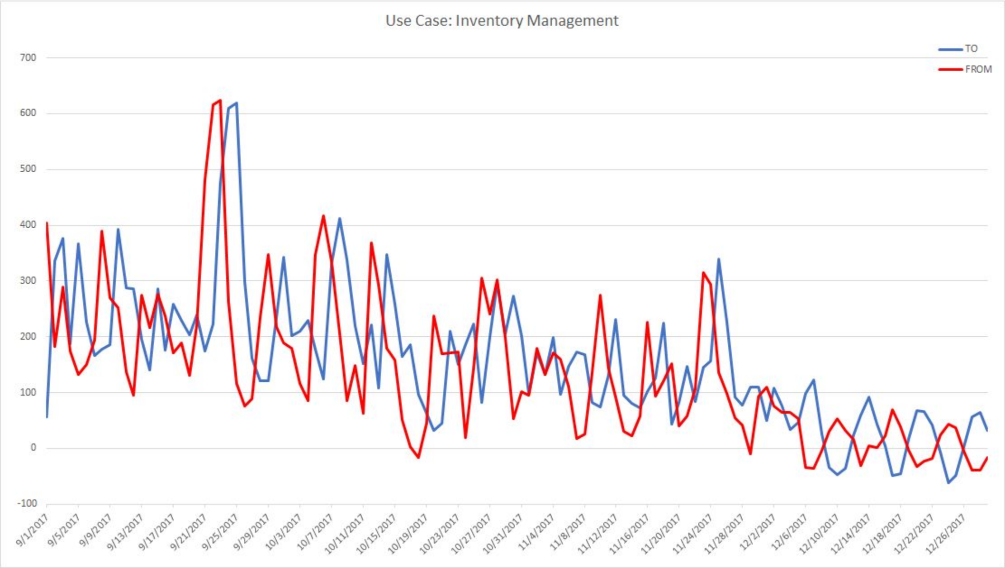
Interestingly enough although the auto.arima model recommended a lower p value compared to the FROM data, the best fit model for the TO data ended up being the same as the FROM data best fit model of **(4,1,4) (0,1,0 p=384).** The table listed above only includes top 3 performing "exploratory" models compared to the models which were mentioned in the previous sections, for a full list of the models which were used, please see the appendix.



Business Use Case
&
Model Improvements

DIVVY

Business Use Case



The graph above depicts the forecasts for TO and FROM. For a business use-case this visualization could help inventory management identify when the demand and supply gap is the highest. For example October 8th & 9th, where the number of bikes taken to the station and from the station vary significantly as depicted by the distance between the TO and FROM lines above, would be 2 days where the demand supply gap varies significantly.

Model Improvements

Continuous improvements to these models are key in order to improve the accuracy of forecasts so that all Divvy Bike users always have a bike when they need it. The following improvements could be made to the model:

- Transforming a forecast so that TO and FROM do not go into the negative since in reality, there can never be negative bikes taken to or from the station. One way to rectify this issue would be possibly using a different model other than Arima, which is known for this issue.
- Although out of scope for the purposes of this capstone the next step would be to use leave one out cross validation (LOOCV) where one point is repeatedly left out of the data set and tested in order to cross-validate in both the TO and FROM datasets.
- Forecasting on an hourly or even minute by minute basis would also improve forecasting accuracy.
- It would be of interest to collect information on why this data set had missing days. One scenario for these missing days is that the sensors recording trips were malfunctioning. A second scenario would be that no trips were taken and therefore there was no start and stop time in the data set. For the purposes of this project I took the mean from the previous and following day for the days which did not have any recorded observations (there were about 99 days added to the TO and FROM data sets each) but receiving this background information on the missing days could possibly change the way missing days were handled.

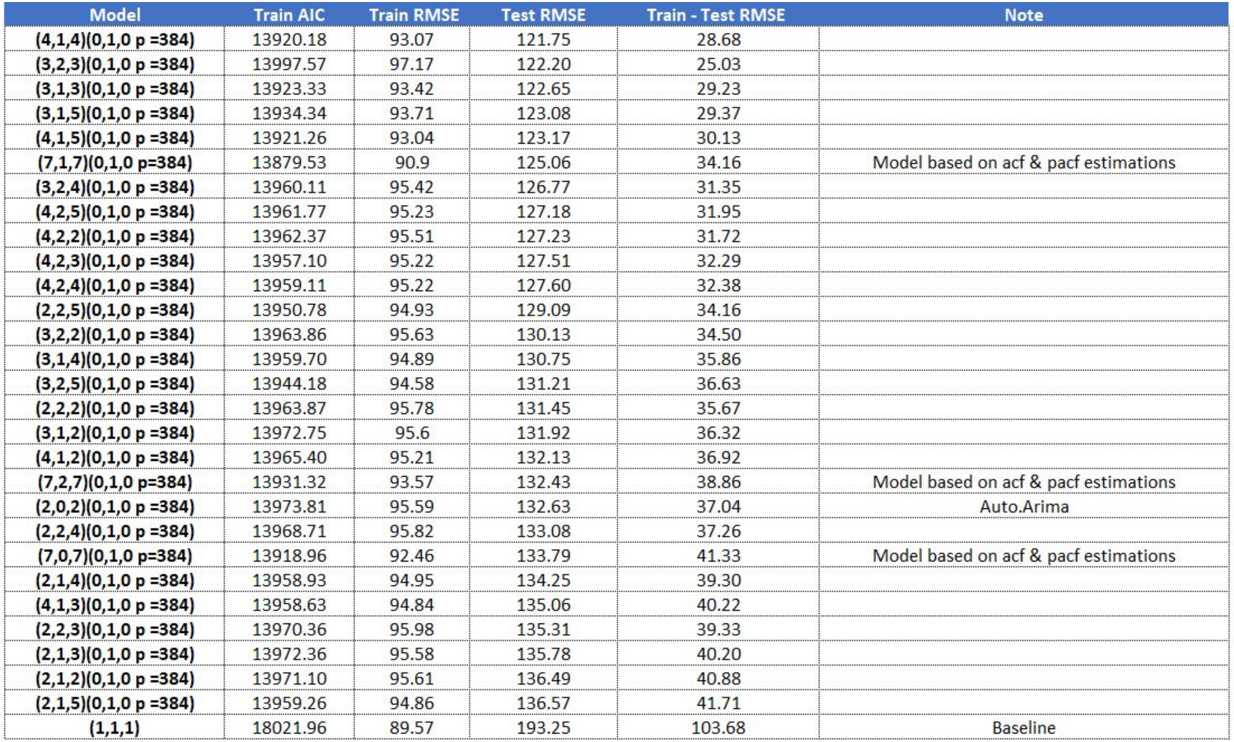
Thank you!

DIVVY

Appendix

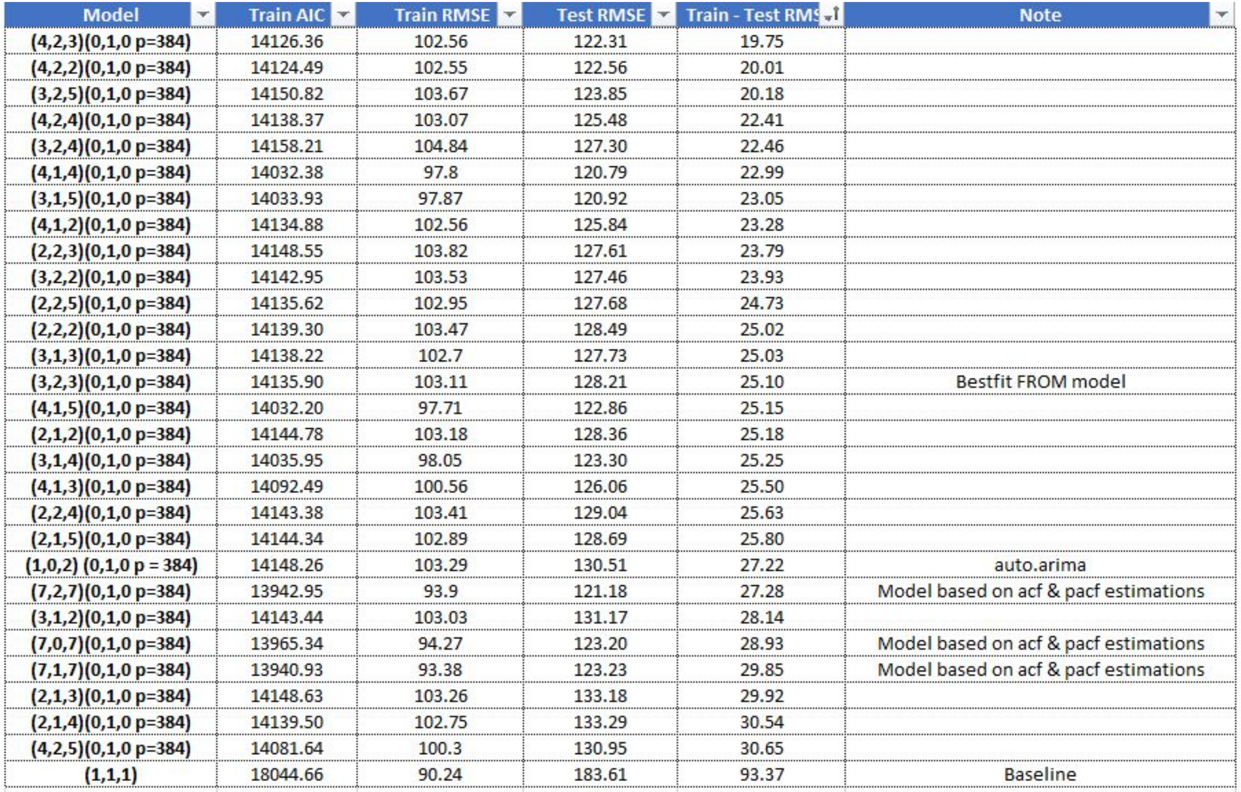
DIVVY

FROM Comprehensive Data Table



Model	Train AIC	Train RMSE	Test RMSE	Train - Test RMSE	Note
(4,1,4)(0,1,0 p =384)	13920.18	93.07	121.75	28.68	
(3,2,3)(0,1,0 p =384)	13997.57	97.17	122.20	25.03	
(3,1,3)(0,1,0 p =384)	13923.33	93.42	122.65	29.23	
(3,1,5)(0,1,0 p =384)	13934.34	93.71	123.08	29.37	
(4,1,5)(0,1,0 p =384)	13921.26	93.04	123.17	30.13	
(7,1,7)(0,1,0 p=384)	13879.53	90.9	125.06	34.16	Model based on acf & pacf estimations
(3,2,4)(0,1,0 p =384)	13960.11	95.42	126.77	31.35	
(4,2,5)(0,1,0 p =384)	13961.77	95.23	127.18	31.95	
(4,2,2)(0,1,0 p =384)	13962.37	95.51	127.23	31.72	
(4,2,3)(0,1,0 p =384)	13957.10	95.22	127.51	32.29	
(4,2,4)(0,1,0 p =384)	13959.11	95.22	127.60	32.38	
(2,2,5)(0,1,0 p =384)	13950.78	94.93	129.09	34.16	
(3,2,2)(0,1,0 p =384)	13963.86	95.63	130.13	34.50	
(3,1,4)(0,1,0 p =384)	13959.70	94.89	130.75	35.86	
(3,2,5)(0,1,0 p =384)	13944.18	94.58	131.21	36.63	
(2,2,2)(0,1,0 p =384)	13963.87	95.78	131.45	35.67	
(3,1,2)(0,1,0 p =384)	13972.75	95.6	131.92	36.32	
(4,1,2)(0,1,0 p =384)	13965.40	95.21	132.13	36.92	
(7,2,7)(0,1,0 p=384)	13931.32	93.57	132.43	38.86	Model based on acf & pacf estimations
(2,0,2)(0,1,0 p =384)	13973.81	95.59	132.63	37.04	Auto.Arima
(2,2,4)(0,1,0 p =384)	13968.71	95.82	133.08	37.26	
(7,0,7)(0,1,0 p=384)	13918.96	92.46	133.79	41.33	Model based on acf & pacf estimations
(2,1,4)(0,1,0 p =384)	13958.93	94.95	134.25	39.30	
(4,1,3)(0,1,0 p =384)	13958.63	94.84	135.06	40.22	
(2,2,3)(0,1,0 p =384)	13970.36	95.98	135.31	39.33	
(2,1,3)(0,1,0 p =384)	13972.36	95.58	135.78	40.20	
(2,1,2)(0,1,0 p =384)	13971.10	95.61	136.49	40.88	
(2,1,5)(0,1,0 p =384)	13959.26	94.86	136.57	41.71	
(1,1,1)	18021.96	89.57	193.25	103.68	Baseline

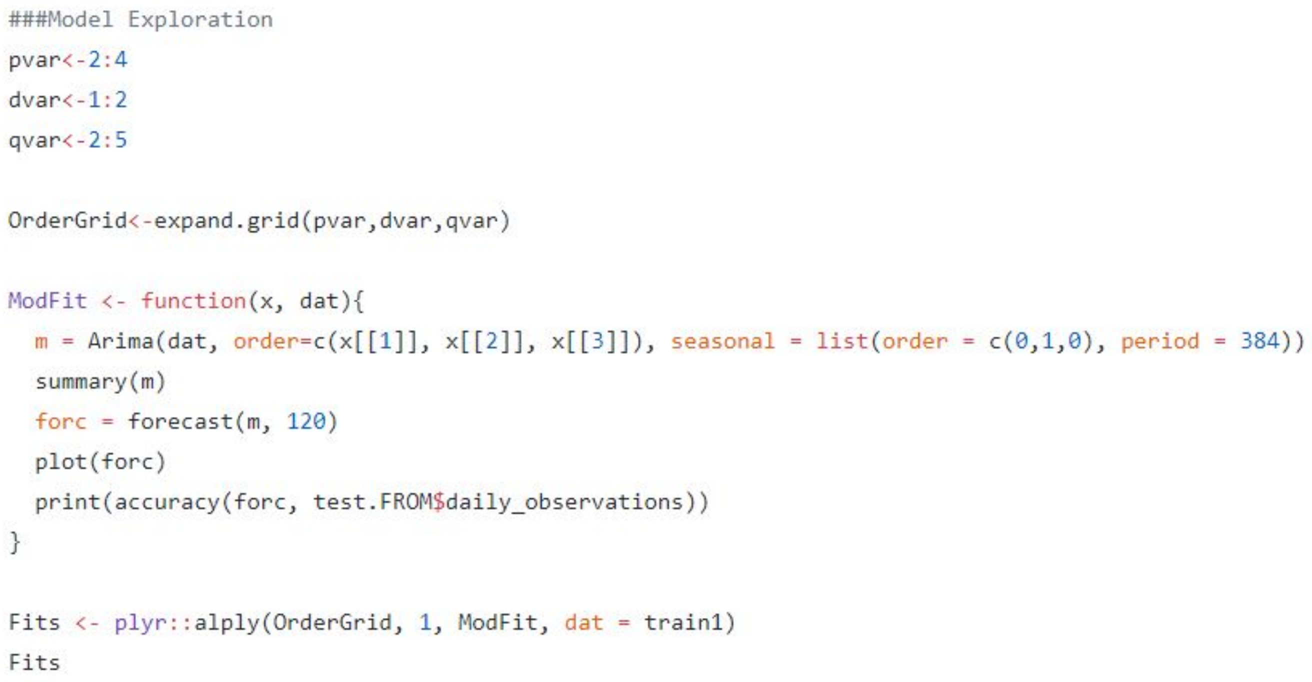
TO Comprehensive Data Table



Model	Train AIC	Train RMSE	Test RMSE	Train - Test RMS	Note
(4,2,3)(0,1,0 p=384)	14126.36	102.56	122.31	19.75	
(4,2,2)(0,1,0 p=384)	14124.49	102.55	122.56	20.01	
(3,2,5)(0,1,0 p=384)	14150.82	103.67	123.85	20.18	
(4,2,4)(0,1,0 p=384)	14138.37	103.07	125.48	22.41	
(3,2,4)(0,1,0 p=384)	14158.21	104.84	127.30	22.46	
(4,1,4)(0,1,0 p=384)	14032.38	97.8	120.79	22.99	
(3,1,5)(0,1,0 p=384)	14033.93	97.87	120.92	23.05	
(4,1,2)(0,1,0 p=384)	14134.88	102.56	125.84	23.28	
(2,2,3)(0,1,0 p=384)	14148.55	103.82	127.61	23.79	
(3,2,2)(0,1,0 p=384)	14142.95	103.53	127.46	23.93	
(2,2,5)(0,1,0 p=384)	14135.62	102.95	127.68	24.73	
(2,2,2)(0,1,0 p=384)	14139.30	103.47	128.49	25.02	
(3,1,3)(0,1,0 p=384)	14138.22	102.7	127.73	25.03	
(3,2,3)(0,1,0 p=384)	14135.90	103.11	128.21	25.10	Bestfit FROM model
(4,1,5)(0,1,0 p=384)	14032.20	97.71	122.86	25.15	
(2,1,2)(0,1,0 p=384)	14144.78	103.18	128.36	25.18	
(3,1,4)(0,1,0 p=384)	14035.95	98.05	123.30	25.25	
(4,1,3)(0,1,0 p=384)	14092.49	100.56	126.06	25.50	
(2,2,4)(0,1,0 p=384)	14143.38	103.41	129.04	25.63	
(2,1,5)(0,1,0 p=384)	14144.34	102.89	128.69	25.80	
(1,0,2) (0,1,0 p = 384)	14148.26	103.29	130.51	27.22	auto.arima
(7,2,7)(0,1,0 p=384)	13942.95	93.9	121.18	27.28	Model based on acf & pacf estimations
(3,1,2)(0,1,0 p=384)	14143.44	103.03	131.17	28.14	
(7,0,7)(0,1,0 p=384)	13965.34	94.27	123.20	28.93	Model based on acf & pacf estimations
(7,1,7)(0,1,0 p=384)	13940.93	93.38	123.23	29.85	Model based on acf & pacf estimations
(2,1,3)(0,1,0 p=384)	14148.63	103.26	133.18	29.92	
(2,1,4)(0,1,0 p=384)	14139.50	102.75	133.29	30.54	
(4,2,5)(0,1,0 p=384)	14081.64	100.3	130.95	30.65	
(1,1,1)	18044.66	90.24	183.61	93.37	Baseline

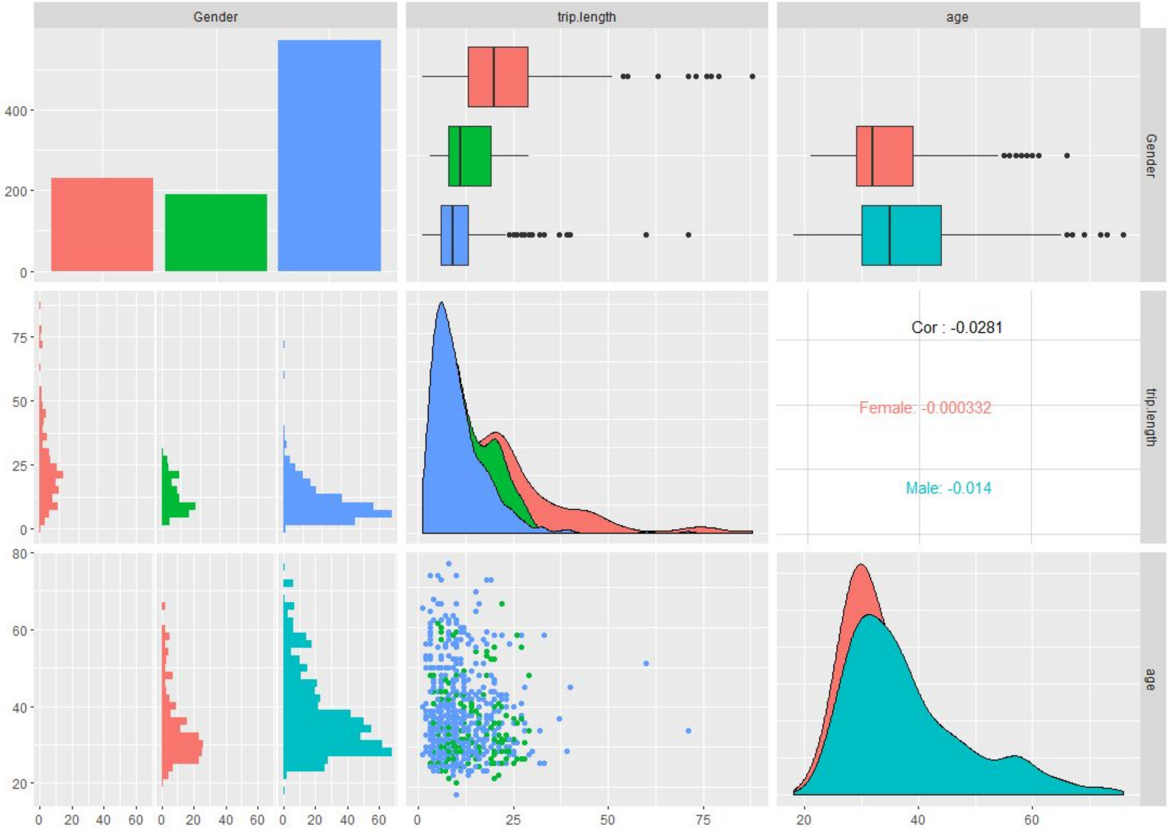
Code to Automate Arima Model Testing

In order to prevent excessive amounts of manual labour in trying out various combinations of the Arima model I ran the following code below:



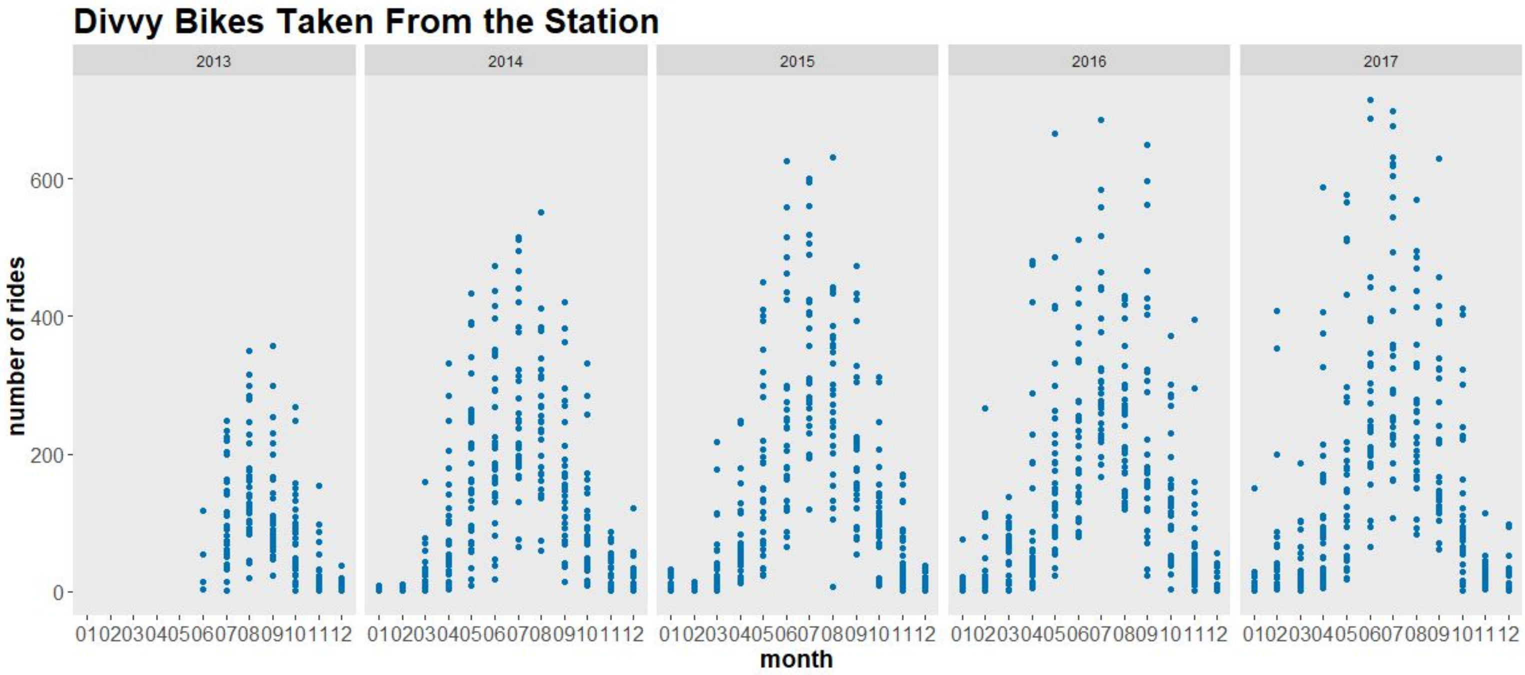
```
###Model Exploration
pvar<-2:4
dvar<-1:2
qvar<-2:5

OrderGrid<-expand.grid(pvar,dvar,qvar)

ModFit <- function(x, dat){
  m = Arima(dat, order=c(x[[1]], x[[2]], x[[3]]), seasonal = list(order = c(0,1,0), period = 384))
  summary(m)
  forc = forecast(m, 120)
  plot(forc)
  print(accuracy(forc, test.FROM$daily_observations))
}

Fits <- plyr::alply(OrderGrid, 1, ModFit, dat = train1)
Fits
```

Exploratory Data Analysis



Visualization based off of a random subset of 1000 rows of the original data set in order to analyze further

Data: Lake Shore Dr & Monroe Station


Divvy Bikes Taken From the Station
number of rides
month