# Divvy

*Tina Carr*

*November 13, 2018*

```
#Import the data
data = read.csv("C:/Users/chris/Desktop/Divvy/Divvy_Trips.csv", header = TRUE, sep =
",")
head(data)
```

```
##   ï..TRIP.ID             START.TIME              STOP.TIME BIKE.ID
## 1   17536701 12/31/2017 11:58:00 PM 01/01/2018 12:03:00 AM    3304
## 2   17536699 12/31/2017 11:54:00 PM 01/01/2018 12:18:00 AM    4906
## 3   17536700 12/31/2017 11:54:00 PM 01/01/2018 12:18:00 AM    5975
## 4   17536698 12/31/2017 11:48:00 PM 12/31/2017 11:53:00 PM    5667
## 5   17536697 12/31/2017 11:42:00 PM 12/31/2017 11:47:00 PM    5353
## 6   17536696 12/31/2017 11:41:00 PM 12/31/2017 11:51:00 PM    5840
##   TRIP.DURATION FROM.STATION.ID                  FROM.STATION.NAME
## 1           284             159           Claremont Ave & Hirsch St
## 2         1,441             145 Mies van der Rohe Way & Chestnut St
## 3         1,402             145 Mies van der Rohe Way & Chestnut St
## 4           315             340           Clark St & Wrightwood Ave
## 5           272             240        Sheridan Rd & Irving Park Rd
## 6           589              93           Sheffield Ave & Willow St
##   TO.STATION.ID                    TO.STATION.NAME  USER.TYPE GENDER
## 1            69              Damen Ave & Pierce Ave Subscriber   Male
## 2           145 Mies van der Rohe Way & Chestnut St   Customer
## 3           145 Mies van der Rohe Way & Chestnut St   Customer
## 4           143           Sedgwick St & Webster Ave Subscriber   Male
## 5           245          Clarendon Ave & Junior Ter Subscriber   Male
## 6           343         Racine Ave & Wrightwood Ave Subscriber   Male
##   BIRTH.YEAR FROM.LATITUDE FROM.LONGITUDE
## 1       1988      41.90778      -87.68585
## 2         NA      41.89859      -87.62192
## 3         NA      41.89859      -87.62192
## 4       1963      41.92955      -87.64312
## 5       1977      41.95425      -87.65441
## 6       1988      41.91369      -87.65286
##                          FROM.LOCATION TO.LATITUDE TO.LONGITUDE
## 1         POINT (-87.685854 41.907781)    41.90940    -87.67769
## 2 POINT (-87.6219152258 41.8985866514)    41.89859    -87.62192
## 3 POINT (-87.6219152258 41.8985866514)    41.89859    -87.62192
## 4         POINT (-87.643118 41.929546)    41.92217    -87.63889
## 5         POINT (-87.654406 41.954245)    41.96100    -87.64960
## 6         POINT (-87.652855 41.913688)    41.92889    -87.65897
##                            TO.LOCATION Boundaries...ZIP.Codes Zip.Codes
## 1 POINT (-87.6776919292 41.9093960065)                     4    21,560
## 2 POINT (-87.6219152258 41.8985866514)                     6    21,182
## 3 POINT (-87.6219152258 41.8985866514)                     6    21,182
## 4         POINT (-87.638888 41.922167)                    16    21,190
## 5         POINT (-87.649603 41.961004)                    53    21,186
## 6         POINT (-87.658971 41.928887)                    16    21,190
##   Community.Areas Wards
## 1              25    41
## 2              37    11
## 3              37    11
## 4              68    34
```

```
## 5              57    39
## 6              68    11
```

```
#Created subsets in order to further explore trips to and from this station
FROM <- subset(data, FROM.STATION.NAME == "Lake Shore Dr & Monroe St")
TO <- subset(data, TO.STATION.NAME == "Lake Shore Dr & Monroe St")
```

```
##FROM Data Prep
#created a df with start.time only
FROMts <- FROM[ -c(1, 3:22) ]

#removed the time element, only date is relevant since this is a daily forecast
library(tidyr)
FROMts <- separate(FROMts, START.TIME, c("start.date", "start.time", "start.time.amp
m"), sep = " ")
FROMts <- FROMts[ -c(2:3) ]

library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
FROMts$start.date <- mdy(FROMts$start.date)
str(FROMts)
```

```
## 'data.frame':    210255 obs. of  1 variable:
##  $ start.date: Date, format: "2017-12-28" "2017-12-28" ...
```

```
FROMtstest <- FROMts

library(lubridate)
FROMts$year <- year(FROMts$start.date)
FROMts$month <- month(FROMts$start.date)
FROMts$day <- day(FROMts$start.date)

library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:lubridate':
##
##     intersect, setdiff, union
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

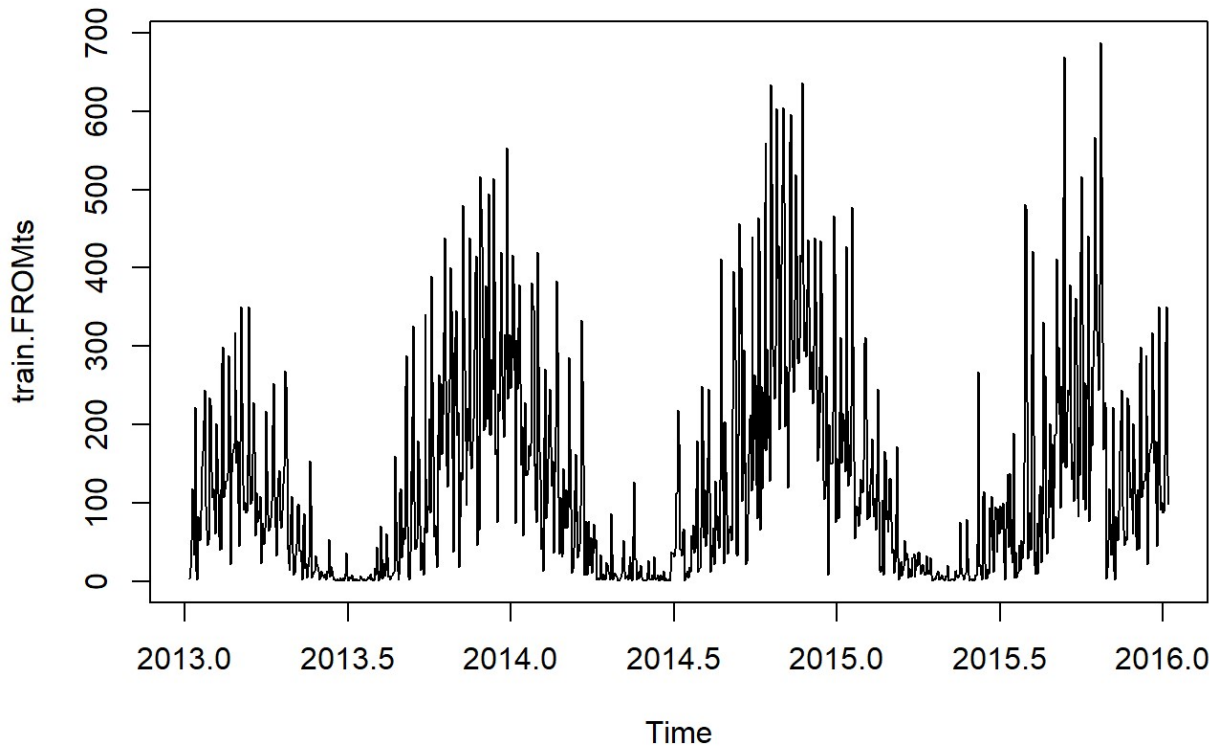
```
dailyFROM <- FROMts %>%
  group_by(year, month, day)

FROMper_day <- summarize(dailyFROM, daily_observations = n())
FROMper_day <- unite(FROMper_day, "date", year,month,day, sep = "-")

#Formatted to a time object
FROMper_day$date <- strptime(FROMper_day$date, format='%Y-%m-%d')
str(FROMper_day)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    1545 obs. of  2 variables:
##  $ date              : POSIXlt, format: "2013-06-27" "2013-06-28" ...
##  $ daily_observations: int  4 17 52 118 75 34 36 222 13 2 ...
```

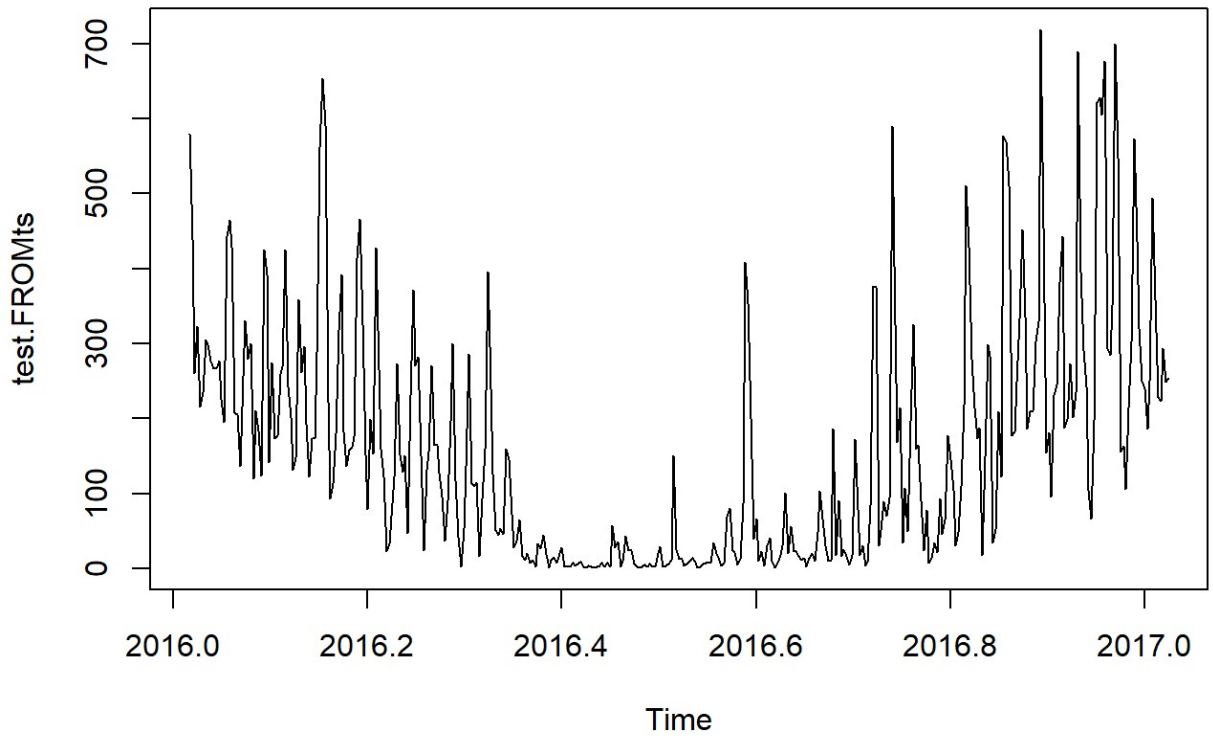
```
sum(FROMper_day$daily_observations) #210255 observations, this matches with the origin
al dataset
```

```
## [1] 210255
```

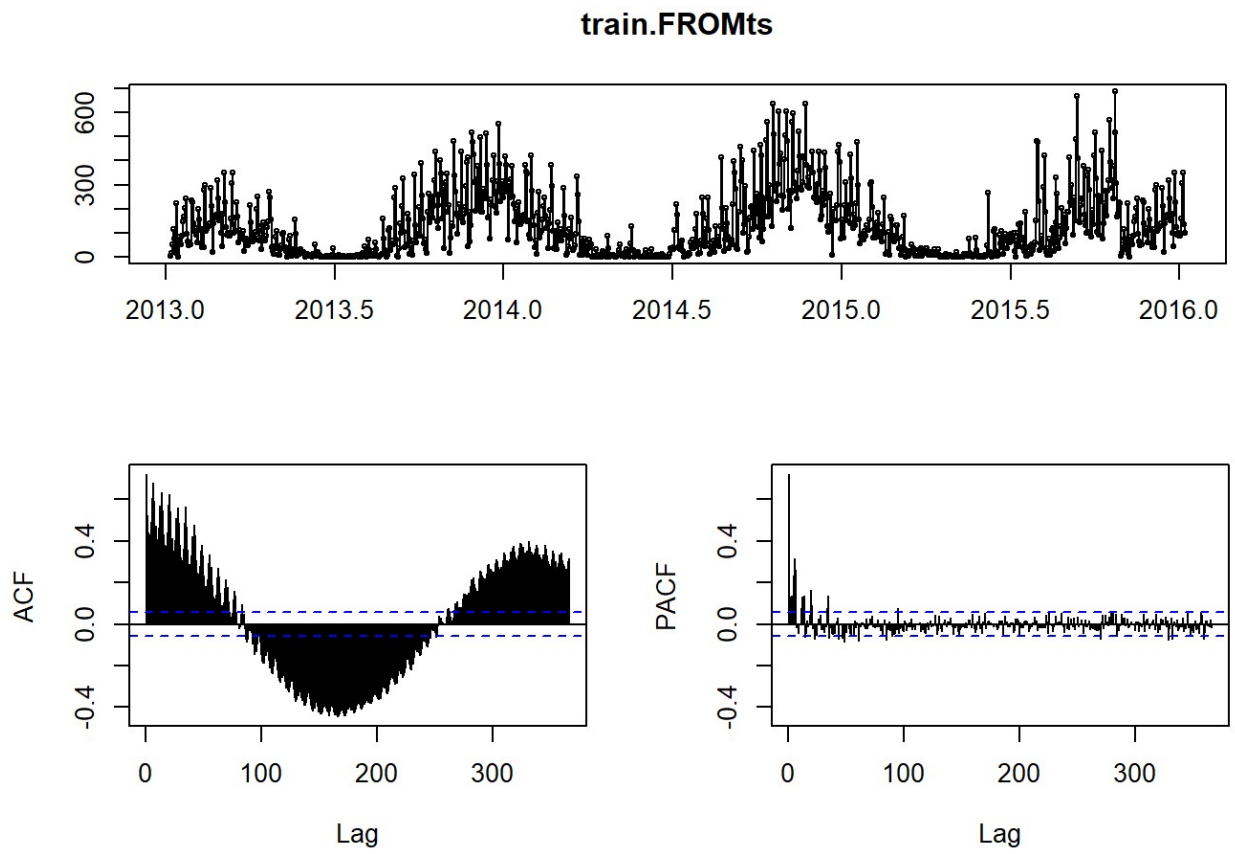
```
#spearated the train and test datasets
train.FROM = FROMper_day[1:1027,]
test.FROM = FROMper_day[1028:1467,]

#created the time series
train.FROMts = ts(train.FROM$daily_observations, start = c(2013, 6,27), end = c(2016,
7,15), frequency = 365 )
plot(train.FROMts)
```




```
test.FROMts = ts(test.FROM$daily_observations, start = c(2016, 7,16), end = c(2017,10,
07), frequency = 365 )
plot(test.FROMts)
```



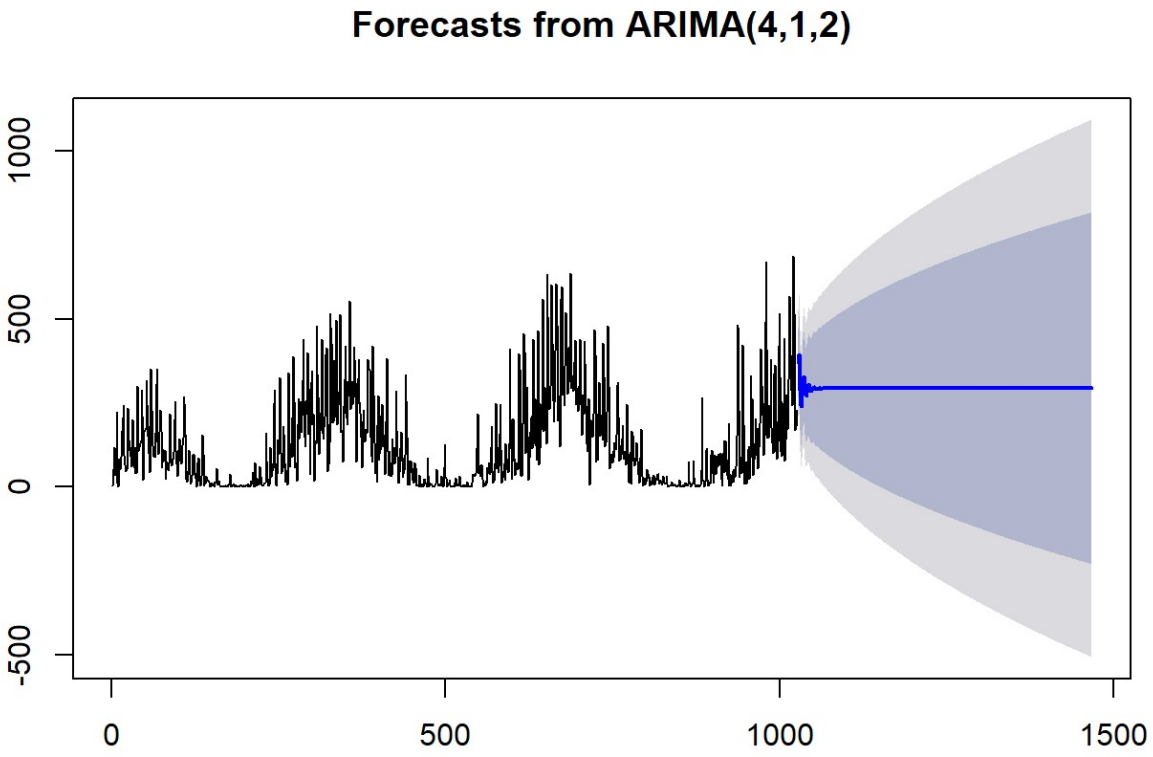

```
#check if the data s stationary
library(forecast)
tsdisplay(train.FROMts) #will account for 7 lags in model 2
```




```
#initial run with "bare" model in auto.arima
library(forecast)
FROMmodel1 <- auto.arima(train.FROM$daily_observations)
summary(FROMmodel1)
```

```
## Series: train.FROM$daily_observations
## ARIMA(4,1,2)
##
## Coefficients:
##          ar1      ar2      ar3      ar4      ma1     ma2
##       0.5261  -0.3830  -0.1045  -0.2405  -1.0164  0.2979
## s.e.  0.0765   0.0877   0.0353   0.0534   0.0685  0.1223
##
## sigma^2 estimated as 6577:  log likelihood=-5963.53
## AIC=11941.05   AICc=11941.16   BIC=11975.59
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 1.095815 80.81898 52.64128 -103.1106 134.8476 0.8289456
##                      ACF1
## Training set -0.01572758
```

```
#created the forecast
library(tseries)
FROMmodel1for = forecast(FROMmodel1, 440)
plot(FROMmodel1for)  ##auto arima seems like it did not account for seasonality, will
incorporate this in model 2
```




```
#tested the accuracy of the "bare" model forecast
accuracy(FROMmodel1for, test.FROM$daily_observations)
```

```
##                       ME      RMSE       MAE        MPE      MAPE
## Training set    1.095815  80.81898  52.64128  -103.1106  134.8476
## Test set     -119.961080 200.22664 174.43833 -1586.4517 1597.3687
##                   MASE        ACF1
## Training set 0.8289456 -0.01572758
## Test set     2.7468917          NA
```