Ottawa Bike Counter Machine Learning Project

Nick Marum

28/10/2020

INTRODUCTION

This report documents the exploratory analysis and machine learning model building process for a HarvardX Data Science Professional Certificate capstone project. It is uses publicly available City of Ottawa bike ride counter data from public pathways in Ottawa, CANADA as well as publicly available weather data from Environment Canada.

The data used in this project were found in zip format on Kaggle at the following address: https://www.kaggle.com/m7homson/ottawa-bike-counters/download To access it on Kaggle you have to create an account and login which is a challenge to do in the R environment.

Since kaggle requires an account and login to access the data, I have shared the information on my github repo. A copy of the final script, including data import and wrangling is included on my github repo: https://github.com/nmarum/ottawa_bike_counters

The reason why I selected this particular dataset for this project is that I live in Ottawa and am a regular bike rider who commutes to and from work on my bike about 8 months of the year. There is an extensive network of multi-use pathways in Ottawa, with several bike counters that are prominently placed. It appealed to me to use a data set to make predictions about something I am very familiar with and in my home town.

EXPLORATORY ANALYSIS

```r
summary(dat)
```

```
##  location_name       location_id        count              day
##  Length:28776       Min.   : 0.00   Min.   :   0.0   Min.   :   0
##  Class :character   1st Qu.: 3.00   1st Qu.:  94.0   1st Qu.:1269
##  Mode  :character   Median : 7.00   Median : 538.0   Median :1948
##                     Mean   : 6.44   Mean   : 822.8   Mean   :1876
##                     3rd Qu.:10.00   3rd Qu.:1311.0   3rd Qu.:2571
##                     Max.   :13.00   Max.   :7617.0   Max.   :3286
##
##   day_of_year     day_of_week       MaxTemp          MeanTemp
##  Min.   :  0.0   Min.   :0.000   Min.   :-24.50   Min.   :-26.800
##  1st Qu.: 88.0   1st Qu.:1.000   1st Qu.:  2.00   1st Qu.: -1.500
##  Median :182.0   Median :3.000   Median : 13.50   Median :  8.700
##  Mean   :179.6   Mean   :3.003   Mean   : 12.37   Mean   :  7.599
##  3rd Qu.:269.0   3rd Qu.:5.000   3rd Qu.: 23.80   3rd Qu.: 18.300
##  Max.   :365.0   Max.   :6.000   Max.   : 36.30   Max.   : 29.800
##                                  NA's   :127      NA's   :127
##     MinTemp         SnowonGrndcm    TotalPrecipmm     TotalRainmm
##  Min.   :-30.900   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
##  1st Qu.: -5.000   1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.000
##  Median :  3.900   Median : 0.000   Median : 0.000   Median : 0.000
```

```
##  Mean   :  2.809   Mean   : 6.214   Mean   : 2.521   Mean   : 2.103
##  3rd Qu.: 12.500   3rd Qu.: 8.000   3rd Qu.: 2.000   3rd Qu.: 1.000
##  Max.   : 23.300   Max.   :66.000   Max.   :84.600   Max.   :84.600
##  NA's   :127       NA's   :142      NA's   :143      NA's   :143
##   TotalSnowcm           date
##  Min.   : 0.0000   Min.   :2010-01-01
##  1st Qu.: 0.0000   1st Qu.:2013-06-23
##  Median : 0.0000   Median :2015-05-02
##  Mean   : 0.4892   Mean   :2015-02-19
##  3rd Qu.: 0.0000   3rd Qu.:2017-01-15
##  Max.   :37.0000   Max.   :2018-12-31
##  NA's   :143
```

A quick summary of the wrangled data set shows that there are a total of 14 columns reflecting 28,776 total entries. The "location_id" and "location_name" are obviously identifiers of bike ride counter locations. The dependent variable for this machine learning project is "count" which is the number of two-way rides past a bike counter recorded on a daily basis. The "day" variable is a unique identifier per day from the first entry in January 2010 until the last entry on Dec 31, 2018; "day_of_week" identifies the day of the week by a numeric identifier starting with Sunday as 0 and Saturday as 6, and "day_of_year" being a count from 0 to 365 of each day in the calendar year (including an extra day for leap years). I added a "date" field using the lubridate package during data wrangling to be able to more clearly associate an entry with a day of the year. Finally, there are a series of weather observations that have been married with the bike counter ride entries which include: "MaxTemp" for maximum daily temperature, "Totalprecipmm" for total precipitation recorded that day in millimetres, among others.

```r
dat %>%
  group_by(location_name) %>%
  summarize(mean_count = mean(count),
            prop_zerocount = mean(count==0),
            entries = n()) %>%
  arrange(desc(entries))
```

```
## # A tibble: 14 x 4
##    location_name mean_count prop_zerocount entries
##    <chr>              <dbl>          <dbl>   <int>
##  1 COBY                867.        0.0166     3259
##  2 ORPY               1223.        0.198      3258
##  3 ALEX                951.        0.164      3136
##  4 LMET               1105.        0.00318    2829
##  5 SOMO                385.        0.00777    2573
##  6 CRTZ                889.        0.0309     2526
##  7 OGLD                483.        0.0501     2097
##  8 OBVW                406.        0.109      2096
##  9 LLYN                790.        0.00469    1919
## 10 OYNG                407.        0.0320     1188
## 11 ADAWE BIKE          927.        0          1184
## 12 ADAWE PED          1215.        0          1184
## 13 LBAY                313.        0.00984     915
## 14 PORTAGE            1362.        0.0212      612
```

Examining the bike counter ride count data, we see that some counters have more entries than others. While some counters seem to cease operations through the winter months, a number of counters which are identified as "winter" counters operate year round. These are a more interesting set of counters with which to make weather-based predictions. Also having more entries should improve the accuracy of these predictions.

The table above shows the bike counter along Colonel By Drive (COBY) near the Corktown footbridge crossing over the Rideau Canal as the counter with the most overall entries/fewest missing entries. COBY also has a relatively low proportion of "0" count/no bike entries (1.7%). While not the busiest counter, the consistent entries seem promising from a predictive perspective.

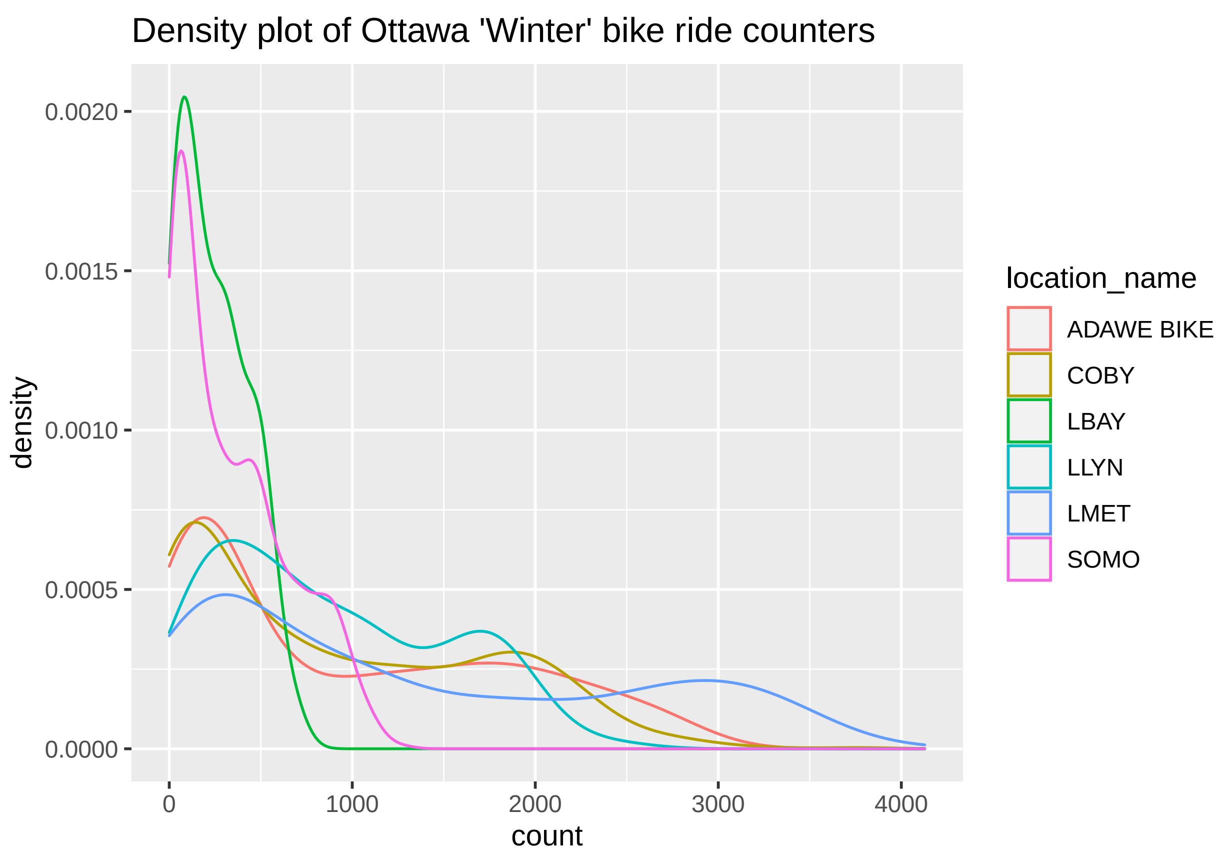
A City of Ottawa legend for the bike counter data provides the following description for the COBY counter:

"3_COBY: National Capital Commission (NCC) Eastern Canal Pathway approximately 100m north of the Corktown Bridge. #WINTER counter

"The data provides counts of bike trips (both directions summed unless otherwise noted)" COBY has not footnote associated with it so the data must be both directions."

This counter also appeals to me since it is close to my workplace and I have passed by this stretch of the Col By pathway near the Corktown Bridge hundreds of times.

```
coby <- dat %>% filter(location_name == "COBY")

dat %>% filter(location_name == c("COBY", "LMET", "LLYN", "LBAY", "SOMO", "ADAWE BIKE")) %>%
  ggplot(aes(count, col=location_name)) + geom_density() +
  ggtitle("Density plot of Ottawa 'Winter' bike ride counters")
```



Selecting for "winter" counters (i.e., counters that operate year round even when there is snow on the ground.), and using a density plot to see the frequency of bike ride counts per entry (i.e., day) we see that not all counters follow the same distribution. There seems to be one group of counters that follow a similar distribution with a wide distribution of count entries, while a couple of other entries have a high number of days with relatively few riders and relatively few busy days. We can see that COBY follows the distribution of most of the winter counters.

```r
coby_rides <- c(median(coby$count), mean(coby$count))
total_rides <- c(median(dat$count), mean(dat$count))
diff <- c(median(coby$count)-median(dat$count), mean(coby$count)-mean(dat$count))
data.frame(coby_rides, total_rides, diff, row.names = c("median", "mean"))
```

```
##        coby_rides total_rides      diff
## median   641.0000     538.000 103.00000
## mean     866.9064     822.769  44.13737
```

Looking at the mean and median of the COBY counter and all of the City of Ottawa bike ride counters, we can see that the mean and median are similar to the overall mean and median of bike ride counts.

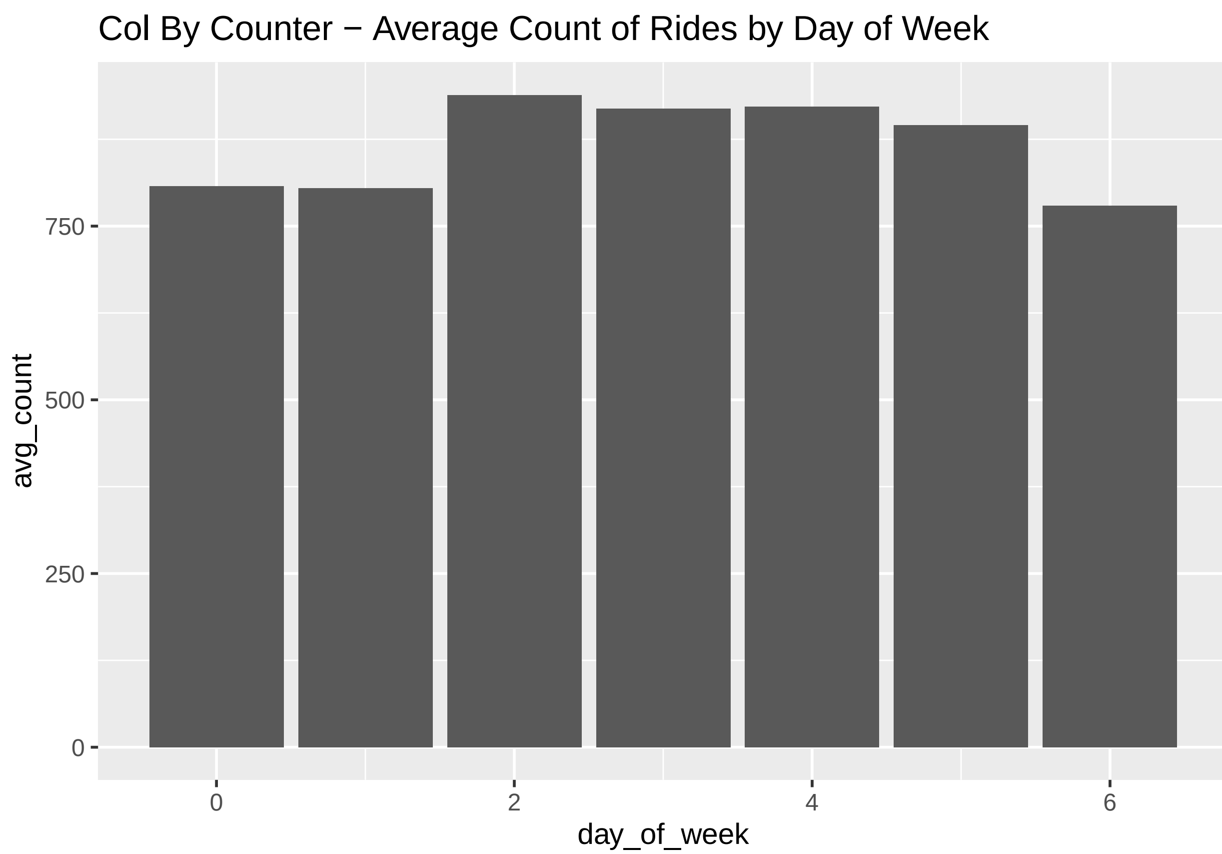
```r
tibble(sd(coby$count), sd(dat$count))
```

```
## # A tibble: 1 x 2
##   `sd(coby$count)` `sd(dat$count)`
##              <dbl>           <dbl>
## 1             814.            861.
```

Looking at standard deviation, we can see it is similar as well at 814 rides. This would suggest that a predictive model created by COBY data should have applicability to other winter counters. That being said, overall it appears that ride counts overall between different counters can vary widely, which could be a challenge to predict.

We will start to look to identify what kind of patterns may exist in the COBY bike counter over time.

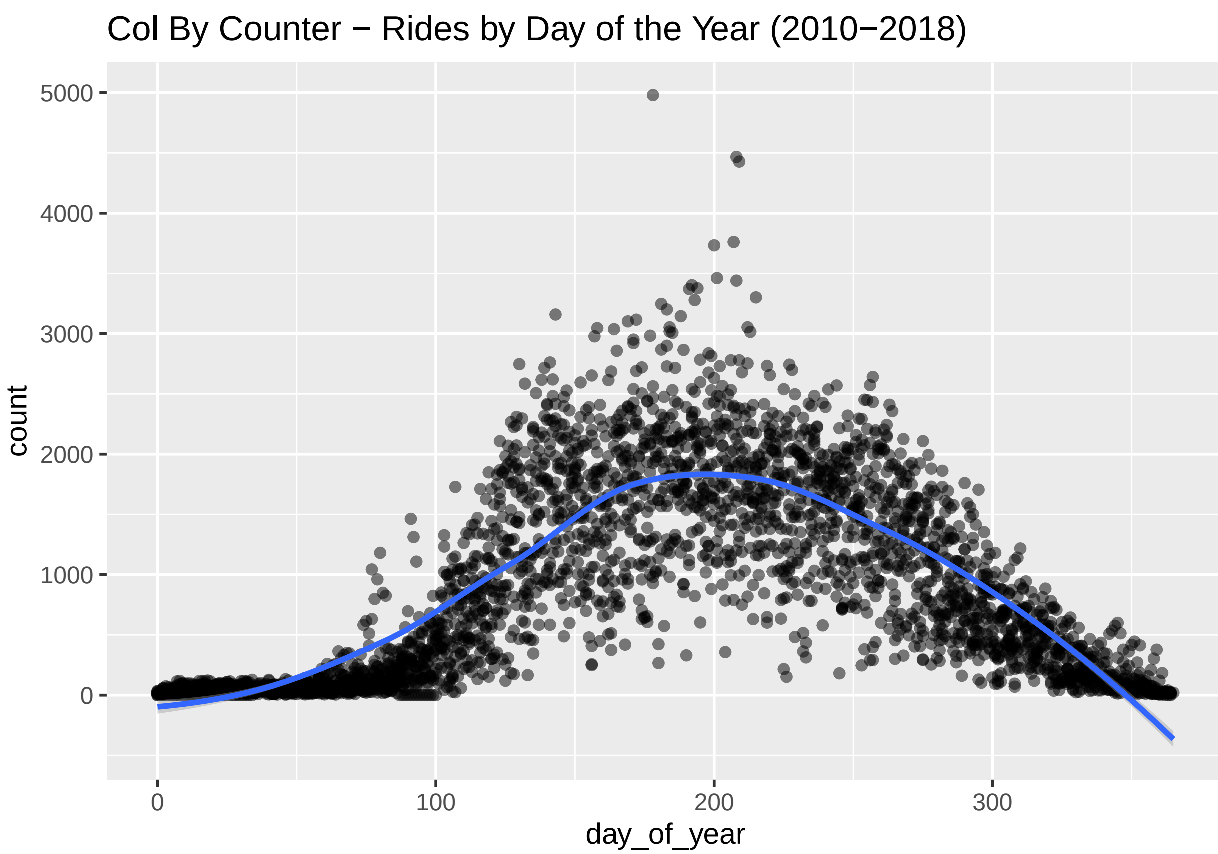
```r
coby %>% group_by(day_of_week) %>%
  summarize(avg_count = mean(count)) %>%
  ggplot(aes(day_of_week, avg_count)) + geom_col() +
  ggtitle("Col By Counter - Average Count of Rides by Day of Week")
```



Looking at the average number of rides by the day of the week (with 0 meaning Sunday and 6 meaning Saturday), we can see a trend of Tuesday to Friday of of higher traffic - likely from commuters heading towards downtown Ottawa along the Colonel By drive pathway. Traffic drops off a little, by about 20%, on the weekends and Mondays.

```
coby %>% mutate(year = year(date)) %>%
  ggplot(aes(day_of_year, count)) +
  geom_point(alpha=.5) +
  geom_smooth(method = "loess") +
  ggtitle("Col By Counter - Rides by Day of the Year (2010-2018)")
```

```
## 'geom_smooth()' using formula 'y ~ x'
```

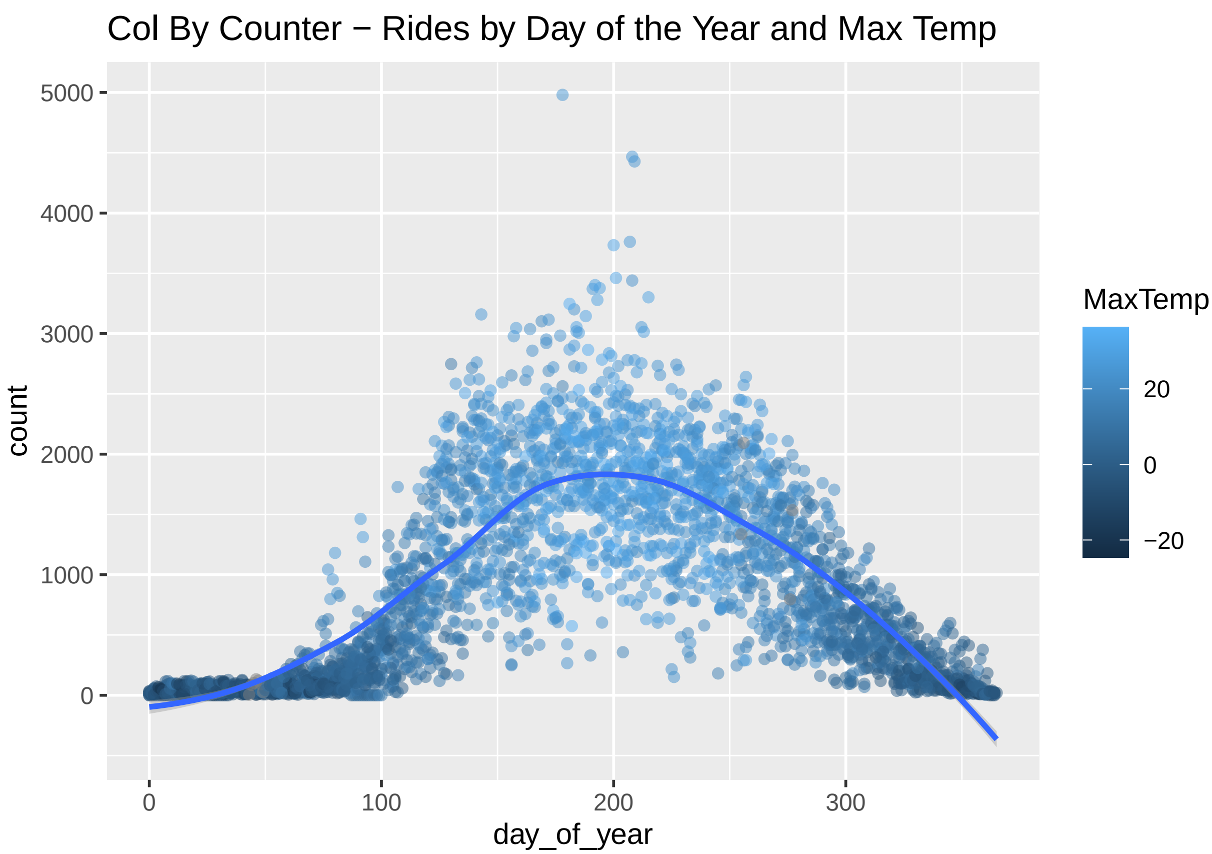


Looking at the ride count data by day of the year, with 0 being New Year's Day running to 364 as New Year's Eve of the same year, we see a clear trend in terms of the number of rides per day over course of the year. The first 100 days of year (January to early April) are the dead of winter in Ottawa - not many bike rides are recorded before the 80 to 90 day mark with the start of spring. From there, bike rides per day rise and peak around June and remain elevated through the summer months and then starts to gradually drop off in the fall and returns to winter levels by the month of December. This trends appears relatively consistent year over year and we can see the pattern by fitting a bin-smoothing line using locally weighted regression.

Obviously, the weather in the summer months in Ottawa is very different from the weather in the winter, so this pattern is not surprising.

```
coby %>% mutate(year = year(date), month = month(date)) %>%
  ggplot(aes(day_of_year, count, col=MaxTemp)) +
  geom_point(alpha=.5) +
  geom_smooth(method = "loess") +
  ggtitle("Col By Counter - Rides by Day of the Year and Max Temp")
```

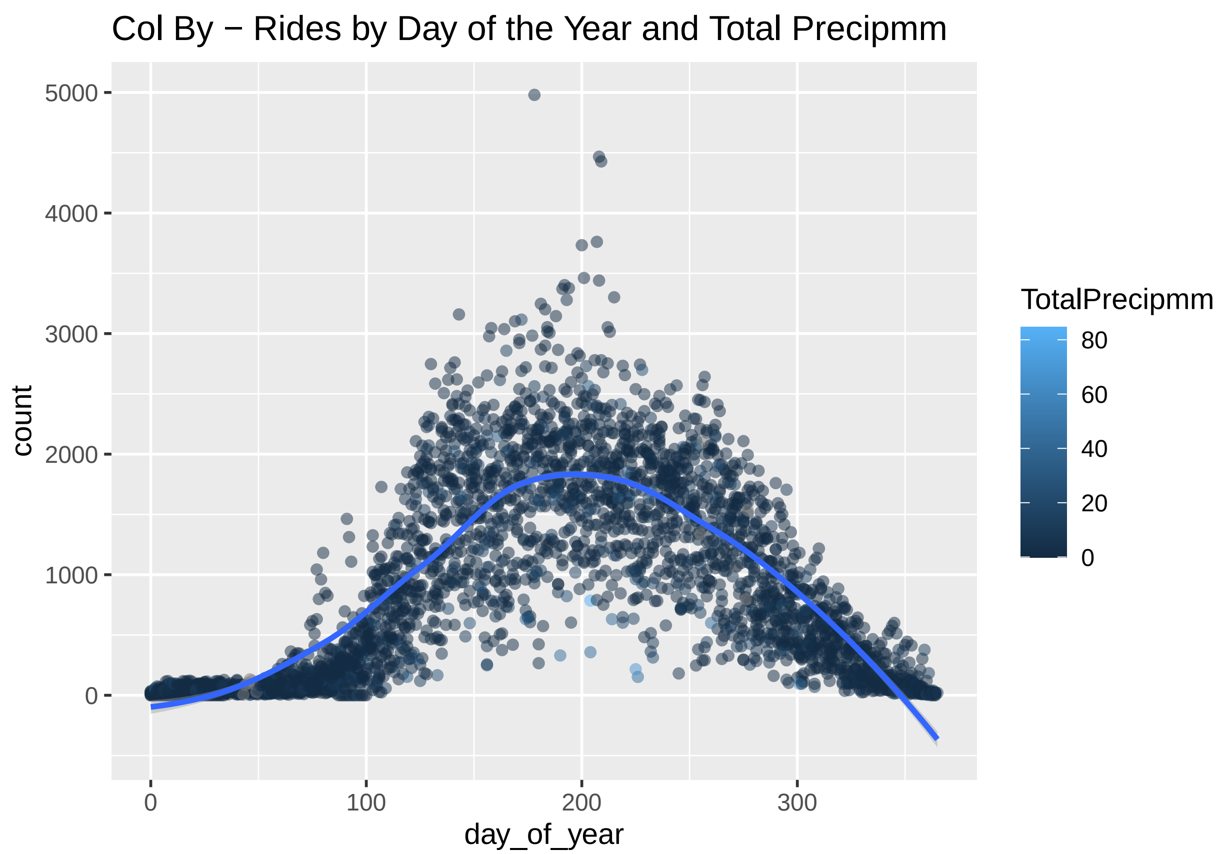
```
## `geom_smooth()` using formula 'y ~ x'
```



When we take the same graph, but colour each dot to represent the maximum temperature on the day in question, we can see that the higher temperature counts (lighter blue dots) are generally found in the summer months and appear associated with more rides. There is however a great deal of variability, some relatively warmer days did not have a very high number of rides recorded that day while there are some colder days in the summer months which did. This may be due to precipitation levels or other factors not seen in this graph.

```
coby %>% mutate(year = year(date), month = month(date)) %>%
  ggplot(aes(day_of_year, count, col=TotalPrecipmm)) +
  geom_point(alpha=.5) +
  geom_smooth(method = "loess")+
  ggtitle("Col By - Rides by Day of the Year and Total Precipmm")
```

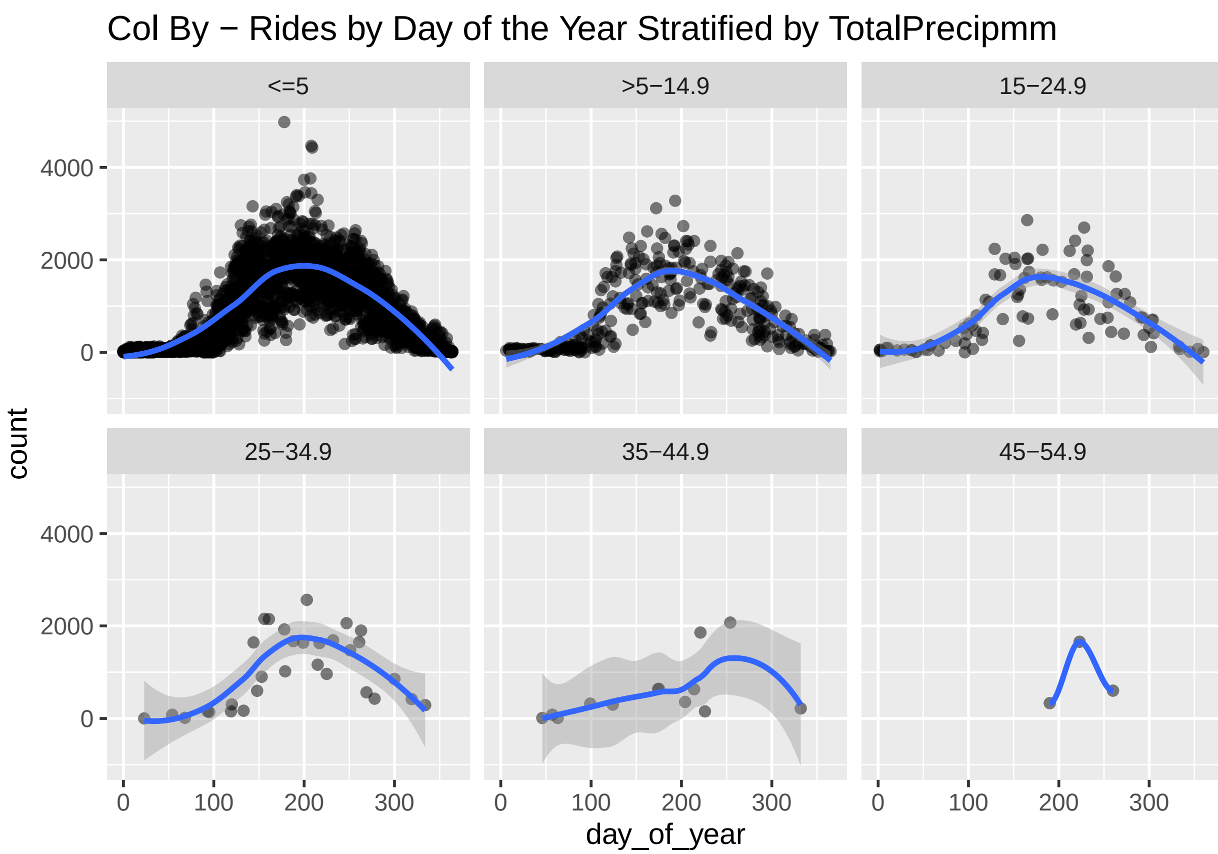
```
## `geom_smooth()` using formula 'y ~ x'
```



Looking once again at the same graph, but this time overlaying precipitation levels, we see that there are a few days of high precipitation appear to be less busy. However, there are also a few days with high levels of precipitation (light blue dots) that had a higher than normal (i.e., above the trend line), however a clear pattern is not easily discernible.

Since it is hard to identify with confidence a clear pattern with respect to precipitation and its impact on bike rides, I decided to stratify data by precipitation amounts to have a closer look.

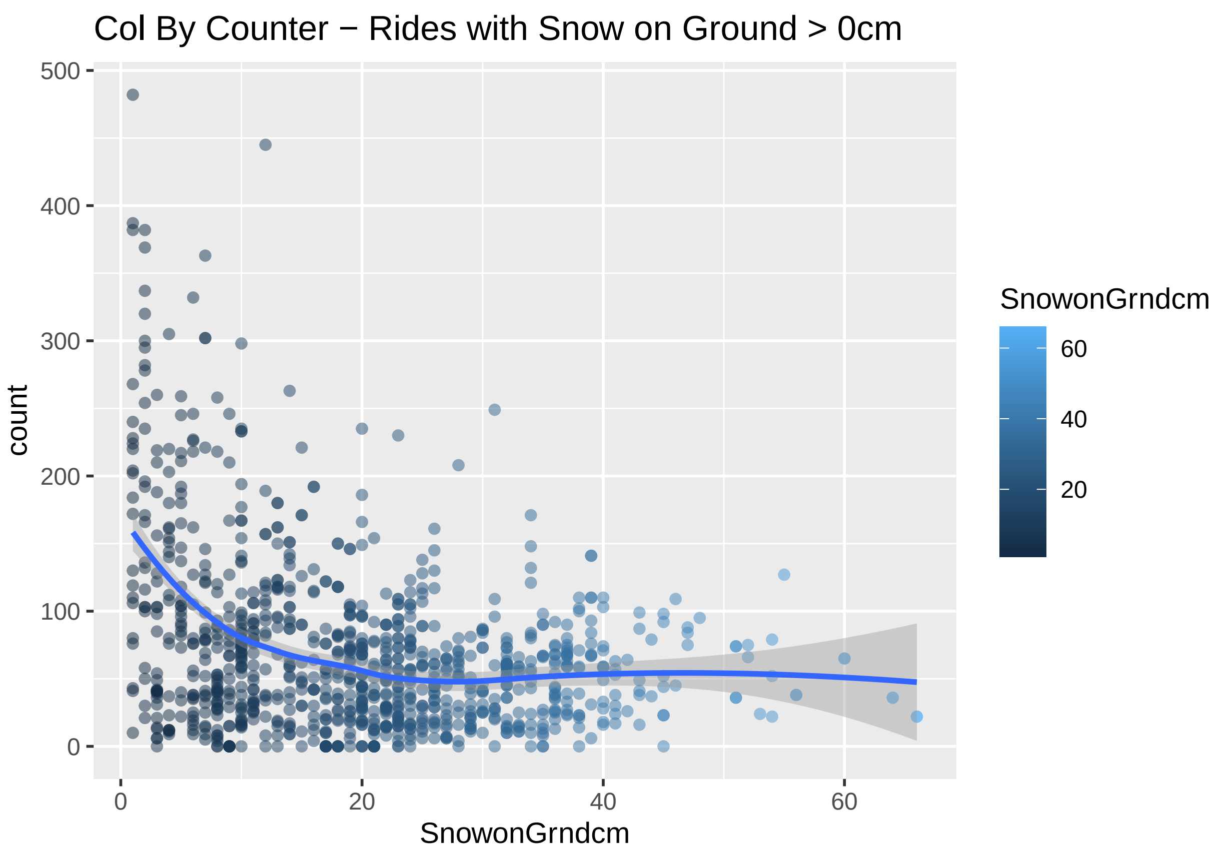
```
#creating a factor with levels
precip_levels <- coby %>%
  mutate(Precipmm_levels = factor(round(TotalPrecipmm/10, digits=0)))

#labelling factor levels
levels(precip_levels$Precipmm_levels) <- c("<=5", ">5-14.9", "15-24.9", "25-34.9", "35-44.9", "45-54.9"

#Visualizing by stratified precipitation levels
precip_levels %>% mutate(year = year(date), month = month(date)) %>%
  filter(TotalPrecipmm<55) %>%
  ggplot(aes(day_of_year, count)) +
  geom_point(alpha=.5) +
  geom_smooth(method = "loess") +
  facet_wrap(facets = Precipmm_levels~.) +
  ggtitle("Col By - Rides by Day of the Year Stratified by TotalPrecipmm")
```



By stratifying the ride entries by the "Totalprecipmm" variable and overlaying a bin smoothing line, we see that only at very high precipitation levels there seems to be an effect on the number of rides per day. The apparent visual difference between the low/no precipitation days and high precipitation days may simply be due to random variation. Day of the year or temperature seem to be stronger indicators - with the warm summer months have more rides even in the rain than during the Ottawa winter months (Dec-March)

```
coby %>% mutate(year = year(date), month = month(date)) %>%
  filter(SnowonGrndcm > 0) %>%
  ggplot(aes(SnowonGrndcm, count, col=SnowonGrndcm)) +
  geom_point(alpha=.5) +
  geom_smooth(method = "loess") +
  ggtitle("Col By Counter - Rides with Snow on Ground > 0cm")
```

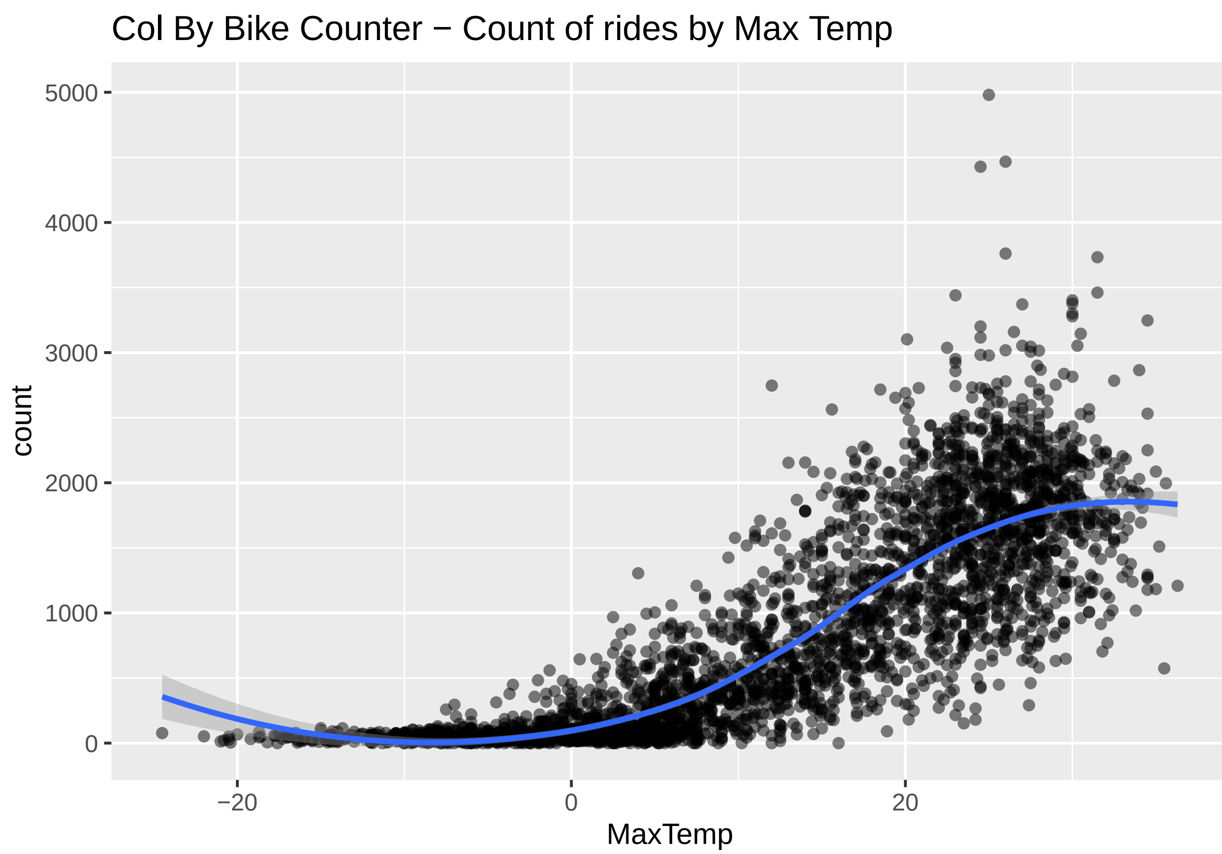
```
## `geom_smooth()` using formula 'y ~ x'
```



Comparing the amount of recorded snow on the ground and the number of rides shows a clear trend that *any* recorded snow on the ground reduces the number of rides significantly, though there are still a handful of hardy Ottawa bike riders that seem to ride their bikes year round along the Colonel By Drive pathway no matter how much snow is on the ground.

```
temp_levels <- coby %>%
  mutate(maxtemp_levels = factor(round(MaxTemp/10, digits=0)))

#labelling factor levels
levels(temp_levels$maxtemp_levels) <- c("-20", "-10", "0", "10", "20", "30", "40","NaN")

temp_levels %>% mutate(year = year(date), month = month(date)) %>%
  filter(maxtemp_levels %in% c("-20", "-10", "0", "10", "20", "30", "40")) %>%
  ggplot(aes(MaxTemp, count)) +
  geom_point(alpha=.5) +
  geom_smooth(method = "loess") +
  ggtitle("Col By Bike Counter - Count of rides by Max Temp")
```

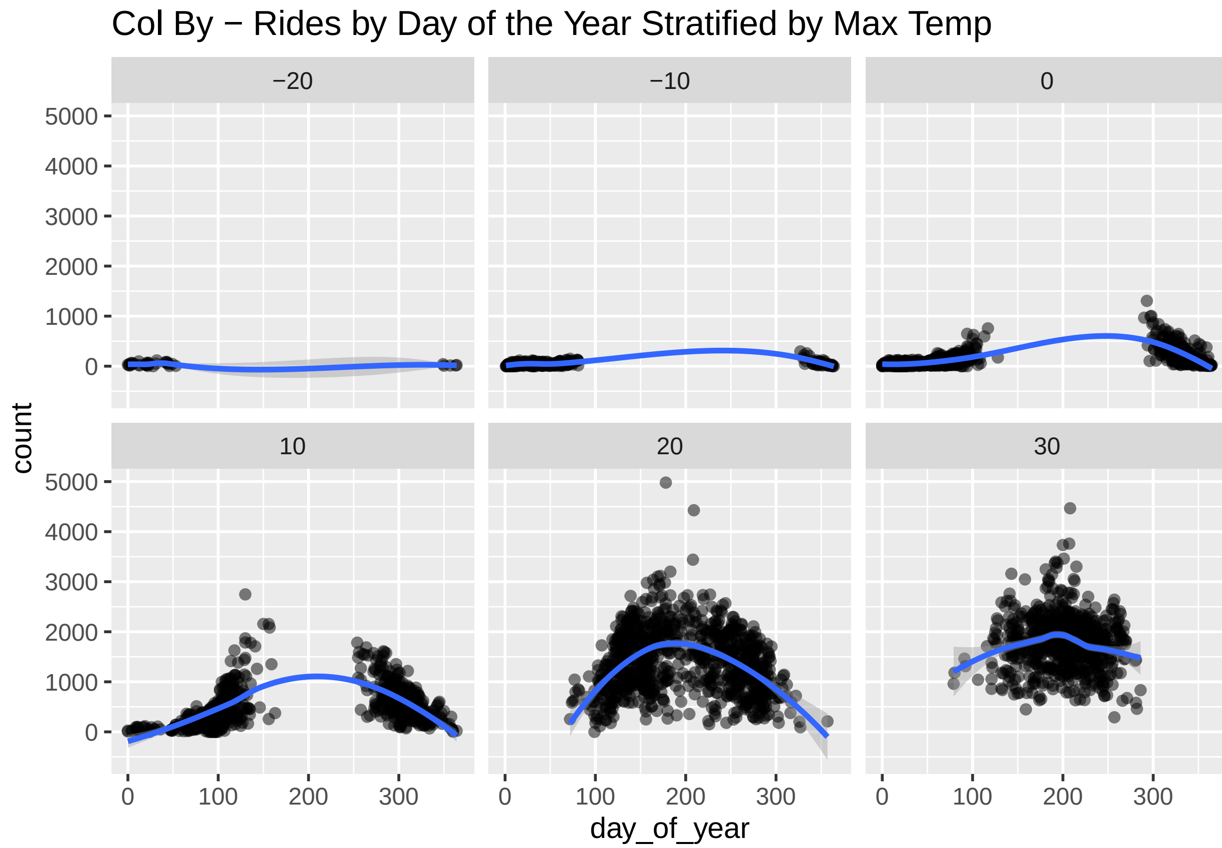
```
## `geom_smooth()` using formula 'y ~ x'
```



Taking a closer look at max daily temperature's impact on rides, we see a smooth relationship between an increase in temperature and more rides however the effect seems to tail off at higher temperatures - presumably too hot for some riders.

```
temp_levels %>% mutate(year = year(date), month = month(date)) %>%
  filter(maxtemp_levels %in% c("-20", "-10", "0", "10", "20", "30")) %>%
  ggplot(aes(day_of_year, count)) +
  geom_point(alpha=.5) +
  geom_smooth(method = "loess") +
  facet_wrap(facets = maxtemp_levels~.) +
  ggtitle("Col By - Rides by Day of the Year Stratified by Max Temp")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Visualizing ride counts over the course of the year stratified by max temperature we can see that warmer days are clustered in summer months and have more rides. However, days in the same broad temperature range still show wide variability.

```
#running a series of pairwise correlations by various predictors
day_of_year <- cor(coby$count, coby$day_of_year, use = "pairwise.complete.obs", method = "spearman")
day_of_week <- cor(coby$count, coby$day_of_week, use="pairwise.complete.obs", method = "spearman")
Max_Temp <- cor(coby$MaxTemp, coby$count, use="pairwise.complete.obs", method = "spearman")
Mean_Temp <- cor(coby$MeanTemp, coby$count, use = "pairwise.complete.obs", method = "spearman")
Min_Temp <- cor(coby$MinTemp, coby$count, use="pairwise.complete.obs", method = "spearman")
Total_Precipmm <- cor(coby$TotalPrecipmm, coby$count, use="pairwise.complete.obs", method = "spearman")
Total_Rainmm <- cor(coby$TotalRainmm, coby$count, use = "pairwise.complete.obs", method = "spearman")
Total_Snowcm <- cor(coby$TotalSnowcm, coby$count, use = "pairwise.complete.obs", method = "spearman")
Snow_on_Grd <- cor(coby$SnowonGrndcm, coby$count, use = "pairwise.complete.obs", method = "spearman")

#create a data.frame with all the Spearman correlations
data.frame(day_of_year, day_of_week, Max_Temp, Mean_Temp, Min_Temp, Total_Precipmm, Total_Rainmm, Total_
```

```
##                        day_of_year day_of_week  Max_Temp Mean_Temp  Min_Temp
## Spear Cor with Rides     0.4036198 0.007505415 0.8704933 0.8704868 0.8439845
##                        Total_Precipmm Total_Rainmm Total_Snowcm Snow_on_Grd
## Spear Cor with Rides      -0.08637889    0.1148095   -0.4195537  -0.7552001
```

While visualizing the data shows that day of the year is a strong predictor looking at pairwise correlations of the various features against rides suggests that using temperature, max temp in particular, and snow on the ground are most closely associated with rides. The Spearman correlation in order to help control for outliers by computing correlation based on the ranks of values, of which there seem to be many in the data.

Higher temps are strongly correlated with more rides (.87), while increased snow on the ground is negatively correlated with rides (-.76). Precipitation, rain or snow, seems less of a predictor. Day of the year is not strongly correlated with rides however the relationship we saw in the visualization was not linear so this would make sense.

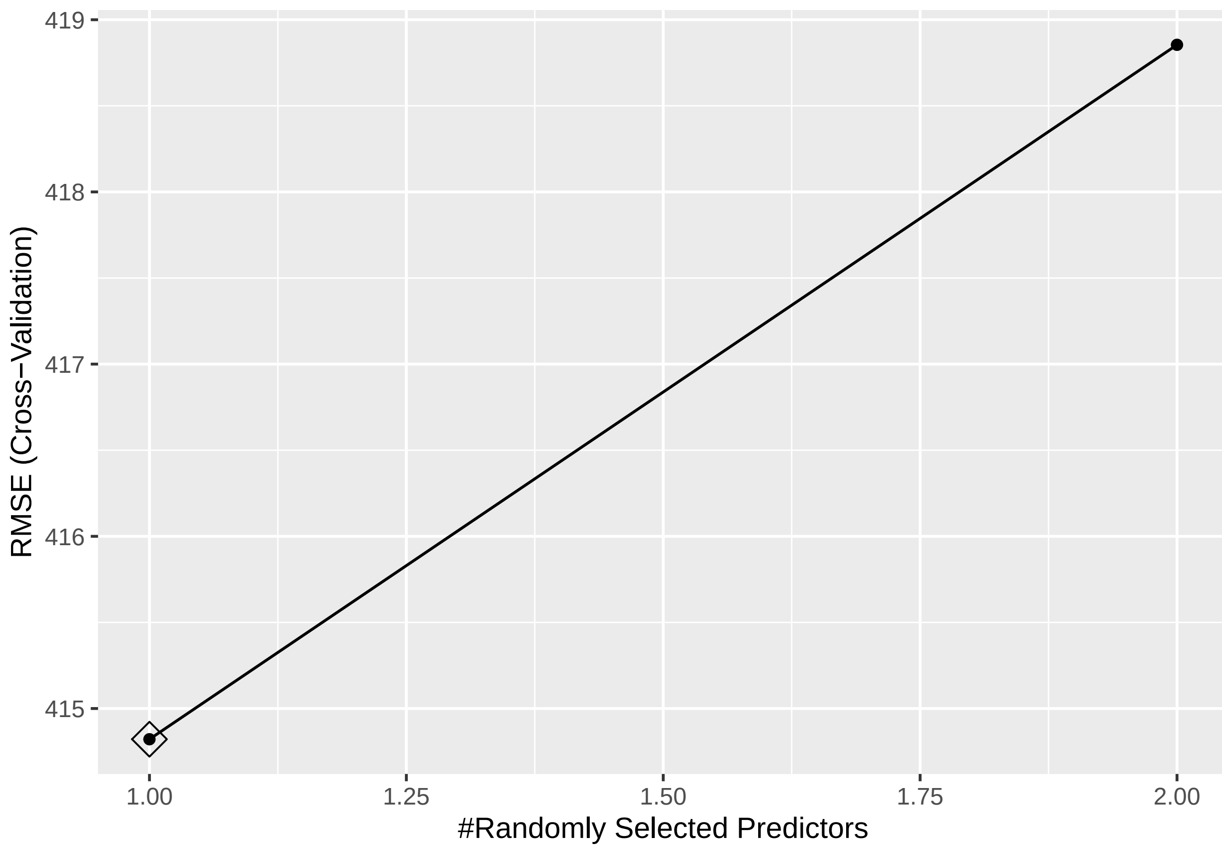
While the causation is not clear (i.e., Is it higher temperature or the fact it is the summer months that drive additional ridership along the pathway?) the pattern with respect to rides is.

Using the insights gained from this exploratory analysis, we will make a Machine Learning model that will focus on using Temperature, Snow on Ground and Day of year as key features to predict number of rides.

MACHINE LEARNING MODEL

For this ML model building exercise, I used the functionality provided in the caret package to leverage a number of different machine learning algorithms. The first step in the ML model building is to partition the COBY dataset into training and test sets. Then I established a training control protocol using 10-fold cross-validation and establishing training grids for tuning for those algorithms that included tuning parameters. The machine learning algorithms I examined included, K Nearest Neighbours (KNN), randomForest (rf), a Naive Bayesian linear algorithm (bayes_glm), linear regression (lm), and two regression tree algorithms (rpart and rpart2). For the first round of trials, I used three features: "MaxTemp", "day_of_year", and "SnowonGrndcm."

```
#select for COBY data and remove some na entries to avoid errors for some algorithms
coby <- dat %>% filter(location_name == "COBY", !is.na(SnowonGrndcm))


#create data partition for training and evaluating of model
set.seed(20201021, sample.kind = "Rounding")
ind <- createDataPartition(coby$count, p=.9, list = FALSE)

train <- coby[ind,]
test <- coby[-ind,]
control <- trainControl(method = "cv", number = 10, p = .9) #establish train control

rf_cv <- train(count ~ MaxTemp + day_of_year + SnowonGrndcm, method = "rf", #randomForest algorithm
               data = train, na.action = na.omit, #omit NAs
               tuneGrid = data.frame(mtry = seq(1:2)),
               trControl = control)
ggplot(rf_cv, highlight = TRUE)
```




```r
rf_cv$bestTune #mtry of 1  #best is about RMSE of 415
```

```
##   mtry
## 1    1
```

```r
bayes_glm_cv <- train(count ~ MaxTemp + day_of_year + SnowonGrndcm, method = "bayesglm", #bayesian algo
                      data = train, na.action = na.omit, #omit NAs
                      trControl = control)
bayes_glm_cv$results #no tuning parameters = RMSE 470
```

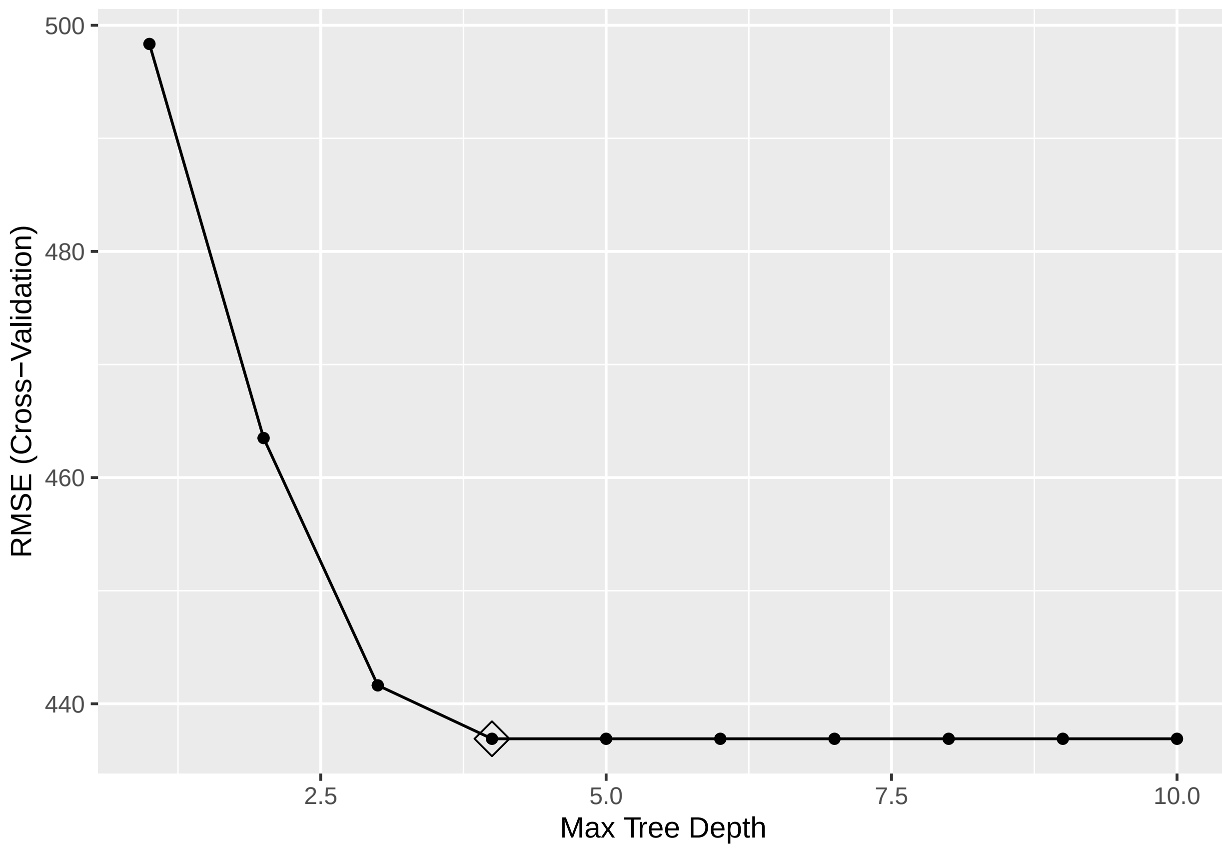
```
##   parameter    RMSE  Rsquared      MAE  RMSESD RsquaredSD    MAESD
## 1      none 470.303 0.6684705 364.1825 20.2464 0.02090554 15.55282
```

```r
lm_cv <- train(count ~ MaxTemp + day_of_year + SnowonGrndcm, method = "lm", #linear regression
               data = train, na.action = na.omit, #omit NAs
               trControl = control)
lm_cv$results #no tuning - simple linear regression - RMSE 470
```

```
##   intercept     RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
## 1      TRUE 470.2278 0.6685722 364.3897 25.16918 0.02268842 18.40105
```

```r
rpart2_cv <- train(count ~ MaxTemp + day_of_year + SnowonGrndcm, method = "rpart2", #regression tree
                   data = train, na.action = na.omit, #omit NAs
```

```r
                    tuneGrid = data.frame(maxdepth = seq(1:10)),
                    trControl = control)

ggplot(rpart2_cv, highlight = TRUE)
```



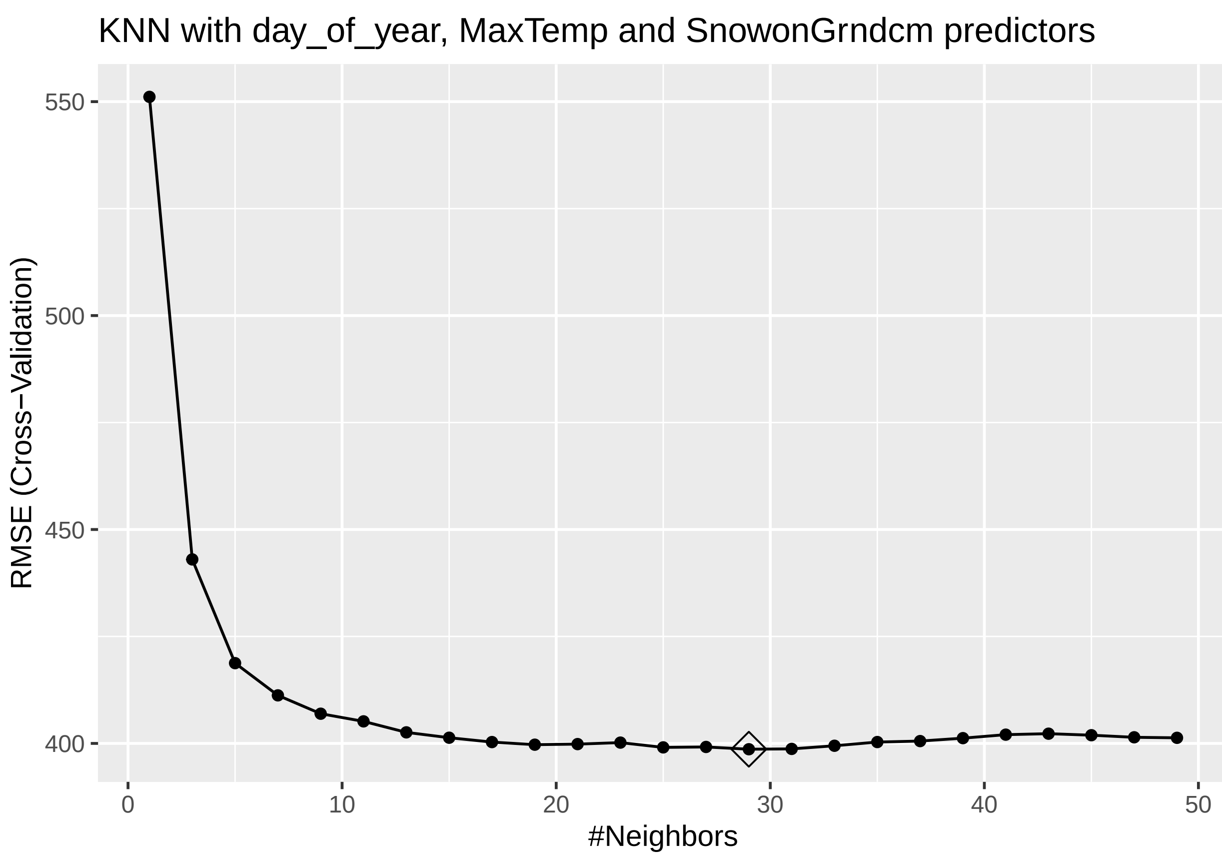

```r
rpart2_cv$results #best tune is Max depth of 4 and RMSE of 436
```

```
##    maxdepth     RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
## 1         1 498.3472 0.6285996 371.5301 16.02767 0.02195494 11.64415
## 2         2 463.4976 0.6786493 326.3923 18.54904 0.02384986 17.14047
## 3         3 441.6298 0.7084448 295.7941 18.32756 0.01740502 11.77473
## 4         4 436.9042 0.7147949 292.7592 17.01434 0.01504777 10.67126
## 5         5 436.9042 0.7147949 292.7592 17.01434 0.01504777 10.67126
## 6         6 436.9042 0.7147949 292.7592 17.01434 0.01504777 10.67126
## 7         7 436.9042 0.7147949 292.7592 17.01434 0.01504777 10.67126
## 8         8 436.9042 0.7147949 292.7592 17.01434 0.01504777 10.67126
## 9         9 436.9042 0.7147949 292.7592 17.01434 0.01504777 10.67126
## 10       10 436.9042 0.7147949 292.7592 17.01434 0.01504777 10.67126
```

```r
knn_cv <- train(count ~ day_of_year + MaxTemp + SnowonGrndcm, method = "knn",
                data = train, na.action = na.omit, #omit NAs
                tuneGrid = data.frame(k = seq(1, 50, 2)),
                trControl = control)
knn_cv$bestTune #K of 29 and RMSE of about 400
```

```
##     k
## 15 29
```

```
ggplot(knn_cv, highlight = TRUE) +
  ggtitle("KNN with day_of_year, MaxTemp and SnowonGrndcm predictors")
```

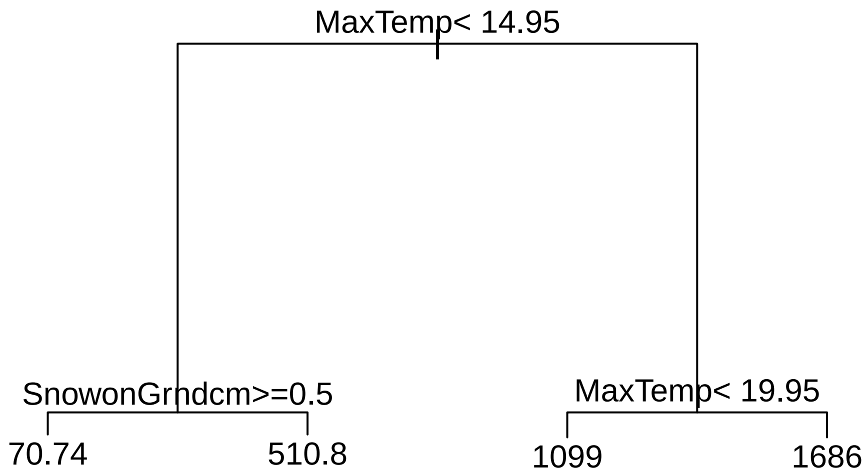



```
#graph of turning based on lowest RMSE for continuous data
```

Intuitively, KNN seemed to be the the best model due to the very different relationships that the three features seem to have on rides this proved to be the case in my first set of algorithm trials. KNN provided the best results with a root mean square error (RMSE) of just under 400 using a k of 25 nearest neighbours. randomForest performed well with an RMSE of 415, but was very computationally intensive and so I decided against using it further. The regression tree algorithm performed well with rpart2 provide an RMSE of 439, while the linear models (lm and bayes_glm) performed the worst at an RMSE of 470 each.

While KNN was the best performing algorithm in terms of predictive power on the training data, the regression tree approach did provide very attractive interpretability and insights into the relationships between the variables in the data.

```
#visualizing a regression tree for this.
fit <- rpart(count ~ MaxTemp + day_of_year + SnowonGrndcm, data = coby)

plot(fit, compress = TRUE, margin = .1)
text(fit)
```



As we can see from the above visualization of a regression tree for the COBY data using the three predictors, the results are easily interpretable based upon temperature and snow on the ground. The day of year varialbe is not used in the model. Given clear pattern from day of year we see in the data, this would seem to impact the effectiveness of the regression tree as a predictor and likely helps explain the better performance of KNN.

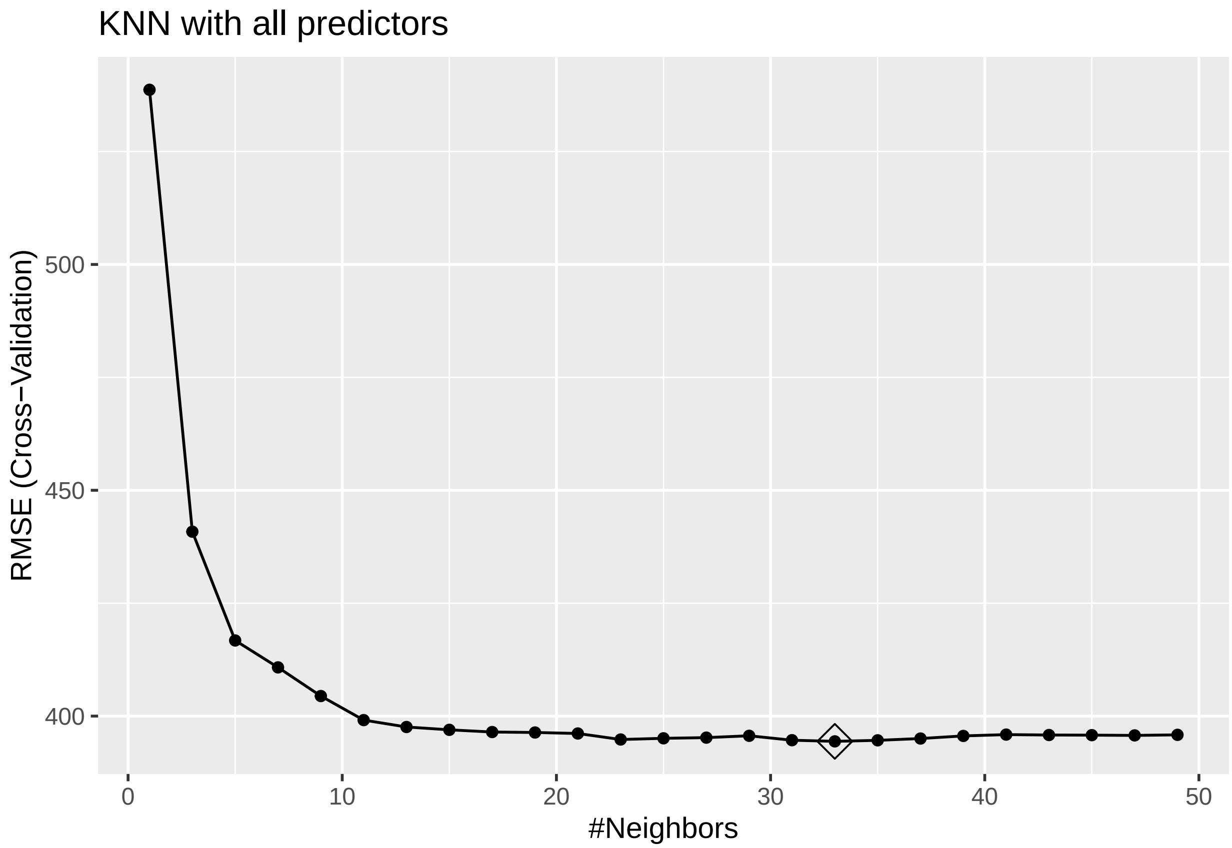
The KNN approach using primary predictors provides for the best results likely because it is better able to handle the non linear relationship it has with ridership. We can easily imagine that regardless of temperature or weather, someone may be more inclined to keep their bike out and go for a ride during the warmer seasons, even on a relatively cold or rainy day, as opposed to when their bike is put away for long periods of time (i.e., winter) and there is a stretch of fair weather.

We will take a closer look to see if we can optimize the results of the regression tree and KNN approaches on the data by looking at more features.

```
knn_all_cv <- train(count ~ day_of_year + MaxTemp + SnowonGrndcm + MeanTemp +MinTemp + TotalRainmm + To
                    data = train, na.action = na.omit, #omit NAs
                    tuneGrid = data.frame(k = seq(1, 50, 2)),
                    trControl = control)
knn_all_cv$bestTune
```

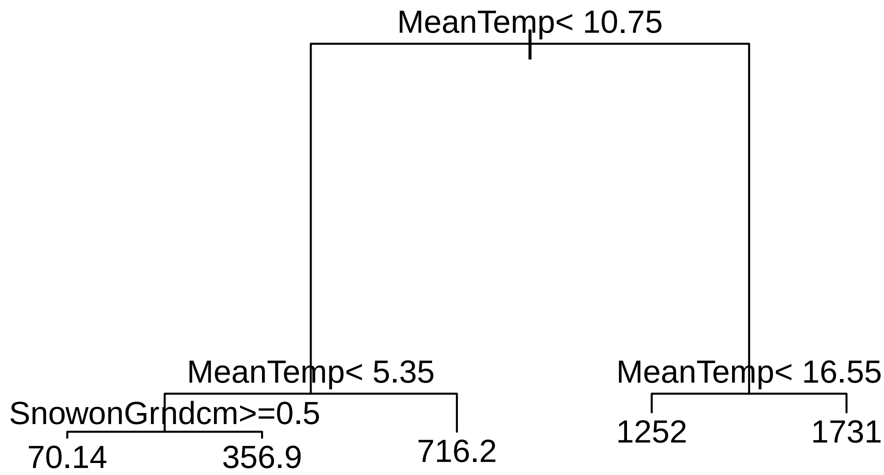
```
##     k
## 17 33
```

```
#K of 33 and RMSE of about 392 - not much of an improvement with precip predictors
ggplot(knn_all_cv, highlight = TRUE,) + ggtitle("KNN with all predictors")
```



Using KNN, we can see there is an improvement using all the features as predictors however it is not a terribly significant one - only by a couple of points.

```
#lets do the same with rtree - all predictors

fit_all <- rpart(count ~ day_of_year + MaxTemp + SnowonGrndcm + MeanTemp +MinTemp + TotalRainmm + TotalS

plot(fit_all, compress = TRUE, margin = .1)
text(fit_all)
```



Looking at a regression tree using all predictors, we can see the results are similar to the previous tree, however there is an extra branch added at lower temperatures and we can see that the algorithm is using mean temperature rather than maximum temperature.

```
variables <- varImp(fit_all)
variables %>% arrange(desc(Overall))
```

```
##                  Overall
## MeanTemp      1.46117412
## MaxTemp       1.41832281
## SnowonGrndcm  1.30323265
## MinTemp       1.19899493
## day_of_year   1.11979614
## TotalPrecipmm 0.05618671
## TotalRainmm   0.00000000
## TotalSnowcm   0.00000000
```

The rpart regression tree has the advantage of being able to provide the importance of variables within the model using a simple function, varImp(). Using this function we see that the mean and max temp features are top predictors followed by snow on ground. day_of_year is a predictor as well but less important than the temperature predictors. Precipitation had very little efffect whatsoever.

```
min <- cor(train$MeanTemp, train$MaxTemp, use = "pairwise.complete.obs")
max <-cor(train$MeanTemp, train$MinTemp, use = "pairwise.complete.obs")

data.frame(min, max, row.names = "Cor with MeanTemp")
```

```
##                           min       max
## Cor with MeanTemp 0.9847495 0.9816308
```

If we look at correlations between min and max temperature and mean temperature, we naturally the temps are highly correlated. Since regression tree suggested MeanTemp is the best predictor we will swap that in for the regression tree and examine it for the KNN approach.

```r
dayofyear <- cor(train$day_of_year, train$MeanTemp, use = "pairwise.complete.obs")
data.frame(dayofyear, row.names = "Cor with MeanTemp")
```

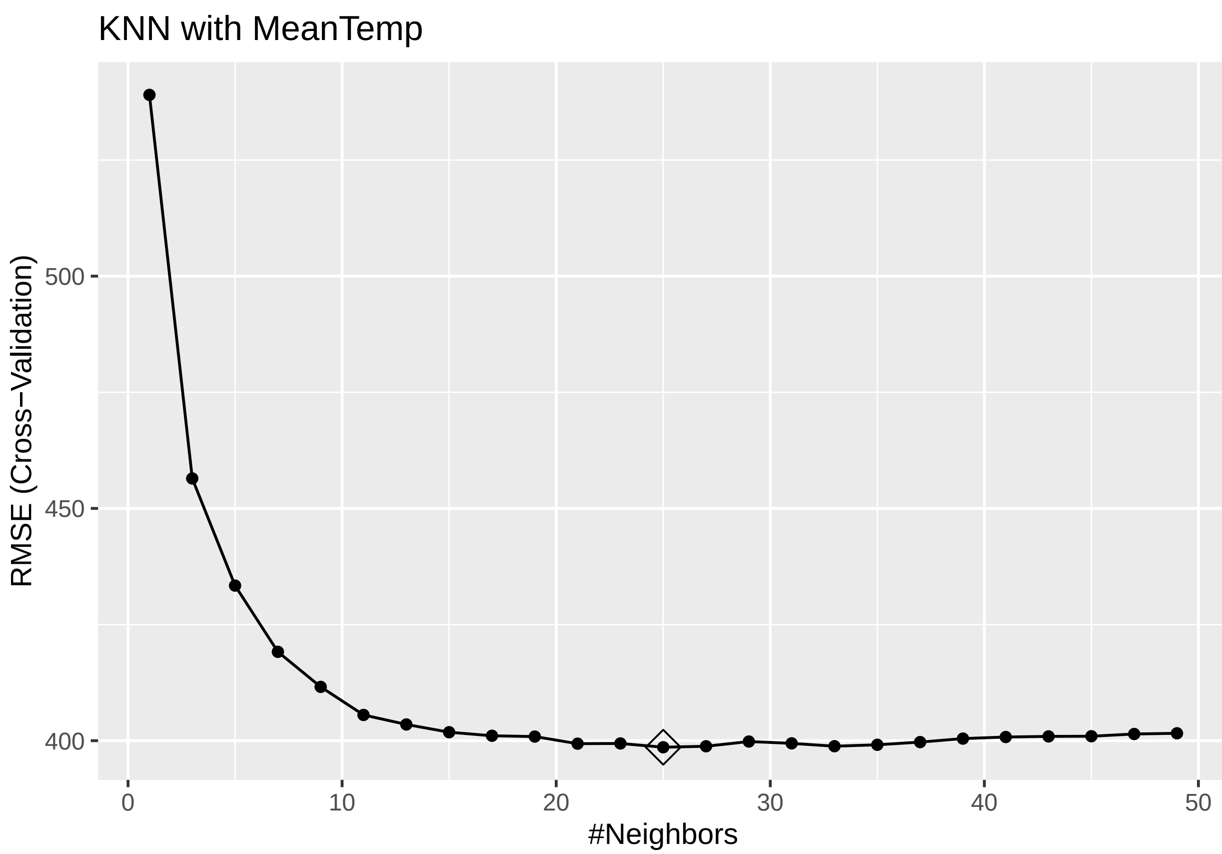
```
##                   dayofyear
## Cor with MeanTemp 0.3691117
```

There is a relatively low correlation with day or year and mean temperature: .369. While weaker in terms of predictive power, the day of year can make a algorithm more robust as it is not highly correlated with temperature overall. Particularly this provide an advantage to the distance based KNN method.

```r
knn_meantemp <-  train(count ~ day_of_year + MeanTemp + SnowonGrndcm, method = "knn",
                       data = train, na.action = na.omit, #omit NAs
                       tuneGrid = data.frame(k = seq(1, 50, 2)),
                       trControl = control)
knn_meantemp$bestTune
```

```
##     k
## 13 25
```

```r
ggplot(knn_meantemp, highlight = TRUE) +ggtitle("KNN with MeanTemp")
```



While close, Maxtemp performs better than MeanTemp using KNN. We will use KNN and the rpart on the test set to get a sense of the algorithms true accuracy.

FINAL MODELS AND RESULTS

```
fit_final <- rpart(count ~ MaxTemp + day_of_year + SnowonGrndcm, data = coby)

rtree_res <- predict(fit_final, newdata = test, type = "vector")
rtree <- RMSE(rtree_res, test$count)

knn_final <- train(count ~ day_of_year + MaxTemp + SnowonGrndcm, method="knn",
                   tune.grid = data.frame(k=25),
                   data = train, na.action = na.omit)
knn_res <- predict(knn_final, newdata=test, type = "raw")
knn_3feat <- RMSE(knn_res, test$count)

knn_final2 <- train(count ~ day_of_year + MaxTemp + SnowonGrndcm + MeanTemp +MinTemp + TotalRainmm + To
                    tune.grid = data.frame(k=33),
                    data = train, na.action = na.omit)
knn_res2 <- predict(knn_final2, newdata=test, type = "raw")
knn_allfeat <- RMSE(knn_res2, test$count)

data.frame(rtree, knn_3feat, knn_allfeat, row.names = "RMSE")
```

```
##         rtree knn_3feat knn_allfeat
## RMSE 441.7062  402.5434    397.9772
```

The best performing model, KNN with all features, has an RMSE of 397.98. The regression tree does not have the same predictive power as KNN but its performance is better than a lot of other algorithms (bayesglm, rf, and lm) and has very attractive interpretability. From a City of Ottawa operations perspective, being able to provide staff a simple heuristic to estimate how many riders one can expect on a given day using the regression tree model would be very attractive and its performance is not that far off (+/-10%) from more sophisticated algorithms.

This model is based on one pathway bike counter in Ottawa. However we can see how well it applies on another winter bike counter in the Ottawa area to validate this approach.

APPLYING MODEL TO ANOTHER BIKE RIDE COUNTER

The Ottawa River pathway bike counter (ORPY) had the second most number of entries of all the bike counters in Ottawa.

```
ORPY <- dat %>% filter(location_name == "ORPY", !is.na(SnowonGrndcm), !is.na(TotalSnowcm))
statistic <- c("min", "1st Qu.", "Median", "Mean", "3rd Qu.", "Max.")
tibble(statistic, summary(ORPY$count), summary(coby$count))
```

```
## # A tibble: 6 x 3
##   statistic `summary(ORPY$count)` `summary(coby$count)`
##   <chr>     <table>               <table>
## 1 min          0.000                 0.0000
## 2 1st Qu.      2.000                93.0000
## 3 Median     867.000               641.5000
## 4 Mean      1224.455               867.5508
## 5 3rd Qu.   2300.500              1559.0000
## 6 Max.      5797.000              4980.0000
```

We can see that on average the Ottawa River pathway has more riders, with higher median, mean and max number of rides than the Col By pathway. However rides appear more skewed with a greater distance between the mean and median.

```
tibble(sd(ORPY$count), sd(coby$count))
```

```
## # A tibble: 1 x 2
##   `sd(ORPY$count)` `sd(coby$count)`
##              <dbl>            <dbl>
## 1            1252.             814.
```

The Ottawa river pathway also has a higher standard deviation, suggesting greater variability.

```
set.seed(20201021, sample.kind = "Rounding")
```

```
## Warning in set.seed(20201021, sample.kind = "Rounding"): non-uniform 'Rounding'
## sampler used
```

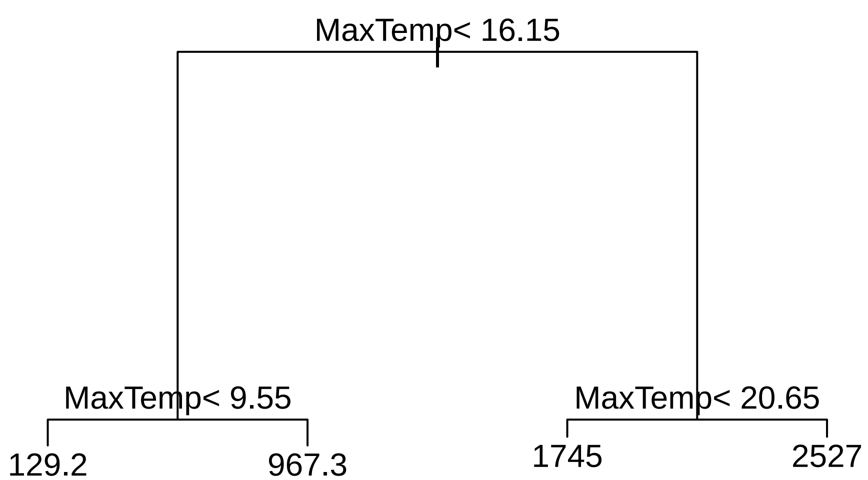
```
ind <- createDataPartition(ORPY$count, p=.9, list = FALSE)#Partitioning ORPY ride data

train_o <- ORPY[ind,]
test_o <- ORPY[-ind,]

fit_final_o <- rpart(count ~ MaxTemp + day_of_year + SnowonGrndcm, data = train_o)#fitting ORPY regress
```

```
rtree_res_o <- predict(fit_final_o, newdata = test_o, type = "vector")
rtree_o <- RMSE(rtree_res_o, test_o$count) #RMSE of 618 - about 200 higher RMSE than the COBY,

knn_final_o <- train(count ~ day_of_year + MaxTemp + SnowonGrndcm + MeanTemp +MinTemp + TotalRainmm + Te
                     data = train_o, na.action = na.omit,
                     tuneGrid = data.frame(k = seq(1, 50, 2)), #best tune from all feature model
                     trControl = control)
knn_res_o <- predict(knn_final_o, newdata=test_o, type = "raw")
knn_o <- RMSE(knn_res_o, test_o$count)

data.frame(rtree_o, knn_o, row.names = "RMSE")
```

```
##       rtree_o    knn_o
## RMSE 618.4653 581.5087
```

Applying the regression tree and KNN algorithms to ORPY results in a higher RMSE score for both algorithms than the results for COBY. However, results are similar in that the RMSE is equal to about about 1/2 a standard deviation of ORPY rides. (ORPY's standard deviation of ride counts was about 1200 and COBY's was about 800.) KNN remains the superior algorithm, however the regression tree is relatively close.

```
plot(fit_final_o, compress = TRUE, margin = .1)
text(fit_final_o)
```



```
#snow on ground is no longer a determinant on the regression tree.
```

Plotting the regression tree for the Ottawa River pathway we can see that snow on the ground is no longer a determinant on the tree. Maximum temperature is the only predictor.

CONCLUSION

The results of applying the model approach to another counter suggests that while the overall approach for forecasting rides based on the weather could still be used across other counters, each counter has slightly different factors that impact the number of rides. Therefore some tuning and selection of optimized parameters would be required for each.

Potential ways to further improve the performance would be to find some way to model the effects of holidays or other significant annual festivals and events would have on bike traffic.

Also, a more complex model using previous year data as a baseline but would adjust for new urban developments that feed traffic to pathways, adjustments to public transit, etc., would likely provide improved accuracy - but is well beyond my capacity to do at this time.