Title: Predicting Total Accidents in the US

Unit Name: Introduction to Data Science

Unit Code: STAT1003

Word Count:

Nelson Verbrugge: 19450393

Jeevanpreet Singh: 20153780

Neda Adim: 17304249

Contents

[Introduction 3](#_Toc42350923)

[Background 3](#_Toc42350924)

[Problem Statement 3](#_Toc42350925)

[Data Preface 3](#_Toc42350926)

[Usability 4](#_Toc42350927)

[Methods 4](#_Toc42350928)

[Description of Data 5](#_Toc42350929)

[Data Cleaning 7](#_Toc42350930)

[Exploratory Analysis 10](#_Toc42350931)

[Miscellaneous Plotting with Time/Severity variables 10](#_Toc42350932)

[Binary Classification 17](#_Toc42350933)

[Results 22](#_Toc42350934)

[Conclusion 24](#_Toc42350935)

[Bibliography 25](#_Toc42350936)

[Appendix 26](#_Toc42350937)

[R code for Exploratory Analysis 26](#_Toc42350938)

[R code for Predictive Modelling 33](#_Toc42350939)

# Introduction

## Background

With a population of over 300 million people situated across multiple states and locations, the United States faces multiple serious issues in accordance to the daily lives of its citizens. As geospatial data becomes more accessible and in abundant amounts, scientists and organizations have attempted to mitigate these issues and provide solutions at a level of efficiency that has never been seen before. An exceptionally large component of every individual’s life is commute and transportation. With more than 273 million vehicles registered in the US in 2018, and around 6 million new car purchases in the same period (SRD 2020), it is no surprise that road safety and traffic accidents have become a direct consequence from the increasing independent commute and spending patterns of American citizens. The increasing number of vehicles on the road is something that has alarmed the government to utilise geospatial and livestreamed data to research and provide regulations or insights for accidents across the US in the future.

## Problem Statement

The data of road accidents in the US from 2016-2019 will be used in a detailed exploratory analysis, identifying key patterns and characteristics in this specific study. We will also attempt to make predictions to accentuate or benefit the efforts of the government in preventing the accident levels in the country from rising. We are specifically predicting the quantitative parameter of the future of total accidents across the United States.

## Data Preface

The dataset was formed in a Cornell University Machine Learning research by Sobhan Moosavi, Mohammad Hossein Samavatian, Srinivasan Parthasarathy, Radu Teodorescu and Rajiv Ramnath. The data comes from several traffic application programming interfaces, displaying accident records in real time, from February 2016 to December 2019 and being updated at least once a year. Specifically, this data is outputted from two APIs streaming traffic data, in which traffic events from various bodies such as departments of transportation (both national and state), traffic cams, law enforcement agencies, and sensors across the road networks, are broadcasted, adding up to a total of nearly three million unique events across the dataset. This data differs from other traffic data in the sense that it is easily obtainable but more sparse, utilising a deep neutral network (named DAP for Deep Accident Prediction) to contain various data elements (examples listed are traffic events, weather data, points-of-interest, and time). As for the reason the data was collected, the dataset creators were frustrated with the inaccuracy from utilising smaller datasets with limited coverage for real-time purposes, and as such, they set out to create a much larger and easily understandable dataset. Prop.(2020)

## Usability

This data has been used to predict rare accident occasions with significantly improved accuracy compared to accident data that existed before this dataset. Furthermore, the data is utilised for not only predictions, but also identifying traffic information, time, weather and POIs, including possible hotspots and the use of cause and effect rules to further predict car accidents involving any external effects such as rain or environmental elements. Prop.(2020) During this project, we see many similarities in the values for outcome prediction that we see in the public kernels and models that have used this data.

# Methods

To successfully perform an exploratory analysis for the data set “US\_accidents\_May19” both an understanding of the data and visual representation of the data was needed.

The first step of our exploratory analysis involved the determination of the variables and how they were measured. To accomplish this, we searched around for further meta data and found it from the source “Us-Accidents: A Countrywide Traffic Accident Dataset” where the header information was further elaborated on. For some of the variables method of obtaining measurements were quite uncomplicated such as the methods needed to determine temperature, wind speed and weather. However, variables such as visibility, which is measured by an optical sensor determining the amount of light delivered over a certain distance, was not as intuitive. To obtain this knowledge internet searches and books were used.

Missing data was also clear in our data set, for variable such as “Wind\_Chill” and “Windspeed” which lead us to question possible reasons for the missing data. Possible reasons for this missing data include: city’s not having the equipment to record the values, equipment failure on the day of recording and finally, the absence of the requirement to obtain such reading when a accident occurs in the particular city.

We also noticed the inconsistency of the data that was collected over the period of 2016-2019. After initial plotting of the data set it became apparent that we had missing data for most of the years. After redownloading the data set and re encountering the same problem, we then looked to our data file size, which was significantly large for the capabilities of excel, hence the reason for the loss of data. To fix this problem we instead loaded the data straight into R where we were able to obtain and observe all the data.

Now that all the data and variables were largely understood, we then turned our attention to plotting the data to find variables that may be significantly related to our output variable that we were trying to predict. To do this we split our data into data sets, those consisting of numeric values and those consisting of categorical values. We then formed a box plot for the variables that contained numeric values and a bar chart for the variables that consisted of categorical value. From this we then decided the variables that may be suitable for predictive modelling.

It was then decided that linear regression and machine learning models would be fitted to our variables.

Linear regression is a model that attempts to model the relationship between 2 variables by fitting a linear equation. For this method to work two variables, the exploratory and dependent variables must have some sort of relationship. It also must have an increasing or decreasing trend for the linear equation of y=a+bx to be fitted. Where y is the dependent variable and x is the exploratory variable.

Machine learning put in simple terms is the application of artificial intelligence. It allows an algorithm to automatically learn without being programmed and is generally used for large volumes of data. For our machine learning prediction, we have specifically used supervised machine learning. Supervised machine learning uses a training set, essentially past data, to make predictions on the future output values. It can also find errors in the predicted vs actual values and correct the algorithm to make it more accurate (Expert System Team 2020).

Supervised machine learning algorithms that we used are briefly outlined below.

Classification and Regression Tree (CART)- an algorithm introduced in 1984 and essentially an extension of the decision tree (Wikipedia 2020). Classification refers to when the target variable is fitted or categorical and the algorithm is used to identify where the class target variable will fall. Whilst the regression tree refers to a target variable that can take on continuous values (real numbers) (Mehta 2019).

K-Nearest Neighbour (KNN)- was created in 1951 by Fix and Hodges (Peterson 2009). It’s non-parametric, meaning no underlying assumption on data distribution has been made, and lazy learning, meaning it does not use training data points to make any generalisations. The data points are separated into several classes based on the feature’s similarity to the training set (Adi 2017).

Random Forest (RF)- was discovered by Ho in 1955 (Wikipedia 2020). The algorithm randomly creates and merges multiple decision trees into one forest. Increasing the amount of decision models improves its accuracy (Deep AI n.d.).

Support Vector Machine (SVM)-invented by Vladimir Vapnik and Alexey Chervonenkis in 1963 (Wikipedia 2020). The algorithm looks at the data and sorts it into one of two categories. After it has done this, the algorithm outputs a map of the sorted data with the margins between the two categories separated as much as possible. The major task for the SVM algorithm is to calculate which category the new data belongs in (Techopedia 2020).

## Description of Data

The dataset spans, from February 2016 to December 2019 including 3 million accidents happening in United States, which after cleaning decreased to 88,024 records (Sobhan,2019). Each record consisted of 49 attributes such as source of the accident report, a Traffic Message Channel (TMC) code, severity of accidents which numbered 1 with least impact 2,3and 4 with highest impact, start and end time of accidents, longitude and latitude points, the distance between two location on road where the accident happened, description of the accidents, detailed addresses, time zone, weather timestamp, wind chill, humidity, temperature,air pressure, wind direction, precipitation, wind speed, weather conditions (rain, snow, thunderstorm, fog), visibility ,annotations which show the presence of amenities, speed bumps or humps, crossings, give way signs, junctions, railways, roundabouts, stations (bus, train, etc.), stop signs, traffic calming means, traffic signals and turning loops, the time of the day (day or night) based on sunrise/sunset, civil twilight, nautical twilight and astronomical twilight(Sobhan,2019). These attributes are listed below. After cleaning number of variables decreased to 47, as End Longitude and End latitude attributes does not have any values.

Character

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| * ID | * Country | * Weather\_Condition |  |
| * Street | * Time zone | * Wind\_Direction |  |
| * Side | * Source | * Airport\_Code |  |
| * City | * Sunrise\_Sunset |  |  |
| * County | * Civil\_Twilight |  |  |
| * State | * Nautical\_Twilight |  |  |
| * Zip code | * Astronomical\_Twilight |  |  |

String Date

|  |  |
| --- | --- |
|  |  |
| * Description | * Start\_Time |
|  | * End\_Time |
|  | * Weather\_Timestamp |
|  | * Start\_Time |

Boolean

|  |  |
| --- | --- |
|  |  |
| * Amenity | * Roundabout |
| * Bump | * Station |
| * Crossing | * Stop |
| * Give\_Way | * Traffic\_Calming |
| * Junction | * Traffic\_Signal |
| * No\_Exit | * Turning\_Loop |
| * Railway | * Roundabout |

Numeric

|  |  |
| --- | --- |
|  |  |
| * TMC | * Temperature(continuous): |
| * Severity(Discrete) | * Wind\_Chill(continuous) |
| * Start\_Lat(continuous) | * Humidity(continuous) |
| * Start\_Lng (continuous) | * Pressure(continuous) |
| * End\_Lat (continuous) | * Visibility (Discrete) |
| * End\_Lng(continuous) | * Wind\_Speed(continuous) |
| * Distance (continuous) | * Precipitation(continuous) |

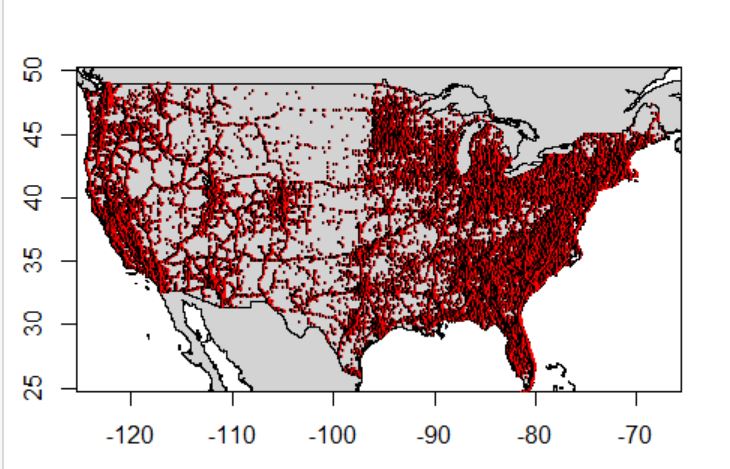


Figure 1. Distribution of accidents in the United StateA close up of a map

Description automatically generated

Figure 2. Accident density in United States

## Data Cleaning

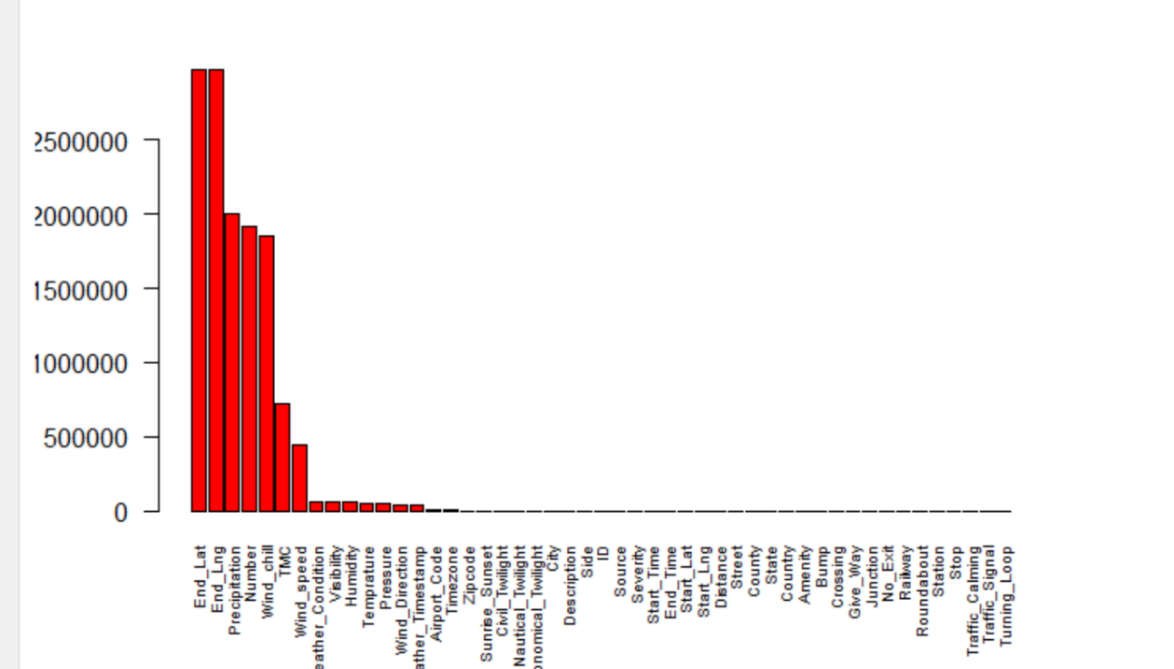
In the data cleaning process, we are going to do the following steps:

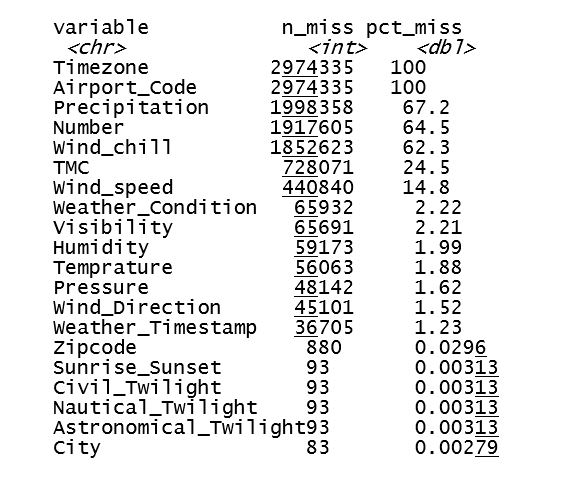
Changing column names which don’t have a proper name.

Converting variable type from character to factor and numeric

Dealing with missing values:

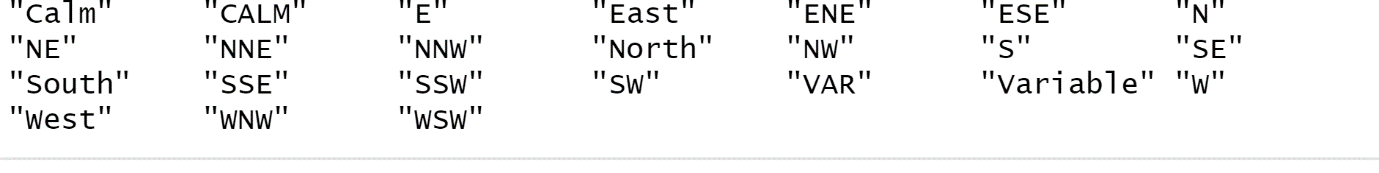
The End latitude and longitude do not have any data, so we removed these two variables. In order to not lose any data, the random forest applied to impute them but because of large dataset it did not work, so they have been removed.





1. Inconsistency in Wind direction variable:

W and West, N and North, E and East, S and South, Variable and VAR, calm and CALM.



There are 24 levels in the wind direction. Removing inconsistency by transforming the above variables, the number of levels decreased to 18.

5- From Start time of accident year, month, date, and hours has been extracted.

6- Normalization:

Rate of accident has been normalized per population.

As the data set don’t include the population of the states and in the data set the abbreviation of state has been used, it has been merged with the excel file which included both abbreviation and full name of states in order to be matched with population data set which has been added in the next step. In the next step the number of accidents per state divided per population and multiplied 100000.

7- Extracting time differences: Time differences has been obtained in order to extract the time spent to clear the accident.

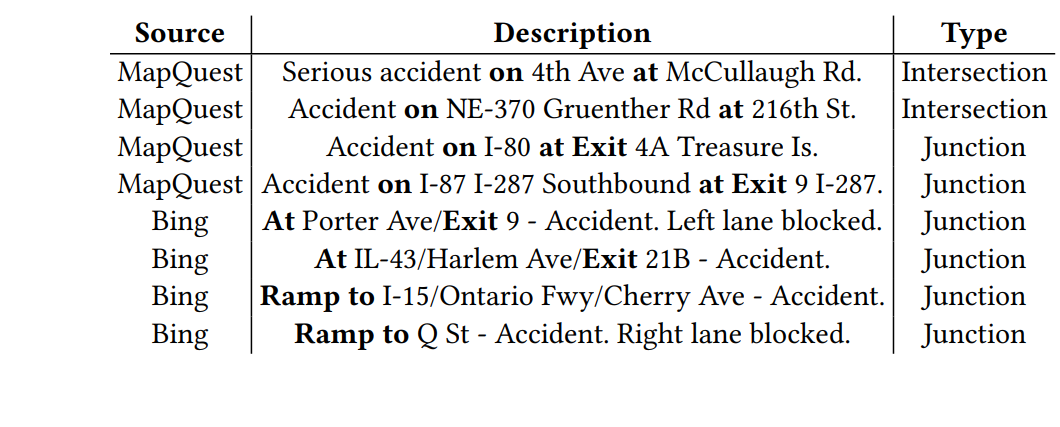
8- Extracting the from Description variable by using regular expression.

Regular expression has been identified in the description of traffic events of type accident which the most common ones are:

Based on the MapQuest patterns, the expression including “. . . on . . . at . . .” correspond to intersection and those include the patterns like : “. . . on . . . at exit . . .”, “at . . . exit . . .” and “…..ramp to” determine as junction(Sobhan,2019).

-Intersection: on . . . at.

-Junction: 1- “at . . . exit . . .” 2- “ramp to…” 3- “on . . . at exit “

‘’

|  |  |
| --- | --- |
| |  | | --- | |  | |
|  |

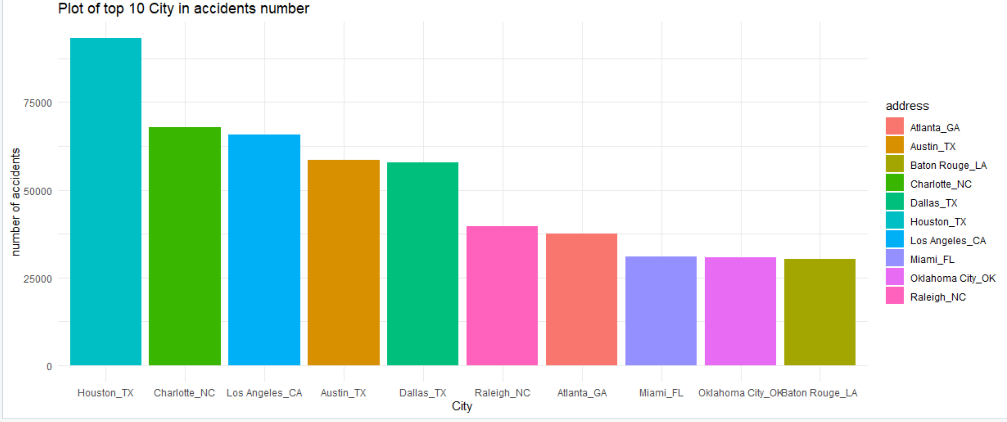
# Exploratory Analysis

For all Figures, they have been created directly from R studio using several codes and functions.

## Miscellaneous Plotting with Time/Severity variables

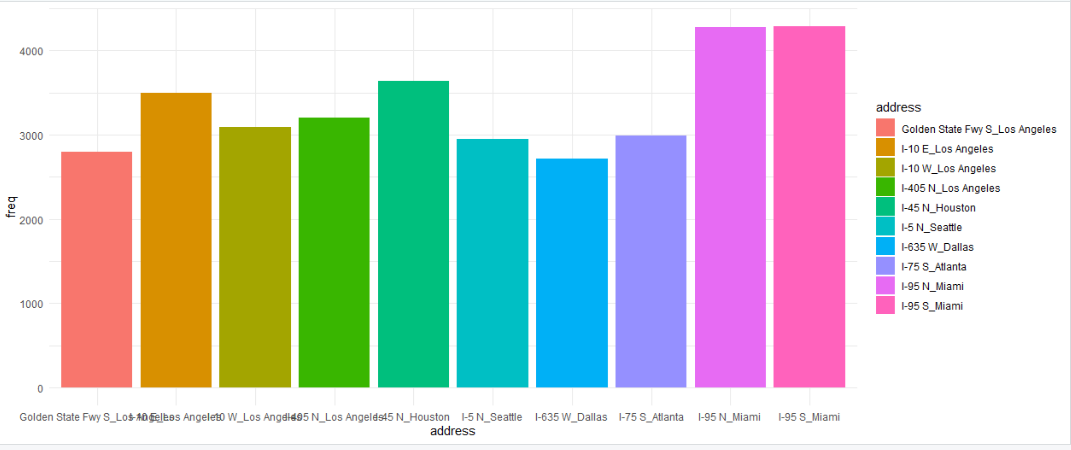
A picture containing drawing

Description automatically generated



A screenshot of a cell phone

Description automatically generated



In our initial analysis, we see the largest number of accidents in that state of California and country of Los Angeles (in California), however, the city with the highest number is Houston, situated in Texas. It gives us an indication that some data is more disperse than others, involving several geospatial and locational variables. This involves major highways, in which Miami I-95 holds the highest number of accidents.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generatedA picture containing fence, computer

Description automatically generated

The number of accidents seems to be increasing every year, with 2019 being around two times as much as only 3 years before in 2016. This can be attributed to the increasing number of vehicles on the roads, and furthermore the large spike in accidents in such a short time can be caused by either a leniency in road regulations or lack of discipline on the road. Months with the highest accidents occur in the Fall season in America (September, October, November). We consider this a direct consequence of a driver’s inability to adapt to the changing conditions, thus the largest number of accidents occurring in October. Accidents throughout the week generally occur in weekdays, possibly due to the large amount of commute as Americans try to get to work. This is confirmed by the adjacent figure of accident hours, where early morning hours of 7-8am and evening hours of 4-5pm see the highest accident rates (going to work in the morning and returning from work in the afternoon). Furthermore, the figure below is conclusive of weekdays having the most accidents and weekends having the least.

A picture containing white, sitting, table, old

Description automatically generated

A picture containing text, map

Description automatically generatedA screenshot of a cell phone

Description automatically generated

A picture containing text, map

Description automatically generatedA picture containing text, map

Description automatically generatedA screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A close up of a map

Description automatically generated A screenshot of a cell phone

Description automatically generated

As the data is analysed in a different approach, accident rate, the data tells a slightly different story than the total accident number figures. South Carolina appears to have the highest accident rate per 100000, as compared to California having the highest total accident rate. This indicates that California’s much larger population affects accident rate less proportionately. The figures above thus visualise and identify the idea that total accident rate does not exactly explain the rate of accidents.

A picture containing screenshot, drawing

Description automatically generated

As we move onto analysing weather data, there is a large number of accidents occurring in clear weather indications, which is unsurprising considering the weather is consistently clear for approximately 80% of the day (this is a logical assumption made in accordance with data). However, excluding clear conditions, there is an extremely large amount of accidents in cloudy conditions, specifically; Mostly Cloudy, Overcast, Partly Cloudy and Scattered Clouds. These all contribute to decreased visibility on the road (NCBI 2016). The physical impact of the climate can be considered negligible considering variables such as rain and snow are not high contributors in total accidents.

A screenshot of text

Description automatically generated

The above Figure displaying the severity of incidents (comparable to magnitude) indicates most accidents at a severity of 2-3, not serious or light, but extremely moderate. Severity relies upon damage to vehicle, injury to person or persons involved, and any individuals who were not involved in the accident (Smoo 2019).

A screenshot of a cell phone

Description automatically generated

Measuring a continuous variable like temperature does not give certain assertions of its impact on traffic accident as it is normally an extraneous variable in these situations. However, we see most accidents at general temperatures of over 70 degrees Fahrenheit.

A screenshot of a cell phone

Description automatically generated

The above Figure indicates that far more accidents occur during the day than at night-time. This is expected as daytime is experienced for 14 hours and there are a limited amount of people travelling during the night or early hours of the morning.

A screenshot of a cell phone

Description automatically generated

An extremely large amount of accidents occurs on the right side of vehicles. Considering the steering wheel is placed on the left side of American vehicles, we could argue that blind spots are quite significant factors in accident occurrence. This can be attributed to either a vehicle’s inability to provide sufficient visual information on a driver’s blind spot (due to dimensions), or the driver’s lack of discipline in not checking their blind spot.

## Binary Classification

A picture containing screenshot

Description automatically generatedA picture containing screenshot

Description automatically generated

A picture containing screenshot

Description automatically generatedA picture containing clock

Description automatically generated

A picture containing clock

Description automatically generated A picture containing clock

Description automatically generated

A picture containing clock

Description automatically generated A picture containing screenshot, clock

Description automatically generated

A picture containing clock

Description automatically generated A picture containing screenshot

Description automatically generated

A picture containing screenshot

Description automatically generated A picture containing screenshot

Description automatically generated

Further exploratory analysis was conducted through binary classification, indicating whether these variables were either present (TRUE) or not present (FALSE) in the accidents. Most of these factors show very insignificant impact on road accidents, except for Crossing, Junctions and Traffic Signal. These are considered high risk variables, as these are present in many accidents across the country. Considering the dataset is in the millions, these variables impact a number of accidents in the 100 000s.

A screenshot of a cell phone

Description automatically generated

This is displayed in a bar plot. Traffic Signal is highly significant in this data as it includes traffic signals and stop lights positioned at intersections and pedestrian crossings to control competing paths of traffic. This purpose of controlling traffic is slightly counter intuitive considering most accidents occur in this situation. While specifically analysing Traffic Signal in the figure below, it is most apparent in accidents with a Severity of 2. There is a slight amount at a severity of three but minimal occurrence at severity 4. In proportion to false values, Traffic signal is not the main factor of severity 3 and 4 accidents.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

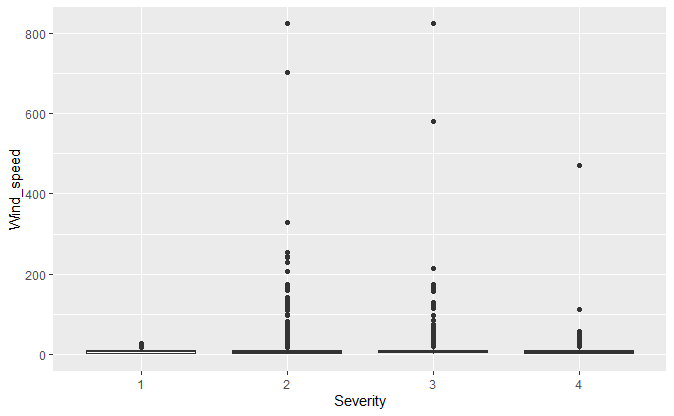
Description automatically generated

We use this bar plot which has been extracted from description variable by regular expression to compare the frequencies of accidents occurred in intersection and junction.

A screenshot of a cell phone

Description automatically generated A picture containing photo, table, boat, parked

Description automatically generated

 A picture containing photo, bird, parked, room

Description automatically generated

Dot plots were utilised to compare discrete values of Severity against environmental elements such as Precipitation, Pressure, Windspeed and Visibility. There are consistent levels of low precipitation, high pressure and low windspeed, apart from a few outlying values. However, there is a large range of visibility when in comparison to Severity. Most accidents occurring at levels slightly above 0 further promotes the notion that low visibility is a high contributor in traffic accidents. This is further confirmed in the figures below. While windspeed remains a consistent factor in accident severity, higher humidity apparent in higher severity. This indicates that lower visibility and decreased responsiveness in the combustion process (both caused by high humidity) are extremely significant in accident occurrence (McNe 2016).

A screenshot of a cell phone

Description automatically generated

A picture containing air, table, man

Description automatically generated

# A picture containing room Description automatically generated

# The final figure above displays the number of accidents against minimum time difference, in other words, the duration between the time of the accident and the affected vehicles being cleared off the roadways. A majority of accidents occur with a large time difference of 360, thus indicating that traffic accidents are not only dangerous for drivers and produce expensive damage to vehicles, but also require a large amount of time before roads are safely opened and traffic is allowed to flow as usual. This should be enough evidence for governments to adjust their focus adequately into mitigating accidents which will directly reduce traffic congestion on major roads.

# Results

In the process of data modelling, it was decided to omit linear regression models from this data. Although the method is quite simple and effective, while carrying out this process, the residuals did not approach a normal distribution (which was required for a good fit) and while checking the homogeneity of the data, a scatter plot displaying residuals against the fitted values showed minimal spread and was apparently linear in appearance. As such, the following models were compared methodically and critically with values that significantly align with what we expect.

Analysing the machine learning algorithms; Classification and Regression Trees, k-Nearest neighbour, Support Vector Machines and Random Forest it was found that Support Vector Machines algorithm was the most accurate. This was determined by running the training set with each of these algorithms and determining the RMSE values relative to each other.

RMSE values were used to compare the algorithms as the output we were trying to predict in the regression was numerical. In contrast if the classification were categorical, we would have used accuracy to compare the algorithms.

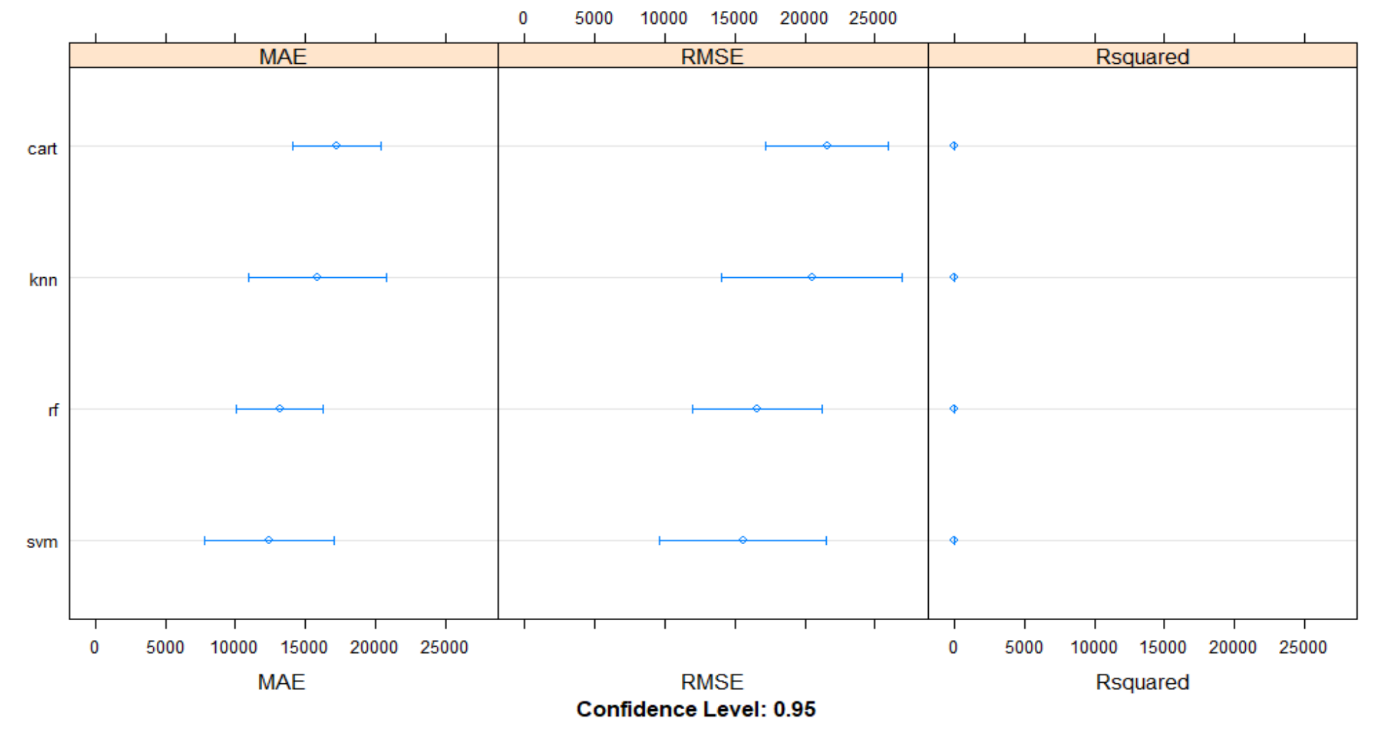


Figure 1

As can be seen in figure 1 the SVM algorithm has the lowest RMSE value. Additionally, to this the mean absolute error is considerably lower than the rest, and the Rsquared value is high compared to the other algorithms suggesting the model produced is more efficient than the others at modelling the data. Although the exact Rsquared value is not determinable in the figure, the function “summary” was used in the r code to extract the exact value of 0.5196. The numbers have been determined using a 95% confidence interval meaning the 95% of the time the train parameters will contain the true variable.

When fitting the SVM algorithm we also looked at which variables were the most influential in determining the model. As you can see in figure 2 the most influential variable in producing this algorithm is humidity. This is somewhat expected as the humid conditions may lead to a higher proportion of people suffering from heat exhaustion or heat stroke, which may lead to their judgement when driving a car to be compromised. Additional humidity may also add precipitation on the road making it more slippery.

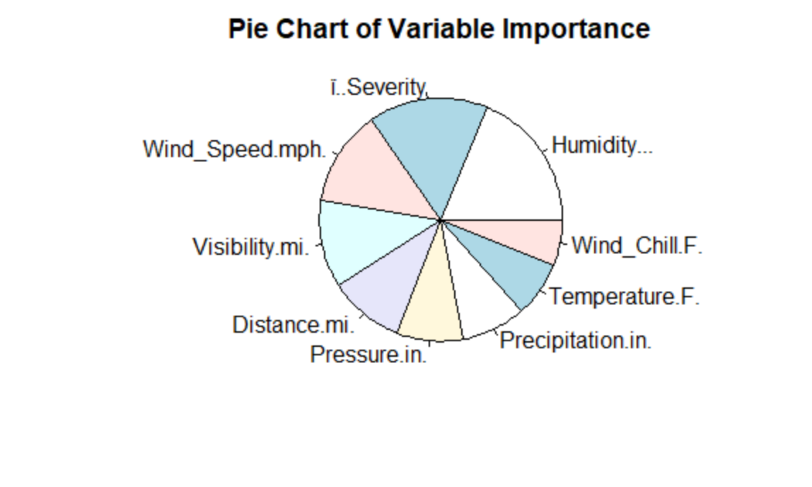


Figure 2

On completion of the built algorithm we then ran it against the test data set. With the results being fairly pleasing considering the difficulty of the task we were trying to achieve. Figure 3.

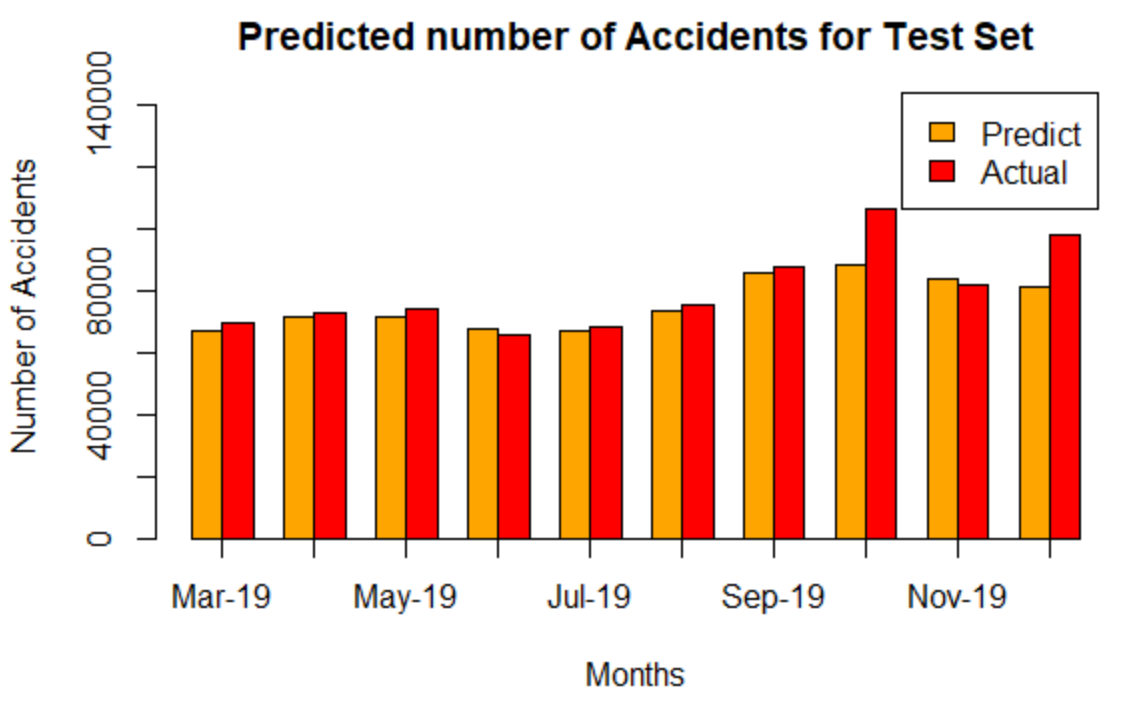


Figure 3

From these results we then went on to evaluate the RMSE. Once calculated the RMSE value we saw that it had decreased its value from the original training set, hence the model was becoming more accurate. This surprised us we thought the RMSE value would relatively stay the same. Figure 4.



Figure 4

# 

# Conclusion

In conclusion, the results obtained from the support vector machine algorithm were pleasing. As stated previously, returning a RMSE value of 8181.213 which may not be significant compared to other predictive models predicting other scenarios, but in our case considering the level of difficulty involved in predicting the number of accidents it was a considerable effort. To improve the result of the predictive model in the future we could modify the code to incorporate both categorical and continuous data, collect more data from the accidents.eg. If it was a public holiday, and finally having a more complete data set.

The counterpart of our investigation with predictive modelling, exploratory analysis, provided sound evidence of environmental and road feature elements having a critical influence on accidents across the US. There were multiple instances of specific trends in traffic accidents in the presence of certain attributes, whether they be climactic conditions outside of human control or design flaws within the road system itself. Improvements could be made if the dataset included more insight in terms of geospatial data; providing more categorical variables than listed and further intricate detail of which components are constitutively involved in each elements listed (eg. what kind of facility is mentioned in public amenities, force of impact, model and make of car, etc.)

# Bibliography

Adi, Bronshtein. 2017. *A Quick Introduction to K-Nearest Neighbors Algorithm.* April 12. Accessed May 23, 2020. https://blog.usejournal.com/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7.

Deep AI. n.d. *Random Forests.* Accessed May 22, 2020. https://deepai.org/machine-learning-glossary-and-terms/random-forest.

Expert System Team. 2020. *What is Machine Learning? A definition.* May 6. Accessed 5 20, 2020. https://expertsystem.com/machine-learning-definition/.

Mehta, Anukrati. 2019. *A Beginner’s Guide to Classification and Regression Trees.* Janruary 3. Accessed MAY 22, 2020. https://www.digitalvidya.com/blog/classification-and-regression-trees/.

Peterson, Leif E. 2009. *K-nearest neighbor.* February 21. Accessed May 24, 2020. http://www.scholarpedia.org/article/K-nearest\_neighbor.

Techopedia. 2020. *Support Vector Machine (SVM).* Accessed May 24, 2020. https://www.techopedia.com/definition/30364/support-vector-machine-svm.

Wikipedia. 2020. *Decision tree learning.* May 18. Accessed May 24, 2020. https://en.wikipedia.org/wiki/Decision\_tree\_learning.

—. 2020. *Random forest.* May 2. Accessed May 24, 2020. https://en.wikipedia.org/wiki/Random\_forest.

—. 2020. *Support vector machine.* May 18. Accessed May 24, 2020. https://en.wikipedia.org/wiki/Support\_vector\_machine.``

Sobhan, M. (2019, Jun 12). A Countrywide Traffic Accident Dataset. Retrieved from

https://smoosavi.org/datasets/us\_accidents

Smoo. (2019). US-Accidents:A Country Wide Traffic Accident Dataset. Retrieved April 25, 2020, from https://smoosavi.org/datasets/us\_accidents`

Arxi. (2019). *A Countrywide Traffic Accident Dataset.*  Retrieved April 25, 2020, from <https://arxiv.org/abs/1906.05409>`

McNe. (2016). *Is Humidity or Dry Air worse for your car?.*  Retrieved April 26, 2020, from <https://www.mcneillnissan.com/blog/is-humidity-or-dry-air-worse-for-your-car/>

NCBI. (2016).*Impact of Relative Humidity on Visibility Degradation.*  Retrieved April 26, 2020, from <https://pubmed.ncbi.nlm.nih.gov/27395081/>

SRD (2020) *Number of Vehicles in the United States since 1990,* retrieved from <https://www.statista.com/statistics/183505/number-of-vehicles-in-the-united-states-since-1990/>.

PROP (2020), STAT1003 Project Proposal, retrieved locally.

# Appendix

## R code for Exploratory Analysis

Untitled

library(tidyverse)

## -- Attaching packages ------------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.0 v purrr 0.3.4  
## v tibble 3.0.0 v dplyr 0.8.5  
## v tidyr 1.0.2 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.5.0

## -- Conflicts ---------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(purrr)  
library(ggplot2)  
library("plyr")

library(dplyr)  
library(naniar)  
library(mice)

library(Hmisc)

library(simputation)

library(forcats)  
library(pals)  
library(maps)

library(mapdata)  
#library(SDMTools)  
library(readxl)

us <- read.csv("US\_Accidents\_Dec19.csv")

names(us)[24] <- "Temprature"  
names(us)[11] <- "Distance"  
names(us)[25] <- "Wind\_chill"  
names(us)[26] <- "Humidity"  
names(us)[27] <- "Pressure"  
names(us)[28] <- "Visibility"  
names(us)[30] <- "Wind\_speed"  
names(us)[31] <- "Precipitation"

us <- mutate\_if(us, is.character, as.factor)  
  
us$Source <- as.factor(us$Source)  
us$TMC <- as.factor(us$TMC)

levels(us$Wind\_Direction)

## [1] "" "Calm" "CALM" "E" "East" "ENE"   
## [7] "ESE" "N" "NE" "NNE" "NNW" "North"   
## [13] "NW" "S" "SE" "South" "SSE" "SSW"   
## [19] "SW" "VAR" "Variable" "W" "West" "WNW"   
## [25] "WSW"

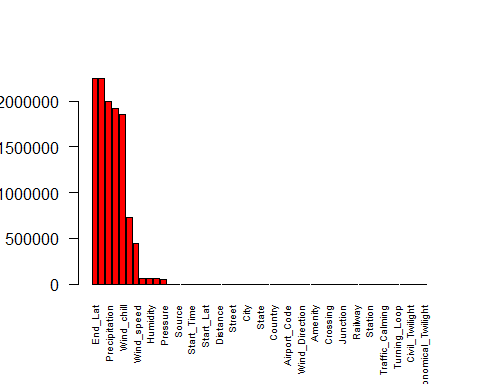
#transforming inconsistency  
us <- transform(us,Wind\_Direction=revalue(Wind\_Direction,c("East"="E")))  
us <- transform(us,Wind\_Direction=revalue(Wind\_Direction,c("North"="N")))  
us <- transform(us,Wind\_Direction=revalue(Wind\_Direction,c("South"="S")))  
us <- transform(us,Wind\_Direction=revalue(Wind\_Direction,c("VAR"="Variable")))  
us <- transform(us,Wind\_Direction=revalue(Wind\_Direction,c("West"="W")))  
us <- transform(us,Wind\_Direction=revalue(Wind\_Direction,c("Calm"="CALM")))  
  
levels(us$Wind\_Direction)

## [1] "" "CALM" "E" "ENE" "ESE" "N"   
## [7] "NE" "NNE" "NNW" "NW" "S" "SE"   
## [13] "SSE" "SSW" "SW" "Variable" "W" "WNW"   
## [19] "WSW"

nlevels(us$Wind\_Direction)

## [1] 19

####my final code for misssng  
missing\_value <-colSums(is.na(us))%>%  
 sort(decreasing = TRUE)  
barplot(missing\_value,cex.names = 0.6,col="red",las=2)

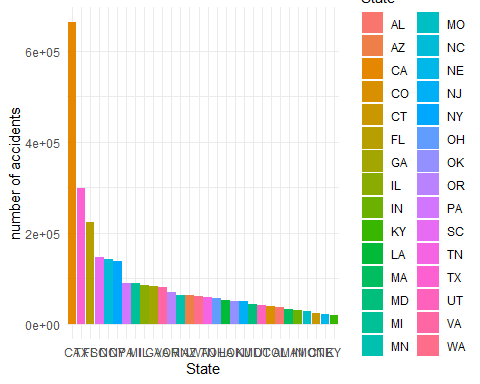


missing\_per <- miss\_var\_summary((us))  
missing\_per

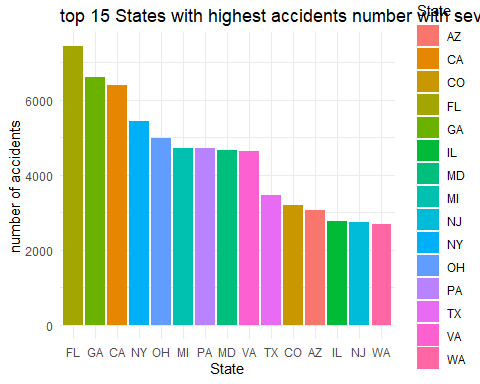
## # A tibble: 49 x 3  
## variable n\_miss pct\_miss  
## <chr> <int> <dbl>  
## 1 End\_Lat 2246264 75.5   
## 2 End\_Lng 2246264 75.5   
## 3 Precipitation 1998358 67.2   
## 4 Number 1917605 64.5   
## 5 Wind\_chill 1852623 62.3   
## 6 TMC 728071 24.5   
## 7 Wind\_speed 440840 14.8   
## 8 Visibility 65691 2.21  
## 9 Humidity 59173 1.99  
## 10 Temprature 56063 1.88  
## # ... with 39 more rows

#removing column with no data,End lat and Endlong  
us <- us[,c(-9,-10)]

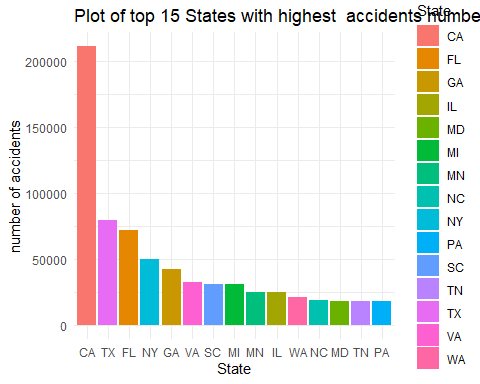
###codes and plot for high risk places  
##state high rate  
library(RColorBrewer)  
State\_h\_accidents<- us%>% dplyr::select(State)%>% group\_by(State)%>% dplyr::count(sort=TRUE)%>%head(30)  
ggplot(State\_h\_accidents,aes(fill=State,x=reorder(State,-n),y=n))+ geom\_bar(stat="identity")+ labs(x="State", y = "number of accidents")+ theme\_minimal()



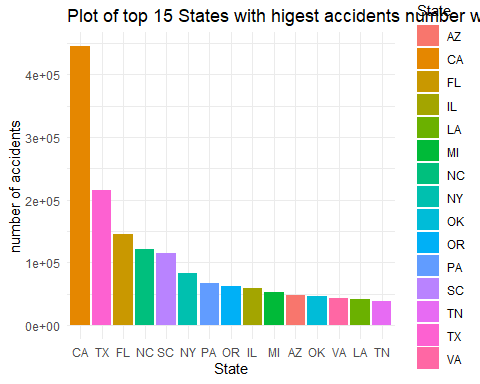
State\_h\_accidents\_severity\_4 <- us%>% select(Severity,State)%>% group\_by(Severity,State)%>%filter(Severity== 4)%>% dplyr::count(sort=TRUE)%>%head(15)  
  
ggplot(State\_h\_accidents\_severity\_4,aes(fill=State,x=reorder(State,-n),y=n))+ geom\_bar(stat="identity")+ labs(title="top 15 States with highest accidents number with severity of 4", x="State", y = "number of accidents")+ theme\_minimal()



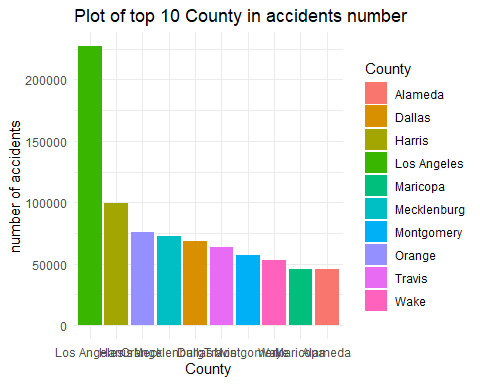
State\_h\_accidents\_severity\_3 <- us%>% select(Severity,State)%>% group\_by(Severity,State)%>%filter(Severity== 3)%>% dplyr::count(sort=TRUE)%>%head(15)  
ggplot(State\_h\_accidents\_severity\_3,aes(fill=State,x=reorder(State,-n),y=n))+ geom\_bar(stat="identity")+ labs(title="Plot of top 15 States with highest accidents number with severity of 3", x="State", y = "number of accidents")+ theme\_minimal()



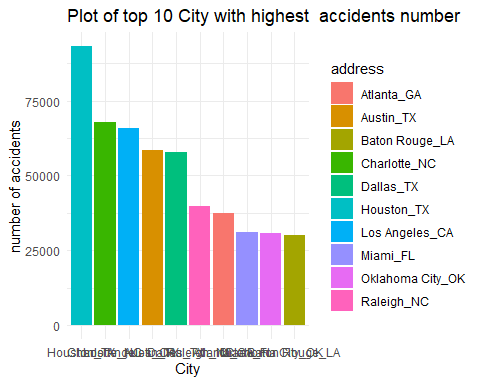
State\_h\_accidents\_severity\_2<- us%>% select(Severity,State)%>% group\_by(Severity,State)%>%filter(Severity== 2)%>% dplyr::count(sort=TRUE)%>%head(15)  
ggplot(State\_h\_accidents\_severity\_2,aes(fill=State,x=reorder(State,-n),y=n))+ geom\_bar(stat="identity")+ labs(title="Plot of top 15 States with higest accidents number with severity of 2", x="State", y = "number of accidents")+ theme\_minimal()



county\_h\_accidents<- us%>% select(County)%>% group\_by(County)%>% dplyr::count(sort=TRUE)%>%head(10)  
ggplot(county\_h\_accidents,aes(fill=County,x=reorder(County,-n),y=n))+ geom\_bar(stat="identity")+ labs(title="Plot of top 10 County in accidents number", x="County", y = "number of accidents")+ theme\_minimal()



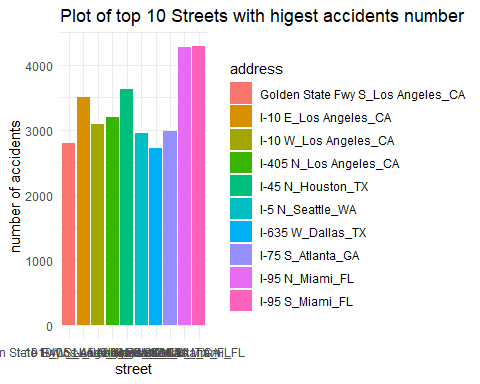
####city high accident  
#Top 10city with highset number of accidents  
city\_h\_accident <- us%>% select(City,State)%>% group\_by(City,State)%>% dplyr::count(sort=TRUE)%>%head(10)  
city\_h\_accident\_unit <- unite(city\_h\_accident,address,City,State)  
ggplot(city\_h\_accident\_unit,aes(fill=address,x=reorder(address,-n),y=n))+ geom\_bar(stat="identity")+ labs(title="Plot of top 10 City with highest accidents number",   
 x="City", y = "number of accidents")+ theme\_minimal()



##################  
#top 10 address  
Street\_high\_accident <- us%>% select(Street,City,State)%>% group\_by(Street,City,State)%>% dplyr::count(sort=TRUE)%>%head(10)  
Street\_high\_accident\_unit <- unite(Street\_high\_accident,address,Street,City,State)  
Street\_high\_accident\_unit

## # A tibble: 10 x 2  
## address n  
## <chr> <int>  
## 1 I-95 S\_Miami\_FL 4287  
## 2 I-95 N\_Miami\_FL 4273  
## 3 I-45 N\_Houston\_TX 3633  
## 4 I-10 E\_Los Angeles\_CA 3499  
## 5 I-405 N\_Los Angeles\_CA 3203  
## 6 I-10 W\_Los Angeles\_CA 3086  
## 7 I-75 S\_Atlanta\_GA 2984  
## 8 I-5 N\_Seattle\_WA 2945  
## 9 Golden State Fwy S\_Los Angeles\_CA 2796  
## 10 I-635 W\_Dallas\_TX 2713

ggplot(Street\_high\_accident\_unit,aes(fill=address,x=address,y=n))+ geom\_bar(stat="identity")+labs(title="Plot of top 10 Streets with higest accidents number ", x="street",y="number of accidents")+theme\_minimal()



######################################################

#Extracting year from start time inorder to count the number of accidents per state  
us$yearr<- as.integer(substr(us$Start\_Time,1,4))  
##merging data set of populationPopulation  
#because the state in the data set is abbreviation, we are going to merge the dataset of state name in order to extract the name os states  
bb <- read.csv("bb.csv")  
str(bb)

## 'data.frame': 51 obs. of 3 variables:  
## $ State : Factor w/ 51 levels "Alabama","Alaska",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ Abbrev: Factor w/ 51 levels "Ala.","Alaska",..: 1 2 3 4 5 6 7 9 8 10 ...  
## $ Code : Factor w/ 51 levels "AK","AL","AR",..: 2 1 4 3 5 6 7 9 8 10 ...

abv <- select(bb,State,Code)  
us2<- merge(us, abv, by.x = c("State"), by.y = c("Code"))

top1 <- read\_excel("population.xlsx", sheet = "pop")

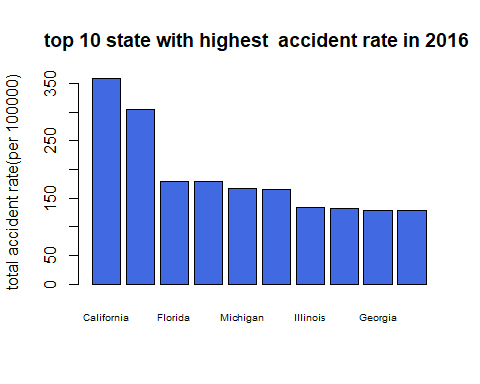
top1$state <- gsub("^.", "", top1$state) # removes preceding dot  
names(top1)[3] <- "Population"  
top1 <- spread(top1,year,Population)  
top2 <- top1[,-c(2,3,4,5,6,7)]  
names(top2)[2] <- "year\_16"  
names(top2)[3] <- "year\_17"  
names(top2)[4] <- "year\_18"  
names(top2)[5] <- "year\_19"  
  
###Removing Alaska and Hawaii,because they are not included in the accident data set  
top3 <- top2[-c(2,12),]

us3 <- us2%>%select(State.y,yearr)  
## State.y yearr

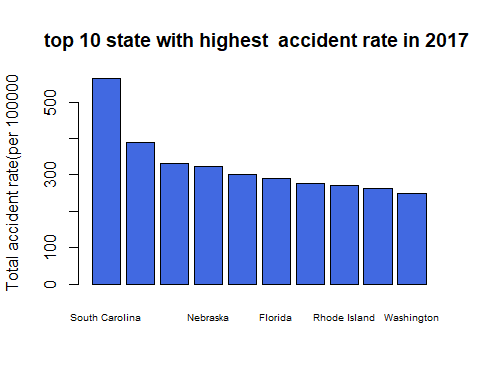
names(us3)[1] <- "state"  
names(us3)

## [1] "state" "yearr"

us4 <- us3%>%  
 dplyr::count(state,yearr)  
#Extracting each year separately then normalize them  
us5\_2016<- subset(us4,subset = (us4$yearr == 2016))  
us5rate2016 <- 100000\*us5\_2016$n/top3$year\_16  
names(us5rate2016) <- us5\_2016$state  
sort16<- sort(us5rate2016)[c(49,48,47,46,45,44,43,42,41,40)]  
barplot(sort16,cex.names = 0.6, main="top 10 state with highest accident rate in 2016 ",ylab="total accident rate(per 100000)",col="royalblue")



us5\_2017<- subset(us4,subset = (yearr == 2017))  
us5rate2017 <- 100000\*us5\_2017$n/top3$year\_17  
names(us5rate2017) <- us5\_2017$state  
sort17<- sort(us5rate2017)[c(49,48,47,46,45,44,43,42,41,40)]  
barplot(sort17,cex.names = 0.6, main="top 10 state with highest accident rate in 2017 ",ylab="Total accident rate(per 100000",col="royalblue")

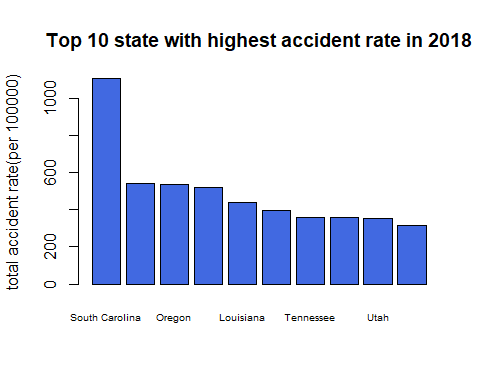


us5\_2018<- subset(us4,subset = (yearr == 2018))  
dim(us5\_2018)

## [1] 49 3

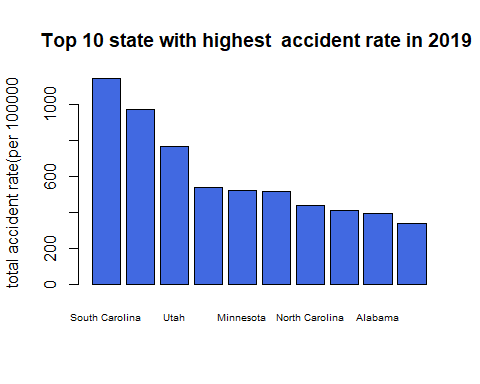
us5rate2018 <- 100000\*us5\_2018$n/top3$year\_18

names(us5rate2018) <- us5\_2018$state  
sort18<- sort(us5rate2018)[c(49,48,47,46,45,44,43,42,41,40)]  
barplot(sort18,cex.names = 0.6, main="Top 10 state with highest accident rate in 2018",ylab="total accident rate(per 100000)",col="royalblue")

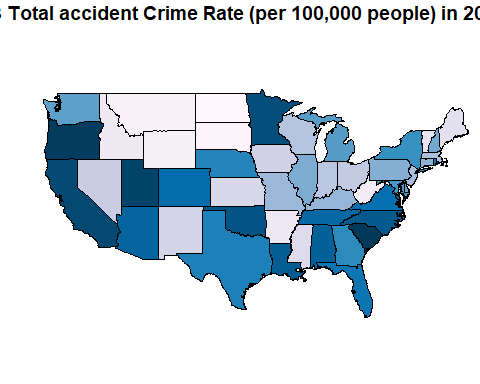


us5\_2019<- subset(us4,subset = (yearr == 2019))

us5rate2019 <- 100000\*us5\_2019$n/top3$year\_19  
names(us5rate2019) <- us5\_2019$state  
sort19<- sort(us5rate2019)[c(49,48,47,46,45,44,43,42,41,40)]  
barplot(sort19,cex.names = 0.6, main="Top 10 state with highest accident rate in 2019 ",ylab="total accident rate(per 100000",col="royalblue")



names(us5rate2019) <- tolower(names(us5rate2019))  
SCols <- brewer.pubu(length(us5rate2019))  
us5rate2019 <- sort(us5rate2019)  
RateCols <- data.frame(State = names(us5rate2019), Rate = us5rate2019,   
 Cols = SCols)  
  
State <- maps::map("state", xlim = c(-125, -66), ylim = c(25, 50), mar = c(0,   
 0, 0, 0), plot = FALSE)$names  
State <- unlist(lapply(strsplit(State, ":"), function(x) x[[1]]))  
State <- data.frame(State = State)  
StateCols <- merge(State, RateCols, all.x = TRUE)  
map("state", xlim = c(-130, -66), ylim = c(25, 52), fill = TRUE, mar = c(0,0, 0, 0), col = as.character(StateCols$Cols))  
par(oma = c(0, 0, 4, 0))  
title(main = " US Total accident Crime Rate (per 100,000 people) in 2019", outer = TRUE,   
 line = 3)

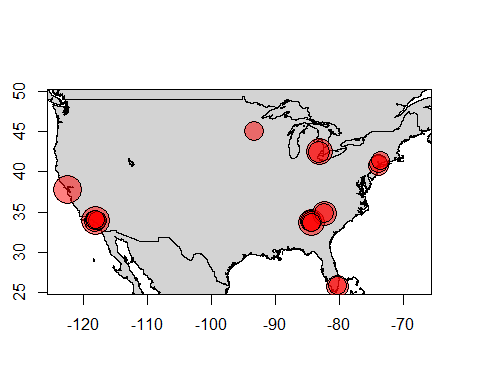


x = c(-20, -18, -18, -20)  
y = c(25, 53, 53, 25)  
#legend.gradient(cbind(x = x - 110, y = y), cols = as.character(RateCols$Cols),   
 # limits = round(range(RateCols$Rate)), title = "")

#finding duplicate address  
sum(duplicated(us$ID))

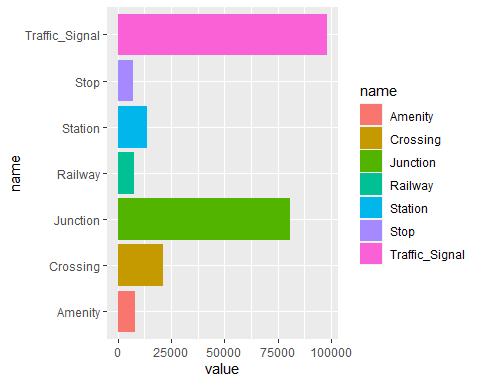
## [1] 0

dup5 <- us%>%  
 select(Start\_Lat,Start\_Lng,Street,City,State)%>%  
 dplyr:: count(Street,City,State,sort=TRUE)  
spot <- us%>%  
 select(Start\_Lat,Start\_Lng,Street,City,State)%>%  
 dplyr::count(Start\_Lat,Start\_Lng,Street,City,State,sort=TRUE)  
spot\_20 <- spot[1:20,]  
df <- data.frame(spot\_20)  
map("worldHires", ylim = c(25, 50), xlim = c(-125, -66), fill = TRUE, col = "lightgrey",   
 mar = c(4.5, 4, 1, 1))  
map.axes()  
points(spot\_20$Start\_Lng, spot\_20$Start\_Lat, cex = spot\_20$n/140, pch = 21, col = "black",   
 bg = rgb(1, 0, 0, 0.5))

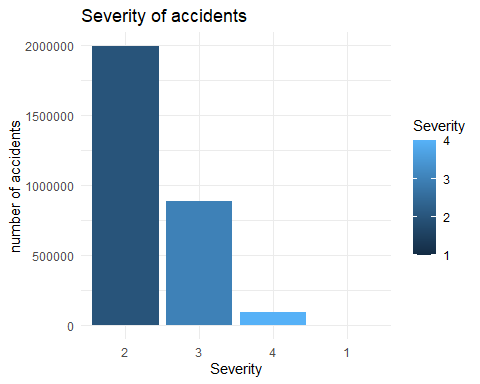


##Point of interest  
#ploting to compare the proportion of Amenity,Crossing,Junction,Railway,Station,Traffic\_Signal,Stop  
us %>% group\_by(Amenity,Crossing,Junction,Railway,Station,Traffic\_Signal,Stop)%>% dplyr::count()

# Create data  
data <- data.frame(  
 name=c("Amenity", "Crossing" ,"Junction" ,"Railway" ,"Station" ,"Traffic\_Signal" ,"Stop") ,   
 value=c(8023,21365,81023,7742,13917,98219,7284))  
  
ggplot(data, aes(fill = name, x=name, y=value)) +   
 geom\_bar(stat = "identity") + coord\_flip()



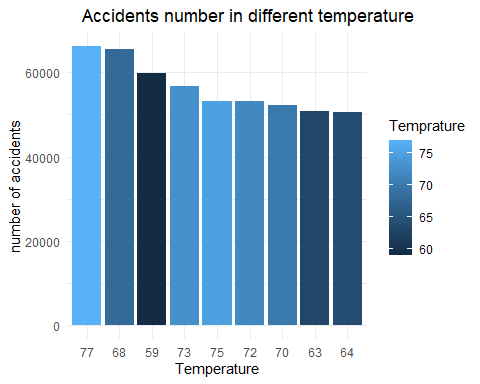
#plot of acceidents per severity  
Severity2<- us%>% dplyr::select(Severity)%>% group\_by(Severity)%>% dplyr::count(sort=TRUE)  
ggplot(Severity2,aes(fill=Severity,x=reorder(Severity,-n),y=n))+ geom\_bar(stat="identity")+ labs(title="Severity of accidents ", x="Severity", y = "number of accidents")+ theme\_minimal()



#plot of accidents per temprature  
Temp<- us%>% select(Temprature)%>% group\_by(Temprature)%>% dplyr::count(sort=TRUE)%>%head(10)

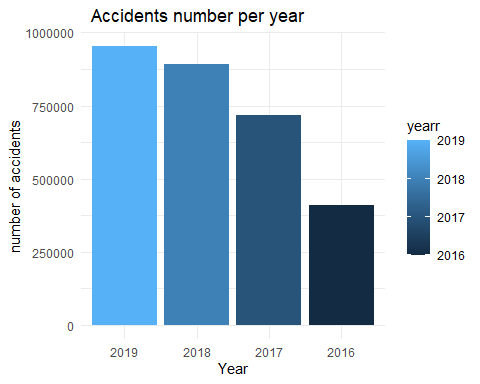
Temp <- Temp[c(-5),]

ggplot(Temp,aes(fill=Temprature,x=reorder(Temprature,-n),y=n))+ geom\_bar(stat="identity")+ labs(title=" Accidents number in different temperature ", x="Temperature", y = "number of accidents")+ theme\_minimal()



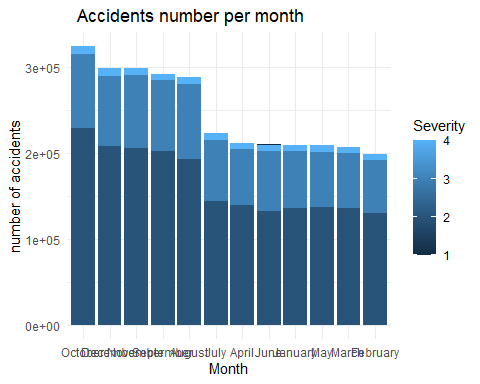
##plotof accident number per year  
year\_us<- us%>% select(yearr)%>% group\_by(yearr)%>% dplyr::count(sort=TRUE)%>%head(4)

ggplot(year\_us,aes(fill=yearr,x=reorder(yearr,-n),y=n))+ geom\_bar(stat="identity")+ labs(title=" Accidents number per year ", x="Year", y = "number of accidents")+ theme\_minimal()

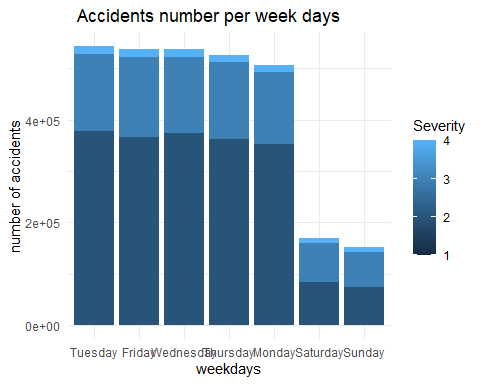


##accident number per month  
us$month <- month.name[as.integer(substr(us$Start\_Time,6,7))]  
us$month <- as.factor(us$month)  
us$month <- factor(us$month,levels=month.name)  
month\_us<- us%>% select(Severity,month)%>% group\_by(Severity,month)%>% dplyr::count(sort=TRUE)  
month\_us

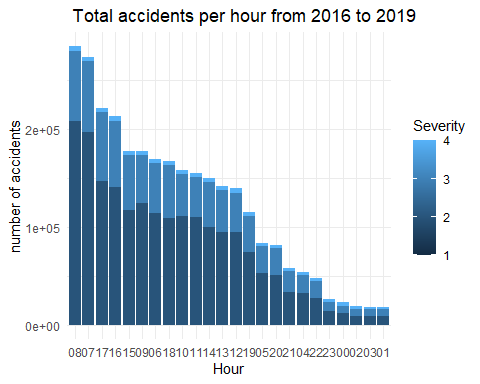
ggplot(month\_us,aes(fill=Severity,x=reorder(month,-n),y=n))+ geom\_bar(stat="identity")+ labs(title=" Accidents number per month ", x="Month", y = "number of accidents")+ theme\_minimal()



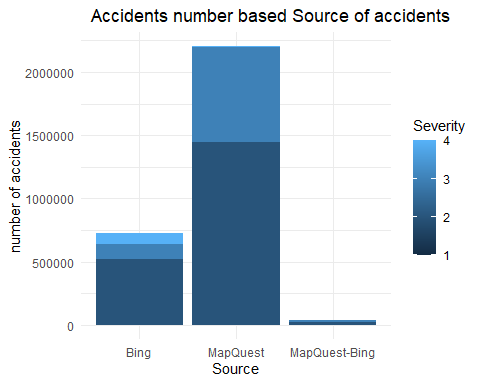
#Accident per weekdays  
us$weekday <- weekdays(as.Date(substr(us$Start\_Time,1,10)))  
weekday\_us<- us%>% select(Severity,weekday)%>% group\_by(Severity,weekday)%>% dplyr::count(sort=TRUE)  
ggplot(weekday\_us,aes(fill=Severity,x=reorder(weekday,-n),y=n))+ geom\_bar(stat="identity")+ labs(title=" Accidents number per week days ", x="weekdays", y = "number of accidents")+ theme\_minimal()



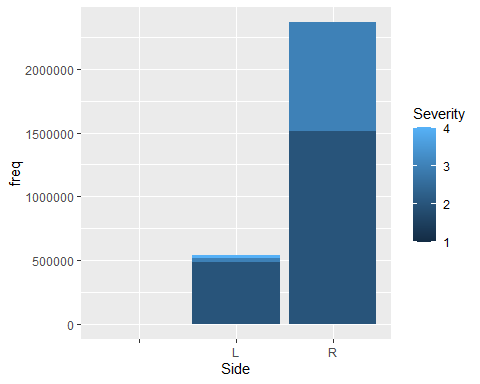
us$hours <- substr(us$Start\_Time,12,13)  
us%>% select(Severity,hours)%>% group\_by(Severity,hours)%>% dplyr::count()%>%  
 ggplot(aes(fill=Severity,x=reorder(hours,-n),y=n))+ geom\_bar(stat="identity")+ labs(title=" Total accidents per hour from 2016 to 2019 ", x="Hour", y = "number of accidents")+ theme\_minimal()



Source\_severity <- us %>% select(Severity,Source)%>%group\_by(Severity,Source)%>% count()  
#Source\_severity <- Sunrise\_Sunset2[-3,]  
 ggplot(Source\_severity,aes(fill=Severity,x=Source,y=freq))+ geom\_bar(stat="identity")+ geom\_bar(stat="identity")+labs(title=" Accidents number based Source of accidents ", y = "number of accidents")+ theme\_minimal()

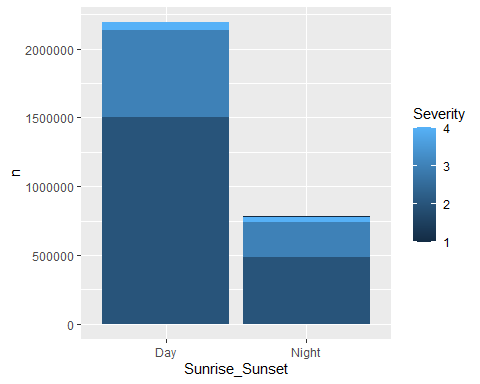


#side and severity  
Side\_severity <- us %>% select(Side,Severity)%>% group\_by(Side, Severity) %>% count()  
Side\_severity <- Side\_severity[-9,]  
ggplot(Side\_severity,aes(fill=Severity,x=Side,y=freq))+ geom\_bar(stat="identity")

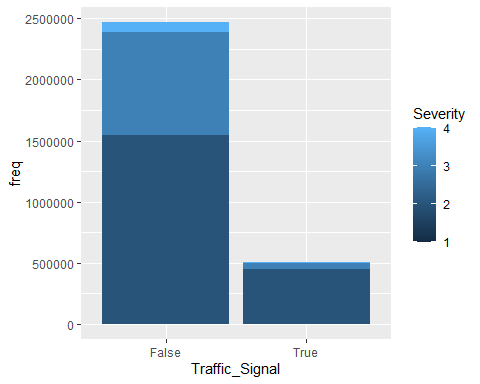


Sunrise\_Sunset2 <- us %>% select(Sunrise\_Sunset,Severity)%>% group\_by(Sunrise\_Sunset, Severity) %>% dplyr::count(sort=TRUE)

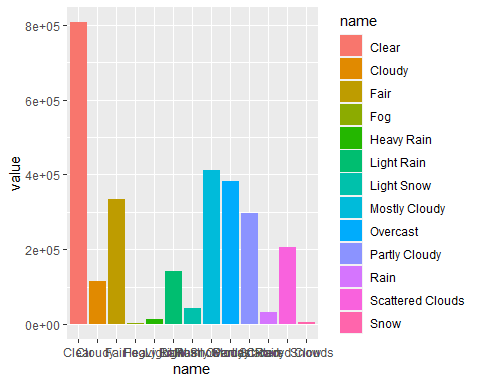
Sunrise\_Sunset2 <- Sunrise\_Sunset2[-c(9,10,11),]  
ggplot(Sunrise\_Sunset2,aes(fill=Severity,x=Sunrise\_Sunset,y=n))+ geom\_bar(stat="identity")



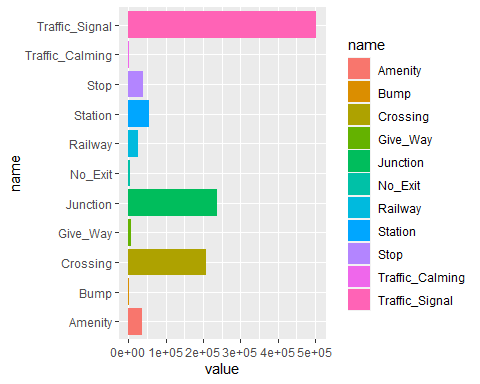
Traffic\_signal\_severity <- us %>% select(Traffic\_Signal,Severity)%>% group\_by(Traffic\_Signal, Severity) %>% count()  
#Traffic\_signal\_severity <- Traffic\_signal\_severity[-9,]  
ggplot(Traffic\_signal\_severity,aes(fill=Severity,x=Traffic\_Signal,y=freq))+ geom\_bar(stat="identity")



#weather condition  
Weather <- us %>% select(Weather\_Condition)%>% group\_by(Weather\_Condition) %>% dplyr::count(sort=TRUE)  
data <- data.frame(  
 name=c("Clear","Mostly Cloudy ","Overcast", "Fair", "Partly Cloudy","Scattered Clouds","Light Rain","Cloudy ","Light Snow ","Rain","Fog","Heavy Rain","Snow") ,   
 value=c(808171, 412528,382480,335289, 295439,204662,141073,115496,42123,32826,2138,12064,4796))  
ggplot(data, aes(fill = name, x=name, y=value)) +   
 geom\_bar(stat = "identity")

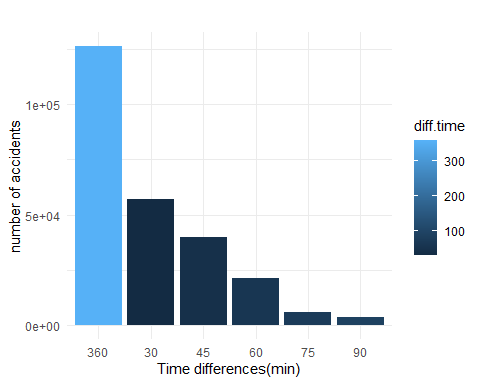


data\_f<- data.frame(  
 name=c("Give\_Way","Amenity","Bump", "Crossing" ,"Junction" ,"Railway" ,"Station" ,"Traffic\_Signal" ,"Stop","Traffic\_Calming","No\_Exit") ,   
 value=c(7627,35220,454 ,207590,238447,25161,56413 ,503383,40160,1111, 3395))  
ggplot(data\_f, aes(fill = name, x=name, y=value)) +   
 geom\_bar(stat = "identity") + coord\_flip()

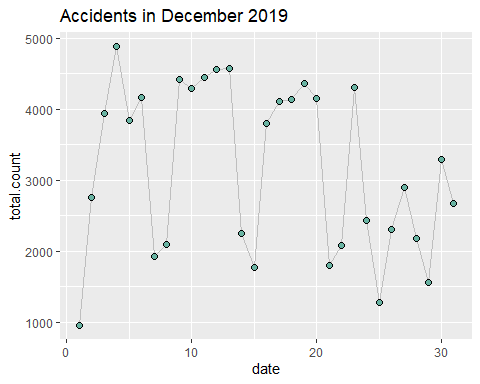


#Extracting time differences  
us$diff.time <- difftime(us$End\_Time, us$Start\_Time, units="min")  
us\_difftime\_18<- us %>% select(diff.time,yearr) %>% filter(yearr == 2016)  
  
ff <- us\_difftime\_18 %>% group\_by(diff.time,yearr)%>% dplyr::count(sort = TRUE)%>% head(6)  
  
ggplot(ff,aes(fill=diff.time,x=reorder(diff.time,-n),y=n))+ geom\_bar(stat="identity")+ labs(title="", x="Time differences(min)", y = "number of accidents")+ theme\_minimal()

## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.



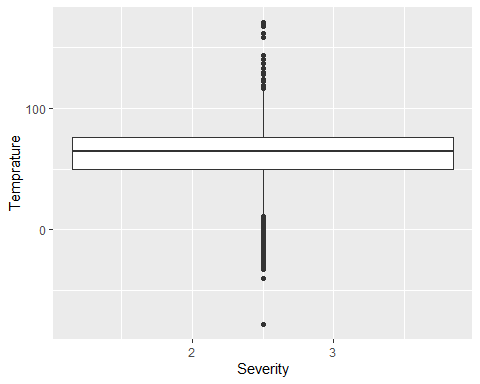
#Line Plot of accident number in 2019  
us$Year <- as.numeric(substr(us$Start\_Time,1,4))  
us$month\_n <- as.integer(substr(us$Start\_Time,6,7))  
us$date <- as.numeric(substr(us$Start\_Time,9,10))  
Date <- us %>% group\_by(month\_n,Year,date) %>% filter(month\_n== 12& Year == 2019)%>% dplyr::summarise(total.count=n())  
  
# Plot  
Date %>%  
 ggplot( aes(x=date, y=total.count)) +  
 geom\_line( color="grey") +  
 geom\_point(shape=21, color="black", fill="#69b3a2", size=2) +  
 ggtitle("Accidents in December 2019")



#box plot of Severity per temprature  
ggplot(us, aes(x= Severity , y= Temprature)) +   
 geom\_boxplot()

## Warning: Continuous x aesthetic -- did you forget aes(group=...)?

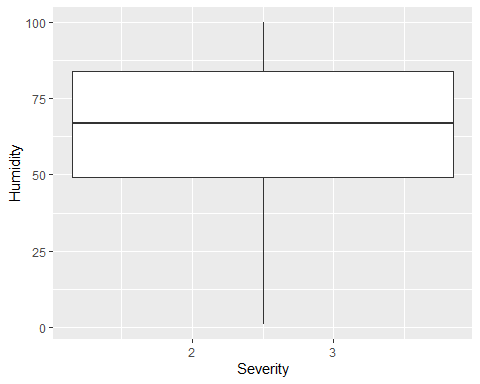
## Warning: Removed 56063 rows containing non-finite values (stat\_boxplot).



#Box plot of Humidity  
ggplot(us, aes(x= Severity , y= Humidity)) +   
 geom\_boxplot()

## Warning: Continuous x aesthetic -- did you forget aes(group=...)?

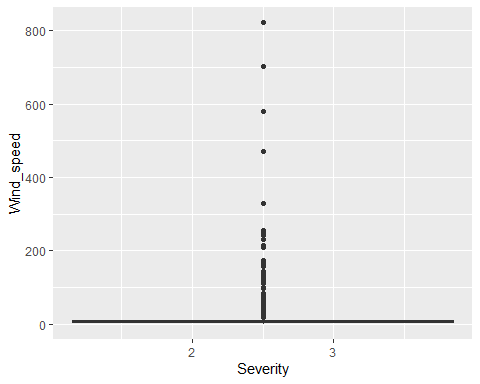
## Warning: Removed 59173 rows containing non-finite values (stat\_boxplot).



#Box plot of Wind\_speed  
ggplot(us, aes(x= Severity , y= Wind\_speed)) +   
 geom\_boxplot()

## Warning: Continuous x aesthetic -- did you forget aes(group=...)?

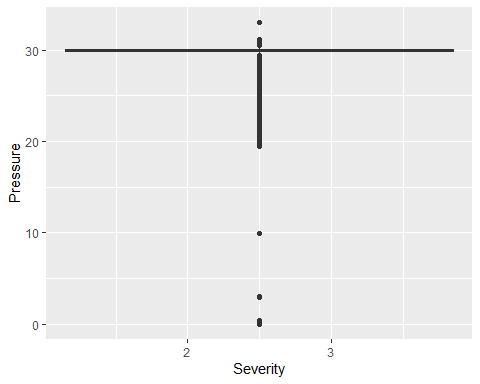
## Warning: Removed 440840 rows containing non-finite values (stat\_boxplot).



#Box plot of Pressure  
ggplot(us, aes(x= Severity , y= Pressure)) +   
 geom\_boxplot()

## Warning: Continuous x aesthetic -- did you forget aes(group=...)?

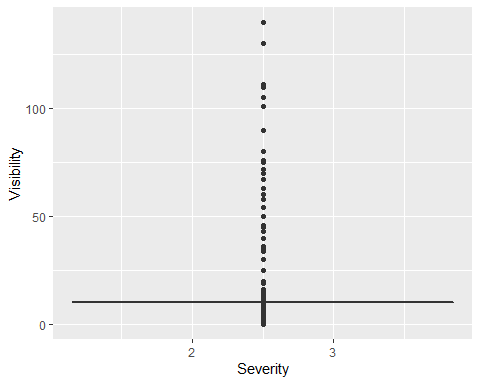
## Warning: Removed 48142 rows containing non-finite values (stat\_boxplot).



#Box plot of Visibility  
ggplot(us, aes(x= Severity , y= Visibility)) +   
 geom\_boxplot()

## Warning: Continuous x aesthetic -- did you forget aes(group=...)?

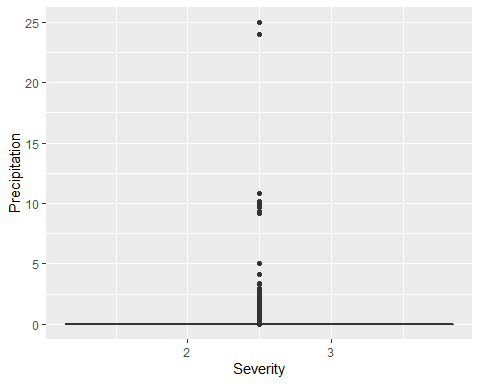
## Warning: Removed 65691 rows containing non-finite values (stat\_boxplot).



#Box plot ofPrecipitation  
ggplot(us, aes(x= Severity , y= Precipitation)) +   
 geom\_boxplot()

## Warning: Continuous x aesthetic -- did you forget aes(group=...)?

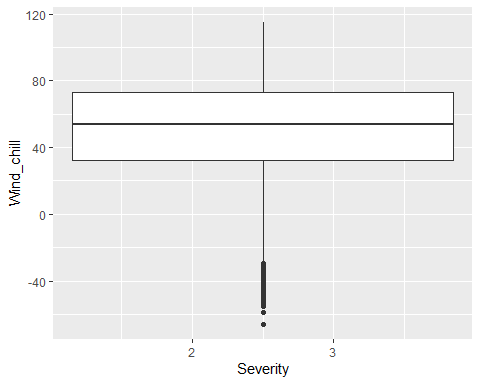
## Warning: Removed 1998358 rows containing non-finite values (stat\_boxplot).



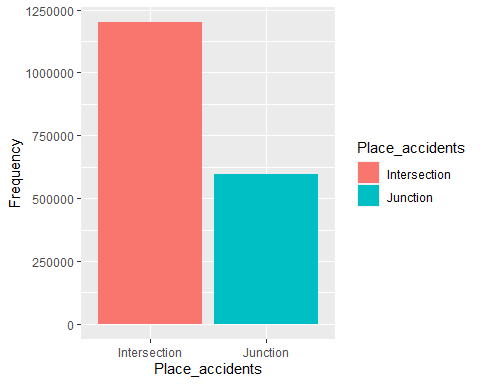
#BOx plot of Wind\_chill  
ggplot(us, aes(x= Severity , y= Wind\_chill)) +   
 geom\_boxplot()

## Warning: Continuous x aesthetic -- did you forget aes(group=...)?

## Warning: Removed 1852623 rows containing non-finite values (stat\_boxplot).



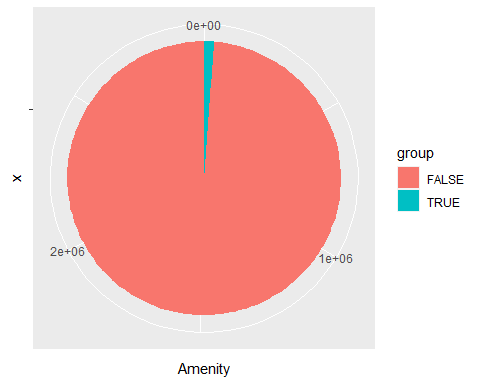
#This bar plot extracted from the information which has been extracted by regular expression from description variables  
data2 <- data.frame(  
 Place\_accidents=c("Junction","Intersection") ,   
 Frequency=c(597599, 1201178))  
ggplot(data2, aes(fill = Place\_accidents, x=Place\_accidents, y=Frequency)) +   
 geom\_bar(stat = "identity")



table(us$Amenity)

##   
## False True   
## 2939115 35220

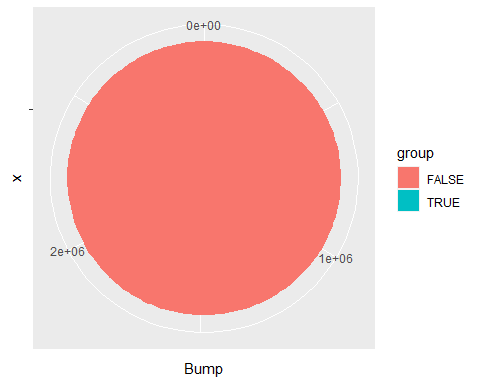
data <- data.frame(  
 group=c("TRUE","FALSE"),  
 Amenity=c(35220,2939115))  
# Basic piechart  
ggplot(data, aes(x="", y=Amenity, fill=group)) +  
 geom\_bar(stat="identity", width=1) +  
 coord\_polar("y", start=0)



table(us$Bump)

##   
## False True   
## 2973881 454

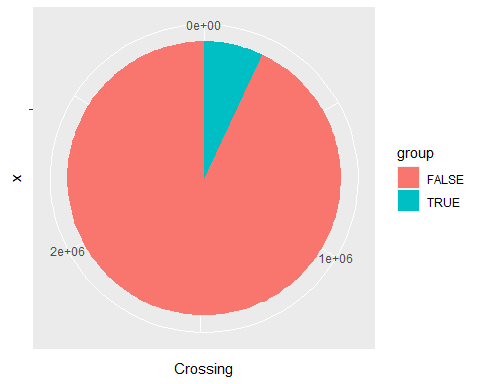
data2 <- data.frame(  
 group=c("TRUE","FALSE"),  
 Bump=c(454,2973881 ))  
# Basic piechart  
ggplot(data2, aes(x="", y=Bump, fill=group)) +  
 geom\_bar(stat="identity", width=1) +  
 coord\_polar("y", start=0)



table(us$Crossing)

##   
## False True   
## 2766745 207590

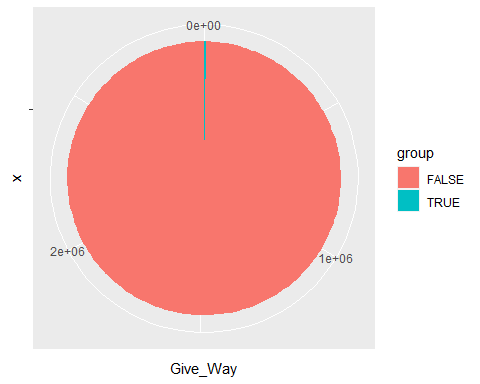
data3 <- data.frame(  
 group=c("TRUE","FALSE"),  
 Crossing=c(207590,2766745 ))  
# Basic piechart  
ggplot(data3, aes(x="", y=Crossing, fill=group)) +  
 geom\_bar(stat="identity", width=1) +  
 coord\_polar("y", start=0)



table(us$Give\_Way)

##   
## False True   
## 2966708 7627

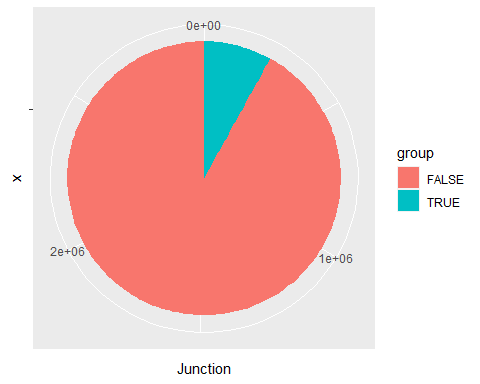
data4<- data.frame(  
 group=c("TRUE","FALSE"),  
 Give\_Way=c(7627,2966708 ))  
# Basic piechart  
ggplot(data4, aes(x="", y=Give\_Way, fill=group)) +  
 geom\_bar(stat="identity", width=1) +  
 coord\_polar("y", start=0)



table(us$Junction)

##   
## False True   
## 2735888 238447

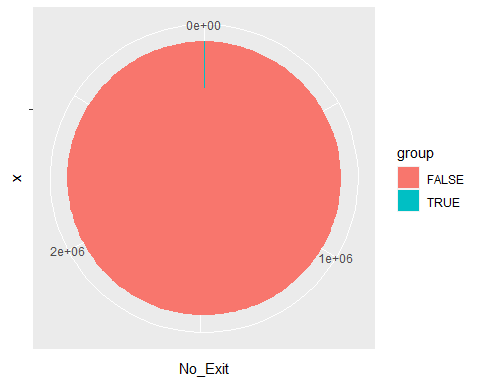
data5<- data.frame(  
 group=c("TRUE","FALSE"),  
 Junction=c(238447,2735888 ))  
# Basic piechart  
ggplot(data5, aes(x="", y=Junction, fill=group)) +  
 geom\_bar(stat="identity", width=1) +  
 coord\_polar("y", start=0)



table(us$No\_Exit)

##   
## False True   
## 2970940 3395

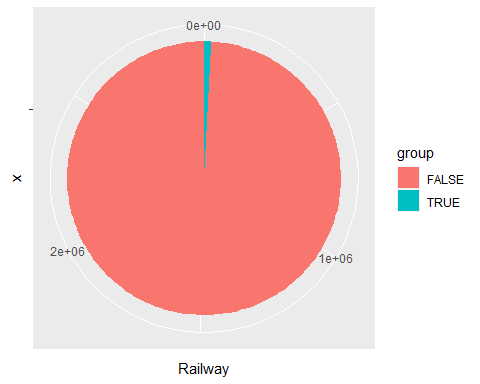
data6<- data.frame(  
 group=c("TRUE","FALSE"),  
 No\_Exit=c( 3395 ,2970940))  
# Basic piechart  
ggplot(data6, aes(x="", y=No\_Exit, fill=group)) +  
 geom\_bar(stat="identity", width=1) +  
 coord\_polar("y", start=0)



table(us$Railway)

##   
## False True   
## 2949174 25161

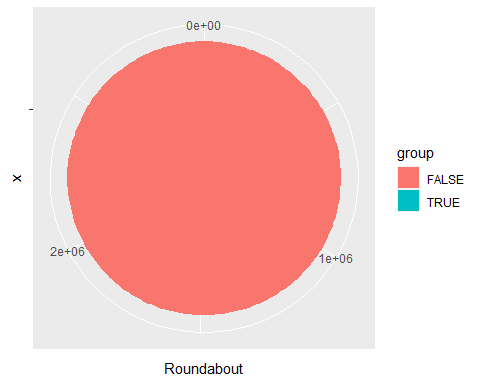
data7<- data.frame(  
 group=c("TRUE","FALSE"),Railway=c( 25161 ,2949174 ))  
# Basic piechart  
ggplot(data7, aes(x="", y=Railway, fill=group)) +  
 geom\_bar(stat="identity",width=1) +  
 coord\_polar("y", start=0)



table(us$Roundabout)

##   
## False True   
## 2974167 168

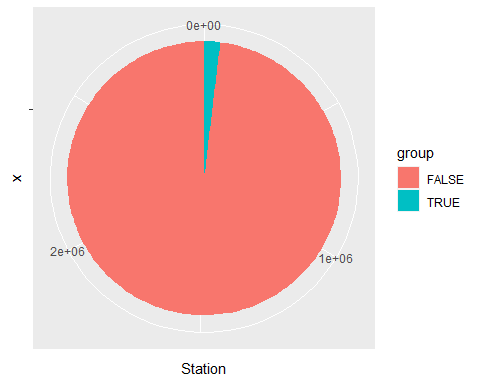
data8<- data.frame(  
 group=c("TRUE","FALSE"),  
 Roundabout=c( 168 ,2974167))  
# Basic piechart  
ggplot(data8, aes(x="", y=Roundabout, fill=group)) +  
 geom\_bar(stat="identity", width=1) +  
 coord\_polar("y", start=0)



table(us$Station)

##   
## False True   
## 2917922 56413

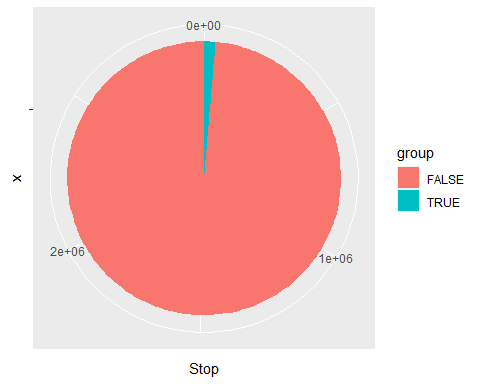
data9<- data.frame(  
 group=c("TRUE","FALSE"),  
 Station=c( 56413 ,2917922))  
# Basic piechart  
ggplot(data9, aes(x="", y=Station, fill=group)) +  
 geom\_bar(stat="identity", width=1) +  
 coord\_polar("y", start=0)



table(us$Stop)

##   
## False True   
## 2934175 40160

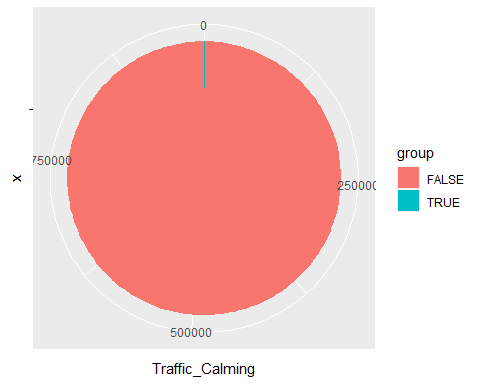
data10<- data.frame(  
 group=c("TRUE","FALSE"),  
 Stop=c( 40160 ,2934175))  
# Basic piechart  
ggplot(data10, aes(x="", y=Stop, fill=group)) +  
 geom\_bar(stat="identity", width=1) +  
 coord\_polar("y", start=0)



table(us$Traffic\_Calming)

##   
## False True   
## 2973224 1111

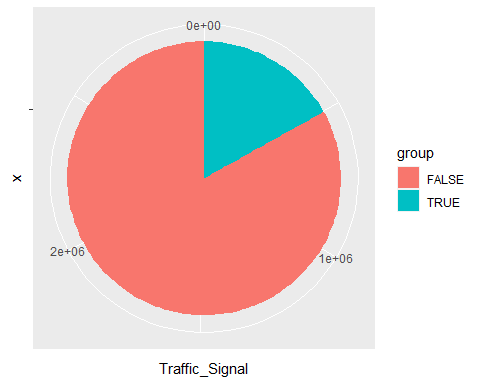
data11<- data.frame(  
 group=c("TRUE","FALSE"),  
 Traffic\_Calming=c( 1111 ,973224))  
# Basic piechart  
ggplot(data11, aes(x="", y=Traffic\_Calming, fill=group)) +  
 geom\_bar(stat="identity", width=1) +  
 coord\_polar("y", start=0)



table(us$Traffic\_Signal)

##   
## False True   
## 2470952 503383

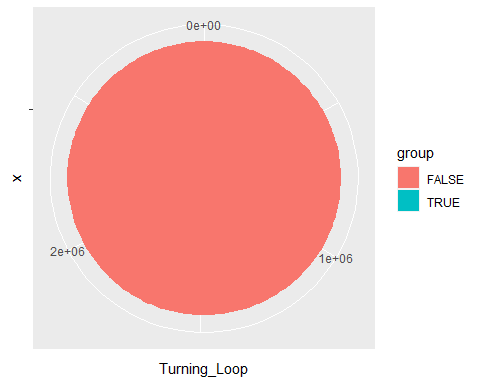
data12<- data.frame(  
 group=c("TRUE","FALSE"),  
 Traffic\_Signal=c( 503383 ,2470952))  
# Basic piechart  
ggplot(data12, aes(x="", y=Traffic\_Signal, fill=group)) +  
 geom\_bar(stat="identity", width=1) +  
 coord\_polar("y", start=0)



table(us$Turning\_Loop)

##   
## False   
## 2974335

data13<- data.frame(  
 group=c("TRUE","FALSE"),  
 Turning\_Loop=c( 1 ,2974335 ))  
# Basic piechart  
ggplot(data13, aes(x="", y=Turning\_Loop, fill=group)) +  
 geom\_bar(stat="identity", width=1) +  
 coord\_polar("y", start=0)



#Extracting from description by regular expression

library(stringr)

tt <- us$Description

df1 <- as.data.frame(tt)

stl8 <- df1 %>%

mutate(typpp = case\_when(

str\_detect(tt,regex("on(.\*)at (.\*)Exit")) ~ "Junction",str\_detect(tt,regex("(.\*)ramp to(.\*)[^at Exit]")) ~ "Junction",str\_detect(tt,regex("(.\*)on(.\*)at[^Exit](.\*)")) ~ "Intersection"))

table(stl8$typpp)

## R code for Predictive Modelling

ml <- read.csv("Data for machine learning final and makes sense.csv")

Validation\_Index <- read.csv("80.csv")

Validation <- read.csv("20.csv")

library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

Index <- sample(nrow(ml), size = 0.80\* nrow(ml), replace = FALSE)

Train <- ml[Index, ]  
Test <- ml[-Index, ]

dim(ml)

## [1] 47 10

sapply(ml, class)

## ï..Severity Distance.mi. Temperature.F. Wind\_Chill.F.   
## "numeric" "numeric" "numeric" "numeric"   
## Humidity... Pressure.in. Visibility.mi. Wind\_Speed.mph.   
## "numeric" "numeric" "numeric" "numeric"   
## Precipitation.in. Total.Accidents   
## "numeric" "integer"

head(ml)

## ï..Severity Distance.mi. Temperature.F. Wind\_Chill.F. Humidity...  
## 1 2.551106 0.9789020 31.61725 19.73722 76.52174  
## 2 2.465122 0.4029051 58.36754 33.56067 56.22283  
## 3 2.387729 0.2402353 63.02905 36.20821 55.56313  
## 4 2.370343 0.2894394 65.56810 56.60138 60.49907  
## 5 2.405823 0.2448246 75.95733 79.26639 57.68210  
## 6 2.376590 0.2365210 80.97316 83.38598 59.86327  
## Pressure.in. Visibility.mi. Wind\_Speed.mph. Precipitation.in. Total.Accidents  
## 1 29.94922 6.954343 10.598400 0.02215548 949  
## 2 29.98245 9.468274 9.900085 0.03008929 6329  
## 3 29.97919 9.583959 9.289587 0.02001865 18088  
## 4 29.94767 9.550544 8.797772 0.02334630 17610  
## 5 29.94701 9.530479 8.949648 0.04976080 30527  
## 6 29.96342 9.737811 8.429407 0.06029963 45759

levels(ml$Total.Accidents)

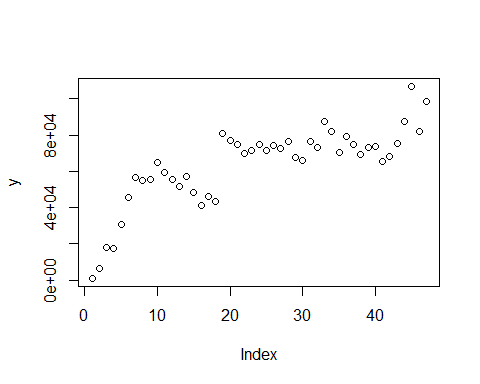
## NULL

summary(ml)

## ï..Severity Distance.mi. Temperature.F. Wind\_Chill.F.   
## Min. :2.231 Min. :0.1820 Min. :31.62 Min. :18.05   
## 1st Qu.:2.358 1st Qu.:0.2513 1st Qu.:50.51 1st Qu.:29.67   
## Median :2.377 Median :0.2842 Median :63.54 Median :44.66   
## Mean :2.373 Mean :0.3020 Mean :62.33 Mean :49.33   
## 3rd Qu.:2.392 3rd Qu.:0.3204 3rd Qu.:75.10 3rd Qu.:72.29   
## Max. :2.551 Max. :0.9789 Max. :80.97 Max. :83.39   
## Humidity... Pressure.in. Visibility.mi. Wind\_Speed.mph.   
## Min. :55.56 Min. :29.13 Min. :6.954 Min. : 6.113   
## 1st Qu.:62.55 1st Qu.:29.94 1st Qu.:8.884 1st Qu.: 7.987   
## Median :64.22 Median :29.99 Median :9.318 Median : 8.669   
## Mean :64.97 Mean :29.87 Mean :9.151 Mean : 8.554   
## 3rd Qu.:67.84 3rd Qu.:30.05 3rd Qu.:9.488 3rd Qu.: 9.280   
## Max. :76.52 Max. :30.18 Max. :9.738 Max. :10.598   
## Precipitation.in. Total.Accidents   
## Min. :0.003121 Min. : 949   
## 1st Qu.:0.025245 1st Qu.: 55146   
## Median :0.045980 Median : 70073   
## Mean :0.049038 Mean : 63283   
## 3rd Qu.:0.073823 3rd Qu.: 75120   
## Max. :0.154874 Max. :106802

x <- ml[,1:9]  
y <- ml[,10]

plot(y)



control <- trainControl(method="cv", number=10)  
metric <- "RMSE"

set.seed(7)  
fit.cart <- train(Total.Accidents~., data=ml, method="rpart",   
metric = metric, trControl=control)

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :  
## There were missing values in resampled performance measures.

set.seed(7)  
fit.knn <- train(Total.Accidents~., data=ml, method="knn", metric=metric, trControl=control)

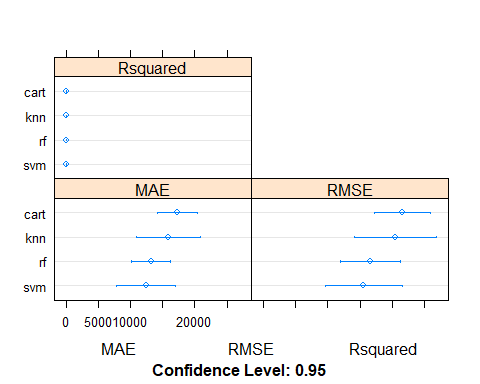
set.seed(7)  
fit.svm <- train(Total.Accidents~., data=ml, method="svmRadial", metric=metric, trControl=control)

set.seed(7)  
fit.rf <- train(Total.Accidents~., data=ml, method="rf", metric=metric, trControl=control)

results <- resamples(list(cart=fit.cart, knn=fit.knn, svm=fit.svm, rf=fit.rf))  
summary(results)

##   
## Call:  
## summary.resamples(object = results)  
##   
## Models: cart, knn, svm, rf   
## Number of resamples: 10   
##   
## MAE   
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's  
## cart 11303.878 13870.064 16950.442 17257.19 18749.61 25112.25 0  
## knn 4405.536 11216.679 14252.524 15893.82 22157.10 25357.68 0  
## svm 4690.867 9010.254 9489.736 12415.19 15520.58 27321.81 0  
## rf 8383.860 10053.446 11297.361 13174.34 15454.76 21608.72 0  
##   
## RMSE   
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's  
## cart 13513.884 18464.42 19730.46 21572.93 27020.44 30429.20 0  
## knn 5862.958 15420.32 20377.80 20489.36 25635.50 36483.24 0  
## svm 5833.927 10532.00 12583.62 15574.29 18616.41 34659.89 0  
## rf 10685.908 11816.95 12864.44 16601.31 21880.02 27939.17 0  
##   
## Rsquared   
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's  
## cart 0.048575911 0.10400647 0.3699648 0.3821868 0.6213552 0.8646717 0  
## knn 0.131063094 0.19187321 0.3380398 0.4182757 0.6477102 0.9023186 0  
## svm 0.008661955 0.09394949 0.6583764 0.5196678 0.8798795 0.9633969 0  
## rf 0.005624063 0.05696605 0.4753039 0.4254946 0.7233425 0.9115442 0

dotplot(results)



print(fit.svm)

## Support Vector Machines with Radial Basis Function Kernel   
##   
## 47 samples  
## 9 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 43, 43, 42, 42, 42, 42, ...   
## Resampling results across tuning parameters:  
##   
## C RMSE Rsquared MAE   
## 0.25 18226.81 0.4759903 14252.24  
## 0.50 17233.39 0.4747465 13679.85  
## 1.00 15574.29 0.5196678 12415.19  
##   
## Tuning parameter 'sigma' was held constant at a value of 0.1107457  
## RMSE was used to select the optimal model using the smallest value.  
## The final values used for the model were sigma = 0.1107457 and C = 1.

library(e1071)

bo <- factor(c(Validation$Total.Accidents))

library(caret)

predictions <- factor(predict(fit.svm, Validation, na.action = na.pass))  
table(predictions, Validation$Total.Accidents)

##   
## predictions 65496 67681.94043 68232 73000 75236 81893 87666 98210 106802  
## 66925.8872559734 0 0 1 0 0 0 0 0 0  
## 67216.2237282746 0 1 0 0 0 0 0 0 0  
## 67681.9404292469 1 0 0 0 0 0 0 0 0  
## 71680.8578434224 0 0 0 1 0 0 0 0 0  
## 73210.7337943905 0 0 0 0 1 0 0 0 0  
## 81122.420966205 0 0 0 0 0 0 0 1 0  
## 84080.1127360598 0 0 0 0 0 1 0 0 0  
## 85485.1926717475 0 0 0 0 0 0 1 0 0  
## 88202.6303138012 0 0 0 0 0 0 0 0 1

library(randomForest)

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':  
##   
## margin

gbmImp <- varImp(fit.rf, scale = FALSE)

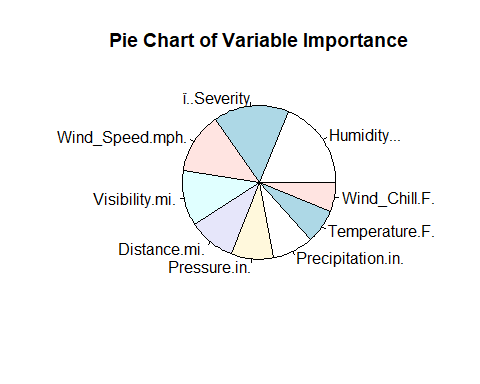
gbmImp

## rf variable importance  
##   
## Overall  
## Humidity... 3.779e+09  
## ï..Severity 3.218e+09  
## Wind\_Speed.mph. 2.568e+09  
## Visibility.mi. 2.349e+09  
## Distance.mi. 1.997e+09  
## Pressure.in. 1.793e+09  
## Precipitation.in. 1.743e+09  
## Temperature.F. 1.463e+09  
## Wind\_Chill.F. 1.245e+09

slices <- c(3778619307,3218230190,   
2567916581   
,2349196620   
,1997491661   
,1792824230   
,1743009925   
,1462530803   
,1245007147)

lbls <- c("Humidity...",   
"ï..Severity"   
,"Wind\_Speed.mph."   
,"Visibility.mi."   
,"Distance.mi."   
,"Pressure.in."   
,"Precipitation.in."   
,"Temperature.F."   
,"Wind\_Chill.F.")

pie(slices, labels=lbls, main = "Pie Chart of Variable Importance")



library(caret)

library(sjstats)

library(forecast)

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

Pred <- c(66925.8872559734,67216.2237282746, 67681.9404292469,71680.8578434224,71680.8578434224,73210.7337943905,81122.420966205,84080.112,85485.1926717475,88202.6303138012)

Act <- c(68232,69396,65496,73000,73868,75236,98210,81893,87666,106802)

hip <-cbind(Pred,Act)

bar <- read.csv("Bar.csv")

accuracy(Pred,Act)

## ME RMSE MAE MPE MAPE  
## Test set 4251.214 8181.213 5125.825 4.380836 5.582478

smoke <- matrix(c(67216.22,  
69396,  
71680.86,  
73000,  
71680.86,  
73868,  
67681.94,  
65496,  
66925.89,  
68232,  
73210.73,  
75236,  
85485.19,  
87666,  
88202.63,  
106802,  
84080.11,  
81893,  
81122.42,  
98210  
),ncol=2,byrow=TRUE)

smoke

## [,1] [,2]  
## [1,] 67216.22 69396  
## [2,] 71680.86 73000  
## [3,] 71680.86 73868  
## [4,] 67681.94 65496  
## [5,] 66925.89 68232  
## [6,] 73210.73 75236  
## [7,] 85485.19 87666  
## [8,] 88202.63 106802  
## [9,] 84080.11 81893  
## [10,] 81122.42 98210

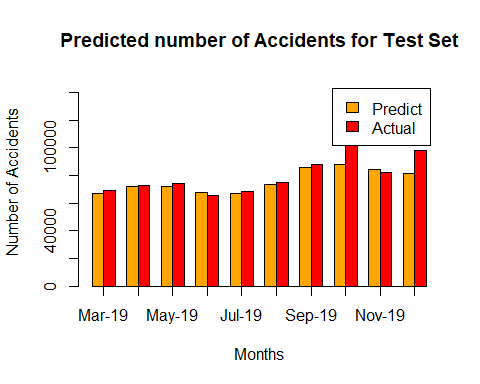
colnames(smoke) <- c("Predict","Actual")  
rownames(smoke) <- c("Mar-19","Apr-19","May-19","Jun-19","Jul-19","Aug-19","Sep-19","Oct-19","Nov-19","Dec-19"  
)

t(smoke)

## Mar-19 Apr-19 May-19 Jun-19 Jul-19 Aug-19 Sep-19  
## Predict 67216.22 71680.86 71680.86 67681.94 66925.89 73210.73 85485.19  
## Actual 69396.00 73000.00 73868.00 65496.00 68232.00 75236.00 87666.00  
## Oct-19 Nov-19 Dec-19  
## Predict 88202.63 84080.11 81122.42  
## Actual 106802.00 81893.00 98210.00

color.names = c("Orange","Red")

barplot(t(smoke),beside=T,ylim=c(0,150000),xlab="Months",ylab="Number of Accidents", col=color.names,axis.lty="solid", legend = TRUE, main = "Predicted number of Accidents for Test Set")

 ```