The Aggregators

Team F

4/25/2021

We are team F “The Aggregators”, we would like to analyse US Accidents data set, which has accidents data from 2016 to 2020.

**Introduction:**

Road accidents have become very common in recent years, Every year, over 38000 people die in road accidents in United States, and 4.4 million people are seriously injured enough to require medical attention. Source- <https://www.asirt.org/safe-travel/road-safety-facts/>

**Research question:**

We will try to find the trend of accident and the factors effecting accidents. We will also analyze the details of US Accidents in different states to be able to check what can be done to reduce accidents and also to avoid its effect on road traffic.

**Objective**

Looking at the severity of the road accidents, for this project we have decided to use the US Accident- A Countrywide Traffic Accident Dataset (2016-2000) to investigate and generate various insights on causes such as time, place etc. In first part, we have analyzed the basic information on accidents through out the United states such as 1. To determine the top 5 states and top 5 cities with respective to number of accidents. 2. To determine whether there is a trend in severity of accidents over passing years? 3. To determine the number of accidents per year while taking severity into account. 4. To determine which month has highest number of accidents. 5. To determine which days of the week has more number of accidents 6. To determine which part of the day has majority of accidents 7. To determine which time majority of accidents occur. 8. To determine accidents based on severity levels. 9. To determine severity of accident in different years. 10. To determine which side of the road has more accidents. 11. What are the top 10 weather conditions that contribute most accidents?

In part two, we have analyzed what all factors affecting on the road accidents. 1. Correlation between various severity and different variables 2. Number of accidents by states based on weather conditions and total accidents. 3. Number of accidents by month based on precipitation and temperature 4. How would you evaluate impact of humidity, wind chill and wind speed on severity of accidents? 5.Temperature affect on severity

Dataset information: Source: Kaggle (<https://www.kaggle.com/sobhanmoosavi/us-accidents>) Source of data- This dataset was collected by Bing, MapQuest & MapQuest-Bing using multiple Traffic APIs.

|  |  |
| --- | --- |
| Column Name | Description |
| ID | A unique identifier of accident record |
| Source | Types of API who reported Accidents |
| TMC | Traffic message channel code which defines the situation in detail. |
| Severity | Indicates effect of accident on road traffic by number between 1 and 4. Where 1 indicates short delays and 4 indicates Long delays |
| Start\_time | Shows start time of the accident in the local time zone. |
| End\_Time | Shows end time of the accident in the local time zone. |
| Start\_Lat | Shows latitude in GPS coordinate of the start point |
| Start\_Lng | Shows longitude in GPS coordinate of the start point. |
| End\_Lat | Shows latitude in GPS coordinate of the end point. |
| End\_Lng | Shows longitude in GPS coordinate of the end point. |
| Distance(mi) | The distance on the road affected by the accident. |
| Description | Accident’s senario is explained in words |
| Number | Specifies the street number in the address variable |
| Street | Indicates the street name in the address column. |
| Side | Shows the relative side of the street (Right/Left) in address field |
| City | Indicates the city in the address field |
| County | Depicts the county in the address field. |
| State | Shows the state in the address column |
| Zipcode | Shows the zipcode in the address field. |
| Country | Shows the country in the address field. |
| Timezone | Indicates time zone depending on the accidents location |
| Airport\_Code | Denotes an airport-based weather station which is the closest accident’s location. |
| Weather\_Timestamp | Specify the time-stamp of the weather observation record (in local time). |
| Temperature(F) | Shows the temperature (in Fahrenheit). |
| Wind\_Chill(F) | Shows the wind chill (in Fahrenheit). |
| Humidity(%) | Shows the humidity (in percentage). |
| Pressure(in) | Shows the air pressure (in inches). |
| Visibility(mi) | Shows visibility (in miles) |
| Wind\_Direction | Shows wind direction. |
| Wind\_Speed(mph) | Shows wind speed (in miles per hour). |
| Precipitation(in) | Indicates precipitation amount in inches, if there is any |
| Weather\_Condition | Specifies the type of Weather (rain, snow, thunderstorm, fog, etc. |
| Amenity | A POI annotation which indicates existence of amenity in a surrounding location. |
| Bump | A POI annotation which specifies presence of speed bump or hump in surrounding accident location. |
| Crossing | Indicates presence of crossing in a nearby location. |
| Give\_Way | Specifies presence of give\_way in a nearby location. |
| Junction | Denotes presence of junction in a at location of accidents |
| No\_Exit | Indicates existence of no\_exit at location of accident. |
| Railway | Implies existence of railway track or crossing |
| Roundabout | existence of roundabout at location of accident |
| Station | Shows the presence of a station in a surrounding incident location. |
| Stop | presence of stop in a nearby location. |
| Traffic\_Calming | A POI annotation which indicates presence of traffic\_calming in a nearby location. |
| Traffic\_Signal | A POI annotation which indicates presence of traffic\_signal within a few miles. |
| Turning\_Loop | Specifies the presence of turning\_loop in a nearby location. |
| Sunrise\_Sunset | Indicates the time of day (i.e.day or night) based on sunrise/sunset. |
| Civil\_Twilight | Shows the period of day (i.e.day or night) based on civil twilight. |
| Nautical\_Twilight | Shows the period of day (i.e.day or night) based on nautical twilight |
| Astronomical\_Twilight | Shows the period of day (i.e.day or night) based on astronomical twilight |

**Description:**

This is a country wide traffic accident data-set which covers 49 states of the US. There are about 4.2 million accident records in this data-set.It contains all sort of information related to each accident like the weather condition during the accident, which side the accident occurred, address, time of accident etc. There are a total of 49 observations.

Description of variables in dataset are mentioned below:

#run library  
  
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.0.4

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

## v ggplot2 3.3.3 v purrr 0.3.4  
## v tibble 3.0.5 v dplyr 1.0.3  
## v tidyr 1.1.2 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.0

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

library(cluster) #clustering algorithms  
library(factoextra) #clustering algorithms & visualization

## Warning: package 'factoextra' was built under R version 4.0.4

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

library(sparklyr)#

##   
## Attaching package: 'sparklyr'

## The following object is masked from 'package:purrr':  
##   
## invoke

library(usmap)#Plot all states of the U.S. to create an empty map.

## Warning: package 'usmap' was built under R version 4.0.5

library(ggplot2)#use ggplot2 to add layer for visualization  
library(plotly)

## Warning: package 'plotly' was built under R version 4.0.5

##   
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':  
##   
## last\_plot

## The following object is masked from 'package:stats':  
##   
## filter

## The following object is masked from 'package:graphics':  
##   
## layout

Import dataset

US\_Accidents <- read\_csv("US\_Accidents.csv")

##   
## -- Column specification --------------------------------------------------------  
## cols(  
## .default = col\_character(),  
## TMC = col\_double(),  
## Severity = col\_double(),  
## Start\_Time = col\_datetime(format = ""),  
## End\_Time = col\_datetime(format = ""),  
## Start\_Lat = col\_double(),  
## Start\_Lng = col\_double(),  
## End\_Lat = col\_logical(),  
## End\_Lng = col\_logical(),  
## `Distance(mi)` = col\_double(),  
## Number = col\_double(),  
## Weather\_Timestamp = col\_datetime(format = ""),  
## `Temperature(F)` = col\_double(),  
## `Wind\_Chill(F)` = col\_double(),  
## `Humidity(%)` = col\_double(),  
## `Pressure(in)` = col\_double(),  
## `Visibility(mi)` = col\_double(),  
## `Wind\_Speed(mph)` = col\_double(),  
## `Precipitation(in)` = col\_double(),  
## Amenity = col\_logical(),  
## Bump = col\_logical()  
## # ... with 11 more columns  
## )  
## i Use `spec()` for the full column specifications.

## Warning: 3032128 parsing failures.  
## row col expected actual file  
## 2716478 End\_Lat 1/0/T/F/TRUE/FALSE 40.11206 'US\_Accidents.csv'  
## 2716478 End\_Lng 1/0/T/F/TRUE/FALSE -83.03187 'US\_Accidents.csv'  
## 2716479 End\_Lat 1/0/T/F/TRUE/FALSE 39.86501 'US\_Accidents.csv'  
## 2716479 End\_Lng 1/0/T/F/TRUE/FALSE -84.04873 'US\_Accidents.csv'  
## 2716480 End\_Lat 1/0/T/F/TRUE/FALSE 39.10209 'US\_Accidents.csv'  
## ....... ....... .................. ......... ..................  
## See problems(...) for more details.

List column names of data set to

colnames(US\_Accidents)

## [1] "ID" "Source" "TMC"   
## [4] "Severity" "Start\_Time" "End\_Time"   
## [7] "Start\_Lat" "Start\_Lng" "End\_Lat"   
## [10] "End\_Lng" "Distance(mi)" "Description"   
## [13] "Number" "Street" "Side"   
## [16] "City" "County" "State"   
## [19] "Zipcode" "Country" "Timezone"   
## [22] "Airport\_Code" "Weather\_Timestamp" "Temperature(F)"   
## [25] "Wind\_Chill(F)" "Humidity(%)" "Pressure(in)"   
## [28] "Visibility(mi)" "Wind\_Direction" "Wind\_Speed(mph)"   
## [31] "Precipitation(in)" "Weather\_Condition" "Amenity"   
## [34] "Bump" "Crossing" "Give\_Way"   
## [37] "Junction" "No\_Exit" "Railway"   
## [40] "Roundabout" "Station" "Stop"   
## [43] "Traffic\_Calming" "Traffic\_Signal" "Turning\_Loop"   
## [46] "Sunrise\_Sunset" "Civil\_Twilight" "Nautical\_Twilight"   
## [49] "Astronomical\_Twilight"

Number of rows in data-set to know about length of dataset

nrow(US\_Accidents)

## [1] 4232541

Sampling US\_Accidents data-set: Since this is huge dataset to analyse as our R is taking time to process huge dataset we decided to sample dataset, since it is easier for us to analyse. The new data-set will contain 1 million rows

US\_Accidents\_Sample <- US\_Accidents[sample(nrow(US\_Accidents), 1000000, replace = FALSE, prob = NULL),]

Checking type of data frame to confirm it is a tibble

is\_tibble(US\_Accidents\_Sample)

## [1] TRUE

Export sample data to share with Team members

write\_csv(US\_Accidents\_Sample,"US\_Accidents\_Sample.csv")

Tidy data-set- This dataset has 1 million rows and it has many anomalies like the date and time format, multiple values for zipcode in the same row, etc. To address all these tidying data is an important step to be followed. Following code tidy up our data set.

#Changing the Start\_Time date format and converting into default format  
  
US\_Accidents\_Sample$Start\_Time <- as.POSIXct(US\_Accidents\_Sample$Start\_Time , format = '%Y/%m/%d %H:%M:%S', tz = 'UTC')

#Adding weekday column to the dataset for analysis  
US\_Accidents\_Sample$weekday <- weekdays(US\_Accidents\_Sample$Start\_Time)

#Separating the time and date value for "Start\_time" and "End\_time" Column using separate function.  
  
US\_Accidents\_Sample <- US\_Accidents\_Sample %>%   
 separate(Start\_Time, into= c("Accident\_year","Start\_Accident\_month", "Start\_Accident\_date", "Start\_Accident\_Hour","Start\_Accident\_min","Start\_Accident\_sec")) %>% separate(End\_Time, into= c("End\_Accident\_year","End\_Accident\_month", "End\_Accident\_date", "End\_Accident\_Hour","End\_Accident\_min","End\_Accident\_sec"))

Removed few columns since they are having same value through out the data for example country column,Also removed columns which doesn’t have any values except null for example end longitude and end latitude,Also removed few columns since they have multiple values which is breaking rule number 3 of tidy data for example zipcode which are not required for analysis. To know the column names or number, we first check the column name and then with select function we remove the unwanted columns. Assigning the changes to the same dataset (US\_Accidents\_Sample) to avoid confusion.

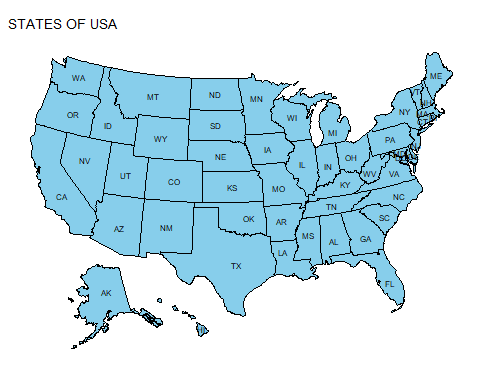
Changing few column names to make them more readable and consistance through out analysis to prevent confusion while performing analysis since some of these column names have parenthesis and different letter case.

colnames(US\_Accidents\_Sample) = c("ID","Source", "TMC","Severity", "Accident\_year", "Start\_Accident\_month","Start\_Accident\_date","Start\_Accident\_hour","Start\_Accident\_min", "Start\_Accident\_sec","End\_Accident\_year","End\_Accident\_month","End\_Accident\_date","End\_Accident\_Hour","End\_Accident\_min","End\_Accident\_sec","Start\_Lat","Start\_Lng","End\_Lat","End\_Lng","Distance", "Description","Number","Street","Side","City","County","State","Zipcode", "Country", "Timezone","Airport\_Code", "Weather\_TimeStamp", "Temperature","Wind\_Chill","Humidity","Pressure","Visibility","Wind\_Direction","Wind\_Speed","Precipitation","Weather\_Condition","Amenity","Bump","Crossing","Give\_Way","Junction","No\_Exit","Railway","Roundabout","Station","Stop","Traffic\_Calming","Traffic\_Signal","Turning\_loop", "Sunrise\_Sunset", "Civil\_Twilight", "Nautical\_Twilight","Astronomical\_Twilight", "Weekday")

US\_Accidents\_Sample <- US\_Accidents\_Sample %>%   
 mutate\_at(c(3,4,5,6,7,8,9,10,11,12,13,14,15,16), as.integer)

##Plotting US map

p<-plot\_usmap(regions = "states",labels = T,label\_color="black", fill="sky blue", color = "black") + labs(title = "STATES OF USA") +   
 theme(panel.background=element\_blank())  
 #Set label font size  
p$layers[[2]]$aes\_params$size <- 2  
print(p)



##To determine Top 5 States and Top 5 Cities with most number of accidents

by\_state <- US\_Accidents\_Sample %>%  
 group\_by(State) %>%  
 summarise(No.of\_accident = n()) %>%  
 arrange(desc(No.of\_accident)) %>%  
 mutate(percentage\_accident = round(No.of\_accident / sum(No.of\_accident) \* 100,2))  
  
  
#Summary of accidents by state- this shows the mean, median, mode of number of accidents per state. This also gives us information on quaterly accidents.  
summary(by\_state)

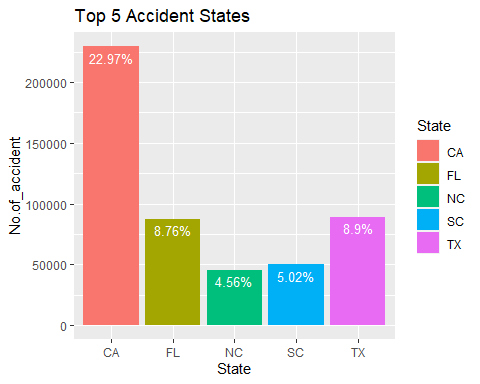
## State No.of\_accident percentage\_accident  
## Length:49 Min. : 50 Min. : 0.00   
## Class :character 1st Qu.: 1843 1st Qu.: 0.18   
## Mode :character Median : 10335 Median : 1.03   
## Mean : 20408 Mean : 2.04   
## 3rd Qu.: 24898 3rd Qu.: 2.49   
## Max. :229707 Max. :22.97

#graph showing top 5 states with highest number of accidents.  
top\_states <- top\_n(by\_state, 5)

## Selecting by percentage\_accident

ggplot(data = top\_states, aes(x = State, y = No.of\_accident, fill = State)) +  
 geom\_histogram(stat = "identity") +  
 ggtitle("Top 5 Accident States") +  
 geom\_text(aes(label=paste0(round(percentage\_accident,2),"%")),  
 vjust = 1.5,  
 color = "white",  
 size = 3.5  
 )+   
 theme\_grey()

## Warning: Ignoring unknown parameters: binwidth, bins, pad

 #Observations: From the graph top 5 accident states we can conclude that California has highest number of accidents with almost 23% of all accidents in USA.The next state after California is Texas with approx 9% of the total number of accidents. It would be interesting to know what factors contribute sharp decline in number of accidents in Texas considering that TX is twice the size of CA. We think one of the reasons could be density of population in california is higher when compared with other states.

#Code Explaination

To know about number of accidents in top 5 states here we have grouped data by state then summarized it to get the count of accidents by state and arranged them in descending order to find top states, we believe, bar graphs are easy to understand better to visualize data we plotted this set of data under bar graphs.

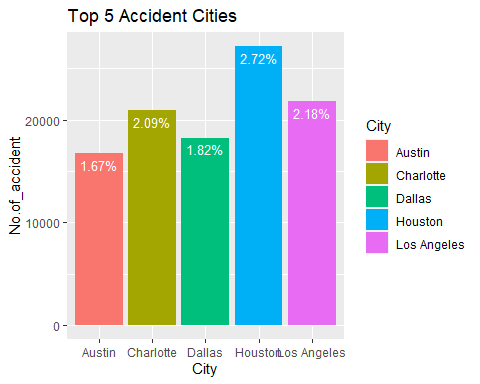
## Top 5 cities with most number of accidents in USA

by\_cities<- US\_Accidents\_Sample %>%   
 group\_by(City) %>%   
 summarise(No.of\_accident = n()) %>%   
 arrange(desc(No.of\_accident)) %>% mutate(percentage\_accident1 = round(No.of\_accident/sum(No.of\_accident)\*100,2))  
  
top\_cities<- top\_n(by\_cities, 5)

## Selecting by percentage\_accident1

ggplot(data = top\_cities, aes(x = City, y = No.of\_accident, fill = City)) +  
 geom\_histogram(stat = "identity") +  
 ggtitle("Top 5 Accident Cities") +  
 geom\_text(aes(label=paste0(round(percentage\_accident1,2),"%")),  
 vjust = 1.5,  
 color = "white",  
 size = 3.5  
 )+   
 theme\_grey()

## Warning: Ignoring unknown parameters: binwidth, bins, pad



## Observation:

From Top 5 accidents cities graph, we observed that most of the accidents are in Houston which is part of Texas and not a part of California, apart from that there are 3 cities from Texas region that are in top 5 cities, but California tops the state wise list. One of the reason we could think of is traffic law enforcement in Texas is different from other states.

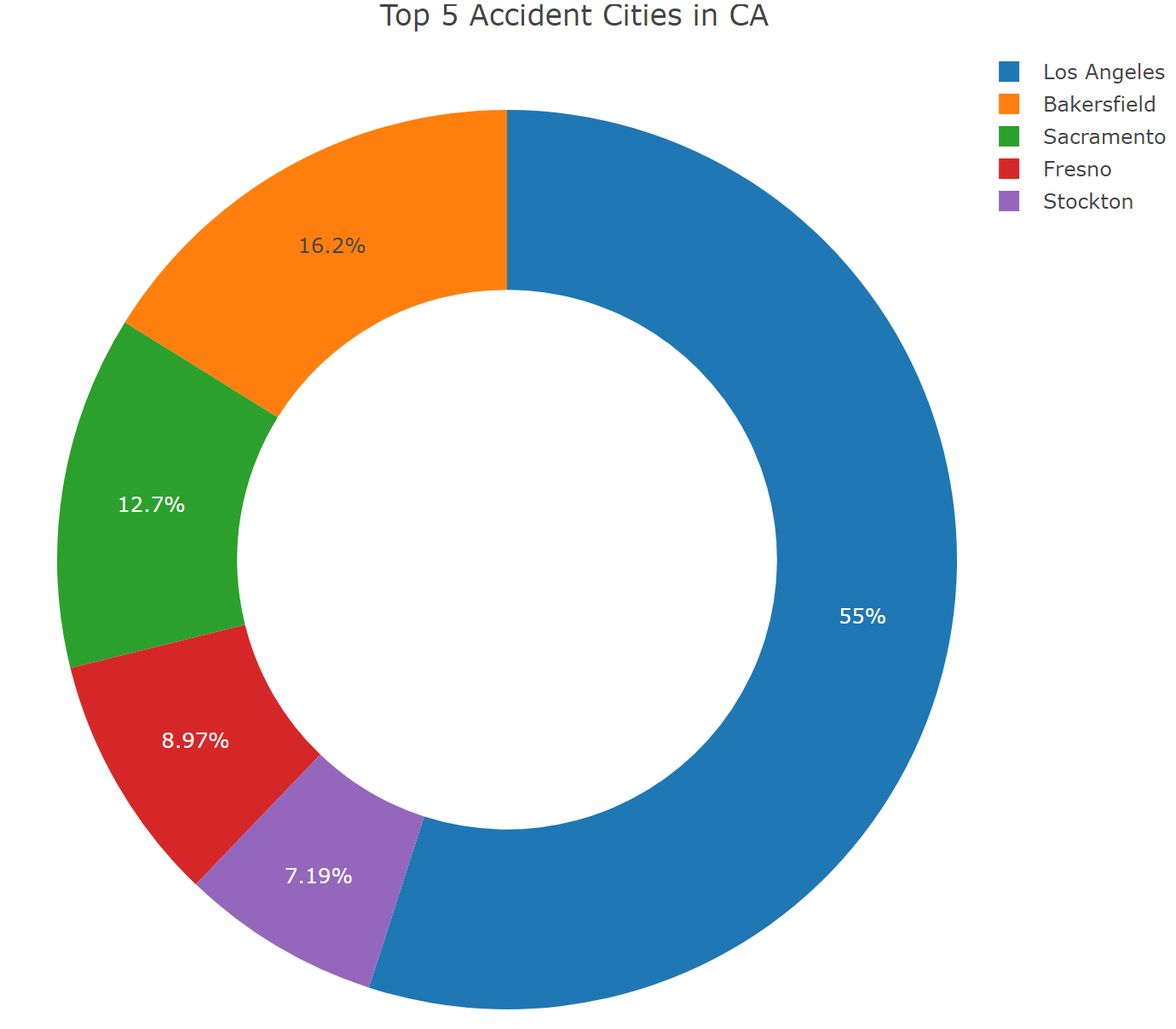
#Code explaination: To know about number of accidents in top 5 cities here we have grouped data by cities then summarized it to get the count of accidents by city and then found the percentage of accidents and arranged them in descending order to find top Cities, we believe, bar graphs are easy to understand better to visualize data we plotted this set of data under bar graphs.

##Top 5 cities in califrnia with highest number of accidents

by\_city<- US\_Accidents\_Sample %>%  
 filter(State =="CA") %>%   
 group\_by(City) %>%  
 summarise(No.of\_accident = n())  
  
top\_california\_cities<- top\_n(by\_city, 5)

## Selecting by No.of\_accident

fig <- top\_california\_cities %>%   
top\_n(6,wt=No.of\_accident)%>% plot\_ly(labels = ~City, values = ~No.of\_accident)  
  
fig <- fig %>% add\_pie(hole = 0.6)  
  
fig <- fig %>% layout(title = "Donut Charts(Number of Accidents as per TOP 5 CA cities )", showlegend = T, xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE), yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE))%>% print(fig)



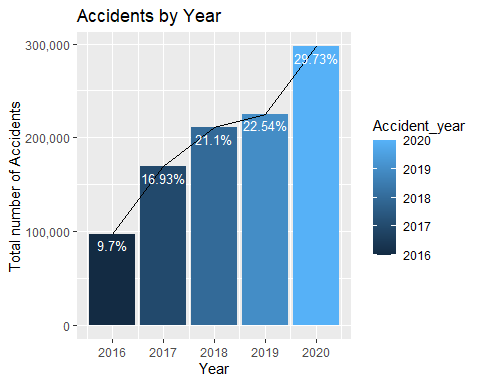
## Observation:

From this top 5 cities in California Donut chart we can observe that Los Angeles is the top city in California with a highest number of accidents and contributes about 55% of all the accidents in California, we think reasons could be population density and road infrastructure #Code Explaination: To know about number of accidents in top 5 cities in California here we have grouped data by cities then filtered data for state California and summarized it to get the count of accidents by city and then found the percentage of accidents and arranged them in descending order to find top Cities, since we are representing a certain percentage of cities in particular state we used donut charts since they are easy to understand better to visualize data, we plotted this set of data under donut chart.

## Accidents Trends by Year, Month, Week, Day & time

###Accident Per Year- To demonestrate and visualize the number of accidents per year.

by\_year<- US\_Accidents\_Sample %>%   
 group\_by(Accident\_year) %>%   
 summarise(total\_accident= n()) %>%   
 mutate(percentage= round(total\_accident/sum(total\_accident)\*100,2))  
  
ggplot(data = by\_year) +  
 geom\_col(mapping = aes(x = Accident\_year, y = total\_accident, fill = Accident\_year))+  
 geom\_line(mapping = aes (x = Accident\_year, y = total\_accident ))+  
 geom\_text(  
 aes(x = Accident\_year, y = total\_accident, label=paste0(round(percentage,2),"%")),  
 vjust = 1.5,  
 color = "white",  
 size = 3.5)+labs(x="Year", y="Total number of Accidents")+  
 ggtitle("Accidents by Year") +scale\_y\_continuous(labels = scales::comma)

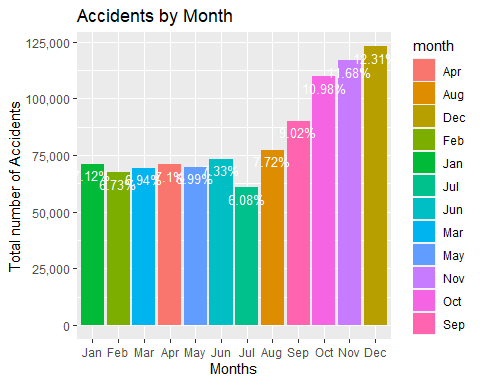


## Observation:

Number of accidents are increasing every passing year , here we can see that from year 2016 to 2020 there is a 3-fold increase in percent of accidents, one of the reason we could think of is Expanding the city limits. #Code explaination: To know about percent of accidents in every year we used Barchat since we want to compare it with every passing year, since This data has a increasing trend we taught line graph will depict the data more accurately since line chart will add depth to the bar chart. ### Accident Per Month- To demonestrate and visualize the number of accidents per month.

US\_Accidents\_Sample <- transform(US\_Accidents\_Sample, month = month.abb[Start\_Accident\_month])  
  
by\_month <- US\_Accidents\_Sample %>%  
 group\_by(month) %>%  
 summarise(Total\_Accident = n()) %>%  
 mutate(Percentage = round(Total\_Accident / sum(Total\_Accident) \*  
 100, 2))  
  
ggplot(data = by\_month) +  
 geom\_col(mapping = aes(x = month, y = Total\_Accident, fill = month))+  
 geom\_line(mapping = aes (x = month, y = Total\_Accident, ))+  
 geom\_text(  
 aes(x = month, y = Total\_Accident, label=paste0(round(Percentage,2),"%")),  
 vjust = 1.5,  
 color = "white",  
 size = 3.5)+ labs(x="Months", y="Total number of Accidents")+  
 ggtitle("Accidents by Month") +scale\_y\_continuous(labels = scales::comma) + scale\_x\_discrete(limits = month.abb)

## geom\_path: Each group consists of only one observation. Do you need to adjust  
## the group aesthetic?



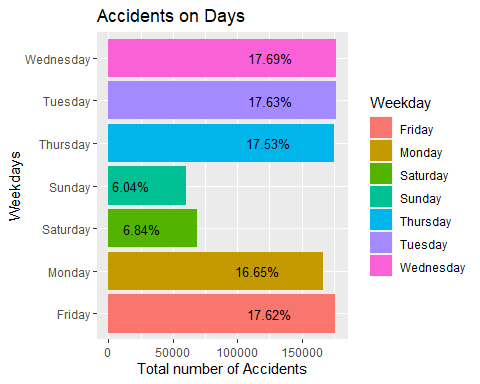
## Observation:

This accidents by month chart shows that July has the least percentage of accidents and December has the highest percentage of accidents which is the holiday season in the USA, it is quite surprising that holidays have an significant effect on accidents. #Code Explaination: To know about number of accidents in different months here we have grouped data by month then summarized it to get the count of accidents by month, we believe, bar graphs are easy to understand better to visualize data we plotted this set of data under bar graphs.

##Accident Per days of Week-To demonestrate and visualize the number of accidents per days of week.

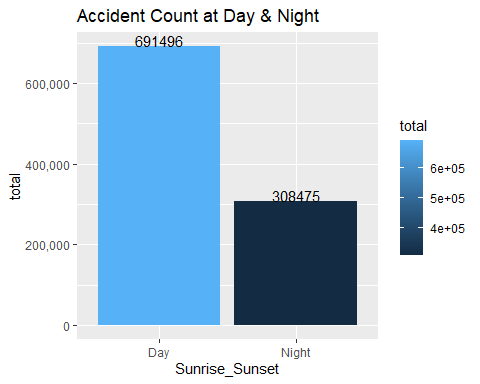
by\_weekday<- US\_Accidents\_Sample %>%  
select(Weekday) %>% group\_by(Weekday) %>%  
summarise(Total\_Accident= n())  
   
ggplot(data = by\_weekday) +  
geom\_col(mapping = aes(x = Weekday, y = Total\_Accident, fill = Weekday))+  
geom\_line(mapping = aes (x = Weekday, y = Total\_Accident))+coord\_flip() +geom\_text(  
aes(x = Weekday, y = Total\_Accident, label=paste0(round(Total\_Accident/sum(Total\_Accident)\*100,2),"%")),  
vjust = 0.5, hjust=2,  
color = "black",  
size = 3.5)+ ggtitle("Accidents on Days") +labs(y="Total number of Accidents", x="Weekdays")

## geom\_path: Each group consists of only one observation. Do you need to adjust  
## the group aesthetic?

 #observation: This accidents on days chart we could see that most of the accidents tend to takes place on weekdays from monday to friday,from the graph we can see that both saturday and sundays total accidents is approximately 13% which is less than daily week day average(from monday to friday), one of the reason we could think of would be most people tend to commute to work on weekdays rather than weekends. #Code Explaination: To know about number of accidents in different days of the week here we have grouped data by days of the week then summarized it to get the count of accidents by days of week, we believe, bar graphs are easy to understand better to visualize data we plotted this set of data under bar graphs. since, days o f the week would overlap if we use regular bar graph so we have decided to use Flipped bar graph

## What part of the day do majority of accidents occur?

day\_night <- US\_Accidents\_Sample %>% select(Severity, Sunrise\_Sunset) %>% group\_by(Sunrise\_Sunset) %>%   
 summarise(total = n()) %>% na.omit  
   
ggplot(data = day\_night)+  
 geom\_col(mapping = aes(x=Sunrise\_Sunset, y= total, fill= total))+  
 scale\_y\_continuous(labels = scales::comma)+  
 geom\_text(aes(x = Sunrise\_Sunset, y = total, label= (total), vjust=0.02))+  
 ggtitle("Accident Count at Day & Night")

 #

**Observation**

From the bar graph we can see that most of the accidents occurs in daytime. One possible reason that we could think of is due to higher commute during day compared to night. #code Explaination: To know about number of accidents in different parts of the day here we have grouped data by day and night then summarized it to get the count of accidents by day and night, we believe, bar graphs are easy to understand better to visualize data, so we have plotted this data set under bar graphs and displayed total count on the top of each bar.

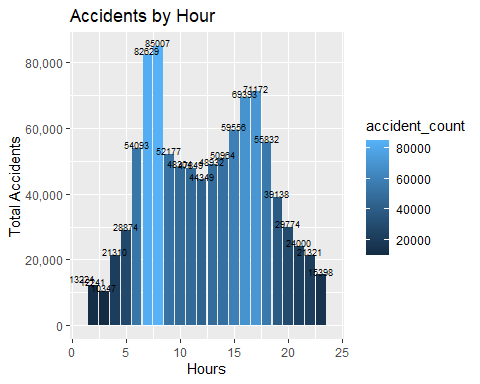
Analyzing the distribution of the accidents and checking at which time majority of these accidents occur.As now we know that day has majority of accidents, now we are trying to find which time of the day has most accidents. # By time- which time of the day has more accidents?

by\_timeofDay<-  
 US\_Accidents\_Sample %>%   
 group\_by(Start\_Accident\_hour) %>%   
 summarise(accident\_count= n()) %>%   
 na.omit  
  
ggplot(data = by\_timeofDay) +  
 geom\_col(mapping = aes(x = Start\_Accident\_hour, y = accident\_count, fill = accident\_count ))+  
 geom\_text(  
 aes(x = Start\_Accident\_hour, y = accident\_count, label= (accident\_count)),  
 vjust = 0.01,  
 color = "black",  
 size = 2.5)+  
 ggtitle("Accidents by Hour")+ labs(y="Total Accidents", x="Hours") +scale\_y\_continuous(labels = scales::comma)+ scale\_x\_continuous(limits = c(1, 24))

## Warning: Removed 1 rows containing missing values (position\_stack).

## Warning: Removed 1 rows containing missing values (geom\_col).

## Warning: Removed 1 rows containing missing values (geom\_text).

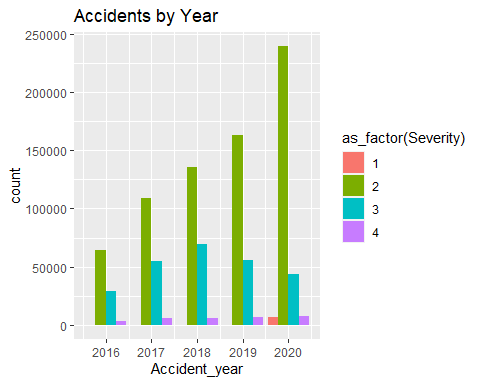


## Observation:

From the accident by hour graph, we can see that most of the accidents occurs at 7am, 8am, 4pm and 5pm. One of the reasons could be these are office starting hours and most of the people leave for work and come back from work at these timings #Code Explaination: To know about number of accidents in hours of the day here we have grouped data by hours of the day and then summarized it to get the count of accidents by hours of the day, we believe, bar graphs are easy to understand better to visualize data, so we have plotted this data set under bar graphs and displayed total count on the top of each bar. ## Accidents based on the severity levels and years.

by\_severity<- US\_Accidents\_Sample %>%  
 group\_by(Severity) %>%   
 summarise(No.of\_accident = n())  
  
severity\_donut <- by\_severity %>% plot\_ly(labels = ~Severity, values = ~No.of\_accident)%>% add\_pie(hole = 0.4)  
  
severity\_donut <- severity\_donut %>% layout(title ="Severity based on Accidents",showlegend = T, xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE), yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE))%>% print(severity\_donut)

by\_year<- US\_Accidents\_Sample %>%   
 group\_by(Accident\_year) %>%   
 summarise(total\_accident= n())  
  
ggplot(data = US\_Accidents\_Sample)+  
 geom\_bar(mapping = aes(x= Accident\_year, fill= as\_factor(Severity)), position = "dodge")+  
 ggtitle("Accidents by Year")+  
 theme\_grey()

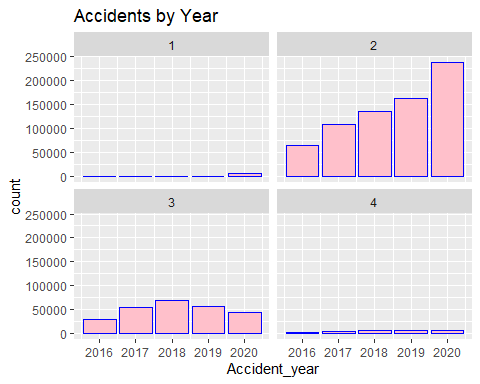


## Observation:

In this dataset severity means how much traffic got affected by accident which means severity 3 and 4 will have higher impact on road traffic when compared with 1 and 2, From severity based on accidents, we see that most of the accidents are of severity level 2. #Code Explaination: since severity is the categorical variable we plotted severity levels in donut chart, which would help to find individual count on categories, and comparing different categories in data.

# Severity of accidents in different years

# to know in depth of details of severiy levels in each year we plotted them individually.  
ggplot(data = US\_Accidents\_Sample)+  
 geom\_bar(mapping = aes(x= Accident\_year,colour = "Severity"), colour = "blue", fill = "pink")+  
 ggtitle("Accidents by Year")+  
 theme\_grey()+  
 facet\_wrap(~Severity)

 #Observation: It also shows that severity 2 accidents tend to increase gradually by every passing year. And number of severity 3 accidents has declined after 2018 which has a “positive effect” since there is a control in severity 3 accidents from 2018. Apart from that Severity 1 and 4 accidents are rare and have been consistent in number over years

#Code explaination: To know about number of accidents with severity into consideration here we have grouped data by year and then summarized it to get the count of accidents by year, and then used facet function to plot them individually we believe, bar graphs are easy to understand better to visualize data, so we have plotted this data set under bar graphs and displayed total count on the top of each bar. #Part-2

# accidents by side- Sparkly R

##Creating Spark Connect

sc <- spark\_connect(master = "local", version = "2.3") #create spark connection  
  
df\_Spark <- sapply(read.csv("US\_Accidents\_Sample.csv"), class)  
  
df\_Spark <- spark\_read\_csv(sc,"US\_Accidents\_Sample.csv")#load data in Spark

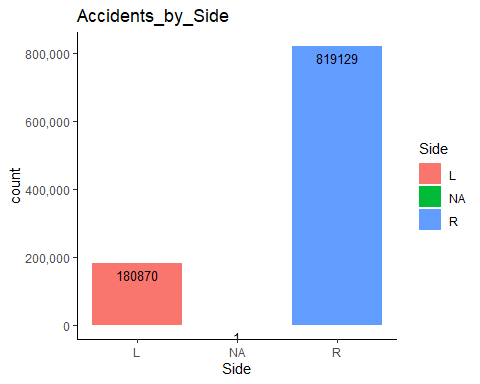
# accidents by side- Sparkly R

#Analyze which side of the road has more accidents

by\_side <- df\_Spark %>%   
 group\_by(Side) %>%   
 summarise(count = n()) %>%   
 collect() %>%  
 print()

## # A tibble: 3 x 2  
## Side count  
## <chr> <dbl>  
## 1 L 180870  
## 2 R 819129  
## 3 NA 1

# Here we have used group\_by() to group Side variable,summarise() will Collapse many values down to a single summary into variable named count.The Pipe %>% operator is used to update a value by first piping it into one or more expressions, and then assigning the result.  
 ggplot(data = by\_side, mapping = aes (x= Side,y= count, fill = Side))+  
 geom\_bar(stat="identity")+  
 ggtitle("Accidents\_by\_Side")+  
 geom\_text(aes(label= count), vjust=1.5, color="black", size=3.5)+  
 scale\_y\_continuous(labels = scales::comma)+  
 theme\_classic()



## Observation:

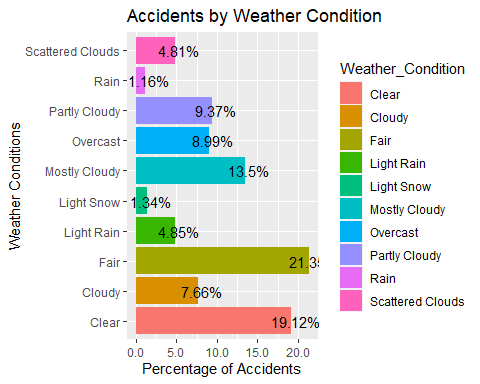
The graph shows that most of the accidents occurs at the right side of the road whereas left side are very less number of accidents. It is quite surprising because the left most lanes is the fastest one. One reason for this could be high number of lane merging that vehicles do while entering or exiting the express. #code explaination: we used sparklyr to perform analysis, we pushed our data from r to spark and performed computations such as summarise and collected the results and displayed them in r # Weather conditions- Analyze number of accidents in different Weather conditions along with to 10 weather conditions that contributes the most.

by\_weather\_condition<- US\_Accidents\_Sample %>%   
 group\_by(Weather\_Condition) %>%   
 summarise(Total\_Accident=n()) %>%   
 mutate(percentage= round(Total\_Accident/sum(Total\_Accident)\*100,2)) %>% na.omit  
  
top10\_WC<- by\_weather\_condition %>% top\_n(10)

## Selecting by percentage

ggplot(data = top10\_WC) +  
 geom\_col(mapping = aes(x = Weather\_Condition, y = percentage, fill = Weather\_Condition ))+ coord\_flip()+  
 geom\_line(mapping = aes (x = Weather\_Condition, y = percentage, ))+  
 geom\_text(  
 aes(x = Weather\_Condition, y = percentage, label=paste0(round(percentage,2),"%")),  
 hjust = 0.4,  
 color = "black",  
 size = 4)+  
 ggtitle("Accidents by Weather Condition")+ labs(y="Percentage of Accidents", x="Weather Conditions") +scale\_y\_continuous(labels = scales::comma)

## geom\_path: Each group consists of only one observation. Do you need to adjust  
## the group aesthetic?



## Observation:

Counter intuitively, most of the accidents appear to occur in fair and clear weather conditions, one of the reason could be most of the people tend to drive fast in clear and fair weather which results in most of the accidents. #code explaination: To know about number of accidents in different weather conditions here we have grouped data by different weather conditions then summarized it to get the count of accidents by different weather conditions, we believe, bar graphs are easy to understand better to visualize data we plotted this set of data under bar graphs. since,different weather conditions would overlap would overlap if we use regular bar graph so we have decided to use Flipped bar graph

**Severity correlation**

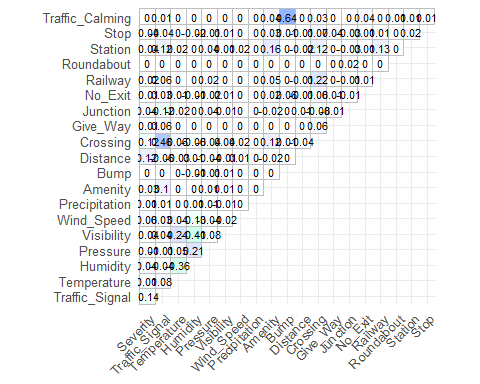
df <- US\_Accidents\_Sample %>%   
 mutate\_at(c(4,54,34,36,37,38,40, 41,43,44,21,45, 46, 47, 48, 49, 50, 51, 52, 53, 54), as.integer) %>%   
 select(4,54,34,36,37,38,40, 41,43,44,21,45, 46, 47, 48, 49, 50, 51, 52, 53, 54) %>% na.omit  
  
library(corrr)

## Warning: package 'corrr' was built under R version 4.0.4

library(ggcorrplot)

## Warning: package 'ggcorrplot' was built under R version 4.0.5

options(repr.plot.width=12, repr.plot.height=12)  
corr <- round(cor(df, use="complete.obs"), 2)  
ggcorrplot(corr, lab = TRUE,colors = c("aquamarine", "white", "dodgerblue"),   
 show.legend = F, outline.color = "gray", type = "upper",   
 tl.cex = 10, lab\_size = 3, sig.level = .1) +  
 labs(fill = "Correlation")

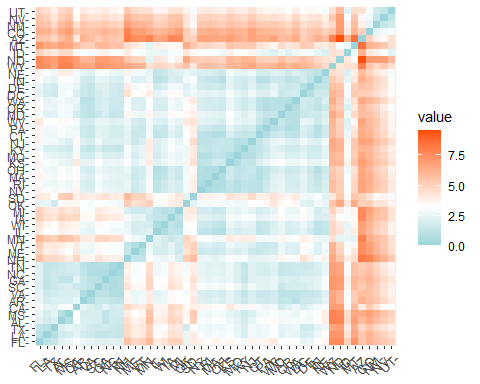
 #

**Observations:**

From this correlation Metrix we can see that traffic signal and crossing, visibility have negative affect on Severity, which is very obvious since if there is more visibility there will be less severity. #Code explaination: here we have used corr function to correlate different variable in the format as shore in the figure, we want to find correlation between various variables and severity.

#CLUSTERING

#Data Manipulation for Analysis  
#selecting the columns that has weather information, removing NAs   
  
weather <- US\_Accidents\_Sample %>%   
 select(c(State, Temperature:Precipitation, -Wind\_Direction)) %>%   
 na.omit()   
  
  
by\_weather\_cluster <- weather %>%   
 group\_by(State) %>%   
 summarise\_all(mean) %>%   
 remove\_rownames %>% column\_to\_rownames(var="State")  
  
by\_weather\_cluster <- weather %>%   
 group\_by(State) %>%   
 summarise\_all(mean) %>%   
 remove\_rownames %>% column\_to\_rownames(var="State") %>%   
 scale()  
  
  
#This starts to illustrate which states have large dissimilarities (red) versus those that appear to be fairly similar (teal)  
  
#get\_dist: for computing a distance matrix between the rows of a data matrix. The default distance computed is the Euclidean  
#fviz\_dist: for visualizing a distance matrix  
  
distance <- get\_dist(by\_weather\_cluster)  
fviz\_dist(distance, gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07"))



##While we now have a statistical and visual understanding of the distance among the observations (rows) in our dataset. We need to use clustering to assign those observations in different groups (clusters)

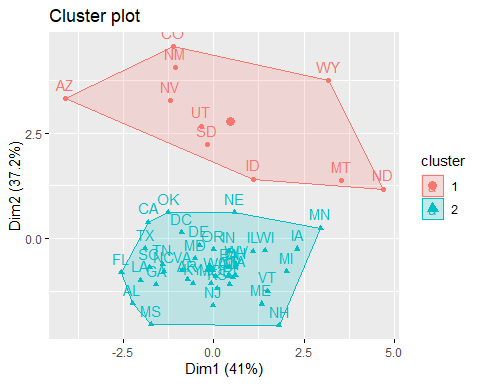
k2 <- kmeans(by\_weather\_cluster, centers = 2, nstart = 25)  
str(k2)

## List of 9  
## $ cluster : Named int [1:49] 2 2 1 2 1 2 2 2 2 2 ...  
## ..- attr(\*, "names")= chr [1:49] "AL" "AR" "AZ" "CA" ...  
## $ centers : num [1:2, 1:7] -0.547 0.14 -0.571 0.146 -1.252 ...  
## ..- attr(\*, "dimnames")=List of 2  
## .. ..$ : chr [1:2] "1" "2"  
## .. ..$ : chr [1:7] "Temperature" "Wind\_Chill" "Humidity" "Pressure" ...  
## $ totss : num 336  
## $ withinss : num [1:2] 107 129  
## $ tot.withinss: num 236  
## $ betweenss : num 100  
## $ size : int [1:2] 10 39  
## $ iter : int 1  
## $ ifault : int 0  
## - attr(\*, "class")= chr "kmeans"

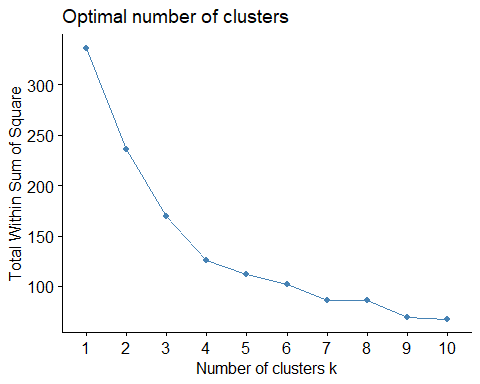
tidy(k2) #the tidy() function summarizes on a per-cluster level

## # A tibble: 2 x 10  
## Temperature Wind\_Chill Humidity Pressure Visibility Wind\_Speed Precipitation  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 -0.547 -0.571 -1.25 -1.71 0.753 0.592 -1.40   
## 2 0.140 0.146 0.321 0.437 -0.193 -0.152 0.360  
## # ... with 3 more variables: size <int>, withinss <dbl>, cluster <fct>

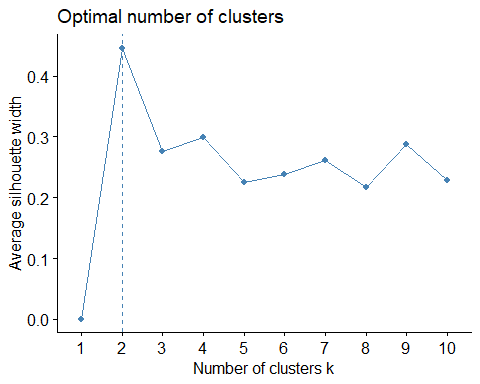
fviz\_cluster(k2, data = by\_weather\_cluster)



### Elbow Method  
  
set.seed(1234)  
fviz\_nbclust(by\_weather\_cluster, kmeans, method = "wss")



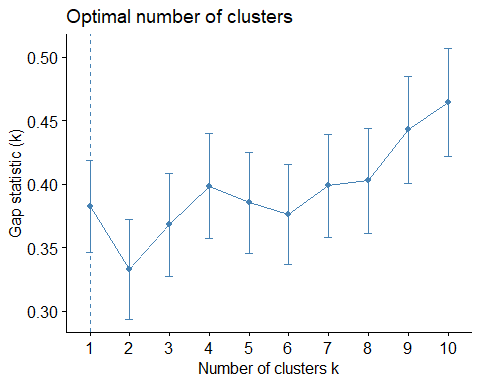
### Average Silhouette Method  
  
fviz\_nbclust(by\_weather\_cluster, kmeans, method = "silhouette")



# compute gap statistic  
set.seed(1234)  
gap\_stat <- clusGap(by\_weather\_cluster, FUN = kmeans, nstart = 25,  
 K.max = 10, B = 50)  
# Print the result  
print(gap\_stat, method = "firstmax")

## Clustering Gap statistic ["clusGap"] from call:  
## clusGap(x = by\_weather\_cluster, FUNcluster = kmeans, K.max = 10, B = 50, nstart = 25)  
## B=50 simulated reference sets, k = 1..10; spaceH0="scaledPCA"  
## --> Number of clusters (method 'firstmax'): 1  
## logW E.logW gap SE.sim  
## [1,] 3.694131 4.076827 0.3826953 0.03630445  
## [2,] 3.499548 3.832552 0.3330034 0.03918193  
## [3,] 3.325403 3.693493 0.3680894 0.04077185  
## [4,] 3.186809 3.585127 0.3983175 0.04130245  
## [5,] 3.113704 3.499256 0.3855521 0.03979898  
## [6,] 3.051609 3.427957 0.3763480 0.03942578  
## [7,] 2.964881 3.363631 0.3987494 0.04062901  
## [8,] 2.899127 3.301938 0.4028111 0.04110099  
## [9,] 2.799963 3.242808 0.4428451 0.04188080  
## [10,] 2.720694 3.185147 0.4644531 0.04241133

fviz\_gap\_stat(gap\_stat)



##Final  
k2 <- kmeans(by\_weather\_cluster, centers = 4, nstart = 25)  
str(k2)

## List of 9  
## $ cluster : Named int [1:49] 4 4 2 4 2 1 4 4 4 4 ...  
## ..- attr(\*, "names")= chr [1:49] "AL" "AR" "AZ" "CA" ...  
## $ centers : num [1:4, 1:7] -0.364 0.402 -1.932 0.875 -0.354 ...  
## ..- attr(\*, "dimnames")=List of 2  
## .. ..$ : chr [1:4] "1" "2" "3" "4"  
## .. ..$ : chr [1:7] "Temperature" "Wind\_Chill" "Humidity" "Pressure" ...  
## $ totss : num 336  
## $ withinss : num [1:4] 36.7 29.7 25.5 33.6  
## $ tot.withinss: num 126  
## $ betweenss : num 210  
## $ size : int [1:4] 21 6 5 17  
## $ iter : int 3  
## $ ifault : int 0  
## - attr(\*, "class")= chr "kmeans"

tidy(k2) #the tidy()

## # A tibble: 4 x 10  
## Temperature Wind\_Chill Humidity Pressure Visibility Wind\_Speed Precipitation  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 -0.364 -0.354 0.395 0.396 -0.714 0.122 0.469  
## 2 0.402 0.391 -2.23 -1.84 1.57 0.222 -1.40   
## 3 -1.93 -1.97 0.296 -1.18 -0.601 1.06 -1.31   
## 4 0.875 0.880 0.212 0.507 0.503 -0.541 0.302  
## # ... with 3 more variables: size <int>, withinss <dbl>, cluster <fct>

library(tidyverse)  
# install.packages(modelr)  
library(modelr)   
# provides easy pipeline modeling  
library(broom)   
# helps to tidy up model outputs

#Data Manipulation for Analysis  
  
US\_Accident\_Sample <- read\_csv("US\_Accident\_Sample.csv")  
  
weather <- US\_Accident\_Sample %>%   
 select(c(State, Temperature:Precipitation, -Wind\_Direction)) %>%   
 na.omit() %>%   
 group\_by(State) %>%   
 summarise\_all(mean)  
  
total\_accident <- US\_Accident\_Sample %>%   
 group\_by(State) %>%   
 summarise(count= n())  
  
#merging weather and total\_accident dataframe  
  
df <- merge(weather, total\_accident)  
  
##Combining population dataset to df using merge function  
  
#importing population data  
US\_Population<- read\_csv("US\_Population.csv")  
  
#combining two dataset  
df\_final <- merge(df,US\_Population)  
  
## Data Preparation  
  
#Here we will use a conventional 60% / 40% split where we train our model on 60% of the data and then test the model performance on 40% of the data that is withheld.  
  
set.seed(123)  
sample <- sample(c(TRUE, FALSE), nrow(df\_final), replace = T, prob = c(0.6,0.4))  
train <- df\_final[sample, ]  
test <- df\_final[!sample, ]

## Linear Regression

#### Model3- Linear Regression

## model building formula, change variables as per dataset  
model3 <- lm(count ~ Population, data = train)  
  
# Check model output   
summary(model3)

##   
## Call:  
## lm(formula = count ~ Population, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -12692 -5040 -546 1390 34996   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.972e+03 2.426e+03 -0.813 0.423   
## Population 6.846e-04 6.011e-05 11.390 5.04e-12 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 9621 on 28 degrees of freedom  
## Multiple R-squared: 0.8225, Adjusted R-squared: 0.8162   
## F-statistic: 129.7 on 1 and 28 DF, p-value: 5.044e-12

tidy(model3)

## # A tibble: 2 x 5  
## term estimate std.error statistic p.value  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) -1972. 2426. -0.813 4.23e- 1  
## 2 Population 0.000685 0.0000601 11.4 5.04e-12

#### Interpretation-

First, we check variables significance. We can see from the summary of our models that our coefficients for Population and Temperature is statistically significant (p-value < 0.05). It also shows that for every 1 unit increase in population there is 0.0006846 unit increase in number of accidents

Next, we want to understand the extent to which the model fits the data i.e. Goodness of fit

Residual standard error (RSE)- The RSE provides an absolute measure of lack of fit of our model to the data. But since it is measured in the units of Y, our response variable, it is not always clear what constitutes a good RSE

R squared (R2)- The result suggests that our model with 2 predictors can explain 83% of the variability in our accident data. Also, the difference between R2 and adj. R2 is not much, which indicates that both variable are significant.

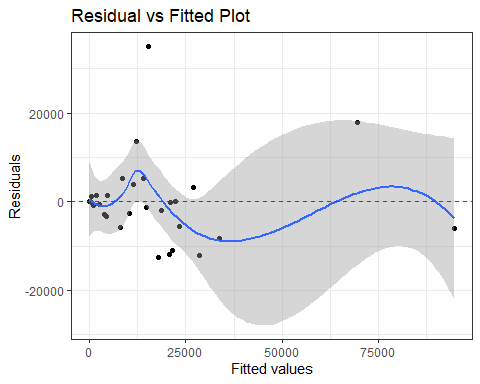
F-statistic- In our summary print out above for model 1 we see that F= 70.76 with p<0.05 suggesting that the variables are related to total number of accidents.

Combined, our RSE, R2, and F-statistic results suggest that our model has a good fit.

confint(model3)

## 2.5 % 97.5 %  
## (Intercept) -6.940428e+03 2.996839e+03  
## Population 5.615166e-04 8.077607e-04

#Assessing Our Model Visually  
  
model3\_results <- augment(model3, train) %>%  
 mutate(Model = "Model 1")  
  
object\_name\_m3 <- ggplot(model3\_results, aes(.fitted, .resid))+   
 geom\_point()+  
 stat\_smooth(method="loess")+  
 geom\_hline(yintercept=0, col="red", linetype="dashed")+  
 xlab("Fitted values")+  
 ylab("Residuals")+  
 ggtitle("Residual vs Fitted Plot")+  
 theme\_bw()  
  
object\_name\_m3



#Applying square root transformation  
   
model3a <- lm(sqrt(count) ~ Population, data = train)  
   
# Check model output   
   
summary(model3a)

##   
## Call:  
## lm(formula = sqrt(count) ~ Population, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -57.685 -32.150 -1.974 23.433 122.888   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.621e+01 1.011e+01 4.57 8.99e-05 \*\*\*  
## Population 2.190e-06 2.506e-07 8.74 1.72e-09 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 40.11 on 28 degrees of freedom  
## Multiple R-squared: 0.7318, Adjusted R-squared: 0.7222   
## F-statistic: 76.39 on 1 and 28 DF, p-value: 1.722e-09

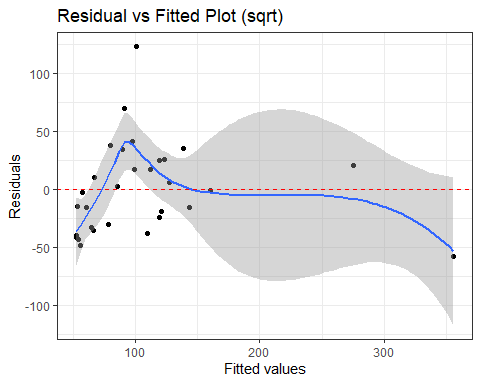
tidy(model3a)

## # A tibble: 2 x 5  
## term estimate std.error statistic p.value  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) 46.2 10.1 4.57 0.0000899   
## 2 Population 0.00000219 0.000000251 8.74 0.00000000172

# Assessing Coefficients  
  
confint(model3a)

## 2.5 % 97.5 %  
## (Intercept) 2.549560e+01 6.692315e+01  
## Population 1.676828e-06 2.703397e-06

#Assessing Our Model Visually  
object\_name\_m3a <- ggplot(model3a, aes(.fitted, .resid))+   
 geom\_point()+  
 stat\_smooth(method="loess")+  
 geom\_hline(yintercept=0, col="red", linetype="dashed")+  
 xlab("Fitted values")+  
 ylab("Residuals")+  
 ggtitle("Residual vs Fitted Plot (sqrt)")+  
 theme\_bw()  
   
object\_name\_m3a



#Try with applying a log transformation  
model3b <- lm(log(count) ~ Population, data = train)  
   
 # Check model output   
summary(model3b)

##   
## Call:  
## lm(formula = log(count) ~ Population, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.7995 -0.5545 0.3320 1.0217 2.2897   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 7.438e+00 3.818e-01 19.483 < 2e-16 \*\*\*  
## Population 4.367e-08 9.461e-09 4.616 7.92e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.514 on 28 degrees of freedom  
## Multiple R-squared: 0.4321, Adjusted R-squared: 0.4118   
## F-statistic: 21.31 on 1 and 28 DF, p-value: 7.924e-05

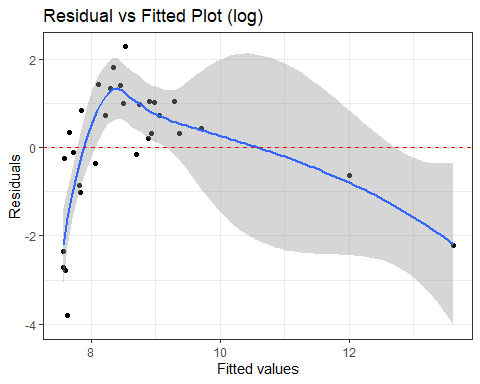
tidy(model3b)

## # A tibble: 2 x 5  
## term estimate std.error statistic p.value  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) 7.44 0.382 19.5 8.06e-18  
## 2 Population 0.0000000437 0.00000000946 4.62 7.92e- 5

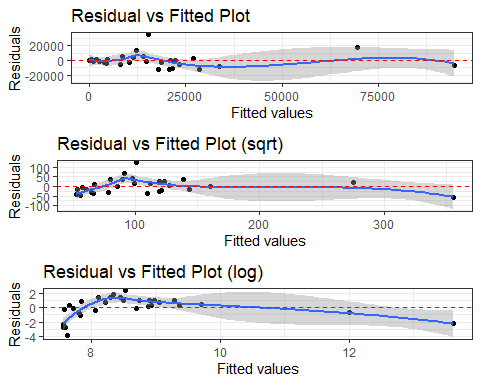
# Assessing Coefficients  
confint(model3b)

## 2.5 % 97.5 %  
## (Intercept) 6.656045e+00 8.220135e+00  
## Population 2.428880e-08 6.304674e-08

#Assessing Our Model Visually  
  
object\_name\_m3b <- ggplot(model3b, aes(.fitted, .resid))+   
 geom\_point()+  
 stat\_smooth(method="loess")+  
 geom\_hline(yintercept=0, col="red", linetype="dashed")+  
 xlab("Fitted values")+  
 ylab("Residuals")+  
 ggtitle("Residual vs Fitted Plot (log)")+  
 theme\_bw()  
   
object\_name\_m3b



gridExtra::grid.arrange(object\_name\_m3, object\_name\_m3a, object\_name\_m3b)



### Comparing standardized residuals versus fitted values.

This is the same plot as above but with the residuals standardized to show where residuals deviate by 1, 2, 3+ standard deviations. This helps us to identify outliers that exceed 3 standard deviations.

The second is the scale-location plot. This plot shows if residuals are spread equally along the ranges of predictors.

p1\_1 <- ggplot(model1, es(.fitted, .std.resid)) +  
 geom\_ref\_line(h = 0) +  
 geom\_point() +  
 geom\_smooth(se = FALSE) +  
 ggtitle("Standardized Residuals vs Fitted")

## Error in ggplot(model1, es(.fitted, .std.resid)): object 'model1' not found

p2\_1 <- ggplot(model1, aes(.fitted, sqrt(.std.resid))) +  
 geom\_ref\_line(h = 0) +  
 geom\_point() +  
 geom\_smooth(se = FALSE) +  
 ggtitle("Scale-Location")

## Error in ggplot(model1, aes(.fitted, sqrt(.std.resid))): object 'model1' not found

gridExtra::grid.arrange(p1, p2, nrow = 1)

## Error in arrangeGrob(...): object 'p1' not found

##Cook’s Distance and residuals versus leverage plot

ggplot(model3\_results, aes(seq\_along(.cooksd), .cooksd))+  
 geom\_bar(stat="identity", position="identity")+  
 xlab("Obs. Number")+  
 ylab("Cook's distance")+  
 ggtitle("Cook's distance")+  
 theme\_bw()  
   
ggplot(model3\_results, aes(.hat, .std.resid))+  
 geom\_point(aes(size=.cooksd), na.rm=TRUE)+  
 stat\_smooth(method="loess", na.rm=TRUE)+  
 xlab("Leverage")+  
 ylab("Standardized Residuals")+  
 ggtitle("Residual vs Leverage Plot")+  
 scale\_size\_continuous("Cook's Distance", range=c(1,5))+  
 theme\_bw()+  
 theme(legend.position="bottom")  
   
ggplot(model3\_results, aes(.hat, .cooksd))+  
 geom\_point(na.rm=TRUE)+  
 stat\_smooth(method="loess", na.rm=TRUE)+  
 xlab("Leverage hii")+

## Error: <text>:22:0: unexpected end of input  
## 20: stat\_smooth(method="loess", na.rm=TRUE)+  
## 21: xlab("Leverage hii")+  
## ^

##These plot helps us to find \*influential cases\* (i.e., subjects) if any. Not all outliers are influential in linear regression analysis. Even though data have extreme values, they might not be influential to determine a regression line. That means, the results would not be much different if we either include or exclude them from analysis.

Checking the top 5 observations with the highest Cook’s distance.

model3\_results %>%  
 top\_n(5, wt = .cooksd)

## # A tibble: 5 x 18  
## .rownames State Temperature Wind\_Chill Humidity Pressure Visibility Wind\_Speed  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 9 FL 74.1 74.0 75.3 30.0 9.48 8.22  
## 2 30 NJ 54.9 52.8 69.4 29.9 8.37 7.71  
## 3 36 OR 52.2 50.5 70.1 29.1 8.78 6.79  
## 4 39 SC 65.0 64.5 70.3 29.5 9.01 5.72  
## 5 42 TX 68.9 68.2 67.1 29.5 9.13 8.26  
## # ... with 10 more variables: Precipitation <dbl>, count <int>,  
## # Population <dbl>, .fitted <dbl>, .resid <dbl>, .hat <dbl>, .sigma <dbl>,  
## # .cooksd <dbl>, .std.resid <dbl>, Model <chr>

**Summary**

Even though the number of low severity and very high severity accidents are almost constant over the years. The severity 2 accidents are sharply increasing resulting in higher total number of accidents. CA has the highest number of accidents as much as 2.5 times that of TX which is the second in the rank. When we check the cities, Houston has the highest number of accidents. It would be interesting to know what factors contribute sharp decline in number of accidents in Texas considering that TX is twice the size of CA. Most of the accidents happen during daytime possibly due to high traffic caused by people commuting to the office and back home. Even though left most lanes is the fastest one, the graph shows that right side of the lane has highest number of accidents. One reason for this could be high number of lane merging that vehicles do while entering or exiting the express Most of the accidents appear to occur in fair and clear weather conditions. It would be interesting to explore the severity of these accidents. December being the holiday season has the highest number of accidents. From clustering we came to conclusion that October, November, and December have highest number of accidents Surprisingly, visibility have no significant affect on severity of accidents. Humidity and Windspeed have positive impact on severity and wind chill has negative impact. #Recommendations: 1.Increase Road Safety messaging during peak driving hours on weekdays. 2.Teaching new drivers, the dangers of holiday travel. 3.Inform on dangers of high visibility driving. 4.Be more careful in high population areas. 5.Increase funding to road maintenance.