Predicting Train Arrival Status - On Time or Late

Introduction

Adanna Alutu

June 6, 2017

At the beginning of the project, it was hard to come up with a good data to analyze and predict the outcome. Initially I wanted to work on data from my job but after we coudn't see much dependence among the fields that made sense, my mentor Dr. Shmuel Naaman advised me to scout for data from other internet sites he recommended.

Since I take the train most of the time and experienced delay issues many times that has ranged from 10 mins to 2 hours, I became interested in working on transportation data for trains. This is because I want to experience the process of predicting outcomes which is made possible through Data Science. I want to focus on the steps that will make it possible for me and my mentor Dr Shmuel Naaman to predict the arrival times of the train. The possibility of cutting down the delays experienced in waiting for the train no longer seems to be far fetched. My mentor agreed with me and the Septa Train data from Kaggle website was a good option to work on. There were 3 different datasets available to work on but I chose the "on time performance" which I felt has more relevant features, variables and observations and also has sufficient data for the analysis, tests involved.

The variables in the dataset include: 1. train_id 2. status 3. origin 4. direction 5. next_station 6. timeStamp 7. date

Several steps were taken to ensure elaborate data analysis and wrangling. Every bit of the data was maximized. We went beyond using the provided variables by creating new ones from some of the observations, removing unnecessary data and testing with reliable tools to get quality, reliable results that can be tested with any dataset.

It was necessary to take the following steps to ensure that all the combinations, dicing and testing would yield a meaningful interpretation and prediction that will help tell us with high confidence when the train will be late:

I. We first tried to plot charts with the entire data but the plots were too crowded and blurry to make any sense. The scales were distorted with big units affected by the outliers.

II. GGPlot bar charts were used to plot and observe the trends and statistics summary but the dataset was too huge for the charts.

III. My mentor suggested shuffling the data and taking the first 20percent as sample to work on. Using the formula below, the row-wise shuffling was done first before the column was then shuffled:

IV. We used the data to fit in several models which include:

- GGPlot with different combination of the variables.
- Linear regression model which was tested with different variations of the variables to determine their value added to the overall performance.
- CART model with focus on the Regression model because the dependent variable is linear. We also tested with a CART classification method since some of the variables are categorical but based on all the variables in the data, it was more meaningful to stick with the Regression model.
- Random Forest model was also used to check the train data just to determine if it will be a better predictor than the other two models.

Each of these models were implemented because the train dataset contains a mixture of numerical and categorical variables. Converting their types to either numeric or factors wasn't sufficient. To get the benefit of all the variables, it was essential to test these models.

Some data manipulations were done which include: + splitting some of the original variables into separate variables. For example, time stamp variable was split into six variables. year, month day, hour, min, seconds. + Irrelevant variables were removed or set to null so they would not appear in the dataframe used for the predictions. + Some observations from the wkday and day of month variables were converted to variables and they significantly improved the results of the models. The additions however increased the number of variables from 11 to 58. + Units attached to the dependent variable observations were removed to enable convesions to different types and allow plotting with only the observations of the same type. + The dependent variable "status" observations of "on Time" were replaced with "0" using gsub so that all the observations for the variable will match and become easier to manipulate."On time" meant the train arrived as scheduled so it made sense to use "0" to represent no delay.

subtitle:

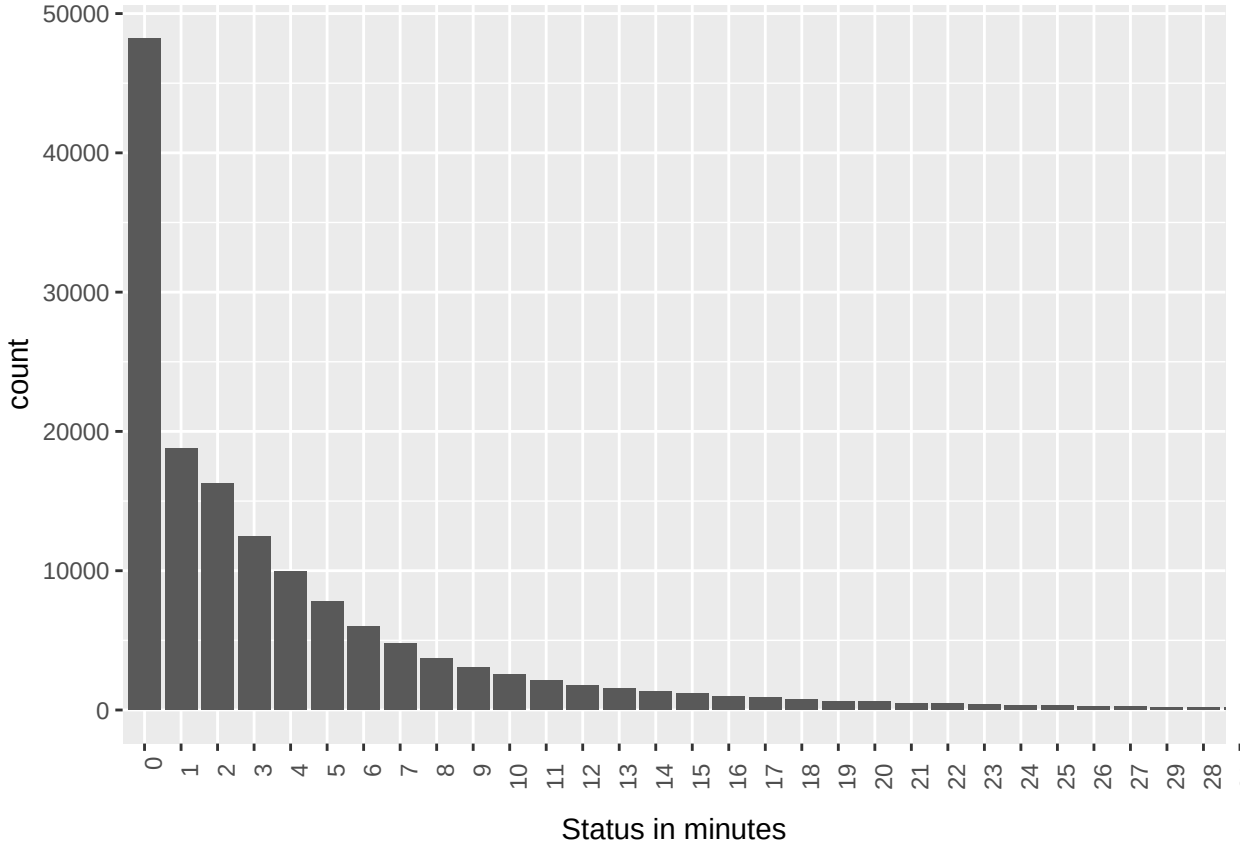
A Peek into some new variables

This section shows the summary of the SEPTA train data and the first few records:

```
##       X.1                X                          next_station    direction
##  Min.   :     1   Min.   :      6   None                :  7019   N:74989
##  1st Qu.: 47150   1st Qu.: 472113   30th Street Station:  6029   S:75020
##  Median : 94032   Median : 942842   Suburban Station   :  5883
##  Mean   : 94116   Mean   : 941520   Jefferson Station  :  5809
##  3rd Qu.:141177   3rd Qu.:1411702   Temple U           :  5596
##  Max.   :188210   Max.   :1882001   University City    :  2865
##                                     (Other)            :116808
##     status                           origin         train_id
##  Length:150009      Doylestown          :10961   5368   :   692
##  Class :character   Airport Terminal E-F: 9573   3542   :   646
##  Mode  :character   Frazer Yard         : 9495   3524   :   622
##                     Elwyn               : 7439   222    :   547
##                     None                : 6994   216    :   541
##                     Roberts Yard        : 6991   5315   :   520
##                     (Other)             :98556   (Other):146441
##    monthday            hour              minute
##  Length:150009      Length:150009      Length:150009
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##     wkday               month
##  Length:150009      Min.   : 3.000
##  Class :character   1st Qu.: 5.000
##  Mode  :character   Median : 7.000
##                     Mean   : 6.843
##                     3rd Qu.: 9.000
##                     Max.   :11.000
##
```

GGplot graphs used initally to see trends and relationships within the datasets.

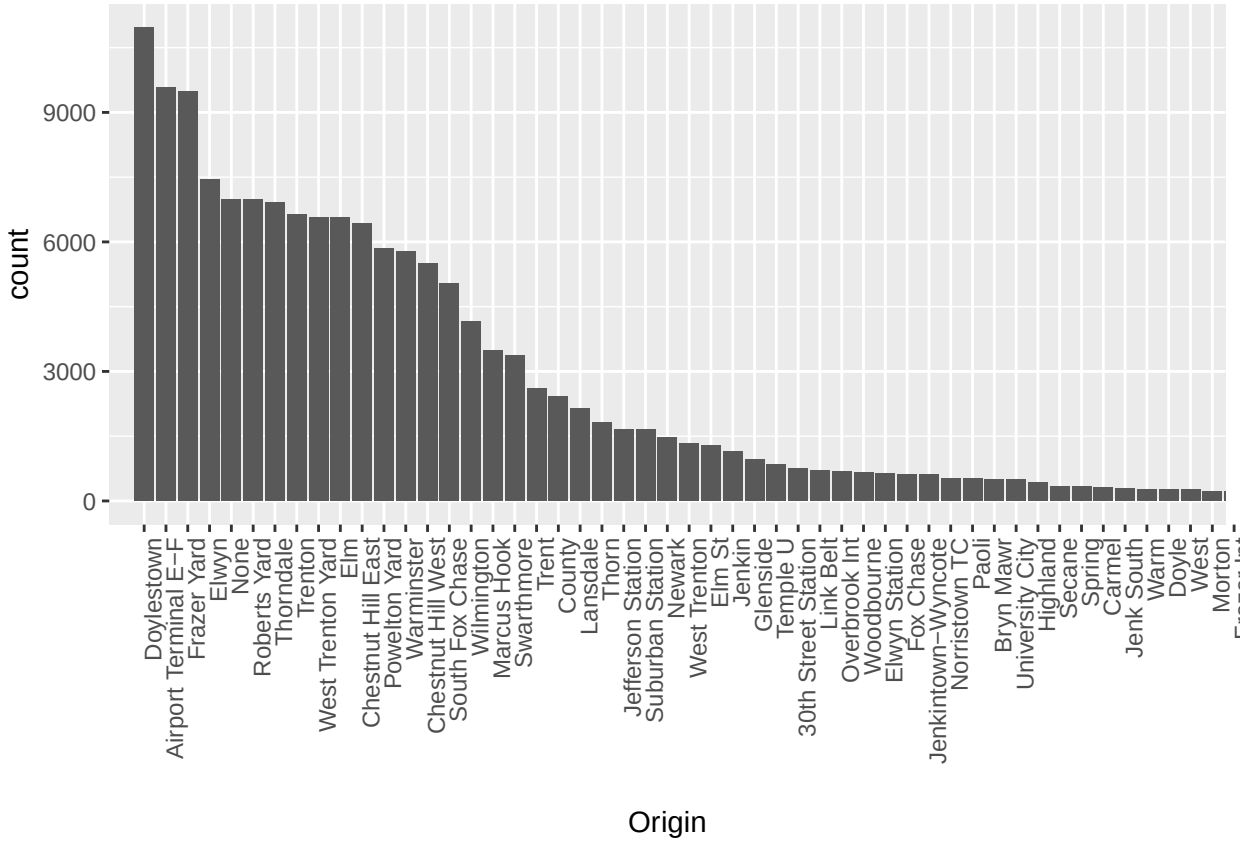
Status variable chart



Status is the name of the dependent variable being predicted in this project. The bar chart shows the frequency and length of the delays experienced by passengers at the train station when the train is late.

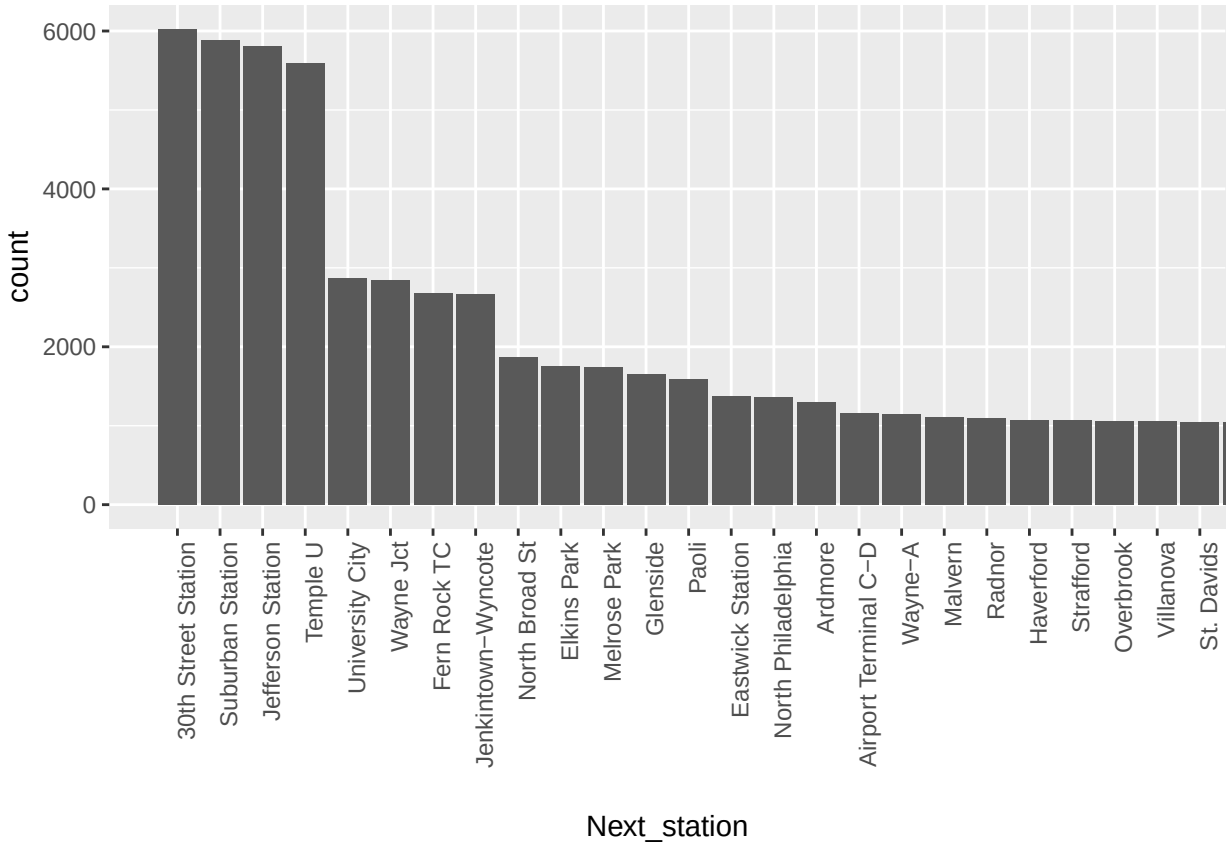
From the chart, we can tell that the trains are on time ~50% of the time and late 50% of the time. In this project, we want to predict when to expect the train to be late and when it will be early to avoid waste of time when possible.

Origin variable bar chart



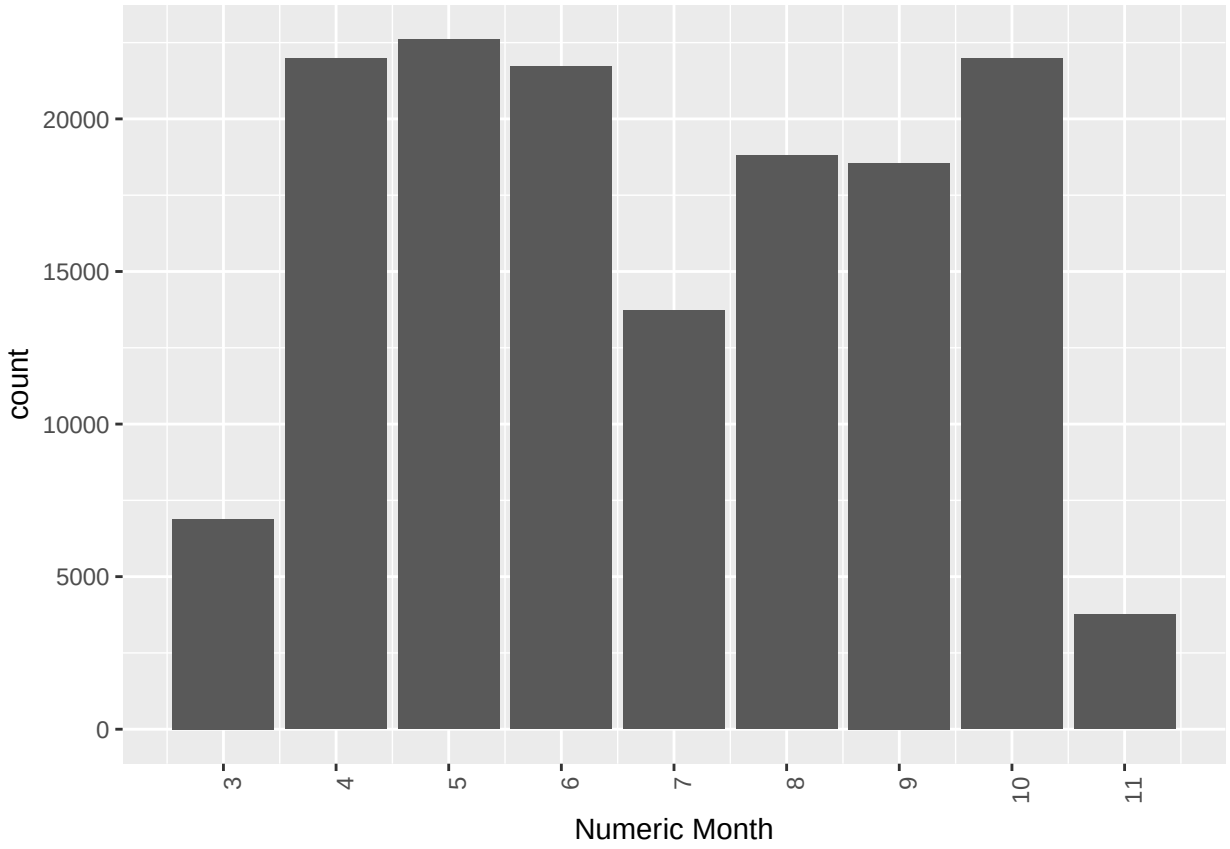
This chart shows the stations that a train's trip can start from and also the number of trains that originate from each station. This also represents the train station where the passenger's journey begings. The Doylestown station is a starting point for most of the trains.

Next Station variable bar chart



NextStation variable represents the next station the train will stop. Most of the trains make a stop at the top four stations in the chart.

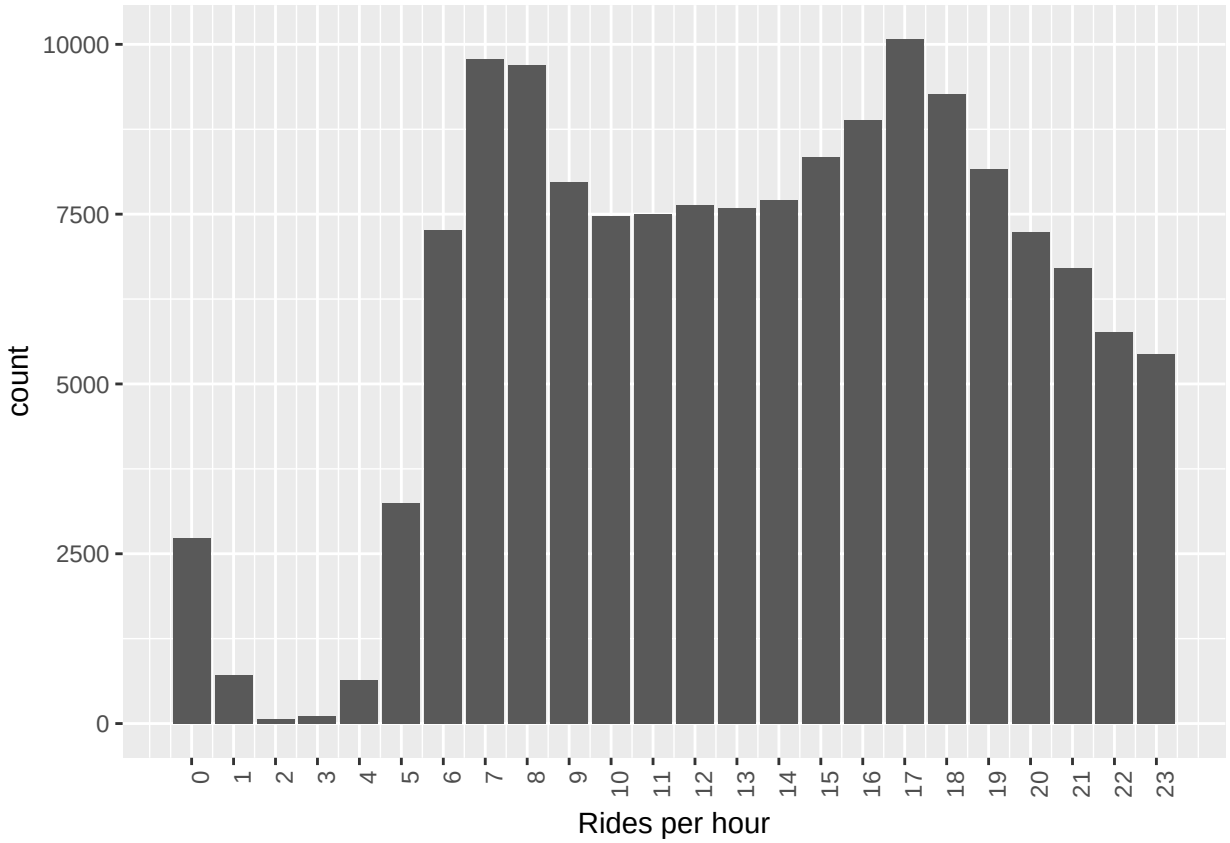
Month bar chart



This chart is used to show the month with the most train rides and the least.

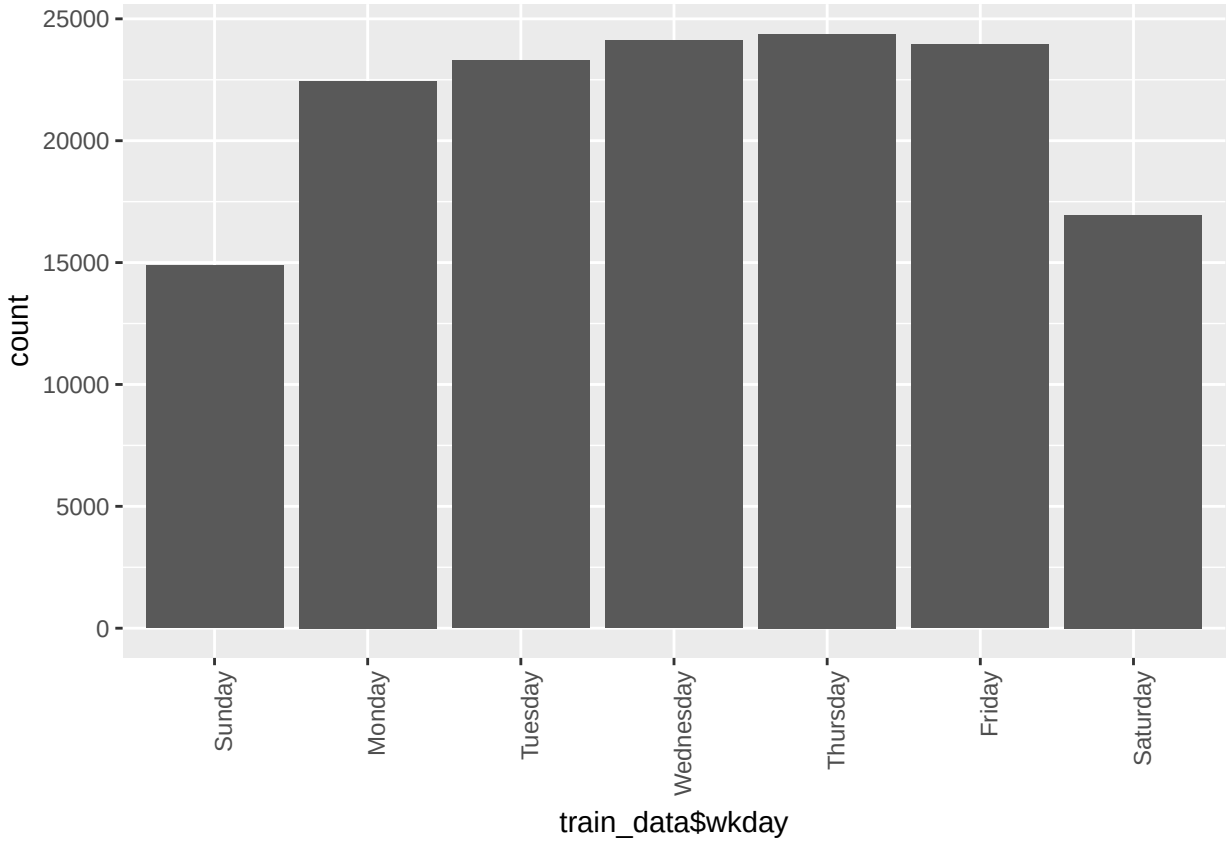
This is one of the new variables improvised by splitting up the timestamp variable.The numbers on the X-axis represent the numbers of real months. The y-axis represent the total number of train rides per month. The highest number of rides occurs in May, June and October.

Hour of the day chart



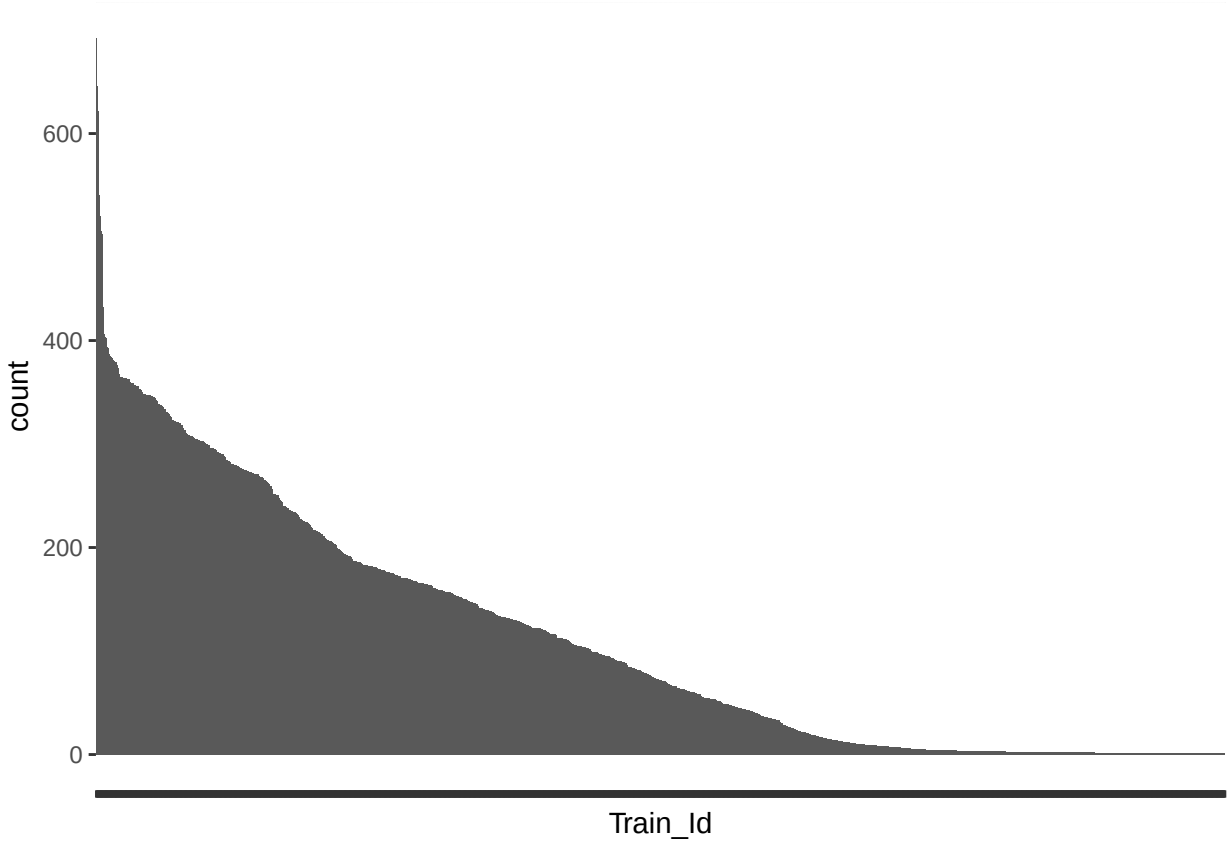
The hour of the day chart shows the time that passengers ride the most.

The Weekday chart



The weekday chart shows the number of trains that run different days of the week. More trains run during the week and fewer trains on the weekends. The busiest day is Thursday.

Train id



This is a chart of all the train ids in descending order. These are all the trains that transported passengers during the period of one year in our dataset.

CART Model /Decision Tree

In this section, the CART model is implemented. The two options considered are Classification and Regression CART models/trees but the regession model is preferred so that the results can be compared with the linear regression model above. It's like comparing apples to apples or oranges to oranges.

I found this site very helpful because they explained in detail the conditions for the variables before a succesful model can be achieved - https://rstudio-pubs-static.s3.amazonaws.com/27179_e64f0de316fc4f169d6ca300f18ee2aa.html. Within this section also a matrix was used to convert the weekday and monthday observations to columns. The purpose is to increase the number of variables that contribute to the status (delay and on time arrivals) of the train.

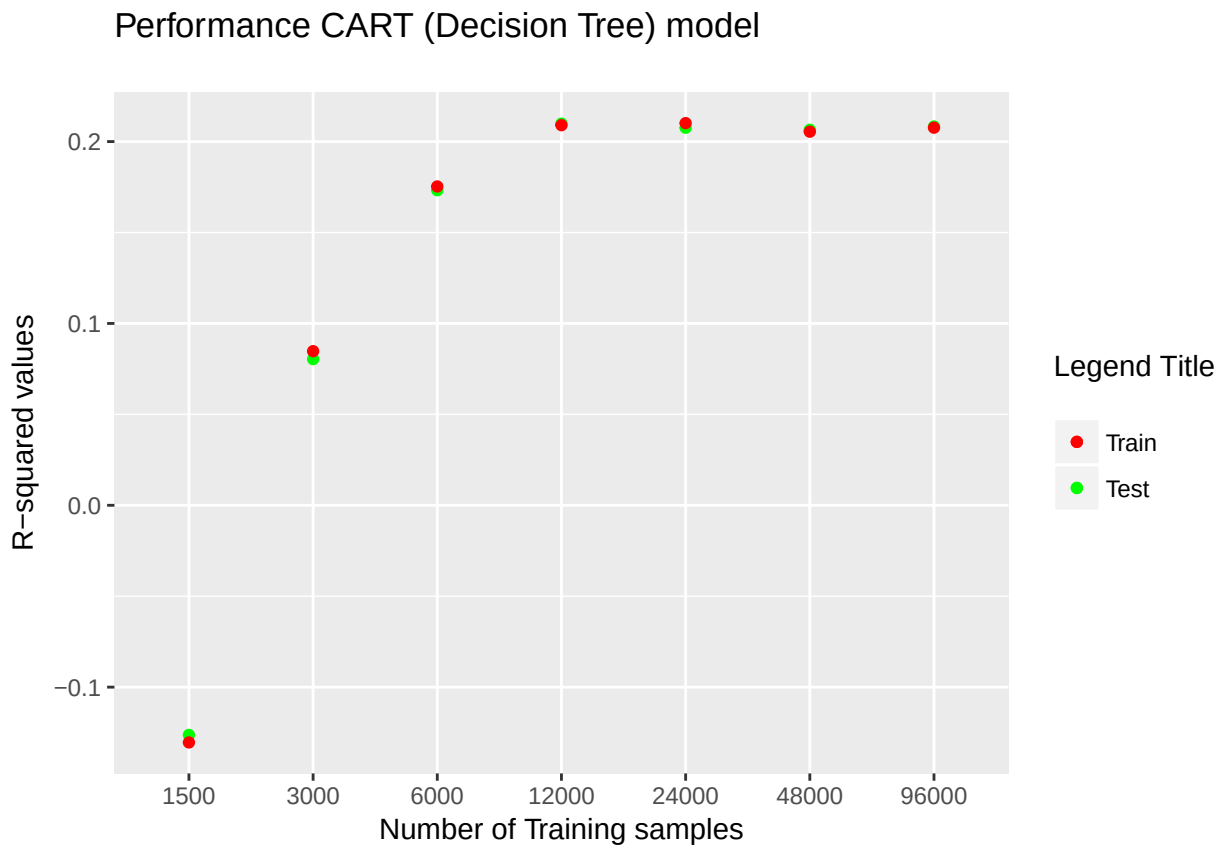
```
## [1] "The performance/r-squared value of the Testing data  0.207282993650837"
```

```
## [1] "The performance/r-squared value of the Training data  0.208606693807776"
```

Check Performance of CART Regression model

```
##      jj      Rsqtrn      Rsqtst
## 1  1500 -0.12627769 -0.13042049
```

```
## 2  3000  0.08038021  0.08474824
## 3  6000  0.17325549  0.17532162
## 4 12000  0.20986960  0.20901182
## 5 24000  0.20760042  0.21014193
## 6 48000  0.20648902  0.20543047
## 7 96000  0.20828764  0.20764966
```



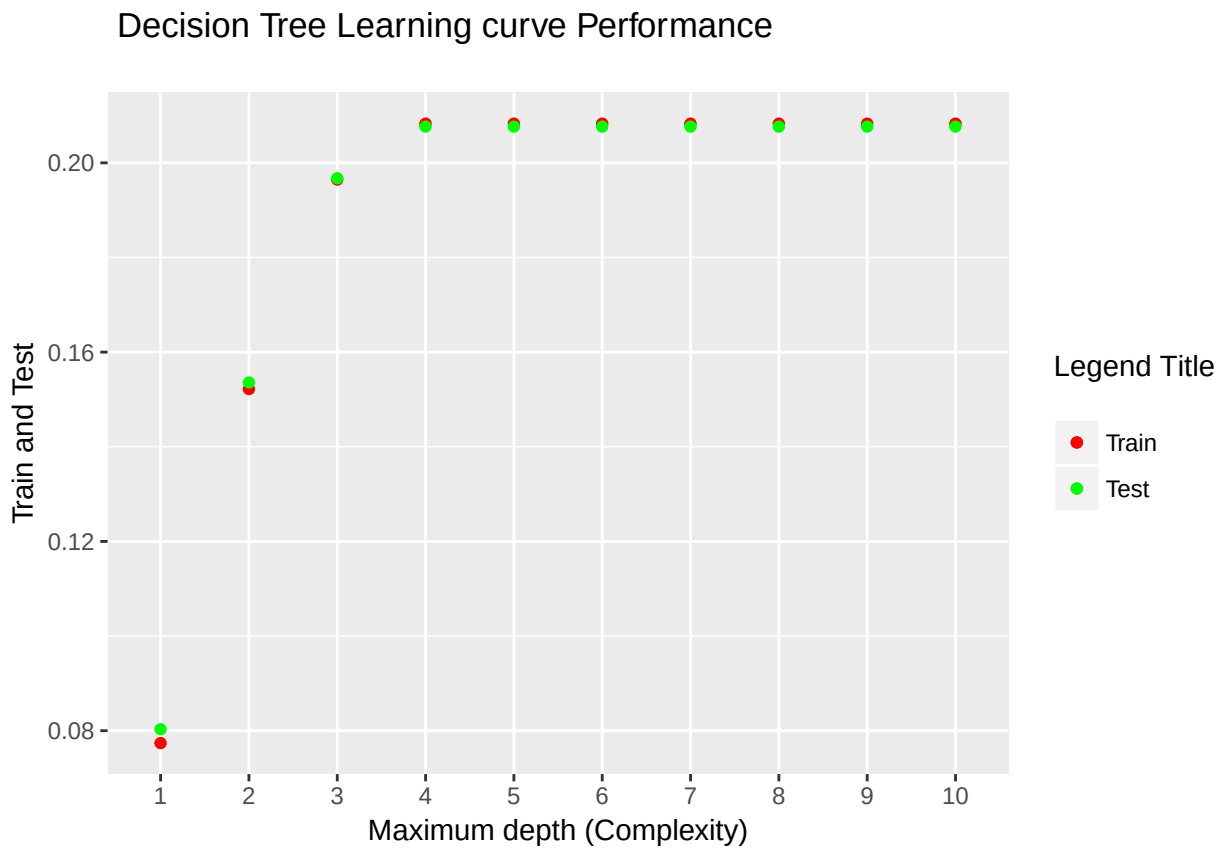
The chart shows the performance of the Training and Test variables at different levels of sample sizes. The result shows that the best performance for both Train and Test set is reached at the record count of 12000 because the performance is at it's peak for both for ~21%.

After this peak, the more data added to the model did not improve the performance rather it decreased in value as displayed in the chart and from the r-squared values.

Checking max depth on the Test set for the CART models

```
##    jj       R2ts       R2tn
## 1   1 0.08031145 0.07738756
## 2   2 0.15358830 0.15218426
## 3   3 0.19674656 0.19646121
## 4   4 0.20764966 0.20828764
## 5   5 0.20764966 0.20828764
## 6   6 0.20764966 0.20828764
## 7   7 0.20764966 0.20828764
## 8   8 0.20764966 0.20828764
## 9   9 0.20764966 0.20828764
```

```
## 10 10 0.20764966 0.20828764
```



As the depth increases, the complexity of the model increases.

The chart shows that the maximum depth performance start leveling around the fourth level for both the Training and testing data. This shows that increasing the complexity of the model or adding more conditions by increasing the maximum depth level more than the 4th level does not add any additional benefit There is no need to increase the complexity further since it doesn't improve the performance of the data.

subtitle:

Random Forest model

Random Forest model implementation. This is the fourth model used in this project to try to come up with a reasonable prediction for the Train delays for any given day. This is the result of the performance at different sample sizes:

```
 jj     rsq_tr        rsqd
```

1 3000 0.04378385 -0.05259217 2 6000 0.12971187 -0.05588601 3 12000 0.07585755 -0.02809949 4 24000 0.05567787 0.02650544 5 48000 0.05327222 -0.04702947 6 96000 0.03164754 -0.02918799

The Training data performed better than the Test data based on the R-squared values. The sample data size of 6000 is sufficient for the model to achieve the best performance for the Training dataset. At this sample size the R-squared values is ~13%.

At 24000 records, the best Test performance is achieved at ~3%.

Check performance on the RandomForest Training and Testing data

The plots show that the Training set performed better than the Test set. **Please see the Random_Forest_chart.pptx for the chart. The chart is not created through the RMD knit tool because it takes too long to generate and sometimes fails because of the duration.

Linear Regression model

The linear regression model is one of the models implemented in this projectvto predict the delays of the train.

```
## [1] "The training data performance is  0.0822650944610929"
```

```
## [1] "The training data performance is  0.0807062477736474"
```

Conclusion from the models

The 3 models used in this project for predicting the Train data are: * The Decision Tree CART model * The Random Forest model * Linear Regression model

The best predictor of the data is the Decision tree with the highest performance value of ~21% for both the Training and Test data. The next better predictor is the Random Forest at 13% for the Training performance and ~ 3% for the Test performance. The Linear regression model's performance is the least for the Training and Test data at ~8%

The overall performances can be argued to be a bit low to be used with confidence but there are other natural factors that could have significant effect that are not included in the dataset like the weather, traffic, accidents, maintenance issues. In a real world project, these should be factored in to get a better gauge of what the actual performance should be.