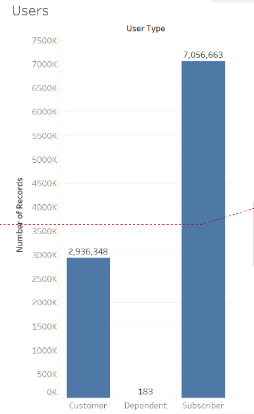
Descriptive data analysis of Divvy Bike ~ Weather data

# Contents

## Demographics

### Who Rides Divvy Bikes and When?

The data categorizes rides by user type. Users are either subscribers, who pay a yearly fee for unlimited 30 minute rides; customers, who pay $9.95 for one day of unlimited 30 minute rides; or dependents. Subscribers make up ~70% of all rides. Customers make up ~30%. ~64% of subscribers are between the ages of 25-40. Subscribers are ~75% male, ~25% female. Ages 25-35 have the largest proportion of females.




**Comment [BL1]:** Recommendation for Divvy, to increase riders target women as they do not make up a normal proportion of the total.

### Are subscribers using Divvy primarily for their work commute?

~39% of all rides take place between 6am and 6pm Monday-Friday and subscribers make up ~80% of those rides.

### What about the weekends?

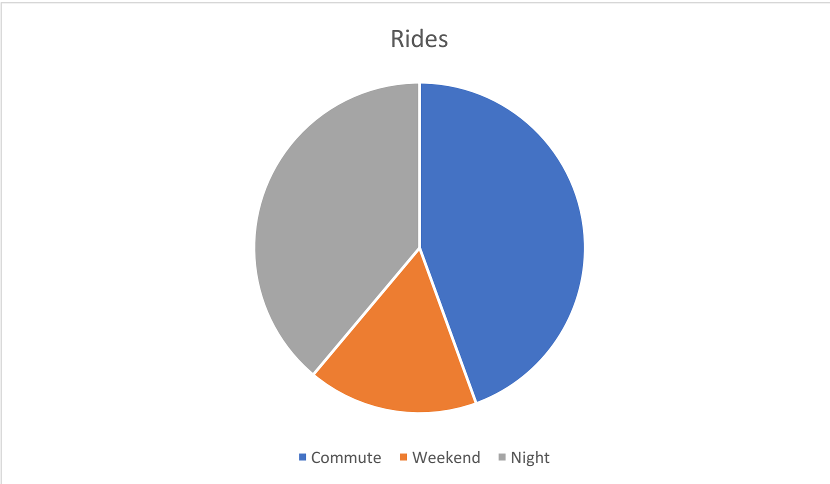
~15% of all rides take place between 6am and 6pm Saturday and Sunday. As might be expected customers were more promentient during this period making up over 50% of these rides.

### Drinking and Biking?

~45% of all rides take place between 6pm and 6am (many bars close at 4am), nearly half of those rides occur over the weekend. To avoid any interaction from workers running late it should be noted that

~35% of all rides take place between 9pm and 6am. The user proportions are the same as the general sample.

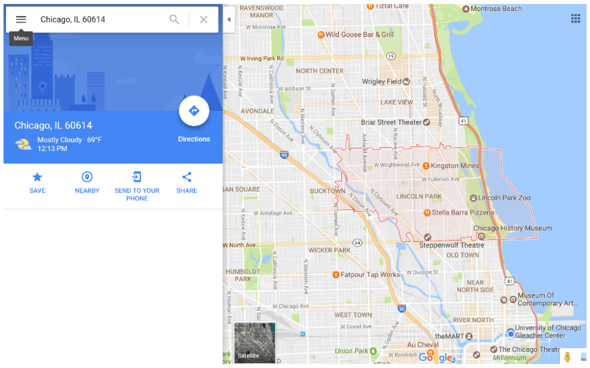
### Total Breakdown



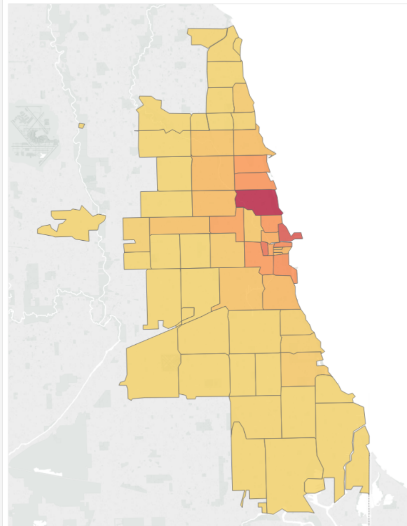
## Where do Rides Go?

### Origin

Zip code 60614 is the most popular ride origin. This is the Lincoln Park neighborhood.

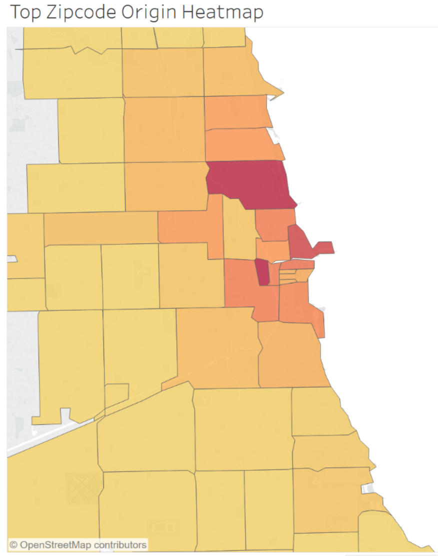



Top Zipcode Origin Heatmap



During weekday, commuting hours (6am to 6pm) 60661 is the most frequent origin. These are the closest Divvy stations to Ogilvie and Union train stations. Lincoln Park is the second most popular origin station during these hours.

Top Zipcode Origin Heatmap




### Destination

In the morning (6am to Noon) 60661 has an even larger proportion of rides. 60611(14.6%), 60601(13.9%), and 60607(11.7%) are the top three destinations for 60661 in the morning. Michigan & Lake(4%), LaSalle & Jackson(4%), and Michigan & Washington(4%) are the top three station destinations for 60661 in the morning. For 60614 the top zip codes and stations are 60614 (30%), 60611(12%), 60610(10%), Sheffield Ave & Fullerton Ave(3%), Streeter Dr. & Grand Ave(2.8%), and Michigan Ave & Oak St(2.6%).

On weekdays from 12-6pm 60611 is the busiest origin with Lincoln Park a close second. 60611 is comprised of the Magnificent Mile, Navy Pier, and the Gleacher Center. For 60611 the top zip codes and stations are 60611 (20.6%), 60661(16.9%), 60614(16%), Streeter Dr. & Grand Ave(4.8%), Canal St & Adams St(4.8%), and Clinton St & Washington Blvd(4.5%). For 60614 the top zip codes and stations are 60614 (40%), 60611(12%), 60657(10.9%), Streeter Dr. & Grand Ave(4.6%), Lake Shore Dr. & North Blvd(2.8%), and Michigan Ave & Oak St(2.7%).

On the weekends (Saturday and Sunday, 6am to 6pm) Lincoln Park is by far the most popular origin. Top zip code destinations are 60614, 60611, and 60610 in the mornings, 60657 in the afternoon. Top station destinations are Streeter Dr. & Grand Ave, Lake Shore Dr. & North Bvld, and Michigan & Oak in the mornings and Clark & Lincoln in the afternoons.

| ZC Origin | 60661a | 60661b | 60661c | | 60614a | 60614b | 60614c | | 60611a | 60611b | 60611c | | 60614w | 60614w | 60614w |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Morning Destination z | 60611 | 60601 | 60607 | | 60614 | 60611 | 60610 | | 60611 | 60614 | 60606 | | 60614 | 60611 | 60610 |
| Afternoon Destination z | 60607 | 60661 | 60614 | | 60614 | 60611 | 60657 | | 60611 | 60661 | 60614 | | 60614 | 60611 | 60657 |
| Morning Destination s | Michigan & Lake | LaSalle & Jackson | Michigan & Washington | | Sheffield & Fullerton | Streeter & Grand | Michigan & Oak | | Streeter & Grand | Michigan & Oak | Millennium Park | | Streeter & Grand | LakeShore & North | Michigan & Oak |
| Afternoon Destination s | Canal & madison | Michigan & Washington | Aberdeen & Monroe | | Streeter & Grand | LakeShore & North | Michigan & Oak | | Streeter & Grand | Canal & Adams | Clinton & Washinton | | Streeter & Grand | clark & lincoln | LakeShore &North |

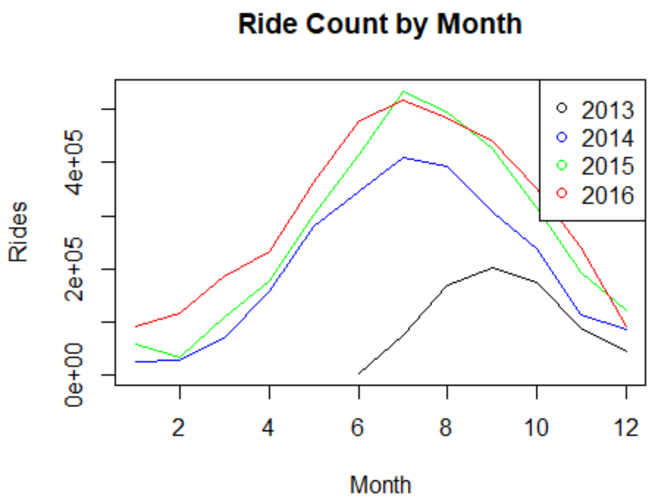
Streeter & Grand is the most popular destination among the 3 different zip code origins. Michigan & Oak, and Lake Shore & North are also present in multiple origins' destinations.

**Comment [BL2]:** Are there enough bike lanes from the zip code origins to these 3 popular stations? Are the enough bike lines in and out of Lincoln Park?

# Exploratory Data Analysis in R

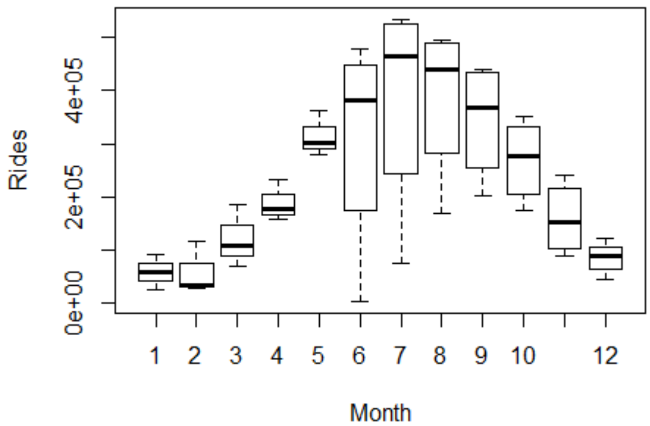
## Ride Count Over the Year(s)

```
myQuery <- "select month(start_time) as monthT,year(start_time) as yeart,
                   count(trip_id) as count1
           from trips t
           group by monthT,yeart
           order by  monthT ,yeart;"
outdata1= dbGetQuery(connection, myQuery)

attach(outdata1)
class(outdata1)

## [1] "data.frame"

with(outdata1, plot(monthT, count1, main= "Ride Count by Month", xlab = "Mont
h", ylab = "Rides", type = "n"))
with(subset(outdata1, yeart== 2013), lines(monthT, count1, col = "black"))
with(subset(outdata1, yeart== 2014), lines(monthT, count1, col = "blue"))
with(subset(outdata1, yeart== 2015), lines(monthT, count1, col = "green"))
with(subset(outdata1, yeart== 2016), lines(monthT, count1, col = "red"))
legend("topright", pch = 1, col= c("black","blue", "green", "red"), legend =
c("2013","2014", "2015", "2016"))
```



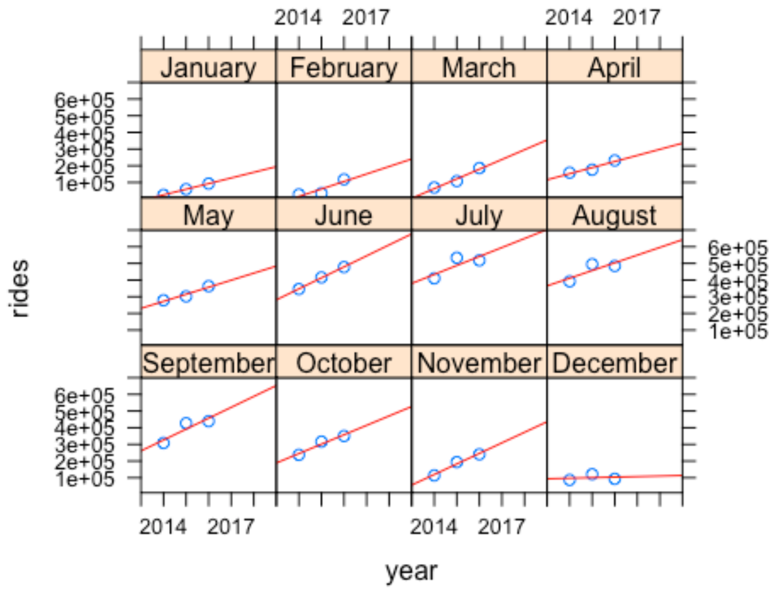

This pattern would seem to indicate that weather is a significant driver for number of rides, the steep drop toward the end of the year could also be due in part to the traditional American holiday periods.

```
outdata2 <- transform(outdata1, monthT = factor(monthT))
boxplot(count1 ~ monthT, xlab = "Month", ylab = "Rides")
```




Large amount of variance in the summer and fall compared to winter and spring. Is this due to the unpredictable nature of weather or perhaps tourism?

```
#lattice example
library(lattice)
outdata3 <- subset(outdata1, yeart != 2013)
y<-outdata3$count1
x<-outdata3$yeart
f<-factor(outdata3$monthT, labels = c("September","October","November","Decem
ber",
                                      "May","June","July","August",
                                 "January","February","March","April"), level
s = c('9','10','11','12','5','6','7','8','1','2','3','4'))
xyplot(y~x|f,xlab="year", ylab="rides",xlim = c(2013.0,2019.0), ylim =c(10000
,700000),panel=function(x,y,...){
  panel.xyplot(x,y,...)
  panel.lmline(x,y,col=2)
})
```

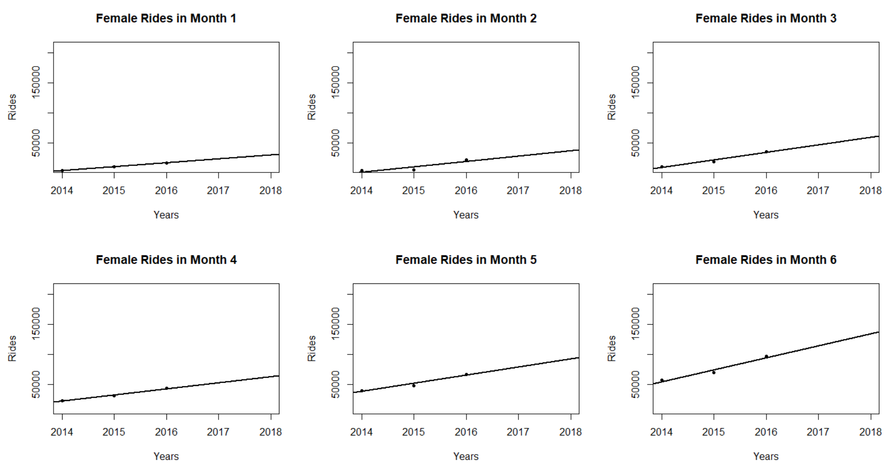



Clear linear growth except for December.

### When do Women Ride Divvy?

```
myQuery <- "select month(start_time) as monthT,year(start_time) as yeart, gen
der,
                   count(trip_id) as count1
            from trips t
            Where user_type = 'Subscriber'
            group by monthT,yeart,gender
            order by  monthT ,yeart;"
outdata_subscribers= dbGetQuery(connection, myQuery)

attach(outdata_subscribers)

## The following objects are masked from outdata1:
##
##     count1, monthT, yeart

class(outdata_subscribers)

## [1] "data.frame"

#create a matrix to hold slope coefficient
trend_female = matrix(, nrow=12, ncol=1)
```

```
#Does the linear increase of female riders mimic the general increase in ride
rs?
for(i in 1:12)
{
predict_month <- i

outdata4 <- subset(outdata_subscribers,monthT==predict_month & yeart != 2013
& gender=="F")
#removing 2013 as it was the first half year of the program and the numbers a
re outliers.
lm2<- lm(count1~yeart,data=outdata4)

month.plot <- with(outdata4, plot(yeart, count1, main = paste("Female Rides i
n Month", i, sep=" "), xlab = "Years", ylab = 'Rides', pch = 20, xlim = c(201
4.0,2018.0), ylim =c(30000*.3,700000*.3)))
abline(lm2, lwd = 2)
trend_female[i,1] = summary(lm2)$coefficients[2,1]
next
}
```




Same signs of growth, but how does the monthly trend compare with all rides?

```
(trends <- cbind(trend_female, trend_male, trend_female/trend_male))
```

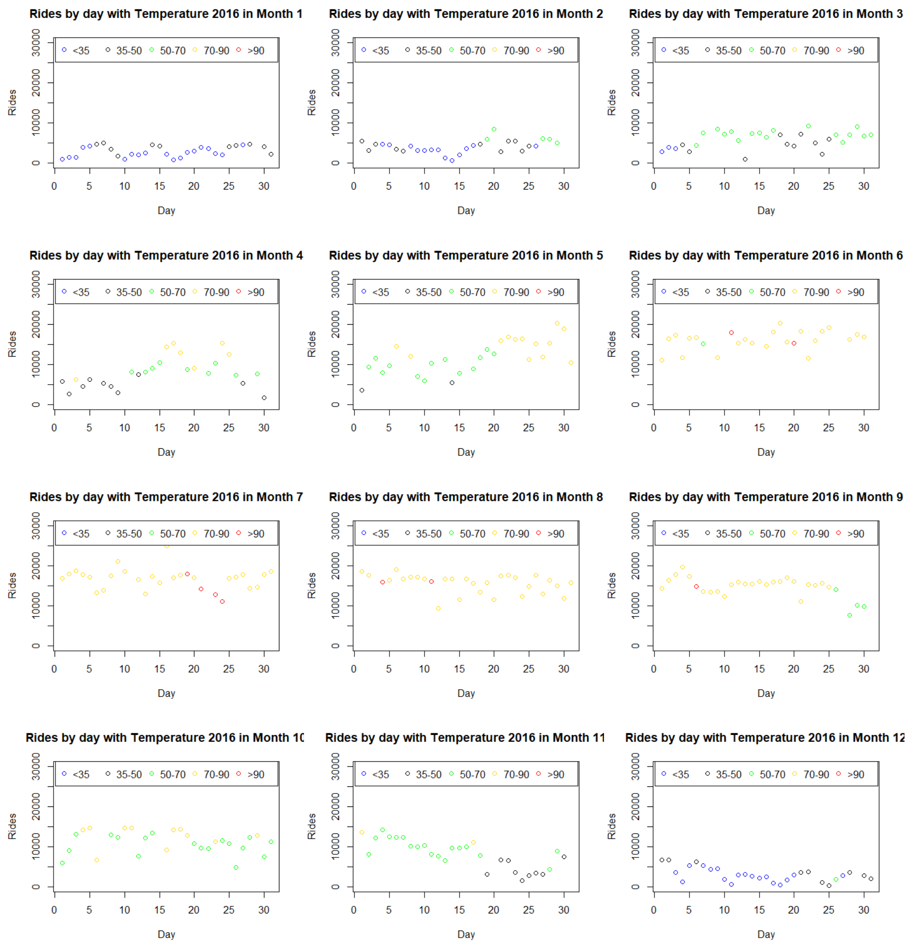
```
##       [,1]   [,2]      [,3]
##  [1,]  849 4585.5 0.1851488
##  [2,]  849 4585.5 0.1851488
##  [3,]  849 4585.5 0.1851488
##  [4,]  849 4585.5 0.1851488
##  [5,]  849 4585.5 0.1851488
##  [6,]  849 4585.5 0.1851488
##  [7,]  849 4585.5 0.1851488
##  [8,]  849 4585.5 0.1851488
##  [9,]  849 4585.5 0.1851488
## [10,]  849 4585.5 0.1851488
## [11,]  849 4585.5 0.1851488
## [12,]  849 4585.5 0.1851488
```

Less forecasted growth of Women rides than male rides. Q1 and Q4 might be a good place to start campaigning.

## Rides and Temperature

```
myQuery <- "select day(t.start_time) as dayt, month(t.start_time) as monthT,y
ear(t.start_time) as yeart, w.max_temp,
                   count(trip_id) as count1
            from trips t
            join weather w on date(t.start_time) = w.record_date
            group by dayt,monthT,yeart
            order by  dayt,monthT ,yeart;"
outdata_temperature= dbGetQuery(connection, myQuery)
```

```
attach(outdata_temperature)

## The following objects are masked from outdata_subscribers:
##
##     count1, monthT, yeart

## The following objects are masked from outdata1:
##
##     count1, monthT, yeart

class(outdata_temperature)

## [1] "data.frame"

for(i in 1:12)
{
predict_month <- i
with(outdata_temperature, plot(max_temp, count1, main = paste("Rides by day w
ith Temperature 2016 in Month", i, sep=" "), xlim = c(1,31), ylim=c(0,30000),
xlab = "Day", ylab = "Rides", type = "n"))
with(subset(outdata_temperature, monthT==i & yeart==2016 & max_temp< 50 & max
_temp> 35), points(dayt, count1, col = "black"))
with(subset(outdata_temperature,monthT==i & yeart==2016 & max_temp< 35), poin
ts(dayt, count1, col = "blue"))
with(subset(outdata_temperature,monthT==i & yeart==2016 & max_temp<70 & max_t
emp>50), points(dayt, count1, col = "green"))
with(subset(outdata_temperature,monthT==i & yeart==2016 & max_temp>70 & max_t
emp<90), points(dayt, count1, col = "gold"))
with(subset(outdata_temperature,monthT==i & yeart==2016 & max_temp>90), point
s(dayt, count1, col = "red"))
legend("top", horiz = TRUE, pch = 1, col= c("blue", "black","green", "gold",
"red"), legend = c("<35","35-50", "50-70", "70-90", ">90"))
next
}
```




January, December, and the end of November show the base traffic pattern due to the die-hard commuters. But rest of the months show the effect of weather.

```
#quantifying temperature's effect
lm_temp <- lm(count1 ~ max_temp,outdata_temperature)
summary(lm_temp)

##
## Call:
## lm(formula = count1 ~ max_temp, data = outdata_temperature)
##
```

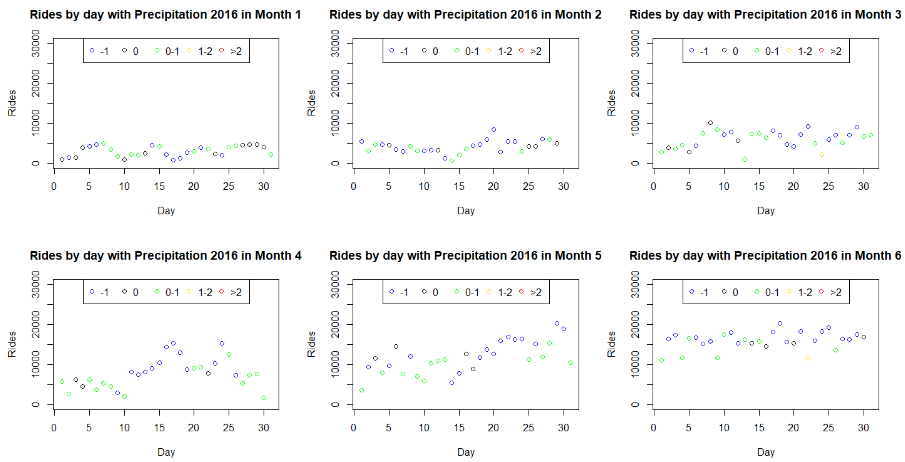
```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8518.9 -1825.1    87.5  1788.5 12443.5
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4902.577    235.974  -20.78   <2e-16 ***
## max_temp      226.599      3.759   60.28   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2705 on 1092 degrees of freedom
## Multiple R-squared:  0.7689, Adjusted R-squared:  0.7687
## F-statistic:  3633 on 1 and 1092 DF,  p-value: < 2.2e-16
```

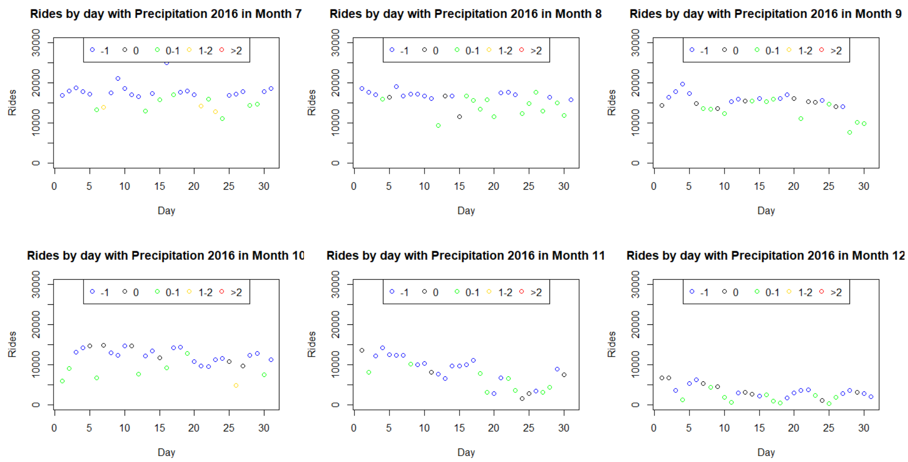
Roughly 77% of variance in daily ride count can be attributed to the day's high temperature, with an increase of 226 rides for every increase of 1 degree Fahrenheit.

### Rides and Precipitation

```
myQuery <- "select day(t.start_time) as dayt, month(t.start_time) as monthT,y
ear(t.start_time) as yeart, w.precipitation,
                   count(trip_id) as count1
            from trips t
            join weather w on date(t.start_time) = w.record_date
            group by dayt,monthT,yeart
            order by  dayt,monthT ,yeart;"
outdata_precip= dbGetQuery(connection, myQuery)

attach(outdata_precip)

## The following objects are masked from outdata_temperature:
##
##     count1, dayt, monthT, yeart

## The following objects are masked from outdata_subscribers:
##
##     count1, monthT, yeart

## The following objects are masked from outdata1:
##
##     count1, monthT, yeart

class(outdata_precip)

## [1] "data.frame"

for(i in 1:12)
{
predict_month <- i
with(outdata_precip, plot(max_temp, count1, main = paste("Rides by day with P
recipitation 2016 in Month", i, sep=" "), xlim = c(1,31), ylim=c(0,30000), xl
```

```
ab = "Day", ylab = "Rides", type = "n"))
with(subset(outdata_precip, monthT==i & yeart==2016 & precipitation==-1), poi
nts(dayt, count1, col = "black"))
with(subset(outdata_precip,monthT==i & yeart==2016 & precipitation==0), point
s(dayt, count1, col = "blue"))
with(subset(outdata_precip,monthT==i & yeart==2016 & precipitation<1 & precip
itation>0), points(dayt, count1, col = "green"))
with(subset(outdata_precip,monthT==i & yeart==2016 & precipitation>1 & precip
itation<2), points(dayt, count1, col = "gold"))
with(subset(outdata_precip,monthT==i & yeart==2016 & precipitation>2), points
(dayt, count1, col = "red"))
legend("top", horiz = TRUE, pch = 1, col= c("blue", "black","green", "gold",
"red"), legend = c("-1","0", "0-1", "1-2", ">2"))
next
}
```






The data set gave continuous precipitation levels except for days where there was "trace" amount of precipitation for our analysis we have given trace a value of 0.0002. As might be expected precipitation seems to explain lower ride counts.

```
#quantifying precipitation's effect
lm_precip <- lm(count1 ~ precipitation,outdata_precip)
summary(lm_precip)

##
## Call:
## lm(formula = count1 ~ precipitation, data = outdata_precip)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -8580  -5163   -521   4848  16408
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     8581.0      181.5  47.288   <2e-16 ***
## precipitation  -1336.7      609.0  -2.195   0.0284 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5614 on 1092 degrees of freedom
## Multiple R-squared:  0.004392,   Adjusted R-squared:  0.00348
## F-statistic: 4.817 on 1 and 1092 DF,  p-value: 0.02839
```

On its own as it is currently scaled, precipitation is a some-what significant factor but it does not account for much of the variation of daily ride count.

## Predicting ride counts

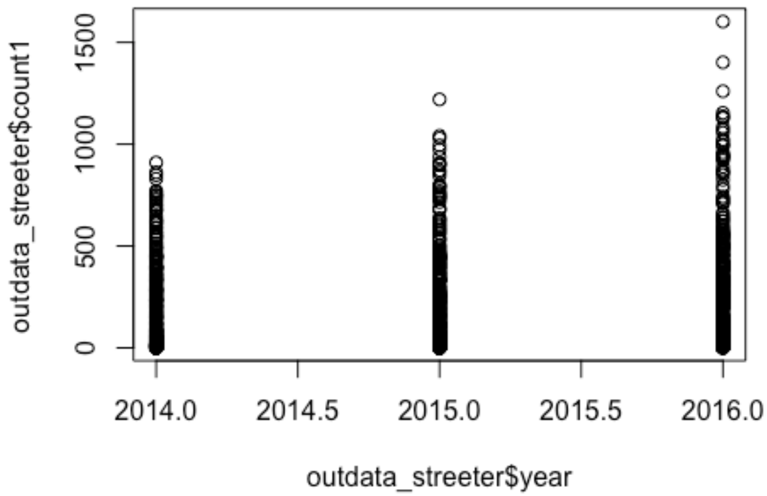
If an accurate prediction can be made for the number of rides originating and arriving at busy stations then perhaps a more efficient method for bike restocking could be implemented.

### Modeling Ride Count on Time, Date, and Weather.

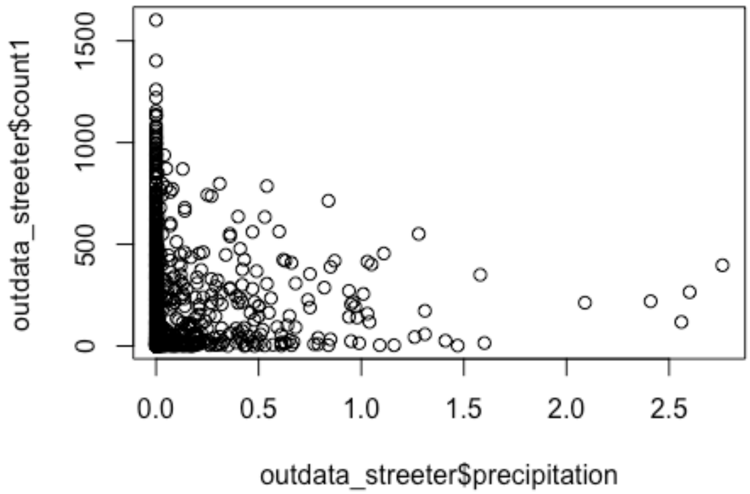
As identified above, the "Streeter & Grand" station is the most popular destination for the busiest Zip code origins. This model will focus on rides from April through September as there appears to be a common trend for those quarters.

```
connection = dbConnect(MySQL(),user="", password=" ", dbname="", host="mysql.
rcc.uchicago.edu");
myQuery <- "select day(t.start_time) as date, dayofweek(t.start_time) as day,
month(t.start_time) as month,year(t.start_time) as year, w.precipitation,w.ma
x_temp as temp,
                   count(trip_id) as count1
            from trips t
            join weather w on date(t.start_time) = w.record_date
            where t.to_station = 35
            group by date,month,year
            order by year,month,date;"
outdata_streeter= dbGetQuery(connection, myQuery)

attach(outdata_streeter)

## The following objects are masked from outdata_precip:
##
##     count1, precipitation

## The following object is masked from outdata_temperature:
##
##     count1

## The following object is masked from outdata_subscribers:
##
##     count1

## The following object is masked from outdata1:
##
##     count1

class(outdata_streeter)

## [1] "data.frame"

# Encoding categorical data
outdata_streeter$date = factor(outdata_streeter$date)
outdata_streeter$day = factor(outdata_streeter$day)
outdata_streeter$month = factor(outdata_streeter$month)
```

```r
plot(outdata_streeter$year, outdata_streeter$count1)
```

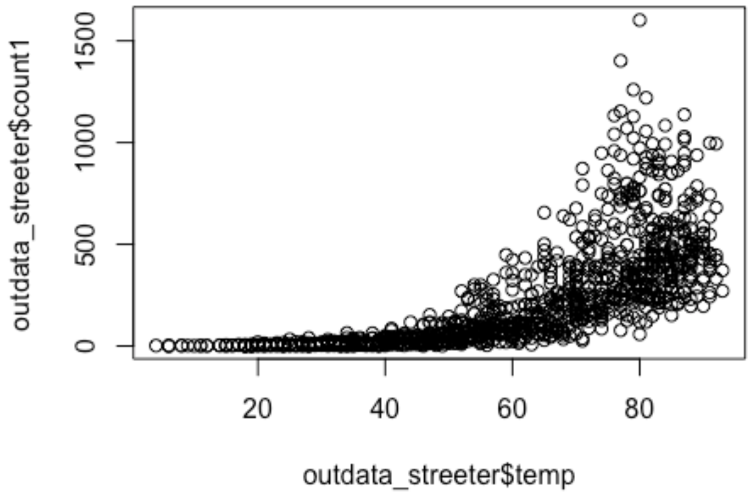



```r
plot(outdata_streeter$precipitation,outdata_streeter$count1)
```




Skewed, most rides occur close to 0.

```
plot(outdata_streeter$temp,outdata_streeter$count1)
```




Exponentially correlated.

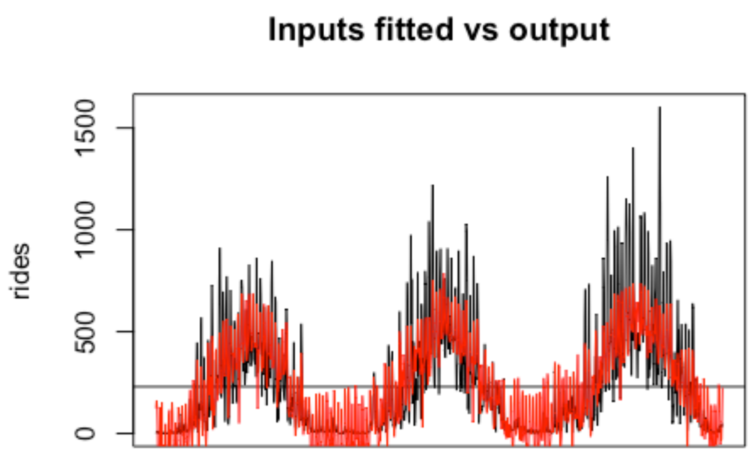
```
# Feature Scaling
outdata_streeter[, 4:6] = scale(outdata_streeter[, 4:6])

#linear model of all inputs
lm_streeter <- lm(count1 ~ .,outdata_streeter)
summary(lm_streeter)

##
## Call:
## lm(formula = count1 ~ ., data = outdata_streeter)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -388.53  -83.53   -6.85   63.11  973.04
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      232.8717    32.8081   7.098 2.37e-12 ***
## date2             12.7649    33.5504   0.380   0.7037
## date3             38.3818    33.0804   1.160   0.2462
## date4             60.2074    32.8627   1.832   0.0672 .
## date5             49.1043    33.5540   1.463   0.1437
```

```
## date6           35.8826    33.3396   1.076   0.2821
## date7           16.1405    33.0938   0.488   0.6259
## date8            1.4818    33.3206   0.044   0.9645
## date9            1.6371    33.3024   0.049   0.9608
## date10          21.1527    33.0824   0.639   0.5227
## date11          22.5179    32.8711   0.685   0.4935
## date12           0.1568    32.8648   0.005   0.9962
## date13          15.1392    32.8636   0.461   0.6451
## date14          -3.5565    33.0879  -0.107   0.9144
## date15          -2.7741    33.0630  -0.084   0.9332
## date16          29.2935    32.8712   0.891   0.3731
## date17          27.0913    32.8676   0.824   0.4100
## date18          35.9270    33.3353   1.078   0.2814
## date19          33.7272    32.8719   1.026   0.3051
## date20          11.5942    33.0795   0.350   0.7260
## date21          10.2182    33.1129   0.309   0.7577
## date22           0.9871    32.8519   0.030   0.9760
## date23          18.1206    33.0825   0.548   0.5840
## date24          27.8145    32.8721   0.846   0.3977
## date25          43.3329    32.8776   1.318   0.1878
## date26          18.6157    33.0801   0.563   0.5737
## date27          16.7939    33.0953   0.507   0.6120
## date28          13.5135    33.0934   0.408   0.6831
## date29          -2.6742    33.5363  -0.080   0.9365
## date30          23.2182    33.5631   0.692   0.4892
## date31           1.9397    38.1458   0.051   0.9595
## day2          -138.0189    15.6400  -8.825  < 2e-16 ***
## day3          -194.1131    15.6328 -12.417  < 2e-16 ***
## day4          -202.5934    15.6630 -12.935  < 2e-16 ***
## day5          -189.2187    15.6958 -12.055  < 2e-16 ***
## day6          -140.1403    15.5543  -9.010  < 2e-16 ***
## day7            36.3793    15.5369   2.341   0.0194 *
## month2          32.5781    22.0498   1.477   0.1399
## month3          -5.1262    22.1881  -0.231   0.8173
## month4          10.0977    24.5422   0.411   0.6808
## month5         121.6747    27.2036   4.473 8.59e-06 ***
## month6         187.2979    30.2692   6.188 8.84e-10 ***
## month7         300.1609    30.6454   9.795  < 2e-16 ***
## month8         245.1174    30.7512   7.971 4.21e-15 ***
## month9         182.8633    29.0915   6.286 4.84e-10 ***
## month10         44.4067    25.3486   1.752   0.0801 .
## month11         -1.7845    22.9560  -0.078   0.9381
## month12          4.3285    21.2663   0.204   0.8388
## year            29.6187     4.2047   7.044 3.44e-12 ***
## precipitation  -39.5553     4.2522  -9.302  < 2e-16 ***
## temp           107.7006     9.1270  11.800  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 135.1 on 1015 degrees of freedom
```

```
## Multiple R-squared:  0.7394, Adjusted R-squared:  0.7266
## F-statistic: 57.61 on 50 and 1015 DF,  p-value: < 2.2e-16
```

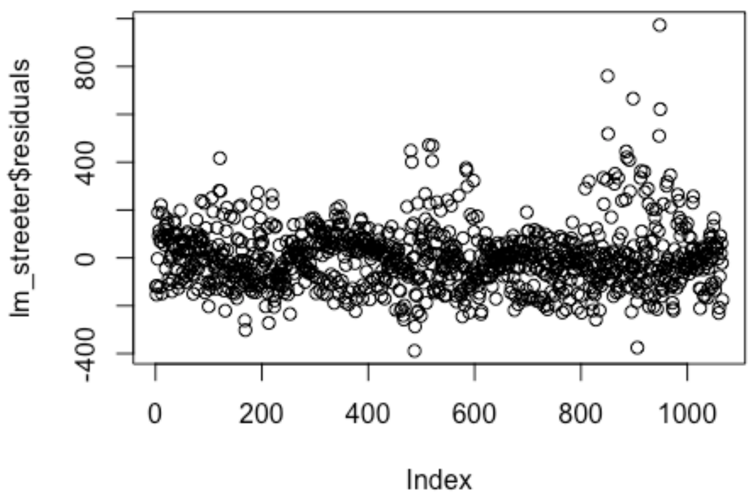
0.7266 Adjusted R-squared, date not relevant

```
#plot model and actual rides
matplot(outdata_streeter[,7],type="l",xaxt="n", ylab = "rides", main="Inputs
fitted vs output")
lines(lm_streeter$fitted.values, abline(h=mean(outdata_streeter$count1)), col
=c("red") )
```

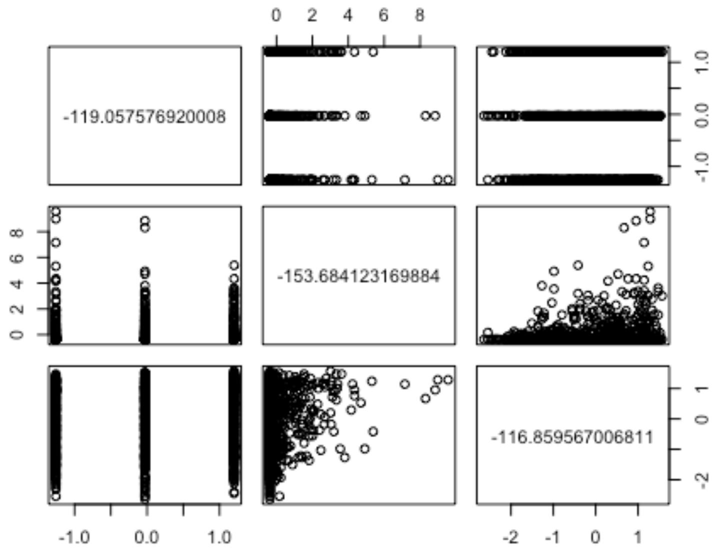



Model seems to do a good job capturing the ebb and flow of the ride pattern but fails to predict the lowest lows and highest spikes.

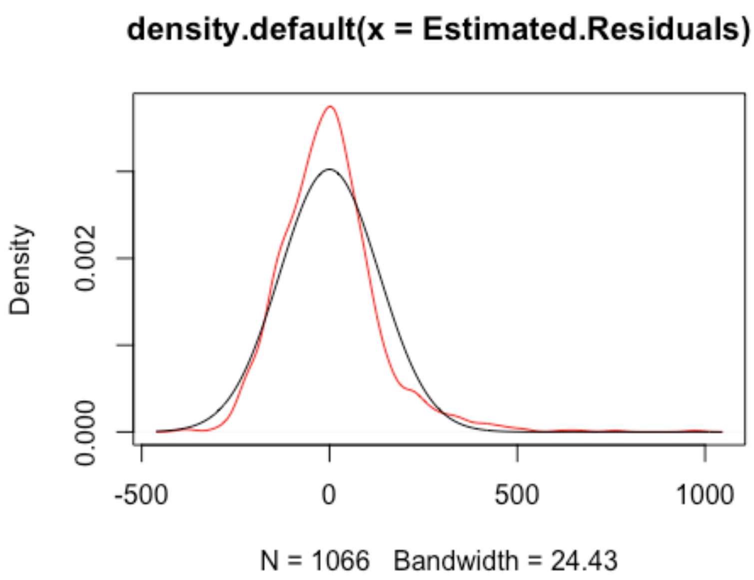
```
plot(lm_streeter$residuals)
```




```
#Observe the residuals, plot them against the input.
Estimated.Residuals <- lm_streeter$residuals
plot(outdata_streeter[,4:6], Estimated.Residuals)
```




```r
#and their probability density in comparison with the normal density

Probability.Density.Residuals <- density(Estimated.Residuals)
plot(Probability.Density.Residuals, col='red')
lines(Probability.Density.Residuals$x,  dnorm(Probability.Density.Residuals$x
,
    mean = mean(Estimated.Residuals), sd = sd(Estimated.Residuals)))
```

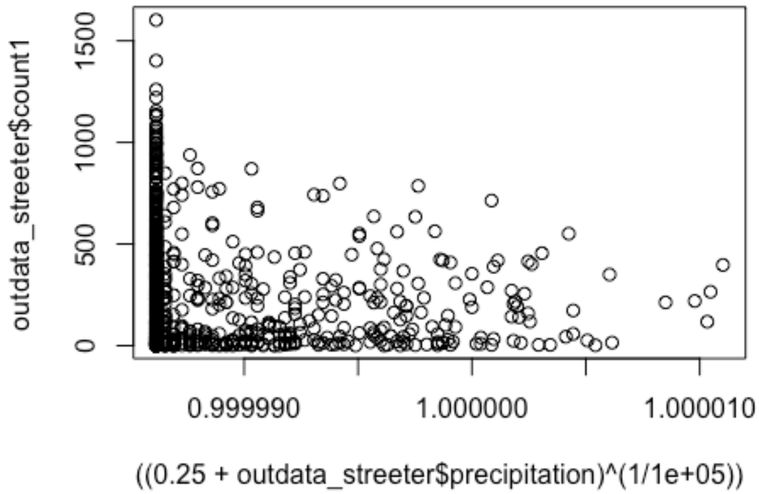



Next round will drop date and transform the skewed continuous variables.

## Linear Model Second Round

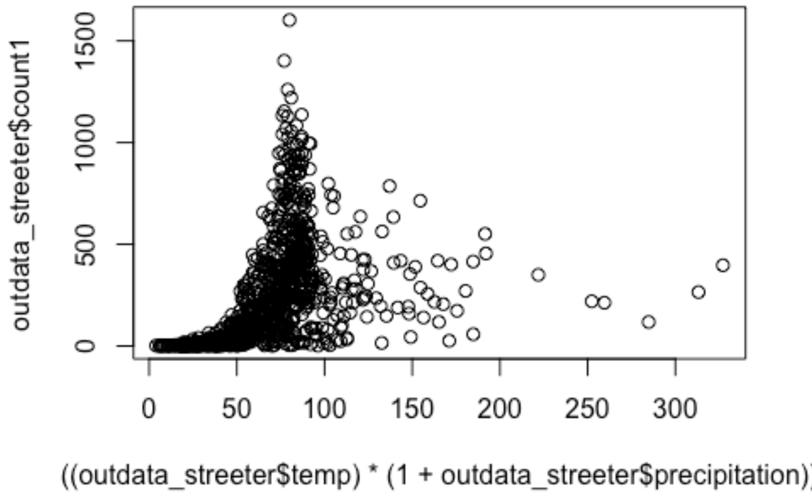
```
connection = dbConnect(MySQL(),user="", password=" ", dbname="", host="mysql.
rcc.uchicago.edu");
myQuery <- "select day(t.start_time) as date, dayofweek(t.start_time) as day,
month(t.start_time) as month,year(t.start_time) as year, w.precipitation,w.ma
x_temp as temp,
                   count(trip_id) as count1
            from trips t
            join weather w on date(t.start_time) = w.record_date
            where t.to_station = 35
            group by date,month,year
            order by year,month,date;"
outdata_streeter= dbGetQuery(connection, myQuery)

attach(outdata_streeter)

## The following objects are masked from outdata_streeter (pos = 3):
##
##     count1, date, day, month, precipitation, temp, year
```

```
## The following objects are masked from outdata_precip:
##
##     count1, precipitation

## The following object is masked from outdata_temperature:
##
##     count1

## The following object is masked from outdata_subscribers:
##
##     count1

## The following object is masked from outdata1:
##
##     count1

class(outdata_streeter)

## [1] "data.frame"

# Encoding categorical data
outdata_streeter$day = factor(outdata_streeter$day)
outdata_streeter$month = factor(outdata_streeter$month)

#best linear approximation, not linear enough, will not transform
plot(((0.25+outdata_streeter$precipitation)^(1/100000)),outdata_streeter$coun
t1)
```




There is probably interaction between precipitation and temp, perhaps the two can be combined.

```
#best approximation, still not linear enough, will not include
plot(((outdata_streeter$temp)*(1+outdata_streeter$precipitation)),outdata_str
eeter$count1)
```




```
#possible polynomial correlation
outdata_streeter$precipitation2 = outdata_streeter$precipitation^2
outdata_streeter$precipitation3 = outdata_streeter$precipitation^3
outdata_streeter$precipitation4 = outdata_streeter$precipitation^4
outdata_streeter$temp2 = outdata_streeter$temp^2
outdata_streeter$temp3 = outdata_streeter$temp^3
outdata_streeter$temp4 = outdata_streeter$temp^4

# Feature Scaling
outdata_streeter[, 4:6] <- scale(outdata_streeter[, 4:6])
outdata_streeter[,8:13] <- scale(outdata_streeter[,8:13])

#linear model
lm_streeter <- lm(count1 ~ day+month+year+temp+precipitation,outdata_streeter
)
summary(lm_streeter)

##
## Call:
## lm(formula = count1 ~ day + month + year + temp + precipitation,
##     data = outdata_streeter)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -383.79  -82.33   -9.12   61.17 1013.85
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    252.249     22.092  11.418  < 2e-16 ***
## day2          -138.654     15.513  -8.938  < 2e-16 ***
## day3          -194.710     15.499 -12.563  < 2e-16 ***
## day4          -202.829     15.532 -13.059  < 2e-16 ***
## day5          -188.988     15.567 -12.140  < 2e-16 ***
## day6          -140.015     15.431  -9.074  < 2e-16 ***
## day7            36.309     15.413   2.356   0.0187 *
## month2          33.765     21.803   1.549   0.1218
## month3          -5.887     22.007  -0.267   0.7891
## month4          10.076     24.285   0.415   0.6783
## month5         121.071     26.948   4.493 7.82e-06 ***
## month6         187.438     29.940   6.261 5.59e-10 ***
## month7         299.416     30.344   9.868  < 2e-16 ***
## month8         244.415     30.448   8.027 2.66e-15 ***
## month9         182.719     28.772   6.351 3.19e-10 ***
## month10         43.780     25.118   1.743   0.0816 .
## month11         -1.707     22.717  -0.075   0.9401
## month12          3.911     21.092   0.185   0.8529
## year            29.528      4.172   7.078 2.68e-12 ***
## temp           107.869      9.018  11.962  < 2e-16 ***
## precipitation  -39.843      4.192  -9.503  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 134.2 on 1045 degrees of freedom
## Multiple R-squared:  0.7354, Adjusted R-squared:  0.7304
## F-statistic: 145.3 on 20 and 1045 DF,  p-value: < 2.2e-16
```

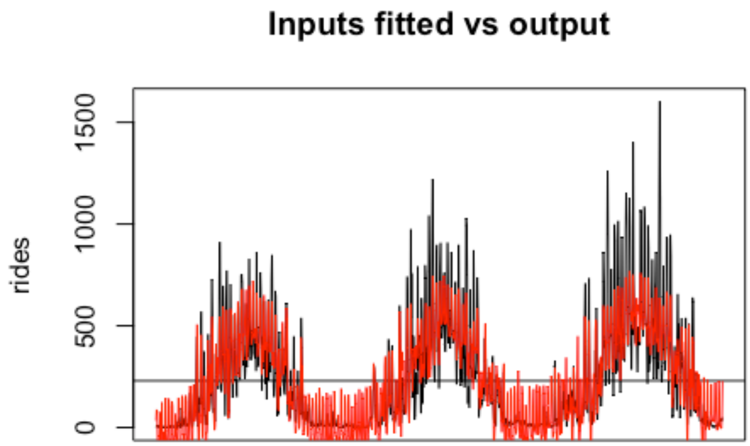
```
#polnomial model
poly_reg = lm(formula = count1 ~ day+month+year+precipitation+temp+precipitat
ion2+temp2+precipitation3+temp3+precipitation4+temp4,
              data = outdata_streeter)
summary(poly_reg)
```

```
##
## Call:
## lm(formula = count1 ~ day + month + year + precipitation + temp +
##     precipitation2 + temp2 + precipitation3 + temp3 + precipitation4 +
##     temp4, data = outdata_streeter)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -353.59  -80.46   -2.70   60.03  972.27
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)      261.558     20.735  12.614  < 2e-16 ***
## day2            -150.637     14.396 -10.464  < 2e-16 ***
## day3            -196.194     14.320 -13.701  < 2e-16 ***
## day4            -200.573     14.367 -13.961  < 2e-16 ***
## day5            -190.652     14.395 -13.244  < 2e-16 ***
## day6            -140.320     14.279  -9.827  < 2e-16 ***
## day7              42.428     14.253   2.977  0.00298 **
## month2             8.669     20.245   0.428  0.66859
## month3            39.839     20.888   1.907  0.05676 .
## month4            46.684     23.432   1.992  0.04660 *
## month5           109.071     25.653   4.252 2.31e-05 ***
## month6           131.391     28.288   4.645 3.84e-06 ***
## month7           238.527     28.657   8.323 2.67e-16 ***
## month8           177.031     28.794   6.148 1.12e-09 ***
## month9           135.910     27.302   4.978 7.52e-07 ***
## month10           61.701     24.456   2.523  0.01179 *
## month11           36.065     21.659   1.665  0.09619 .
## month12           34.399     19.669   1.749  0.08060 .
## year              31.201      3.868   8.067 1.97e-15 ***
## precipitation   -142.855     24.209  -5.901 4.89e-09 ***
## temp             488.022    157.957   3.090  0.00206 **
## precipitation2   318.430    110.433   2.883  0.00401 **
## temp2          -2774.537    619.600  -4.478 8.37e-06 ***
## precipitation3  -403.564    201.626  -2.002  0.04559 *
## temp3           4611.677    816.243   5.650 2.07e-08 ***
## precipitation4   186.695    114.984   1.624  0.10475
## temp4          -2200.806    354.161  -6.214 7.46e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 123.9 on 1039 degrees of freedom
## Multiple R-squared:  0.7756, Adjusted R-squared:   0.77
## F-statistic: 138.1 on 26 and 1039 DF,  p-value: < 2.2e-16
```

0.77 Adjusted R-squared, not all months are relevant

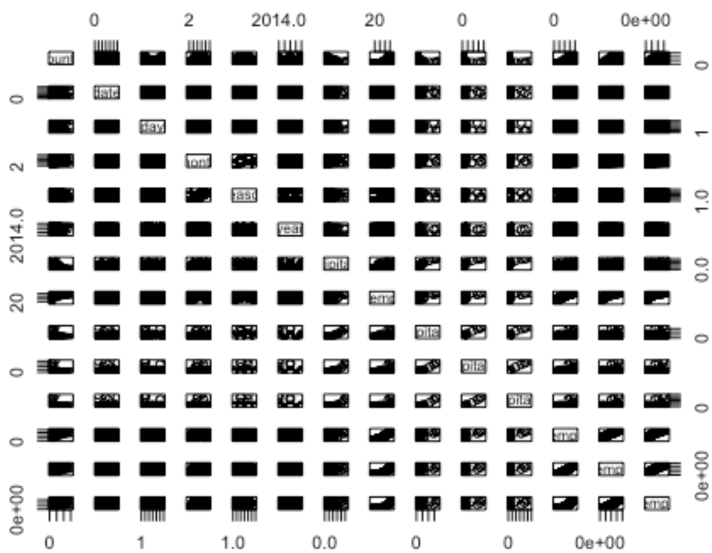
```
#plot model and actual rides
matplot(outdata_streeter[,7],type="l",xaxt="n", ylab = "rides", main="Inputs
fitted vs output")
lines(poly_reg$fitted.values, abline(h=mean(outdata_streeter$count1)), col=c(
"red") )
```




### Third round now polynomial, add Season

```
connection = dbConnect(MySQL(),user="", password=" ", dbname="", host="mysql.
rcc.uchicago.edu");
myQuery <- "select day(t.start_time) as date, dayofweek(t.start_time) as day,
month(t.start_time) as month,
CASE when month(t.start_time) between 2 and 4 then 1
when month(t.start_time) between 5 and 7 then 2
when month(t.start_time) between 8 and 10  then 3
when month(t.start_time) between 11 and 12 then 4
when month(t.start_time) = 1 then 4 END as season, year(t.start_time) as year
, w.precipitation,w.max_temp as temp,
                   count(trip_id) as count1
            from trips t
            join weather w on date(t.start_time) = w.record_date
            where t.to_station = 35
            group by date,month,year
            order by year,month,date;"
outdata_streeter= dbGetQuery(connection, myQuery)

attach(outdata_streeter)
```

```
## The following objects are masked from outdata_streeter (pos = 3):
##
##     count1, date, day, month, precipitation, temp, year

## The following objects are masked from outdata_streeter (pos = 4):
##
##     count1, date, day, month, precipitation, temp, year

## The following objects are masked from outdata_precip:
##
##     count1, precipitation

## The following object is masked from outdata_temperature:
##
##     count1

## The following object is masked from outdata_subscribers:
##
##     count1

## The following object is masked from outdata1:
##
##     count1

class(outdata_streeter)

## [1] "data.frame"

# Encoding categorical data
outdata_streeter$date = factor(outdata_streeter$date)
outdata_streeter$day = factor(outdata_streeter$day)
outdata_streeter$month = factor(outdata_streeter$month)
outdata_streeter$season = factor(outdata_streeter$season)

#possible polynomial correlation
outdata_streeter$precipitation2 = outdata_streeter$precipitation^2
outdata_streeter$precipitation3 = outdata_streeter$precipitation^3
outdata_streeter$precipitation4 = outdata_streeter$precipitation^4
outdata_streeter$temp2 = outdata_streeter$temp^2
outdata_streeter$temp3 = outdata_streeter$temp^3
outdata_streeter$temp4 = outdata_streeter$temp^4

pairs(count1 ~., data=outdata_streeter)
```

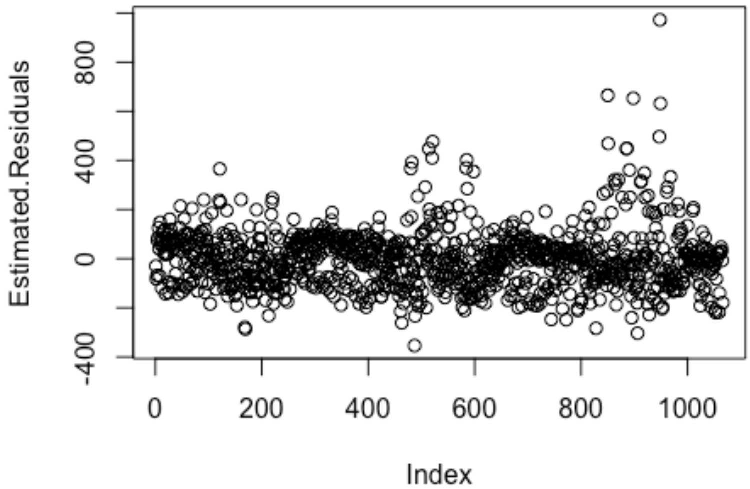


```r
# Feature Scaling
outdata_streeter[, 5:7] <- scale(outdata_streeter[, 5:7])
outdata_streeter[,9:14] <- scale(outdata_streeter[,9:14])

#polnomial model
poly_reg = lm(formula = count1 ~ day+month+season+year+precipitation+temp+pre
cipitation2+temp2+precipitation3+temp3+precipitation4+temp4,
              data = outdata_streeter)
summary(poly_reg)

##
## Call:
## lm(formula = count1 ~ day + month + season + year + precipitation +
##     temp + precipitation2 + temp2 + precipitation3 + temp3 +
##     precipitation4 + temp4, data = outdata_streeter)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -353.59  -80.46   -2.70   60.03  972.27
##
## Coefficients: (3 not defined because of singularities)
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      261.558     20.735  12.614  < 2e-16 ***
```

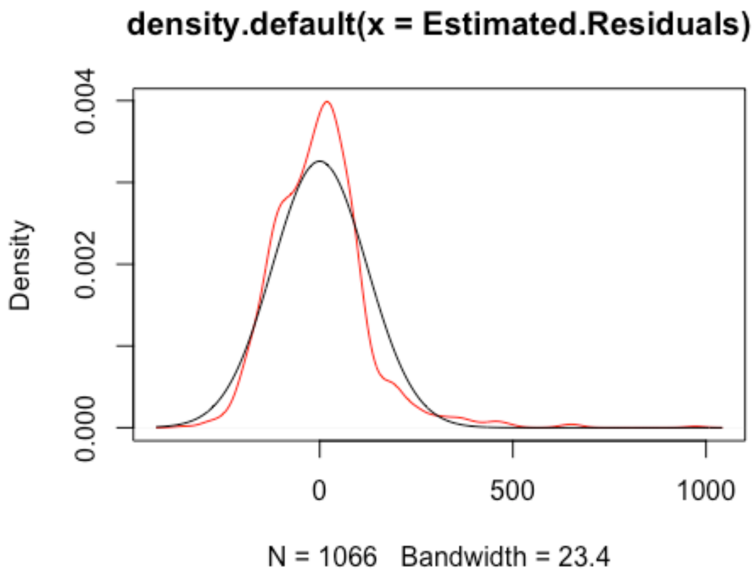
```
## day2             -150.637     14.396 -10.464  < 2e-16 ***
## day3             -196.194     14.320 -13.701  < 2e-16 ***
## day4             -200.573     14.367 -13.961  < 2e-16 ***
## day5             -190.652     14.395 -13.244  < 2e-16 ***
## day6             -140.320     14.279  -9.827  < 2e-16 ***
## day7               42.428     14.253   2.977  0.00298 **
## month2              8.669     20.245   0.428  0.66859
## month3             39.839     20.888   1.907  0.05676 .
## month4             46.684     23.432   1.992  0.04660 *
## month5            109.071     25.653   4.252 2.31e-05 ***
## month6            131.391     28.288   4.645 3.84e-06 ***
## month7            238.527     28.657   8.323 2.67e-16 ***
## month8            177.031     28.794   6.148 1.12e-09 ***
## month9            135.910     27.302   4.978 7.52e-07 ***
## month10            61.701     24.456   2.523  0.01179 *
## month11            36.065     21.659   1.665  0.09619 .
## month12            34.399     19.669   1.749  0.08060 .
## season2                NA         NA      NA       NA
## season3                NA         NA      NA       NA
## season4                NA         NA      NA       NA
## year               31.201      3.868   8.067 1.97e-15 ***
## precipitation    -142.855     24.209  -5.901 4.89e-09 ***
## temp              488.022    157.957   3.090  0.00206 **
## precipitation2    318.430    110.433   2.883  0.00401 **
## temp2           -2774.537    619.600  -4.478 8.37e-06 ***
## precipitation3   -403.564    201.626  -2.002  0.04559 *
## temp3            4611.677    816.243   5.650 2.07e-08 ***
## precipitation4    186.695    114.984   1.624  0.10475
## temp4           -2200.806    354.161  -6.214 7.46e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 123.9 on 1039 degrees of freedom
## Multiple R-squared:  0.7756, Adjusted R-squared:   0.77
## F-statistic: 138.1 on 26 and 1039 DF,  p-value: < 2.2e-16
```

No difference in adjusted R-squared, season must interact too much with month but on their own month provides a better adjusted R-squared.

```
Estimated.Residuals <- poly_reg$residuals
plot(Estimated.Residuals)
```




```
#and their probability density in comparison with the normal density

Probability.Density.Residuals <- density(Estimated.Residuals)
plot(Probability.Density.Residuals, col='red')
lines(Probability.Density.Residuals$x,  dnorm(Probability.Density.Residuals$x
,
    mean = mean(Estimated.Residuals), sd = sd(Estimated.Residuals)))
```




Residuals show there are yet more patterns to be discovered.

## Machine Learning Methods

### Random Forests

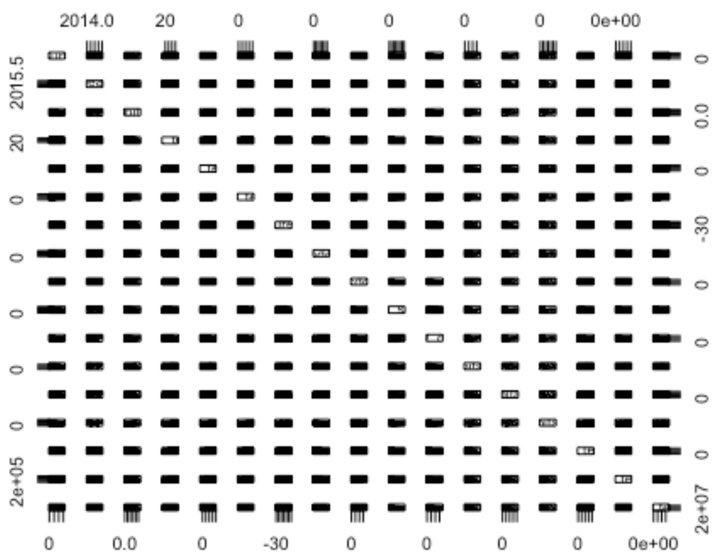
There were more variables gathered in the weather data. Perhaps some machine learning algorithms can find the patterns.

```
connection = dbConnect(MySQL(),user="", password=" ", dbname="", host="mysql.
rcc.uchicago.edu");
myQuery <- "select * from weather limit 10;"
weather= dbGetQuery(connection, myQuery)
```

```
connection = dbConnect(MySQL(),user="", password=" ", dbname="", host="mysql.
rcc.uchicago.edu");
myQuery <- "select count(trip_id) as count1, day(t.start_time) as date, dayof
week(t.start_time) as day, month(t.start_time) as month,
CASE when month(t.start_time) between 2 and 4 then 1
when month(t.start_time) between 5 and 7 then 2
when month(t.start_time) between 8 and 10  then 3
when month(t.start_time) between 11 and 12 then 4
when month(t.start_time) = 1 then 4 END as season, year(t.start_time) as year
```

```
, w.precipitation,w.max_temp,w.min_temp,w.avg_temp,w.departure_temp,w.hdd,w.c
dd,w.new_snow,w.snow_depth

            from trips t
            join weather w on date(t.start_time) = w.record_date
            where t.to_station = 35
            group by date,month,year
            order by year,month,date;"
outdata_streeter_all= dbGetQuery(connection, myQuery)

attach(outdata_streeter_all)

## The following objects are masked from outdata_streeter (pos = 3):
##
##     count1, date, day, month, precipitation, season, year

## The following objects are masked from outdata_streeter (pos = 4):
##
##     count1, date, day, month, precipitation, year

## The following objects are masked from outdata_streeter (pos = 5):
##
##     count1, date, day, month, precipitation, year

## The following objects are masked from outdata_precip:
##
##     count1, precipitation

## The following objects are masked from outdata_temperature:
##
##     count1, max_temp

## The following object is masked from outdata_subscribers:
##
##     count1

## The following object is masked from outdata1:
##
##     count1

class(outdata_streeter_all)

## [1] "data.frame"

#possible polynomial correlation
outdata_streeter_all$precipitation2 = outdata_streeter_all$precipitation^2
outdata_streeter_all$precipitation3 = outdata_streeter_all$precipitation^3
outdata_streeter_all$precipitation4 = outdata_streeter_all$precipitation^4
outdata_streeter_all$max_temp2 = outdata_streeter_all$max_temp^2
outdata_streeter_all$max_temp3 = outdata_streeter_all$max_temp^3
outdata_streeter_all$max_temp4 = outdata_streeter_all$max_temp^4
```

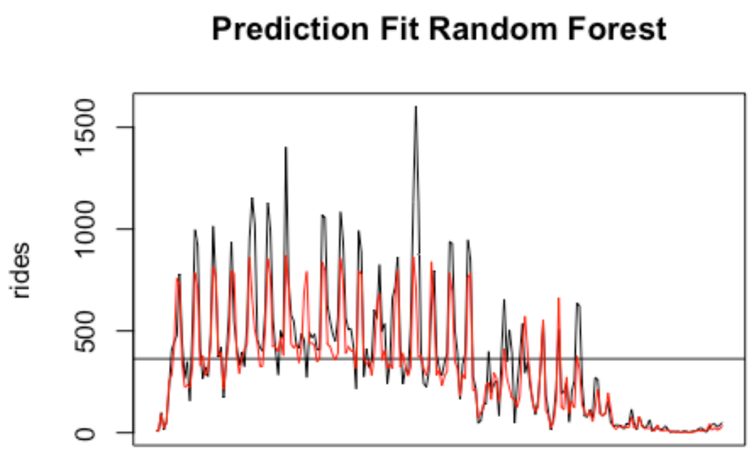
```
# Encoding categorical data
outdata_streeter_all$date = factor(outdata_streeter_all$date)
outdata_streeter_all$day = factor(outdata_streeter_all$day)
outdata_streeter_all$month = factor(outdata_streeter_all$month)
outdata_streeter_all$season = factor(outdata_streeter_all$season)
facs <- sapply(outdata_streeter_all, is.factor)
outdata_streeter_factors <- outdata_streeter_all[,facs]
nums <- sapply(outdata_streeter_all, is.numeric)
outdata_streeter_numeric <- outdata_streeter_all[,nums]
pairs(count1 ~ ., data=outdata_streeter_numeric)
```




```
#extract to CSV to perform modeling in Python
write.csv(outdata_streeter_all, file="streeter_all.csv")
```

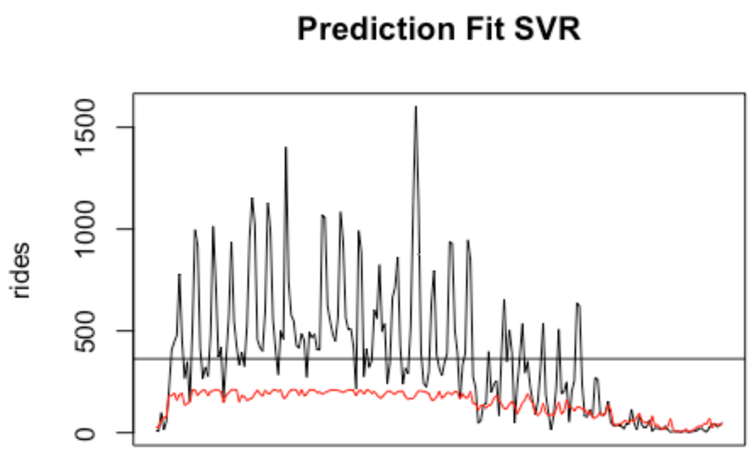
The second half of 2016 was chosen as the test set and a number of models were evaluated. Random forest regression performed the best.

```
rfrModelData<-read.csv(file="streeter_rfr.csv")
#plot model and actual rides
matplot(rfrModelData$count1,type="l",xaxt="n", ylab = "rides", main="Predicti
on Fit Random Forest")
lines(rfrModelData$pred, abline(h=mean(rfrModelData$count1)), col=c("red") )
```

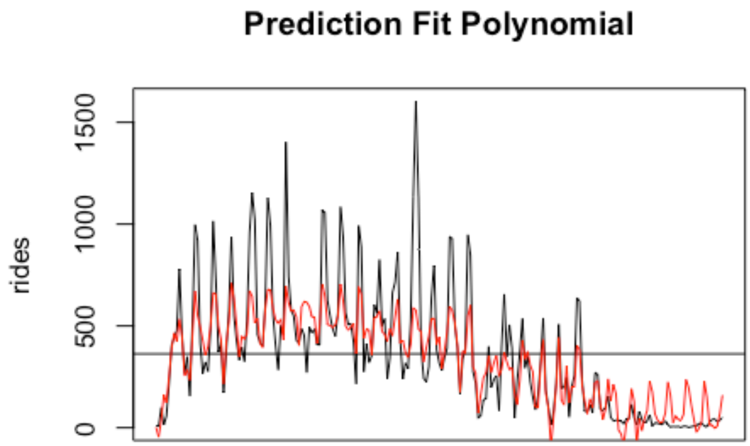



In contrast SVR and Polynomial models did not predict the second half of 2016 very well.

```
svrModelData<-read.csv(file="streeter_svr.csv")
#plot model and actual rides
matplot(svrModelData$count1,type="l",xaxt="n", ylab = "rides", main="Predicti
on Fit SVR")
lines(svrModelData$pred, abline(h=mean(svrModelData$count1)), col=c("red") )
```




```
polyModelData<-read.csv(file="streeter_poly_reg.csv")
#plot model and actual rides
matplot(polyModelData$count1,type="l",xaxt="n", ylab = "rides", main="Predict
ion Fit Polynomial")
lines(polyModelData$pred, abline(h=mean(polyModelData$count1)), col=c("red")
)
```


Prediction Fit Polynomial
rides
1500
1000
500
0