Predicting Rain in Seattle

Abstract

This project is of interest because my work schedule and the amount of money I earn are directly related to the amount of construction and rainfall in the Seattle area throughout the year. At this time, we as a company are unable to predict our schedules any better than a current weather forecast. This method of prediciting rain uses a logistic regression to predict if it will rain based on factors including date, temperature, humidity, and more. Using this dataset, the method presented here successfully predicts rain between 66% and 82% of the time, depending on the model. The data consists of 1000 days of weather data collected from the Dark Sky API.

Introduction

Seattle receives about 38 inches of rainfall per year. I work for a water treatment company in the Seattle area where I treat rainwater run-off from construction sites. Washington has a reputation for receiving a lot of rainfall and it is even home to temperate rain forests. The state is essentially divided into two major biomes, separating the Western and Eastern sides. The divide in weather is a direct result of the Cascades mountains range which divides the state vertically. The majority of rainfall accumulates on the Western side of this mountain range and can reach up to 200 inches of rain per year in some places. Water trickles down the mountains and runs off into the Puget Sound which feeds the Pacific Ocean–home to many important fish and marine mammals.

Human interaction has a direct effect on the ecology of Washington State and interrupting the natural water cycles negatively affects the unique flora and fauna in Washington State. The Washington Department of Ecology has mandated that construction sites should analyze and treat the rain water that falls onto the disturbed areas before being discharged back into the environment. With construction sites popping up everywhere due to the tech boom, it is important to have systems in place to protect the environment.

- Primary objective: Predicting Rain near Seattle
  - Predictions: Rain or No Rain
  - Successes: if the actual recorded rainfall was > 0.01 inches and prediction was "Rain"

Methods (Code and Documentation)

This dataset also has the potential to be used further with other classification algorithms including K-Nearest Neighbors, Neural Networks, and Support Vector Machines.

Data source

This dataset consists of 1000 days of weather data collected from the Dark Sky API. 80% of the data was split into training data and the other 20% was used to test and assess the model. The code and documentation of the data collection is included as a supplement to this document but has had the API key removed for security.

The weather data location was based on the GPS coordinates of Seattle, WA. Latitude: 47.608013 Longitude: -122.335167

Data Prep

```
# Load packages
require("lubridate")
```

```
## Loading required package: lubridate
```

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

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

```
require("ggplot2")
```

```
## Loading required package: ggplot2
```

```
require("reshape2")
```

```
## Loading required package: reshape2
```

```
require("psych")
```

```
## Loading required package: psych
```

```
## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called 'psych'
```

```
# Load data set
data <- read.csv("10kdays.csv")

# Subset the more important columns
weather <- subset(data, select = c("moonPhase", "precipIntensity", "precipProbability", "temperatureMin

# Convert datetimes to dates
date <- unique(as_datetime(data$time))

# Remove duplicate dates and bind to the date column
weather <- cbind(date = as_date(date), unique(weather))

#Change NA to zeros
weather[is.na(weather)] <- 0

# Reindex rownames
rownames(weather) <- NULL

head(weather)
```

```
##         date moonPhase precipIntensity precipProbability temperatureMin
## 1 2014-04-08      0.29          0.0119              0.81          48.13
## 2 2014-04-07      0.26          0.0000              0.00          52.63
## 3 2014-04-06      0.23          0.0000              0.00          48.01
## 4 2014-04-05      0.20          0.0037              0.63          47.01
## 5 2014-04-04      0.17          0.0000              0.00          45.05
## 6 2014-04-03      0.14          0.0045              0.70          46.61
##   temperatureMax dewPoint humidity windSpeed windBearing cloudCover
## 1          59.68    49.18     0.84      5.84         177       0.00
## 2          69.61    48.12     0.71      2.47         166       0.08
## 3          56.52    45.11     0.80      1.23         213       0.00
## 4          53.87    46.85     0.88      5.72         174       0.00
## 5          55.71    41.83     0.76      7.35         182       0.00
## 6          55.11    42.52     0.77      6.55         163       0.09
##   pressure
## 1  1020.61
## 2  1023.08
## 3  1024.14
## 4  1016.35
## 5  1014.25
## 6  1014.30
```

Exploratory Data Analysis

```
names(weather)
```

```
##  [1] "date"              "moonPhase"         "precipIntensity"
##  [4] "precipProbability" "temperatureMin"    "temperatureMax"
##  [7] "dewPoint"          "humidity"          "windSpeed"
## [10] "windBearing"       "cloudCover"        "pressure"
```

```
str(weather)
```

```
## 'data.frame':    1000 obs. of  12 variables:
##  $ date             : Date, format: "2014-04-08" "2014-04-07" ...
##  $ moonPhase        : num  0.29 0.26 0.23 0.2 0.17 0.14 0.11 0.07 0.04 0.01 ...
##  $ precipIntensity  : num  0.0119 0 0 0.0037 0 0.0045 0 0 0 0.0005 ...
##  $ precipProbability: num  0.81 0 0 0.63 0 0.7 0 0 0 0.3 ...
##  $ temperatureMin   : num  48.1 52.6 48 47 45 ...
##  $ temperatureMax   : num  59.7 69.6 56.5 53.9 55.7 ...
##  $ dewPoint         : num  49.2 48.1 45.1 46.9 41.8 ...
##  $ humidity         : num  0.84 0.71 0.8 0.88 0.76 0.77 0.71 0.71 0.75 0.71 ...
##  $ windSpeed        : num  5.84 2.47 1.23 5.72 7.35 6.55 5.48 2.57 4.51 8.34 ...
##  $ windBearing      : int  177 166 213 174 182 163 180 327 344 182 ...
##  $ cloudCover       : num  0 0.08 0 0 0 0.09 0 0.05 0.06 0.24 ...
##  $ pressure         : num  1021 1023 1024 1016 1014 ...
```

```
summary(weather)
```

```
##       date              moonPhase      precipIntensity
##  Min.   :2011-07-14   Min.   :0.0100   Min.   :0.000000
##  1st Qu.:2012-03-19   1st Qu.:0.2500   1st Qu.:0.000000
##  Median :2012-11-24   Median :0.5100   Median :0.000000
##  Mean   :2012-11-24   Mean   :0.5031   Mean   :0.003638
##  3rd Qu.:2013-08-01   3rd Qu.:0.7500   3rd Qu.:0.003500
##  Max.   :2014-04-08   Max.   :1.0000   Max.   :0.077700
##  precipProbability temperatureMin  temperatureMax     dewPoint
##  Min.   :0.0000    Min.   :20.80   Min.   :30.01   Min.   : 1.78
##  1st Qu.:0.0000    1st Qu.:41.01   1st Qu.:48.98   1st Qu.:38.49
##  Median :0.0000    Median :47.11   Median :55.92   Median :44.31
##  Mean   :0.2969    Mean   :47.43   Mean   :57.38   Mean   :43.79
##  3rd Qu.:0.6600    3rd Qu.:54.75   3rd Qu.:66.05   3rd Qu.:49.68
##  Max.   :0.8800    Max.   :65.45   Max.   :85.87   Max.   :62.37
##     humidity        windSpeed        windBearing       cloudCover
##  Min.   :0.3400   Min.   : 0.020   Min.   :  0.0   Min.   :0.0000
##  1st Qu.:0.6900   1st Qu.: 2.220   1st Qu.:166.8   1st Qu.:0.0000
##  Median :0.7700   Median : 3.545   Median :197.0   Median :0.4600
##  Mean   :0.7562   Mean   : 4.050   Mean   :226.1   Mean   :0.4446
##  3rd Qu.:0.8300   3rd Qu.: 5.442   3rd Qu.:323.0   3rd Qu.:0.8300
##  Max.   :0.9600   Max.   :13.080   Max.   :359.0   Max.   :1.0000
##     pressure
##  Min.   : 996.4
##  1st Qu.:1014.1
##  Median :1018.2
##  Mean   :1018.1
##  3rd Qu.:1022.4
##  Max.   :1038.5
```

```r
# Average high
mean(weather$temperatureMax)
```

```
## [1] 57.3804
```

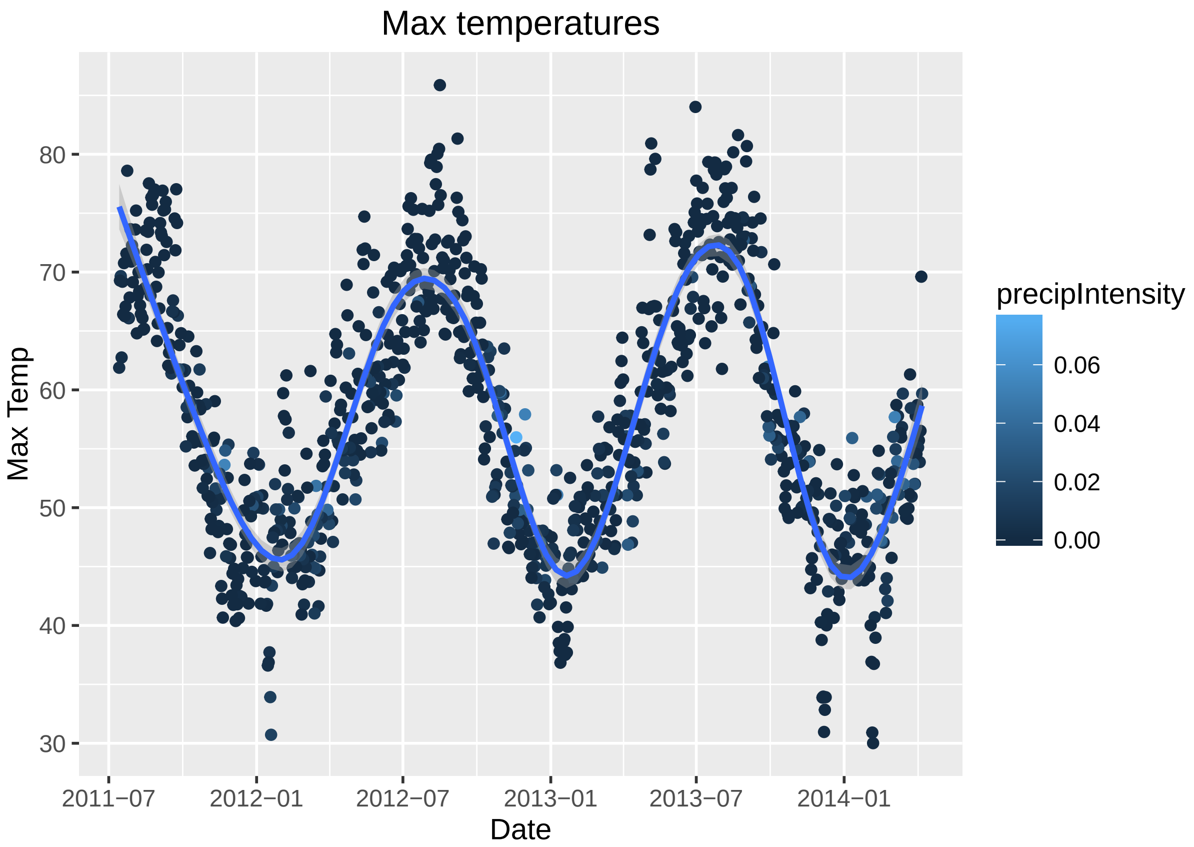
```r
# Average low
mean(weather$temperatureMin)
```

```
## [1] 47.43365
```

```r
# Average rain probability
mean(weather$precipProbability)
```

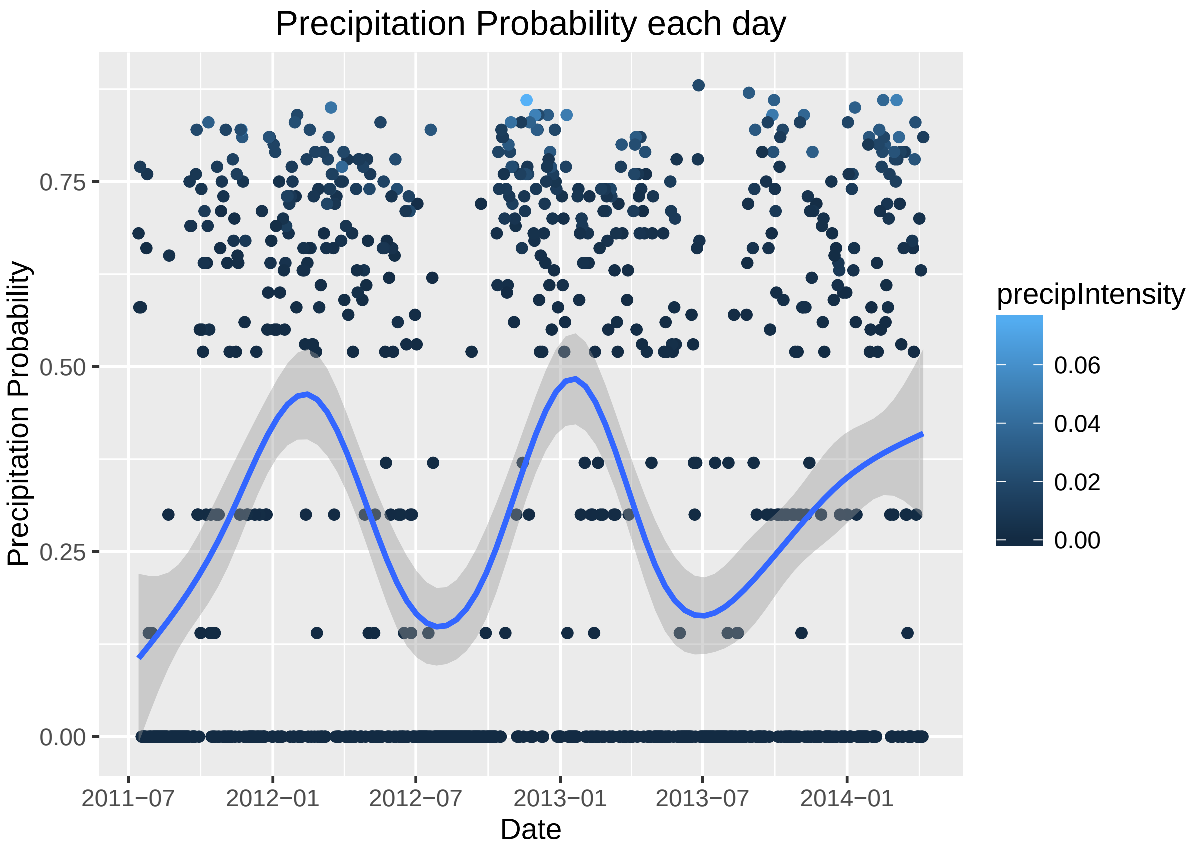
```
## [1] 0.29686
```

```r
ggplot(data = weather) + aes(x = date, y = temperatureMax,  color = precipIntensity) + geom_point() + g
```



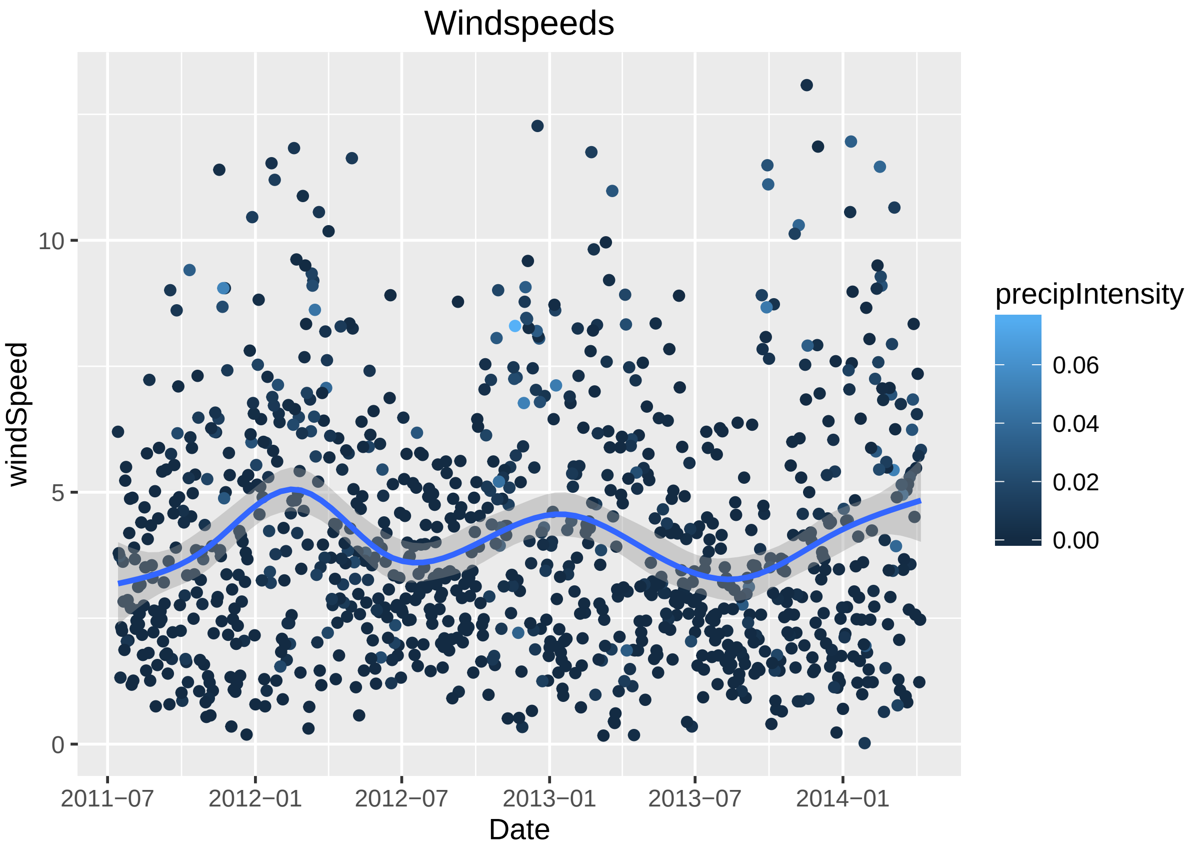
There is a little bit of a pattern to precipitation here. It seems like precipitation generally occurs when max temperatures are between 40 and 60 degrees. This makes sense considering the seasonality of rain in Seattle becasue rain is anecdotally the most common in the colder months.

```
ggplot(data = weather) + aes(x = date, y = precipProbability, color = precipIntensity) + geom_point() +
```



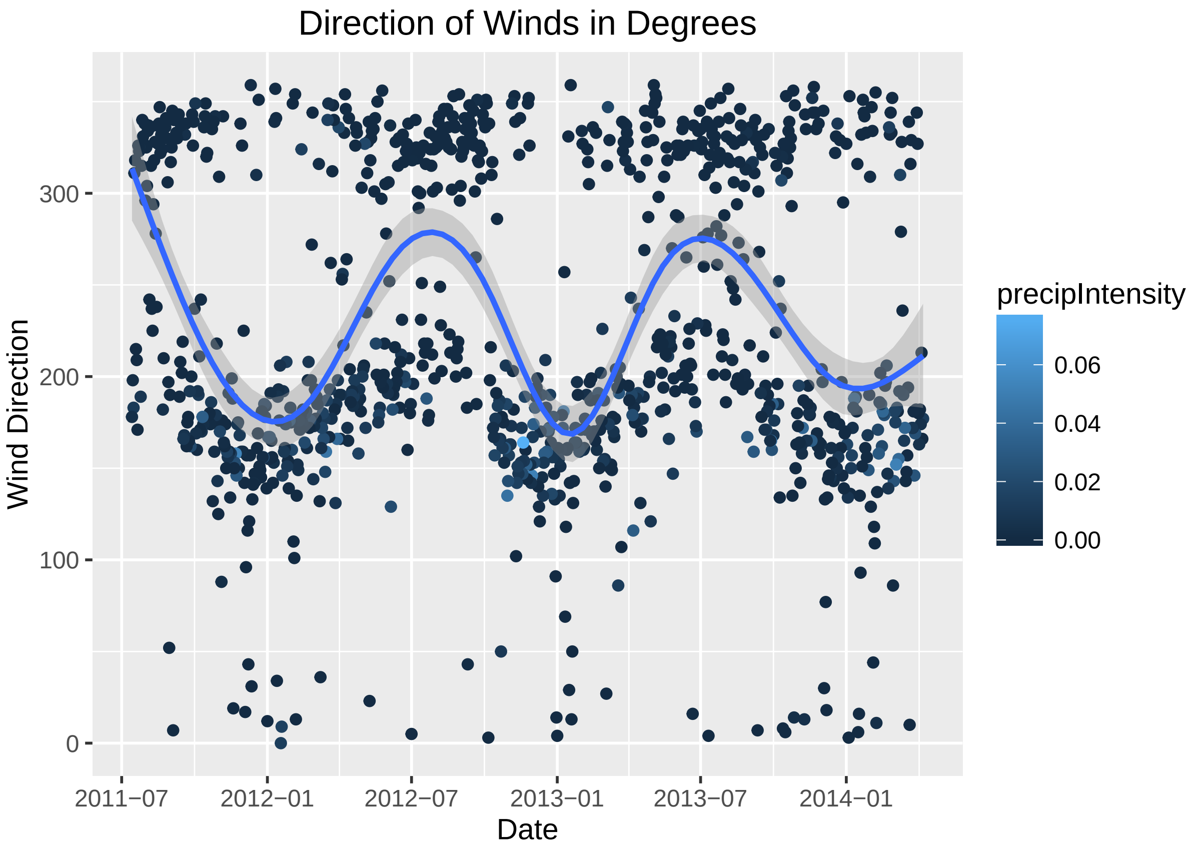
The precipitation probability is very seasonal here. Colder months have higher probability of rain than warmer months.

```
ggplot(data = weather) + aes(x = date, y = windSpeed, color = precipIntensity) + geom_point() + geom_smo
```



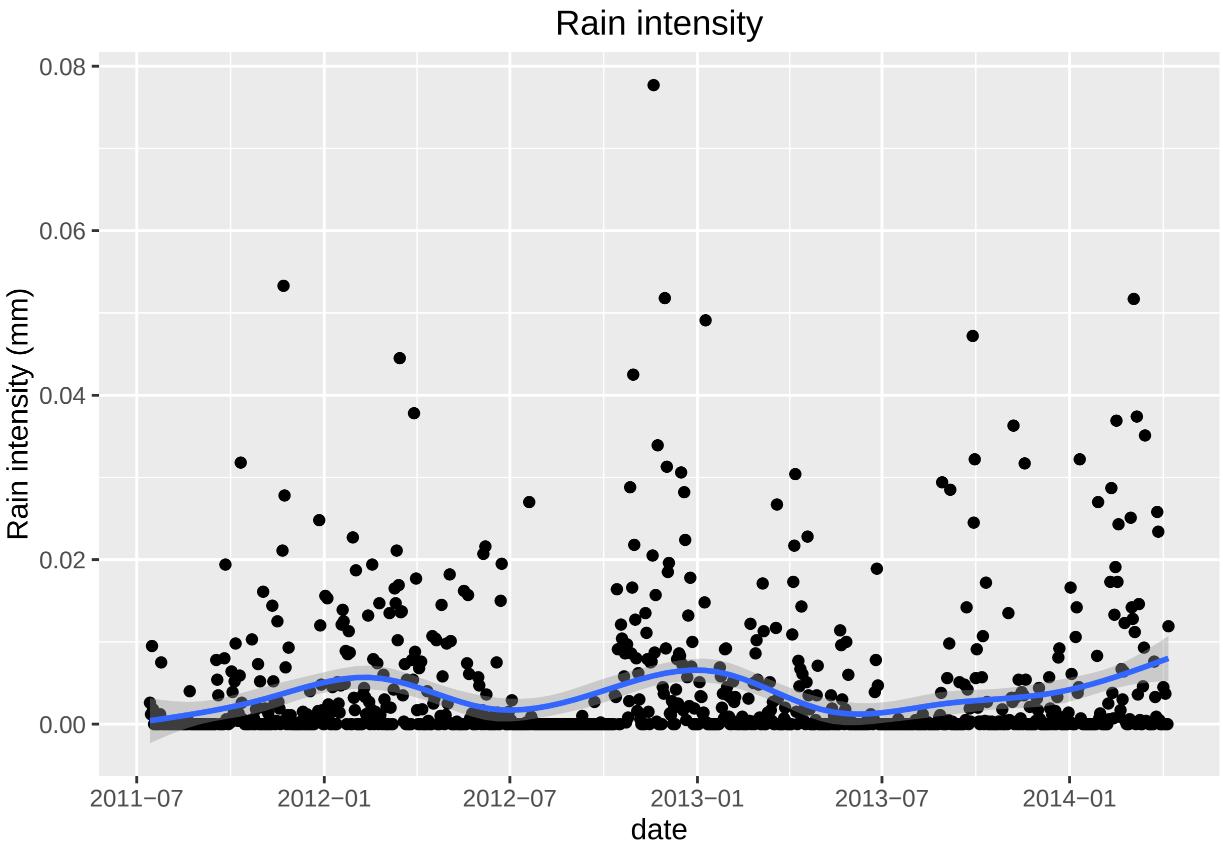
Windspeed has a little bit of a seasonal pattern, but not a lot. Ir does seem to be seasonal and somewhat associated with precipitation intensity. Higher intensities look to be on the days where windspeeds are above 5 mph.

```
ggplot(data = weather) + aes(x = date, y = windBearing, color = precipIntensity) + geom_point() + geom_s
```



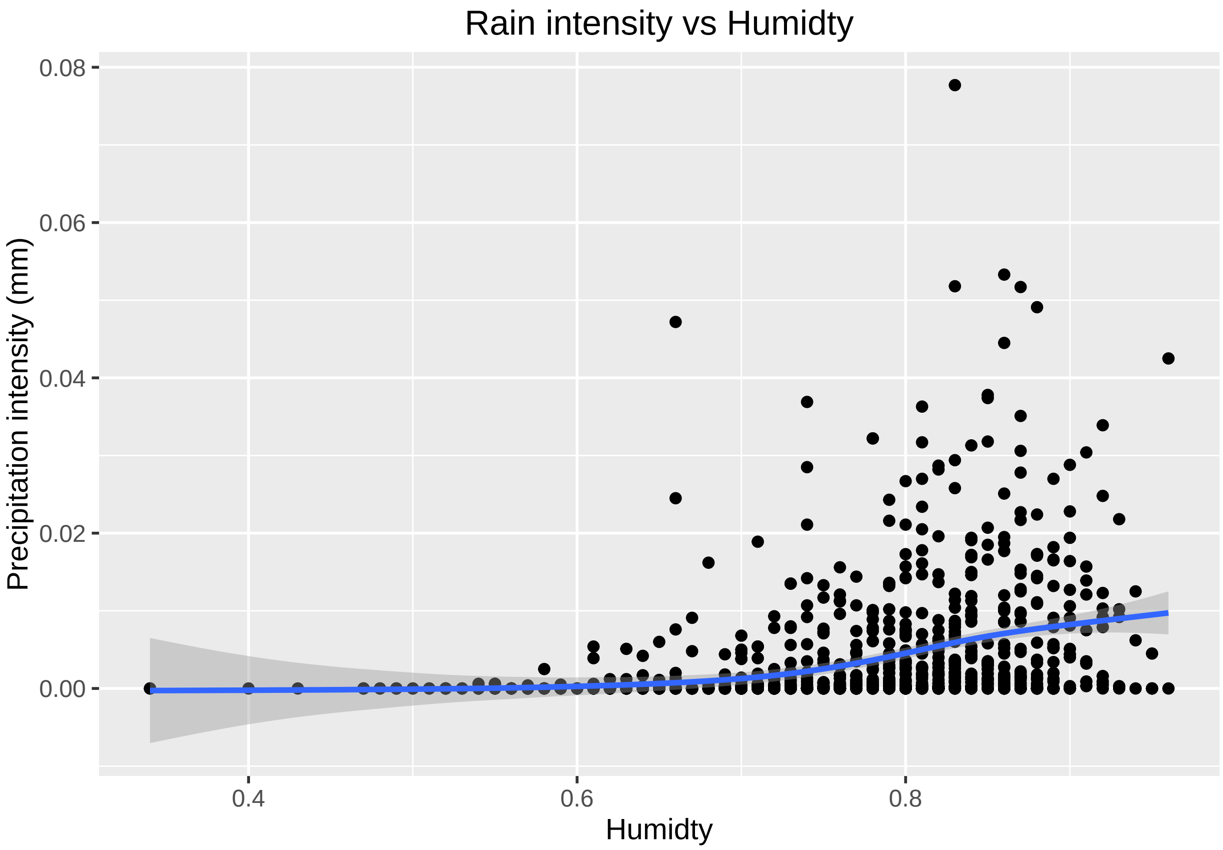
Interestingly, wind direction seems to be seasonal and precipitation intensity appears to be highest when the wind direction is between 100 degrees and 200 degrees which corresponds to winds blowing to the South during the colder months.

```
ggplot(data = weather) + aes(y = precipIntensity, x = date) + geom_point() + geom_smooth() + xlab("date
```



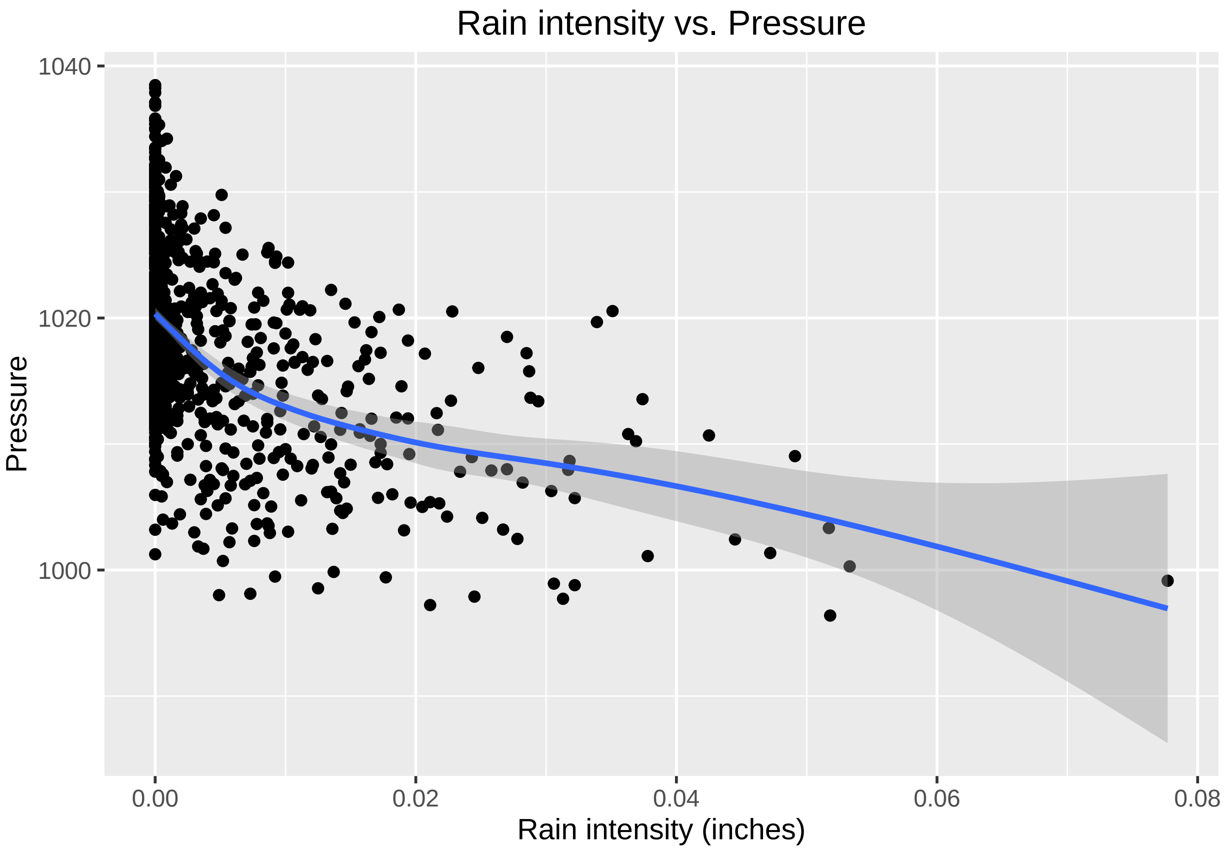
This plot also shows that there is a seasonality to the rain in Seattle with the highest intensities corresponding to cold months like January.

```
ggplot(data = weather) + aes(y = precipIntensity, x = humidity) + geom_point() + geom_smooth() + xlab("I
```



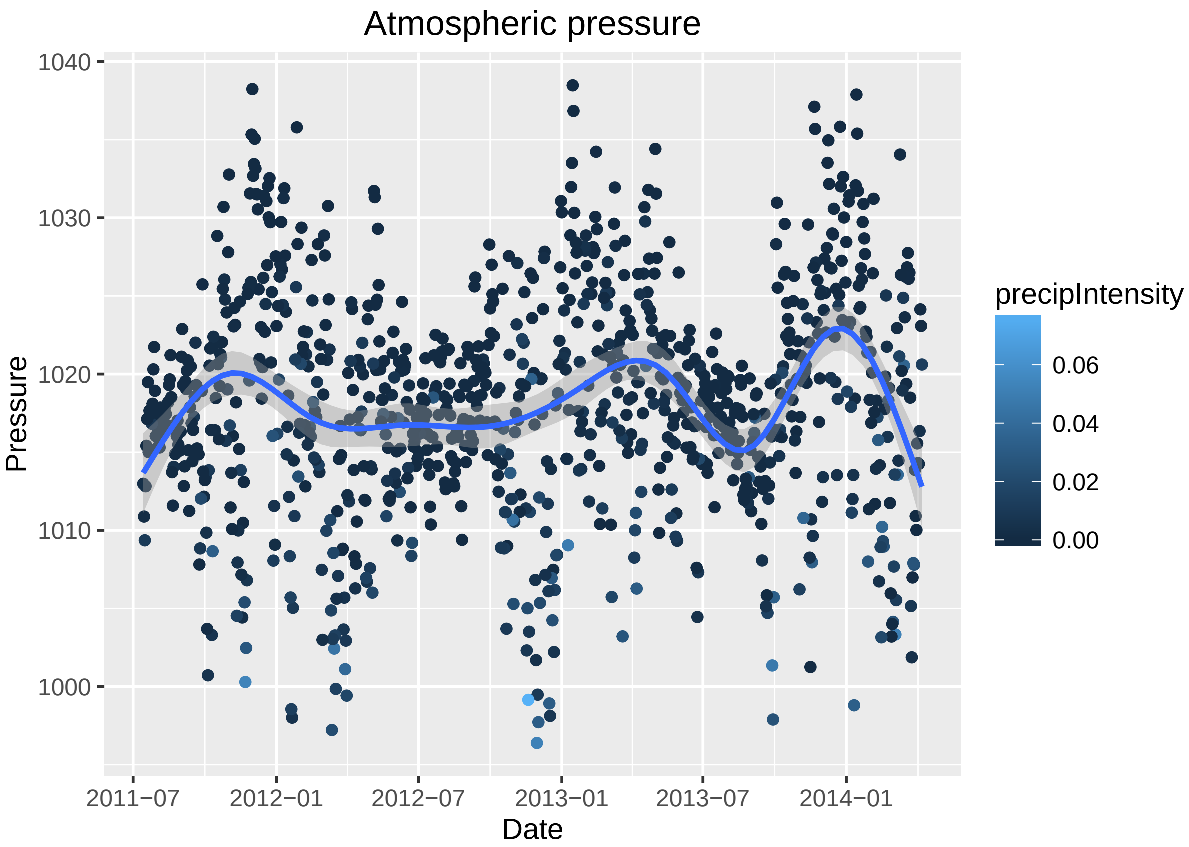
Not surprisingly, higher rain intensity seems to be associated with higher humidty levels.

```
ggplot(data = weather) + aes(x = precipIntensity, y = pressure) + geom_point() + geom_smooth() + ylab("
```



I know very little about the effects of atmospheric pressure on rain, but I find it to be interesting that lower pressures correspond to higher precipitation intensities.

```
ggplot(data = weather) + aes(y = pressure, x = date, color = precipIntensity) + geom_point() + geom_smoo
```



This graph shows that low pressures are probably not entirely responsible for high precipitation intensities. However, it does still have some kind of relationship becasue the lowest pressures of the year have high precipitation intensities sprinkled throughout, whereas the highest pressures do not.

```
# Examine some linear models
linear_PI_all <- lm(precipIntensity ~ ., data = weather)
linear_PI_month <- lm(precipIntensity ~ month(weather$date), data = weather)
linear_PI_tempMax <- lm(precipIntensity ~ temperatureMax, data = weather)
linear_PI_humidity <- lm(precipIntensity ~ humidity, data = weather)
linear_PI_pressure <- lm(precipIntensity ~ pressure, data = weather)
linear_PI_windBearing <- lm(precipIntensity ~ windBearing, data = weather)
linear_PI_windSpeed <- lm(precipIntensity ~ windSpeed, data = weather)
```

```
# All
summary(linear_PI_all)
```

```
##
## Call:
## lm(formula = precipIntensity ~ ., data = weather)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.011104 -0.002863 -0.000288  0.001620  0.061127
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          2.151e-01  3.406e-02   6.317 4.03e-10 ***
## date               1.225e-06  7.825e-07   1.565 0.117789    
## moonPhase         -1.222e-04  6.221e-04  -0.196 0.844291    
## precipProbability  1.096e-02  7.546e-04  14.520  < 2e-16 ***
## temperatureMin     1.712e-04  1.504e-04   1.138 0.255466    
## temperatureMax     3.645e-04  9.643e-05   3.780 0.000166 ***
## dewPoint          -5.840e-04  2.305e-04  -2.534 0.011441 *  
## humidity           3.472e-02  9.028e-03   3.846 0.000128 ***
## windSpeed          3.605e-04  8.550e-05   4.217 2.71e-05 ***
## windBearing       -4.167e-06  2.435e-06  -1.711 0.087358 .  
## cloudCover        -8.066e-04  6.692e-04  -1.205 0.228390    
## pressure          -2.591e-04  3.033e-05  -8.542  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.005602 on 988 degrees of freedom
## Multiple R-squared:  0.4963, Adjusted R-squared:  0.4907 
## F-statistic: 88.52 on 11 and 988 DF,  p-value: < 2.2e-16
```

Looking at all of the features, it's surprising that the date doesn't seem to be very imporant even though dates are directly related to seasons. The prcipitation probability is highly useful in predicting the rain, but it's also cheating the system a little bit. So I won't be using that feature. The Max temperature, windspeed, humidity, and pressure all seem to have the most importance from a combined linear model perspective.

```r
# Month
summary(linear_PI_month)
```

```
## 
## Call:
## lm(formula = precipIntensity ~ month(weather$date), data = weather)
## 
## Residuals:
##       Min        1Q    Median        3Q       Max 
## -0.004092 -0.003608 -0.003285 -0.000291  0.074415 
## 
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)    
## (Intercept)          4.173e-03  5.229e-04   7.980 3.99e-15 ***
## month(weather$date) -8.067e-05  6.937e-05  -1.163    0.245    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.007849 on 998 degrees of freedom
## Multiple R-squared:  0.001353,   Adjusted R-squared:  0.0003526 
## F-statistic: 1.352 on 1 and 998 DF,  p-value: 0.2451
```

A p-value here of 0.2451 is suggests that that the month is not an important aspect and is very surprising to me. It's possible that becasue the rain season is 8 to 10 months long, it leaves relatively few months that rain-free.

```r
# Temp max
summary(linear_PI_tempMax)
```

```
## 
```

```
## Call:
## lm(formula = precipIntensity ~ temperatureMax, data = weather)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.007088 -0.003727 -0.002259 -0.000553  0.073884
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.087e-02  1.333e-03   8.157 1.02e-15 ***
## temperatureMax -1.261e-04  2.283e-05  -5.522 4.28e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.007737 on 998 degrees of freedom
## Multiple R-squared:  0.02964,    Adjusted R-squared:  0.02867
## F-statistic: 30.49 on 1 and 998 DF,  p-value: 4.284e-08
```

The max temperature has a very low p-value here. Linealry speaking, a day with a certain range of maximum temp has a pretty good chance of seeing rain.

```
# Humidity
summary(linear_PI_humidity)
```

```
##
## Call:
## lm(formula = precipIntensity ~ humidity, data = weather)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.008781 -0.003803 -0.001967  0.001061  0.072200
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.015445   0.001783  -8.661   <2e-16 ***
## humidity     0.025235   0.002338  10.795   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.007433 on 998 degrees of freedom
## Multiple R-squared:  0.1045, Adjusted R-squared:  0.1037
## F-statistic: 116.5 on 1 and 998 DF,  p-value: < 2.2e-16
```

Humidty also seems to be good linear predictor of rain.

```
# Pressure
summary(linear_PI_pressure)
```

```
##
## Call:
## lm(formula = precipIntensity ~ pressure, data = weather)
##
```

```
## Residuals:
##       Min        1Q    Median        3Q       Max 
## -0.012072 -0.003960 -0.001827  0.001908  0.064578 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  0.5127000  0.0310519   16.51   <2e-16 ***
## pressure    -0.0005000  0.0000305  -16.39   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.006972 on 998 degrees of freedom
## Multiple R-squared:  0.2122, Adjusted R-squared:  0.2114 
## F-statistic: 268.8 on 1 and 998 DF,  p-value: < 2.2e-16
```

The pressure linear model has a low p-value and should also be a good predictor of rain in Seattle.

```
# Wind Direction
summary(linear_PI_windBearing)
```

```
## 
## Call:
## lm(formula = precipIntensity ~ windBearing, data = weather)
## 
## Residuals:
##       Min        1Q    Median        3Q       Max 
## -0.008898 -0.003937 -0.001307 -0.000499  0.072597 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  8.969e-03  6.718e-04  13.350   <2e-16 ***
## windBearing -2.357e-05  2.775e-06  -8.495   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.007585 on 998 degrees of freedom
## Multiple R-squared:  0.06743,    Adjusted R-squared:  0.0665 
## F-statistic: 72.17 on 1 and 998 DF,  p-value: < 2.2e-16
```

The wind direction turns out to be a good predictor on it's own. Likely becasue of the seasonality of wind directions.

```
anova(linear_PI_all, linear_PI_tempMax, linear_PI_humidity, linear_PI_pressure, linear_PI_windBearing, l
```

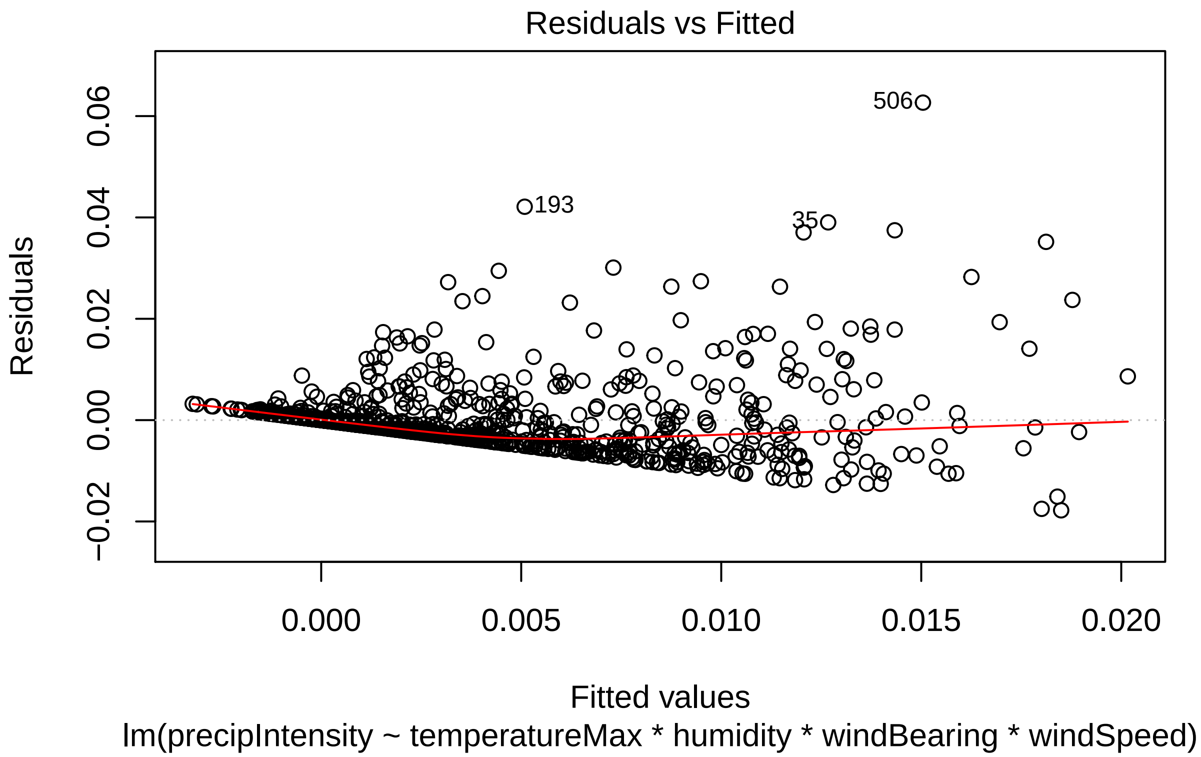
```
## Analysis of Variance Table
## 
## Model 1: precipIntensity ~ date + moonPhase + precipProbability + temperatureMin + 
##     temperatureMax + dewPoint + humidity + windSpeed + windBearing + 
##     cloudCover + pressure
## Model 2: precipIntensity ~ temperatureMax
## Model 3: precipIntensity ~ humidity
## Model 4: precipIntensity ~ pressure
## Model 5: precipIntensity ~ windBearing
## Model 6: precipIntensity ~ windSpeed
##   Res.Df      RSS  Df  Sum of Sq      F    Pr(>F)
## 1    988 0.031011
## 2    998 0.059747 -10 -0.0287358 91.552 < 2.2e-16 ***
## 3    998 0.055135   0  0.0046120
## 4    998 0.048508   0  0.0066266
## 5    998 0.057420   0 -0.0089118
## 6    998 0.053090   0  0.0043296
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

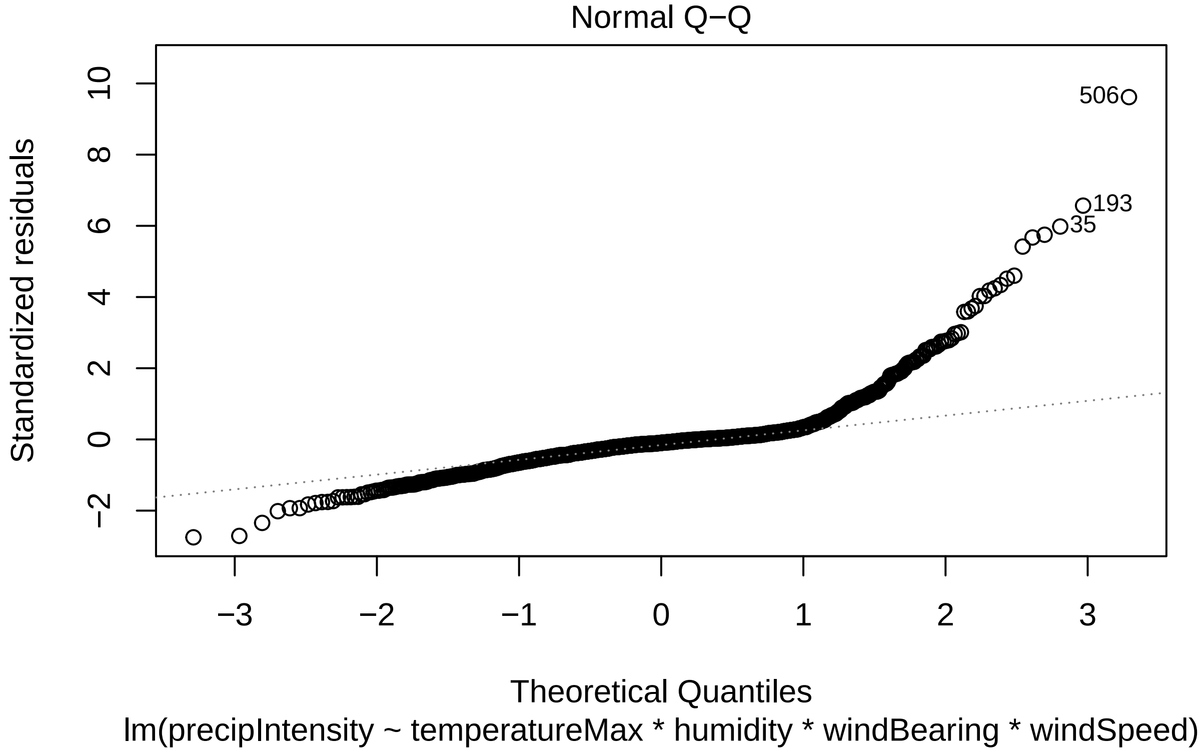
Singularly, only the Max temeprature model is significant.

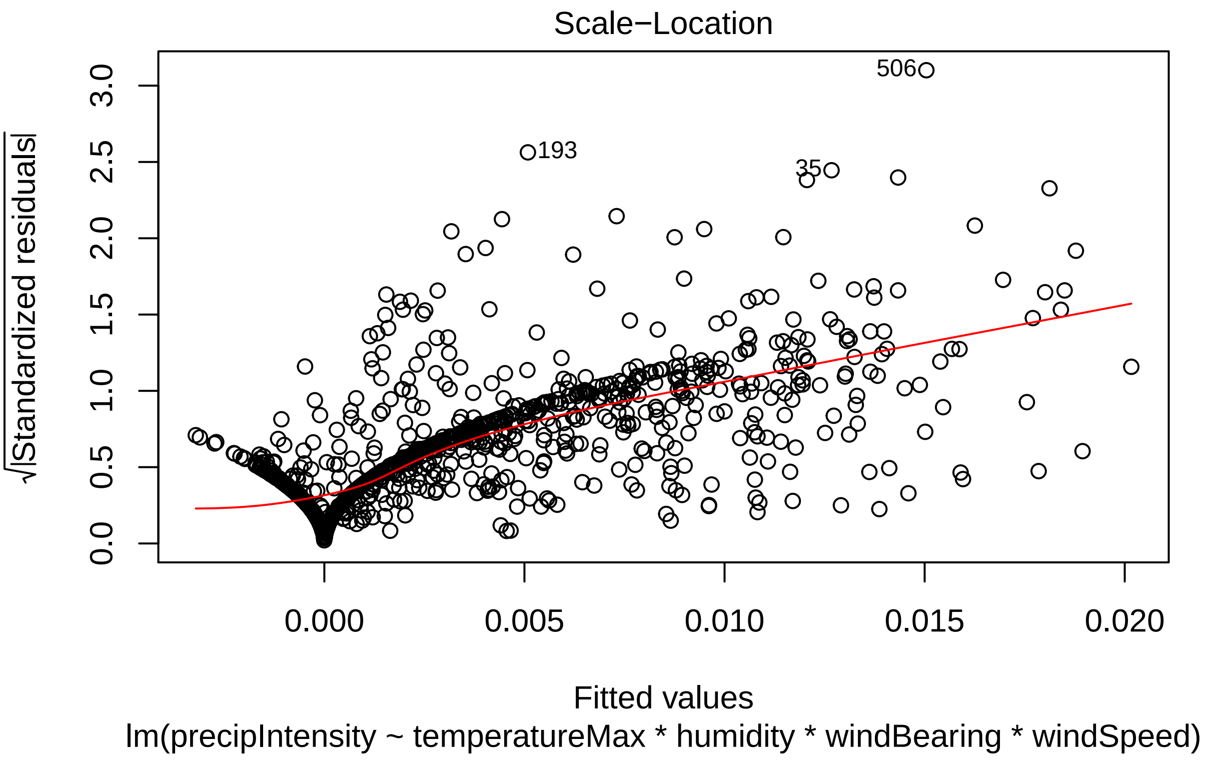
```r
combo_model <- lm(precipIntensity ~ temperatureMax*humidity*windBearing*windSpeed, data = weather)
anova(combo_model)
```

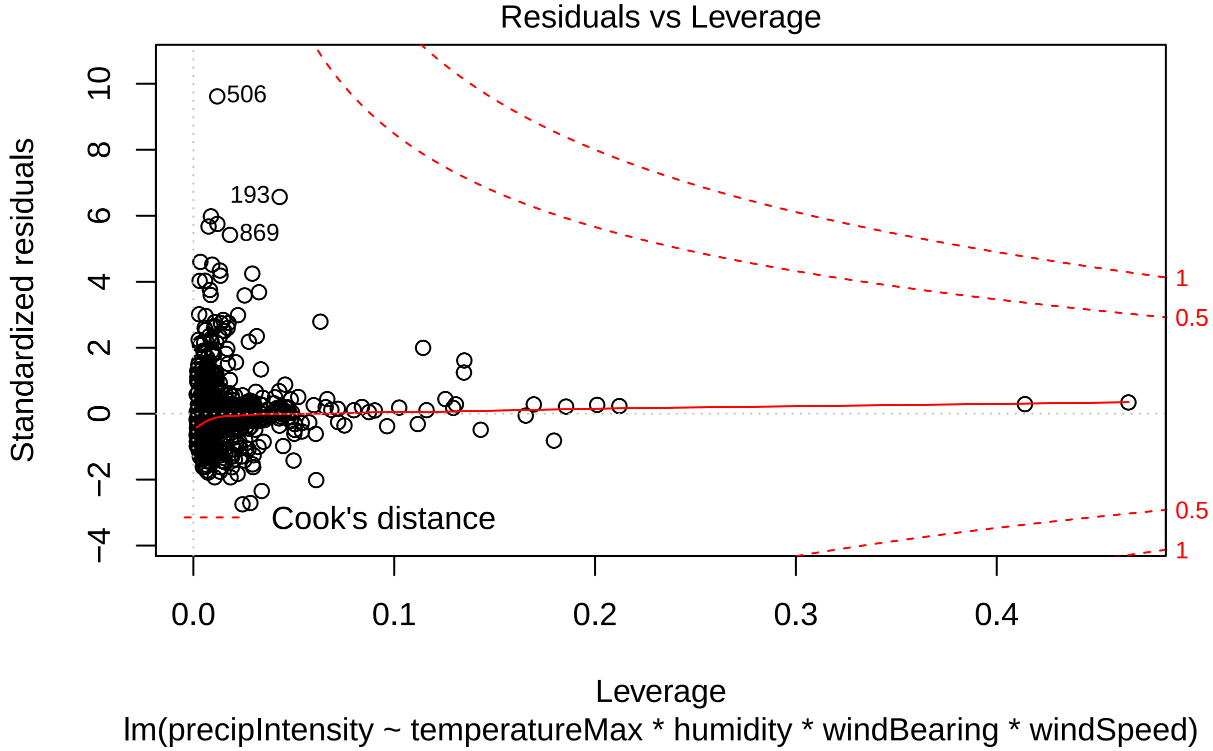
```
## Analysis of Variance Table
##
## Response: precipIntensity
##                                               Df   Sum Sq   Mean Sq
## temperatureMax                                 1 0.001825 0.0018252
## humidity                                       1 0.004621 0.0046212
## windBearing                                    1 0.001656 0.0016559
## windSpeed                                      1 0.006698 0.0066981
## temperatureMax:humidity                        1 0.000004 0.0000037
## temperatureMax:windBearing                     1 0.000104 0.0001041
## humidity:windBearing                           1 0.000377 0.0003767
## temperatureMax:windSpeed                       1 0.000141 0.0001411
## humidity:windSpeed                             1 0.001052 0.0010524
## windBearing:windSpeed                          1 0.001340 0.0013404
## temperatureMax:humidity:windBearing            1 0.000672 0.0006718
## temperatureMax:humidity:windSpeed              1 0.000003 0.0000026
## temperatureMax:windBearing:windSpeed           1 0.000000 0.0000000
## humidity:windBearing:windSpeed                 1 0.000603 0.0006032
## temperatureMax:humidity:windBearing:windSpeed  1 0.000200 0.0001999
## Residuals                                    984 0.042275 0.0000430
##                                               F value    Pr(>F)
## temperatureMax                                42.4829 1.137e-10 ***
## humidity                                     107.5625 < 2.2e-16 ***
## windBearing                                   38.5427 7.887e-10 ***
## windSpeed                                    155.9047 < 2.2e-16 ***
## temperatureMax:humidity                        0.0861 0.7692673
## temperatureMax:windBearing                     2.4238 0.1198298
## humidity:windBearing                           8.7689 0.0031377 **
## temperatureMax:windSpeed                       3.2837 0.0702760 .
## humidity:windSpeed                            24.4949 8.761e-07 ***
## windBearing:windSpeed                         31.1993 3.013e-08 ***
## temperatureMax:humidity:windBearing           15.6356 8.229e-05 ***
## temperatureMax:humidity:windSpeed              0.0601 0.8064331
## temperatureMax:windBearing:windSpeed           0.0006 0.9809148
## humidity:windBearing:windSpeed                14.0396 0.0001894 ***
## temperatureMax:humidity:windBearing:windSpeed  4.6528 0.0312438 *
## Residuals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot(combo_model)
```









Preliminary results: Linear models

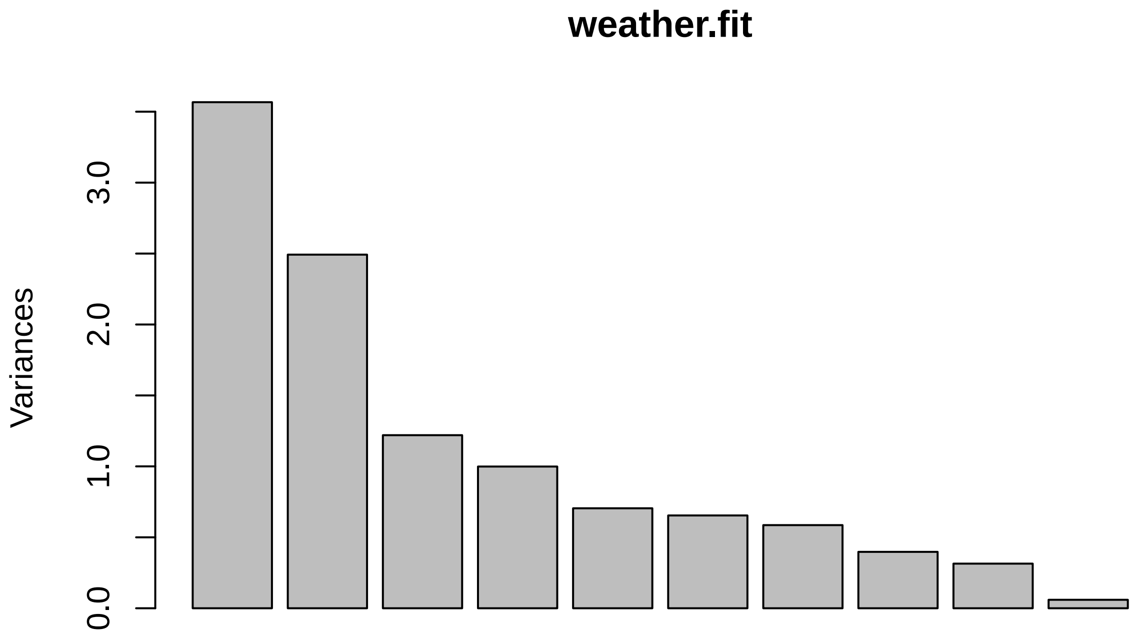
The Residuals vs. Fitted plot shows some definite non-linear / homoscedastic anomalies, suggesting that there are some non-linear relationships which are not accounted for in these linear models. The Normal Q-Q plot shows that there is some normality here but not the best. The Location-Spread plot shows that the residuals are not randomly dispersed - in fact, there is a very specific v-shaped pattern to some of the residuals.

Barring the fact that there really are no strong, linear relationships in this data, these linear models do suggest that the max temperature, humidity, pressure, and wind direction are all possibly important for predicting the precipitation intensity. The anova analysis suggests that the max temperatures and pressures might be the best predictors which would make sense considering the seasonality of them and the rain seasons in Seattle. That is to say, I've got a good chance of success if I predict rain for any given day in October - June. In general, it seems like rain intensity will have a higher chance of being > 0 when max temperature is between 40 and 60 degrees, Pressure is low, and humidity is high.

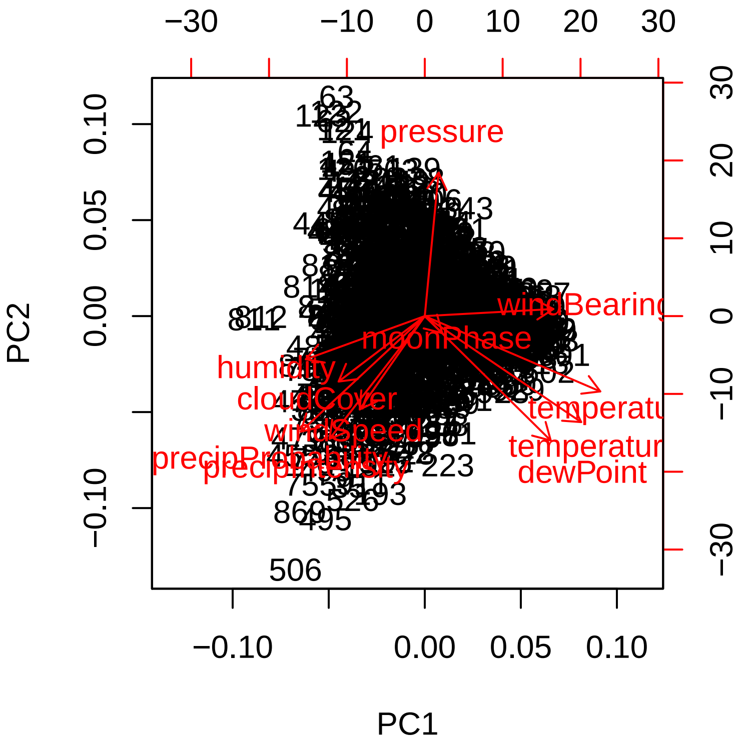
```
#Principle component analysis
weather.fit <- prcomp(weather[-1], center = TRUE, scale = TRUE)
summary(weather.fit)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6
## Standard deviation     1.8886 1.5787 1.1045 0.99944 0.83943 0.80876
## Proportion of Variance 0.3243 0.2266 0.1109 0.09081 0.06406 0.05946
## Cumulative Proportion  0.3243 0.5508 0.6617 0.75256 0.81662 0.87608
##                            PC7     PC8    PC9    PC10    PC11
## Standard deviation     0.76552 0.63052 0.5609 0.24429 0.07262
## Proportion of Variance 0.05327 0.03614 0.0286 0.00543 0.00048
## Cumulative Proportion  0.92936 0.96550 0.9941 0.99952 1.00000
```

```
plot(weather.fit)
```



```
biplot(weather.fit)
```



Principle Components

There are 11 principle components in the data set. The first 5 PC's account for about 80% of the variance but the first two PC's alone account for more than 50%. In the biplot of principle components, the fetures seem to split in half on the first axis. Humidty, cloud cover, precipitation intensity and precipitation probability lean negtively on both axes suggesting they are important in PC1 and PC2. Wind bearing, temperatures, and dew do not appear to be as useful for the first principle component but still suggest that they are important in the second principle component.

Preliminary results of exploratory analysis

There are interesting relationships between the the different weather variables and you can clearly see there is a seasonal cycle. Typically in Seattle, the rainy season is from late September to about June which is apparent in the exploratory data analysis and graphs.

Classification

Split the data into test and train data

```r
# prep the test and training data, split into 80% train, 20% test
size80 <- floor(0.8 * nrow(weather))
set.seed(69)
train_index <- sample(seq_len(nrow(weather)), size = size80)

weathertrain <- weather[train_index, ]
weathertest <- weather[-train_index, ]
```

Modeling

```
#####Logistic regression
# When precipIntensity is >0.01, rained = true
newcol = data.frame(rained = weathertrain$precipIntensity > 0.01)
# add to weather df
weathertrain <- cbind(weathertrain, newcol)

#Setup the forumla - train "rained" on everything else except precipIntensity
formula <- rained ~ date + moonPhase +  temperatureMin + temperatureMax + dewPoint + humidity + windSpe
# This model uses the precipitation probabilities, and I consider it cheating the system a little bit.
formula_cheat <- rained ~ precipProbability + date + moonPhase +  temperatureMin + temperatureMax + dew

# Fit the models
logisticModel <- glm(formula, data=weathertrain, family = "binomial")
logisticModel_cheat <- glm(formula_cheat, data=weathertrain, family = "binomial")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
# Predict the probabilities for the test data
prob <- predict(logisticModel, newdata = weathertest, type="response")
prob_cheat <- predict(logisticModel_cheat, newdata = weathertest, type="response")
```

I'm using two different logistic models here. The first one uses every feature except for the actual precipitation intensity and precipitation probability. The intensity is witheld becasue that is what I am using as a response variable, i.e. that's what I am trying to classify. The pobability is witheld becasue it is a prediction of rain in itself. The second model includes the precipitation probabilities and it is interesting to see the differences because it makes a dramatic difference on the models classification success.

Results

```
# Show the row number of the predictions with probability > 0.9
predictedYes <- as.numeric(names(prob[round(prob,1) > 0.80]))
rainYes <- as.integer(rownames(weather[predictedYes,]))
rainYes
```

```
## [1]  35 506
```

```
# Assess the predictive ability of the model
fitted.results <- predict(logisticModel,newdata=weathertest,type='response')
fitted.results <- ifelse(fitted.results > 0.5,1,0)
results <- data.frame(Prediction = fitted.results, Rained = weathertest$precipIntensity > 0.01)
table1 <- table(results)
table1
```

```
##           Rained
## Prediction FALSE TRUE
##          0   170   15
##          1     5   10
```

This table shows that rain was only predicted correctly 10 times out of 15 (66% correct) and there were no probabilities above 0.9. So not very accurate and not very probable. Keep in mind this does not include the precipitation probabilities from the data.

```r
# Show the row number of the predictions with probability > 0.9
predictedYes <- as.numeric(names(prob_cheat[round(prob_cheat,1) > 0.90]))
rainYes <- as.integer(rownames(weather[predictedYes,]))
rainYes
```

```
##  [1]  11  32  35  57 287 355 367 491 502 506 739 834
```

```r
# Assess the predictive ability of the model
fitted.results <- predict(logisticModel_cheat,newdata=weathertest,type='response')
fitted.results <- ifelse(fitted.results > 0.5,1,0)
results <- data.frame(Prediction = fitted.results, Rained = weathertest$precipIntensity > 0.01)
table2 <- table(results)
table2
```

```
##            Rained
## Prediction FALSE TRUE
##          0   171    6
##          1     4   19
```

This model has 12 days where the probabilities are higher than 0.9, as opposed to the previous model where none were above 0.9 and only two were above 0.8. This second table shows that rain was also predicted correctly 19 out of 23 times (82% correct vs 66% correct).

The first table has a couple of interesting features. The first is that by omitting the prediction probabilities, the model classified fewer days as rainy over all and was not as good at classifying correctly, i.e there were more false positives and false negatives. By including the precipitation probabilities, fewer days were classified as rainy overall and the accuracy was %16 better.

Discussion

Exploratory anlyses showed that linear models aren't the best for predicting the rain, and that's not much of a surprise. But using only a logistic regression based on previous weather data, this method successfully predicted rain between 66% and 82% of the time depending on if the precipitation probabilities were included in the model. Even though this is a relatively simple proof-of-concept method, I am quite surprised at how high the success rate was. One interesting implication is that I may be able to actually use this method in the real world to improve scheduling for my company. This analysis was used in a very controlled environment so it will be interesting to apply it to the business. Evenso, there is potential to further this anlysis with other classification algorithms including K-Nearest Neighbors, Neural Networks, and Support Vector Machines. The data mining portion of this project proved to be my most troublesome area. Even after I decided on a source of data, actually aquiring it and cleaning it took a significant amount of time. I think there is definitely room for improvement in my data wrangling - particularly in using transformations and omptimizations.I expected the dates to have much more impact on the predictions.

Interstingly, the dates did not affect the models I tried fitting, however, the exploratory data analysis clearly shows that there are patterns of rain that are highly associated with the month of the year. I think one potential problem stems from my data wrangling, perhaps the dates in this dataset need to be formatted differently with a seasonal component in mind.

This has truly been a rewarding project. There were many challeneges to overcome and some things had to be abandoned. The biggest hurdle was finding a source of data that could answer my specific question - "Is it going to rain?”. The original objective was to rate the accuracy of weather reporting agencies but was abandoned due to a lack of suitable data. However, there is a silver lining in the clouds. Out of this project I have discovered a possible niche in data collection that would be of interest to many other people.

References

1. “ARIMA FORECASTING TUTORIAL (PART 1).” DataScienceKumar. N.p., 05 Oct. 2014. Web. 19 Nov. 2016.
2. Berrocal, Veronica J. “Package ‘ProbForecastGOP’.” (2012): n. pag. Web. 15 Nov. 2016.
3. “CRAN - Package EnsembleBMA.” CRAN - Package EnsembleBMA. N.p., n.d. Web. 15 Nov. 2016.
4. “Forecasting the Weather with R.” R-bloggers. N.p., 22 Dec. 2009. Web. 7 Dec. 2016. https://www.r-bloggers.com/forecasting-the-weather-with-r/.
5. Mcpherron, Robert L., and George Siscoe. “Probabilistic Forecasting of Geomagnetic Indices Using Solar Wind Air Mass Analysis.” Space Weather 2.1 (2004): Web.
6. “Part 4a: Modelling – Predicting the Amount of Rain.” R-bloggers. 06 Apr. 2015. Web. 19 Nov. 2016. https://www.r-bloggers.com/part-4a-modelling-predicting-the-amount-of-rain/.
7. R-Bloggers. http://www.r-bloggers.com.
8. Schereur, Michael. “Probabilistic Quantitative Precipitation Forecasting Using Ensemble Model Output Statistics.” Scheuerer - 2013 - Quarterly Journal of the Royal Meteorological Society - Wiley Online Library. N.p., n.d. Web. 12 Dec. 2016.

Supplement 1

Data collection script

```r
# Dark sky API
rm(list=ls())

# Load Dark Sky R wrapper
require("darksky")
require("RCurl")
require("jsonlite")
require("lubridate")
require("plyr")


# Seattle Latitude and Longitude
lat <- "47.608013"
lon <- "-122.335167"


# Set Dark Sky API key
DARKSKY_API_KEY = NA


# Get the dates for the number of days of data
```

```
listofdates <- NULL
numdays <- 1000
  for (i in 1:numdays){
  listofdates <- append(x = listofdates, values = date-i)
  }


# init storage
write_all <- NULL
write_hourly <- NULL
write_daily <- NULL
df_all <- NULL
df_hourly <- NULL
df_daily <- NULL


# Iterate through list and do an API call for each
for (i in 1:length(listofdates)){

  possibleError <- tryCatch({
    #Convert to UNIX time for API
    UNIXtime <- as.integer(as.POSIXct(listofdates[i]))

    # Create/fetch the URL
    root <- "https://api.darksky.net/forecast/{DARK_SKY_API}/"
    #u <- paste(root, lat, ",", lon, ",", UNIXtime, "?exclude=currently,flags", sep = "")
    u <- paste(root, lat, ",", lon, ",", UNIXtime, "?exclude=currently,flags", sep = "")
    url <- URLencode(u)

    # Parse the JSON
    json <- getURL(url)
    simple <- fromJSON(txt = json, simplifyDataFrame = TRUE, flatten = TRUE)
    all <- simple
    hourly <- simple$hourly
    daily <- simple$daily

    # Convert UNIX time to datetime
    time <- lapply(hourly$data["time"], as_datetime, tz = Sys.timezone())

    # Diagnostics
    print(paste("Iteration: ",i))
    #print(names(hourly$data))
    #print(names(daily$data))

    df_hourly <- data.frame( Time = time, hourly$data, check.rows = FALSE )
    write_hourly <- rbind.fill(write_hourly, df_hourly)
    write.csv( write_hourly, file="../Research\ project/10khours.csv", row.names = FALSE, append = TRUE

    df_daily <- data.frame( daily$data, check.rows = FALSE )
    write_daily <- rbind.fill(write_daily, df_daily)
    write.csv( write_daily, file="../Research\ project/10kdays.csv", row.names = FALSE, append = TRUE )
  }, error = function(err) {
    err
    # error handler picks up where error was generated
    print(paste("Iteration:", i,"MY_ERROR:", err))
  })

  if(inherits(possibleError, "error")) next

}
```