US Road Accidents

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 throughout 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 scenario 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 zip code in the address field.
Country	Shows the country in the address field.
Timezone	Indicates time zone depending on the accident's location
Airport_Code	Denotes an airport-based weather station which is the closest accident's location.
Weather_Timestamp	Specify the timestamp 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 dataset which covers 49 states of the US. There are about 4.2 million accident records in this dataset. 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_conflict
s() --
## 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://g
oo.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 dataset: Since this is huge dataset to analyze as our R is taking time to process huge dataset we decided to sample dataset, since it is easier for us to analyze. The new dataset will contain 1 million rows

```
US_Accidents_Sample <- US_Accidents[sample(nrow(US_Accidents), 1000000, repla
ce = 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 us
ing 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_s
ec")) %>% separate(End_Time, into= c("End_Accident_year","End_Accident_month"
, "End_Accident_date", "End_Accident_Hour","End_Accident_min","End_Accident_s
ec"))
```

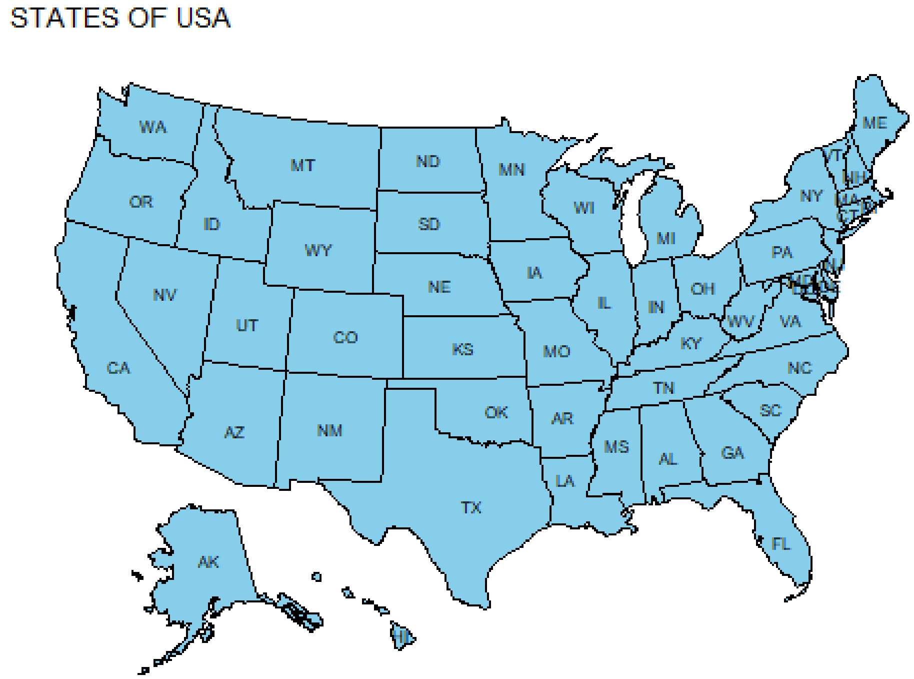
Removed few columns since they are having same value throughout the data for example country column, also removed columns which doesn’t have any values except null for example end longitude and end latitude, also removed few columns since they have multiple values which is breaking rule number 3 of tidy data for example zip code 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 consistent throughout 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","St
art_Accident_min", "Start_Accident_sec","End_Accident_year","End_Accident_mon
th","End_Accident_date","End_Accident_Hour","End_Accident_min","End_Accident_
sec","Start_Lat","Start_Lng","End_Lat","End_Lng","Distance", "Description","N
umber","Street","Side","City","County","State","Zipcode", "Country", "Timezon
e","Airport_Code", "Weather_TimeStamp", "Temperature","Wind_Chill","Humidity"
,"Pressure","Visibility","Wind_Direction","Wind_Speed","Precipitation","Weath
er_Condition","Amenity","Bump","Crossing","Give_Way","Junction","No_Exit","Ra
ilway","Roundabout","Station","Stop","Traffic_Calming","Traffic_Signal","Turn
ing_loop", "Sunrise_Sunset", "Civil_Twilight", "Nautical_Twilight","Astronomi
cal_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 bl
ue", 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 the greatest 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) * 1
00,2))


#Summary of accidents by state- this shows the mean, median, mode of number o
f accidents per state. This also gives us information on quaterly accidents.
summary(by_state)
```

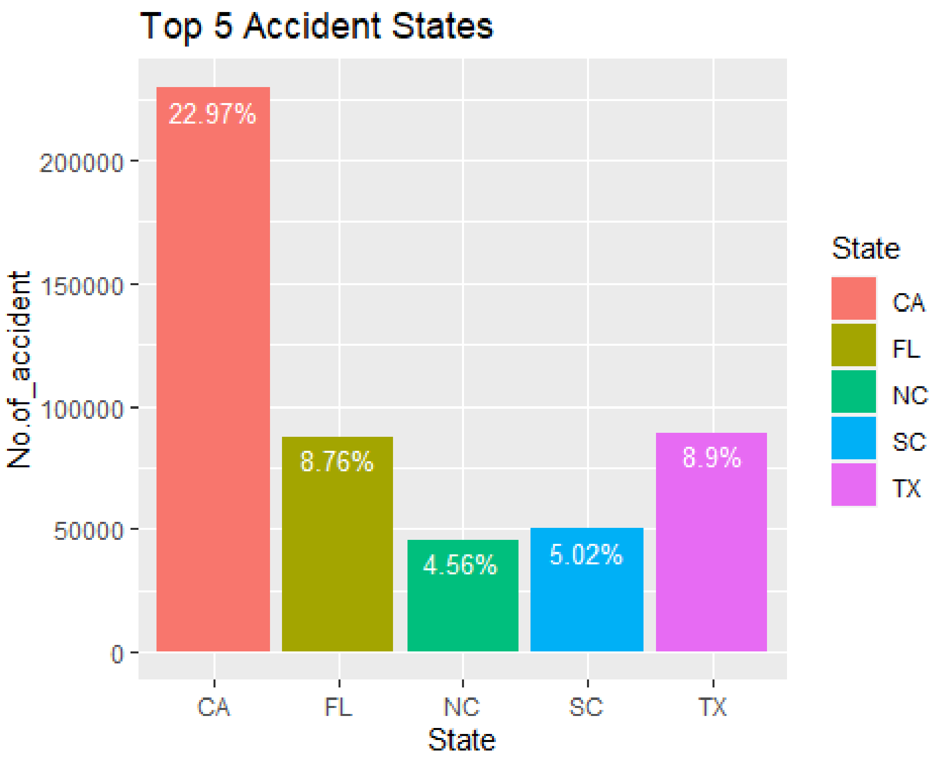
```
##     State            No.of_accident   percentage_accident
##  Length:49          Min.   :    50   Min.   : 0.00
##  Class :character   1st Qu.:  1843   1st Qu.: 0.18
##  Mode  :character   Median : 10335   Median : 1.03
##                     Mean   : 20408   Mean   : 2.04
##                     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
```



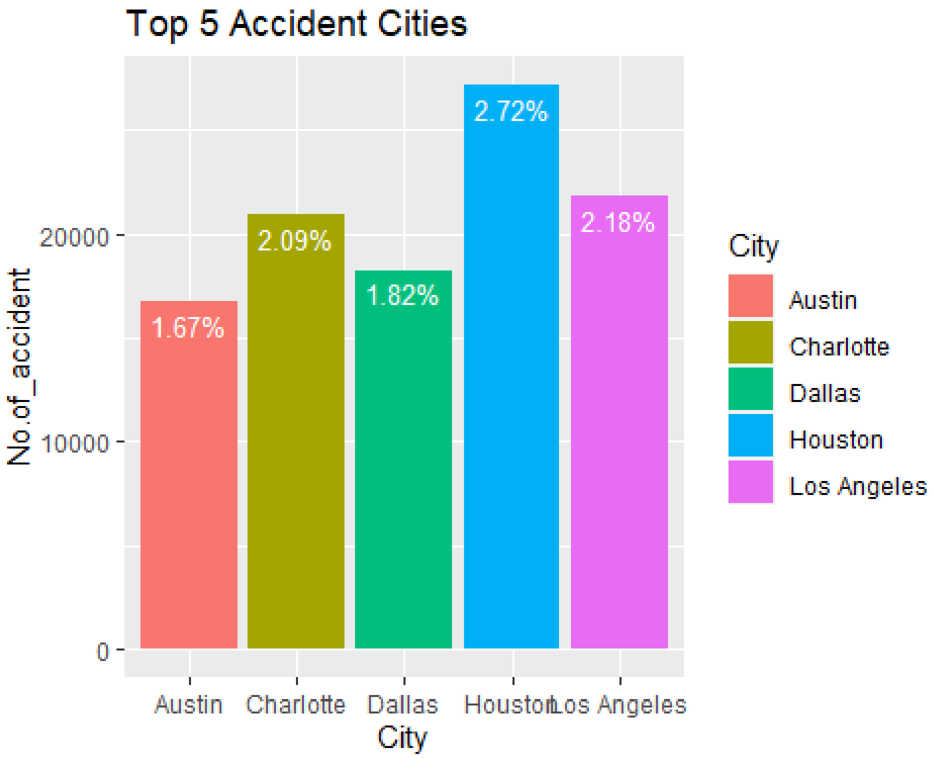
Observations:

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.

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
```



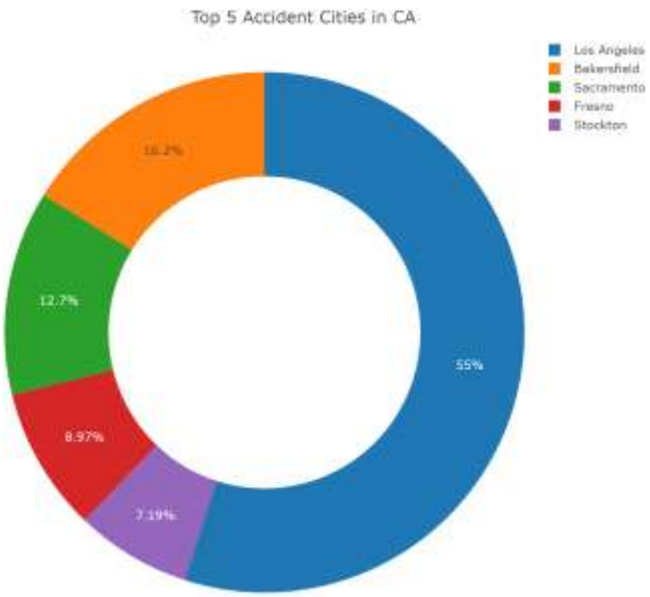
Observation:

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 reasons we could think of is traffic law enforcement in Texas is different from other states.

Top 5 cities in California 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_acciden
t)

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, s
howticklabels = FALSE))%>% print(fig)
```



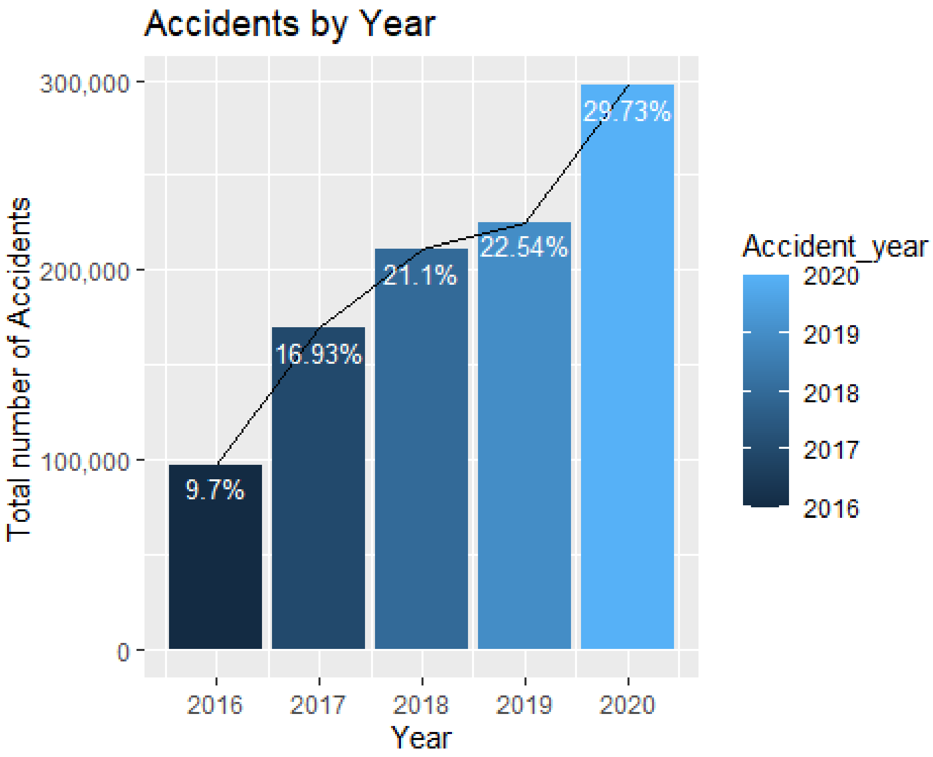
Observation:

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

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 = Accide
nt_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:

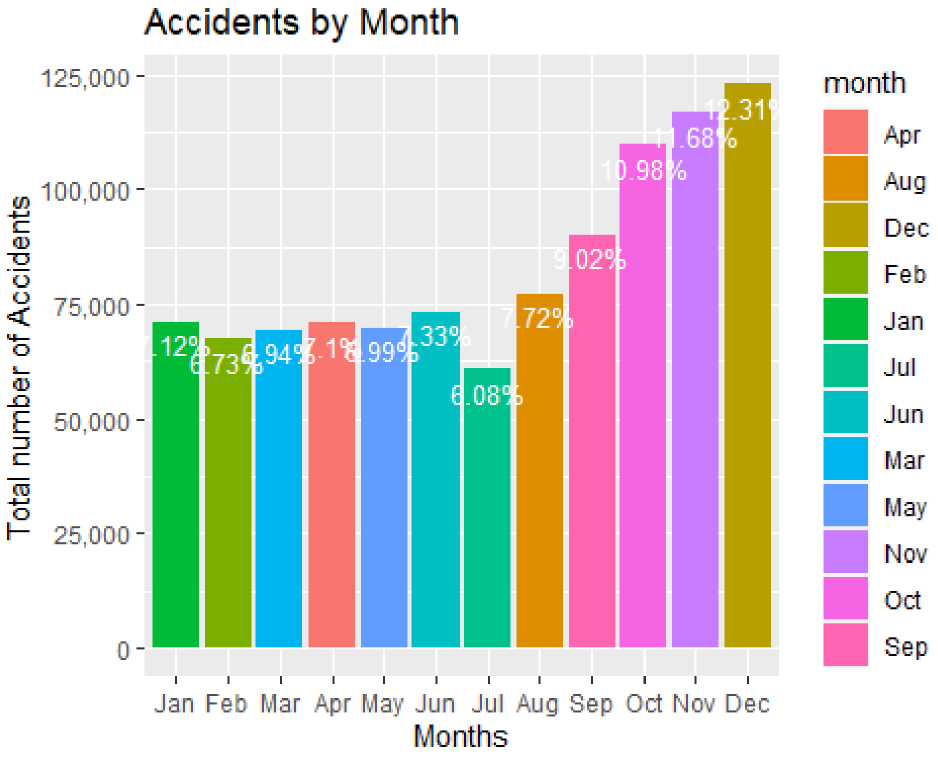
Number of accidents are increasing every passing year, here we can see that from year 2016 to 2020 there is a 3-fold increase in percent of accidents, one of the reasons we could think of is Expanding the city limits.

Accident Per Month- To demonstrate 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 adj
ust
## the group aesthetic?
```

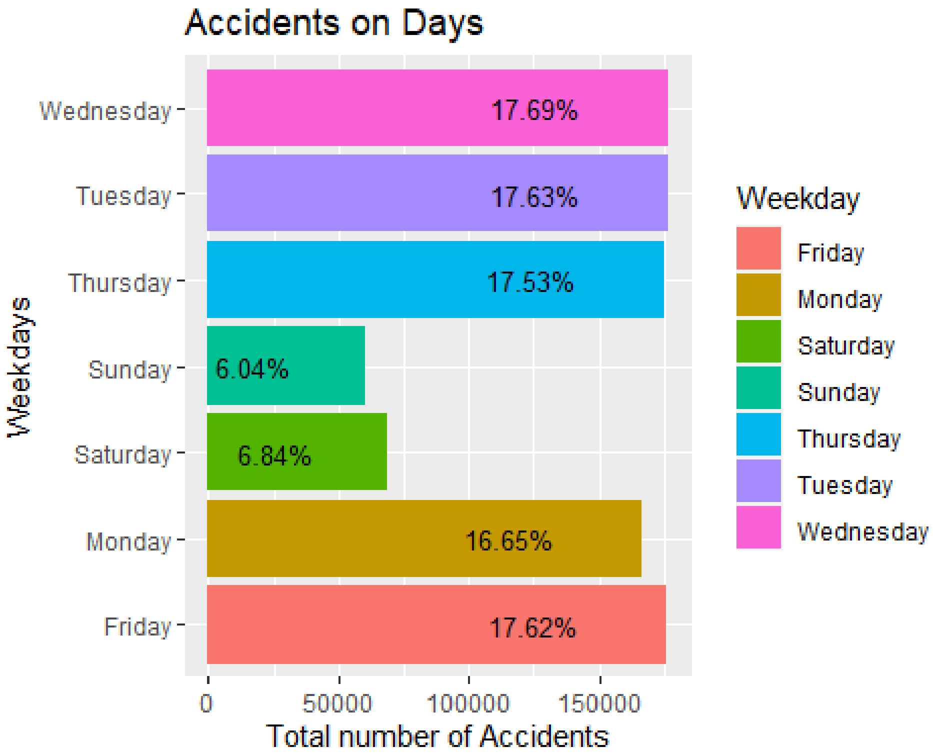


Observation:

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.

Accident Per days of Week- To demonstrate 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(To
tal_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 adj
ust
## the group aesthetic?
```

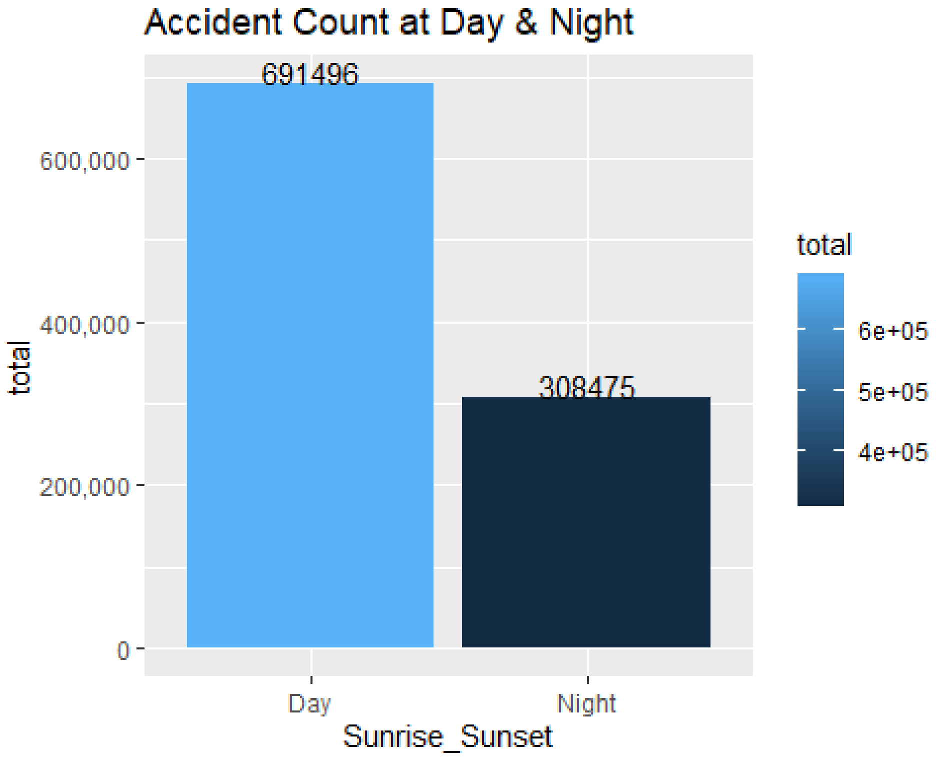


Observation:

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 Sunday 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.

What part of the day do majority of accidents occur?

```
day_night <- US_Accidents_Sample %>% select(Severity, Sunrise_Sunset) %>% gro
up_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")
```



#

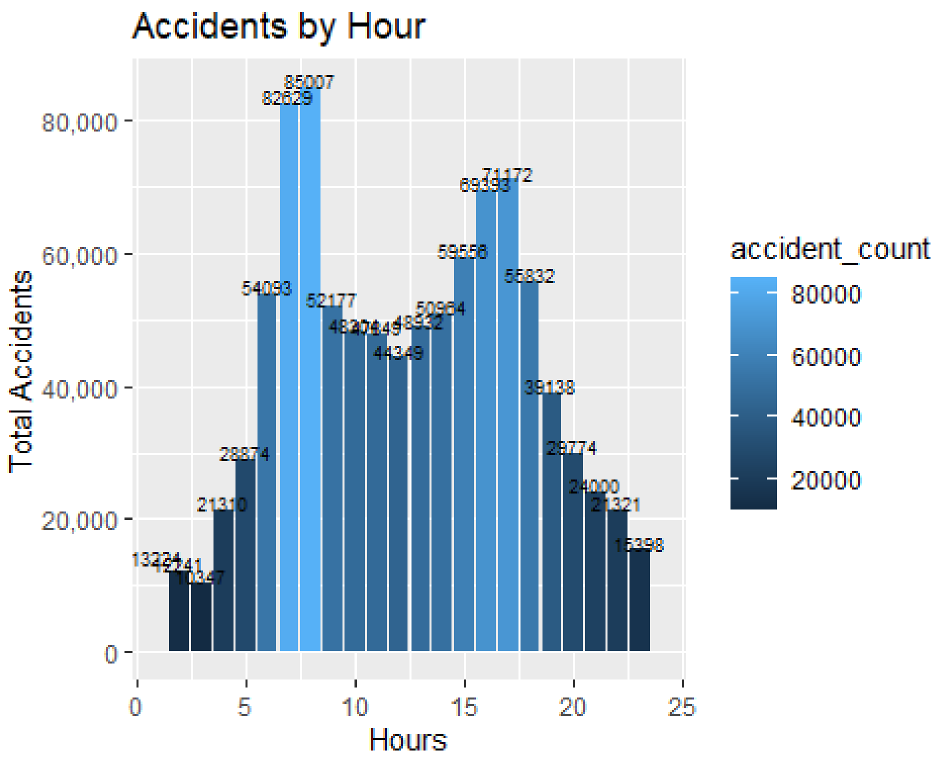
Observation

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.

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.

Which time of the day has more accidents?

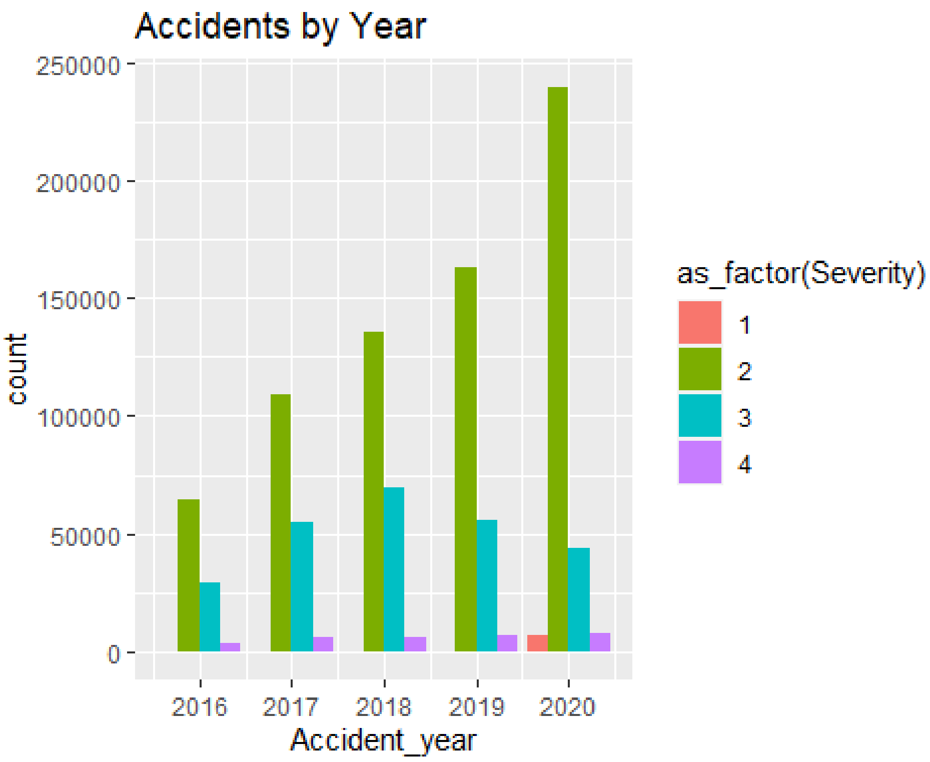
```
by_timeofDay<-
  US_Accidents_Sample %>%
  group_by(Start_Accident_hour) %>%
  summarise(accident_count= n()) %>%
  na.omit

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

```r
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 Acciden
ts",showlegend = T, xaxis = list(showgrid = FALSE, zeroline = FALSE, showtick
labels = FALSE), yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklab
els = 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)), positi
on = "dodge")+
  ggtitle("Accidents by Year")+
  theme_grey()
```

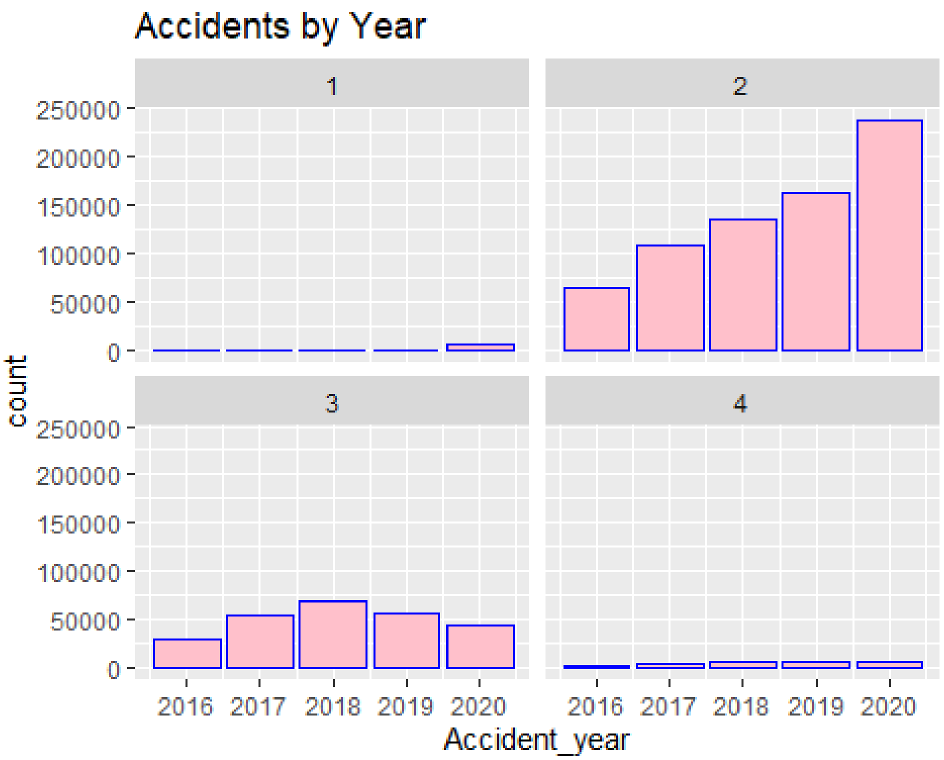


Observation:

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

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 = "blu
e", 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

Accidents by side- Sparkly R

##Creating Spark Connect

```
sc <- spark_connect(master = "local", version = "2.3") #create spark connecti
on

df_Spark <- sapply(read.csv("US_Accidents_Sample.csv"), class)

df_Spark <- spark_read_csv(sc,"US_Accidents_Sample.csv")#Load data in Spark
```

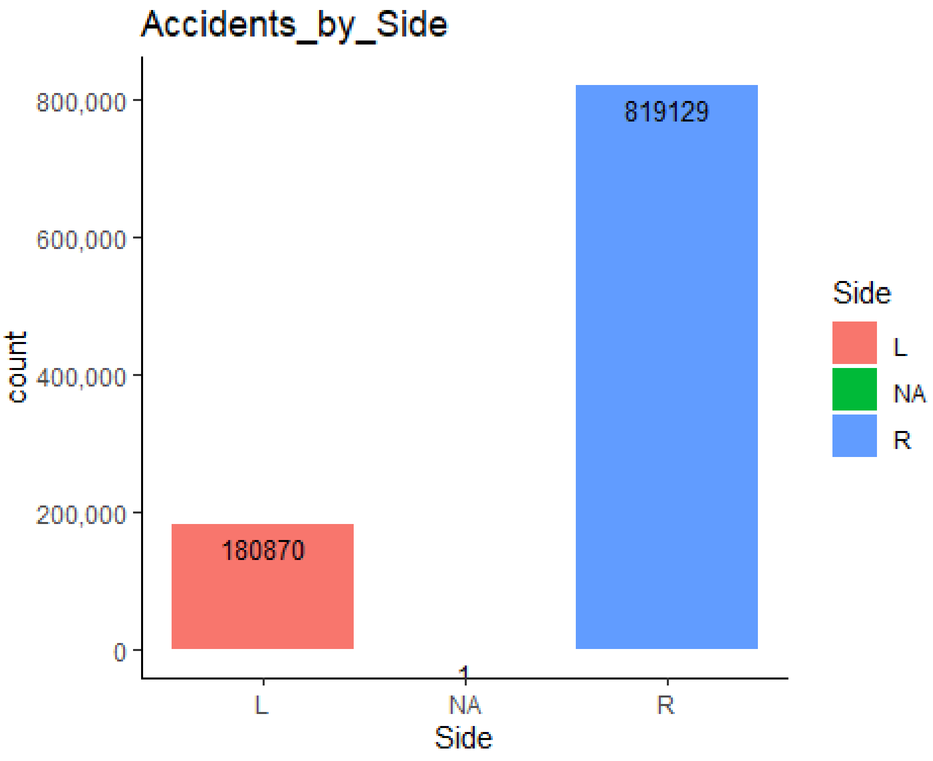
Accidents by side- Sparkly R

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()
```

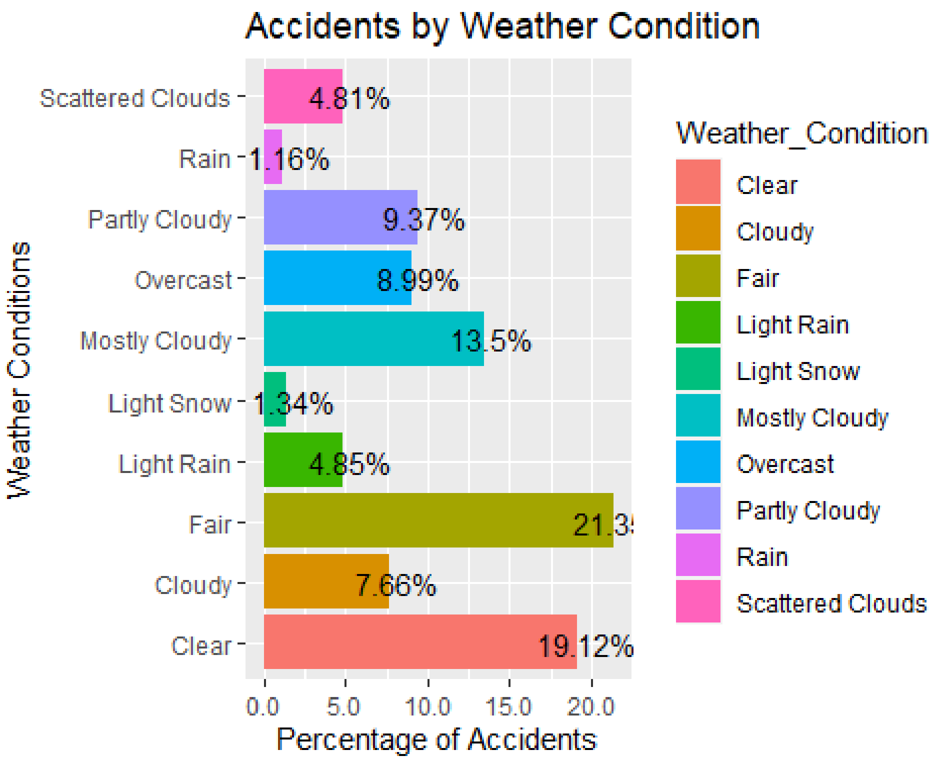


Observation:

The graph shows that most of the accidents occurs at the right side of the road whereas left side are very a smaller 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 lanes merging that vehicles do while entering or exiting the express. #code explanation: we used sparklyr to perform analysis, we pushed our data from r to spark and performed computations such as summarize 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)) %>%   n
a.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 = Weathe
r_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 adj
ust
## the group aesthetic?
```

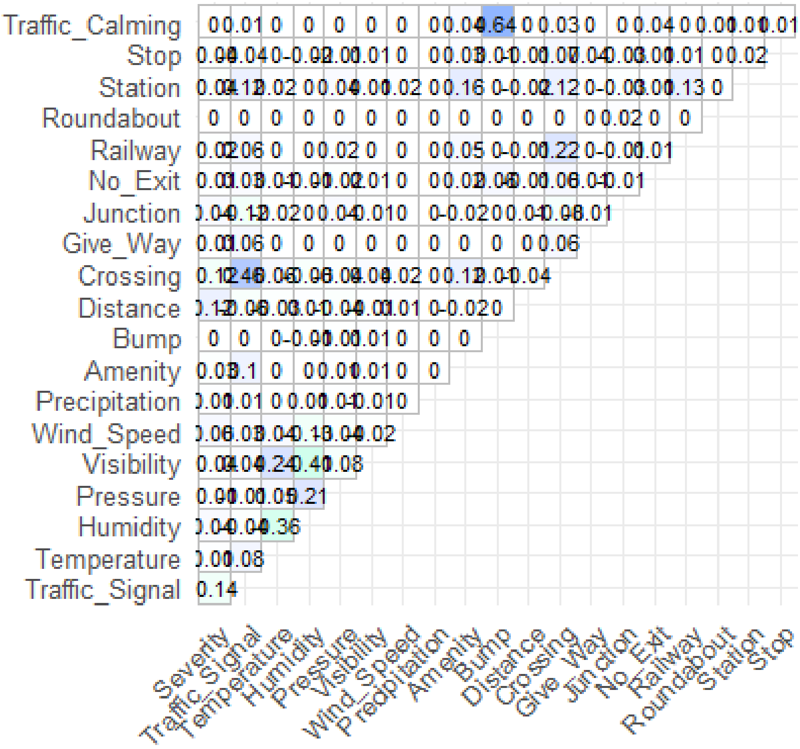


Observation:

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.

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")
```



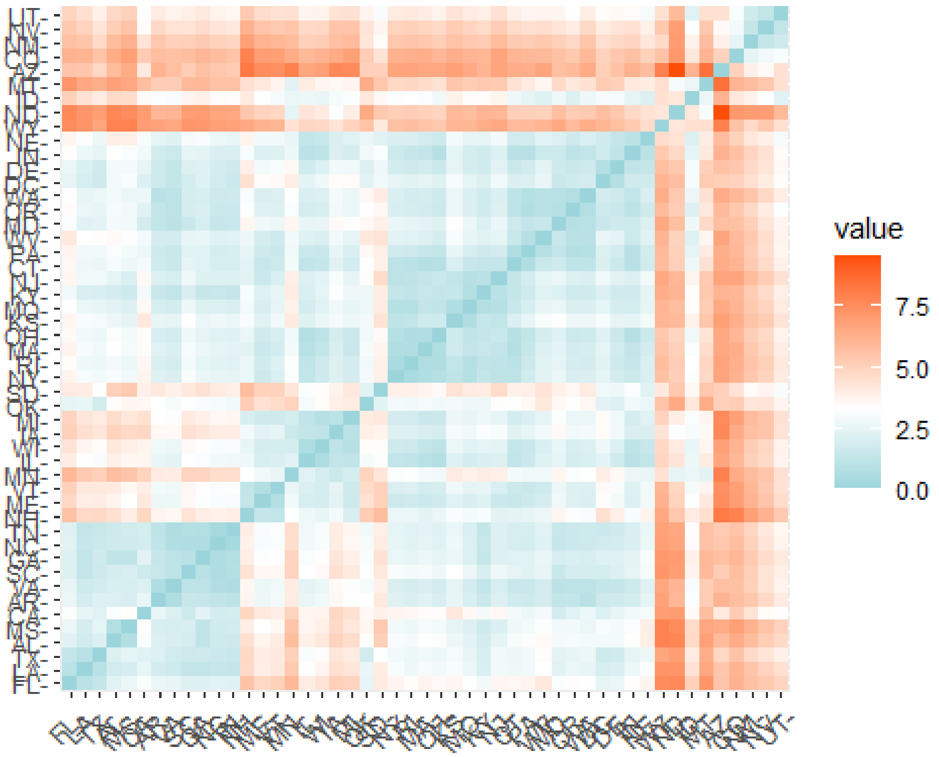
#

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.

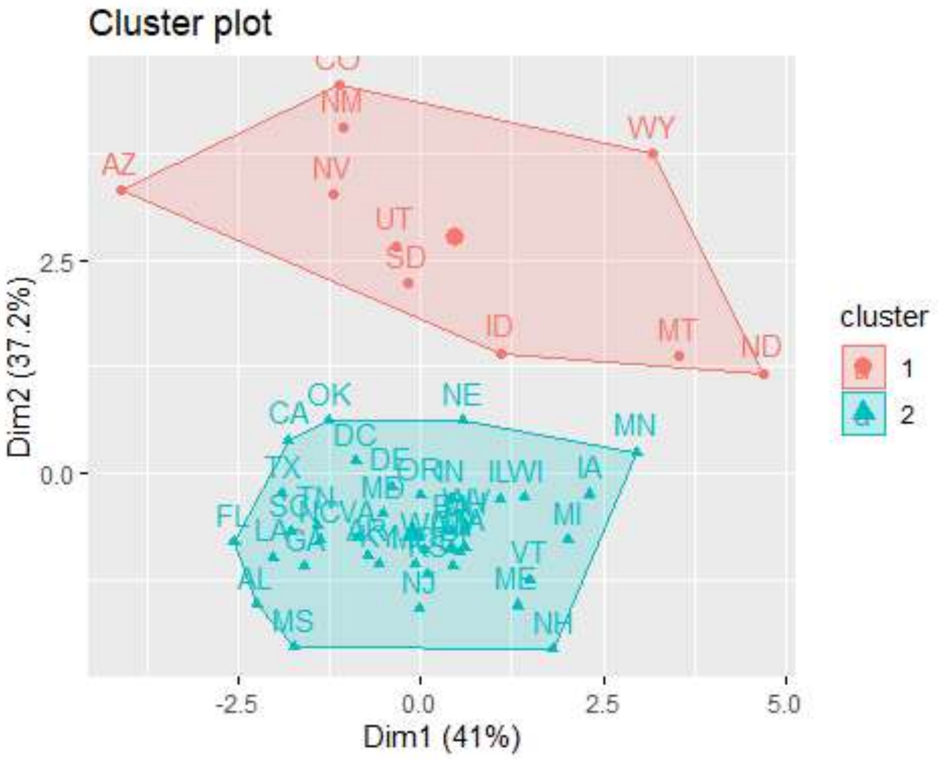
CLUSTERING

```
#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"))
```

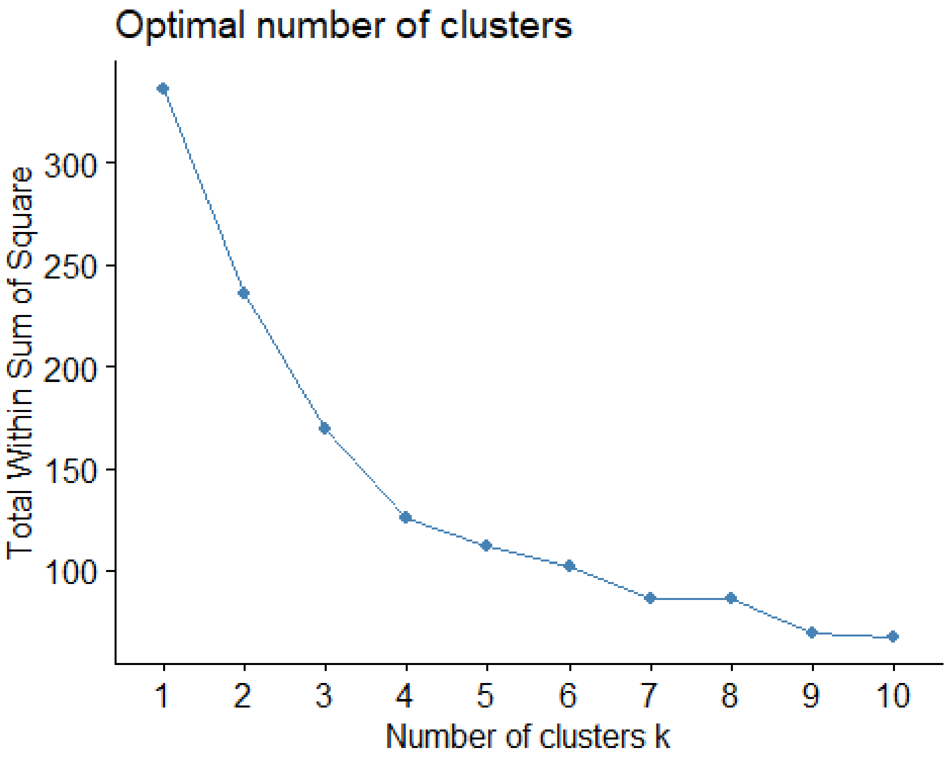


While we now have a statistical and visual understanding of the distance among the observations (rows) in our dataset. We need to use clustering to assign those observations in different groups (clusters)

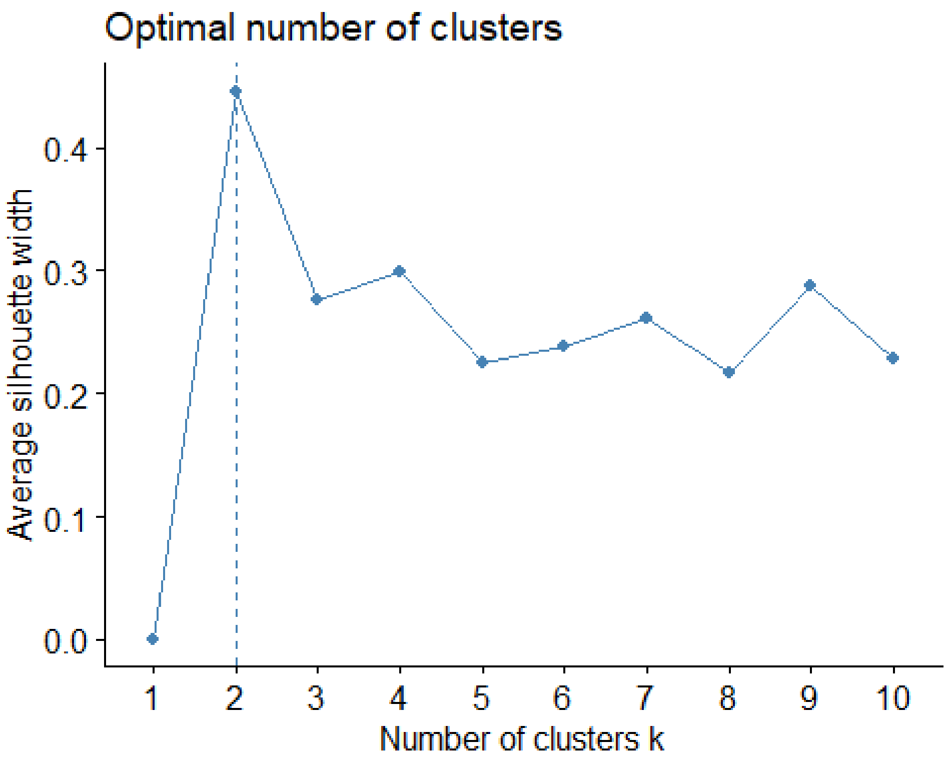
```
k2 <- kmeans(by_weather_cluster, centers = 2, nstart = 25)
str(k2)

## 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 Precipita
tion
##         <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)
```



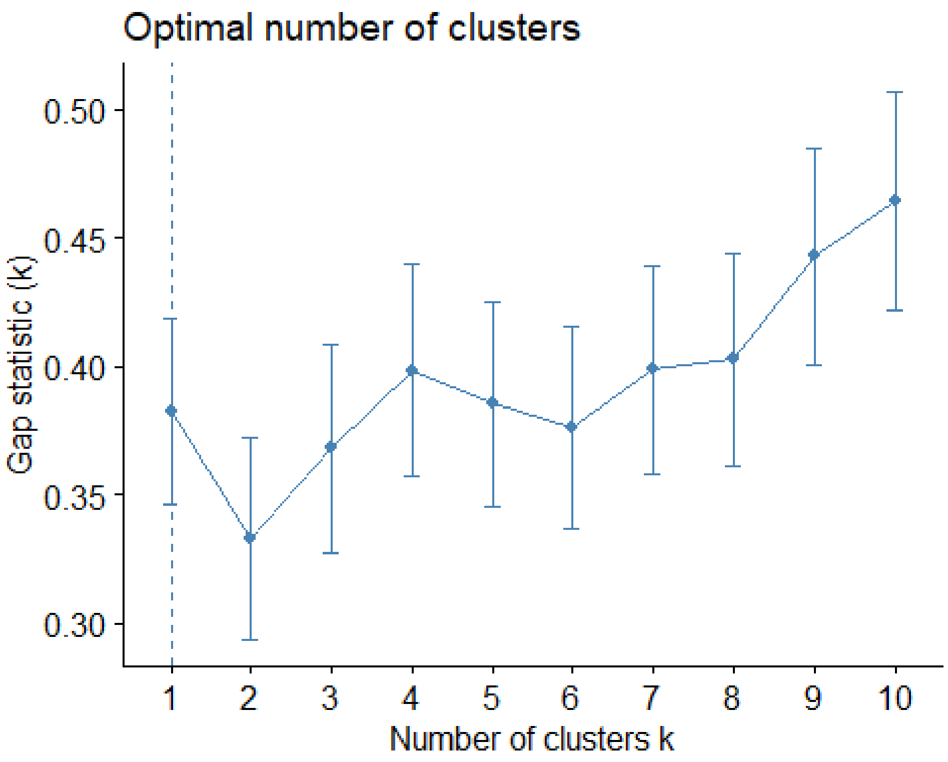
```r
### Elbow Method

set.seed(1234)
fviz_nbclust(by_weather_cluster, kmeans, method = "wss")
```



```r
### 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 = 5
0, 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 Precipita
tion
##         <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 i
s 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

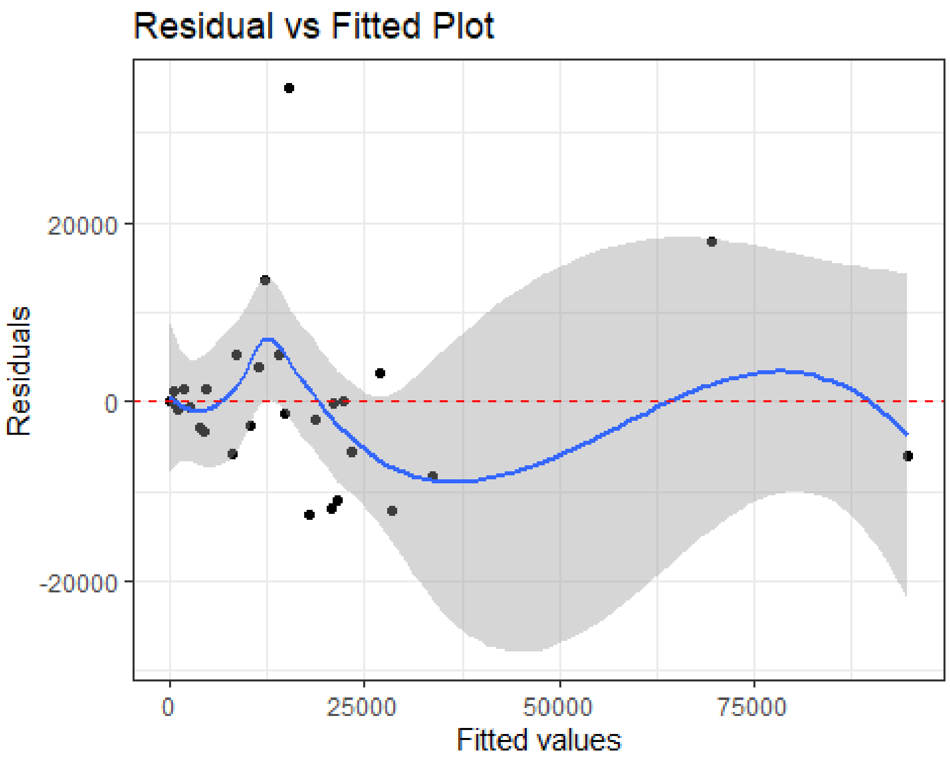
Residual standard error (RSE)- The RSE provides an absolute measure of lack of fit of our model to the data. But since it is measured in the units of Y, our response variable, it is not always clear what constitutes a good RSE

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.

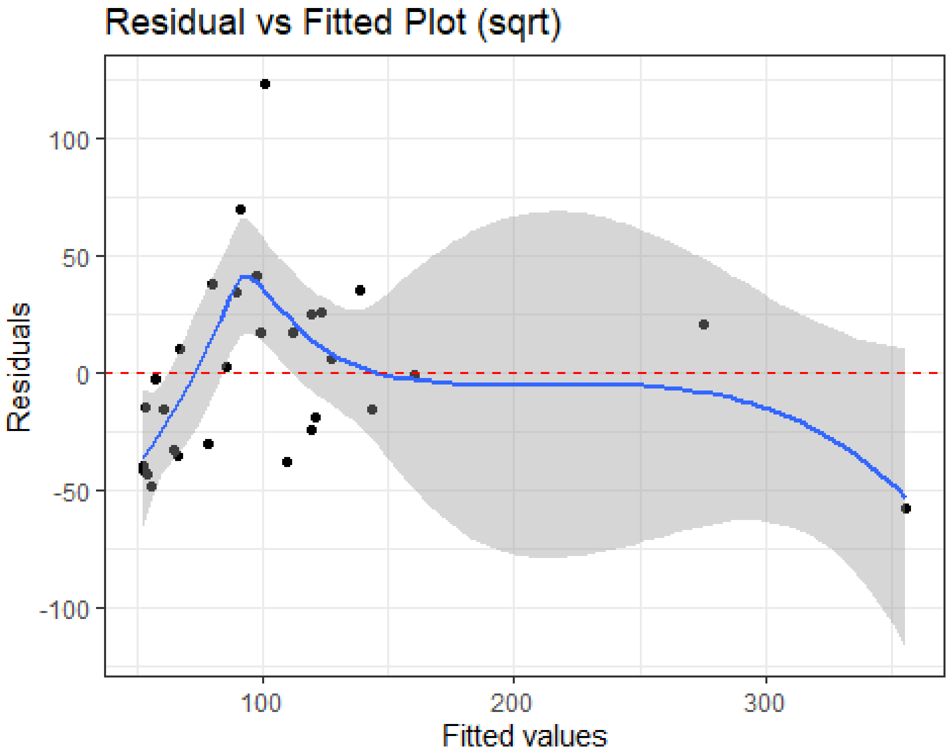
Combined, our RSE, R2, and F-statistic results suggest that our model has a good fit.

```
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
```



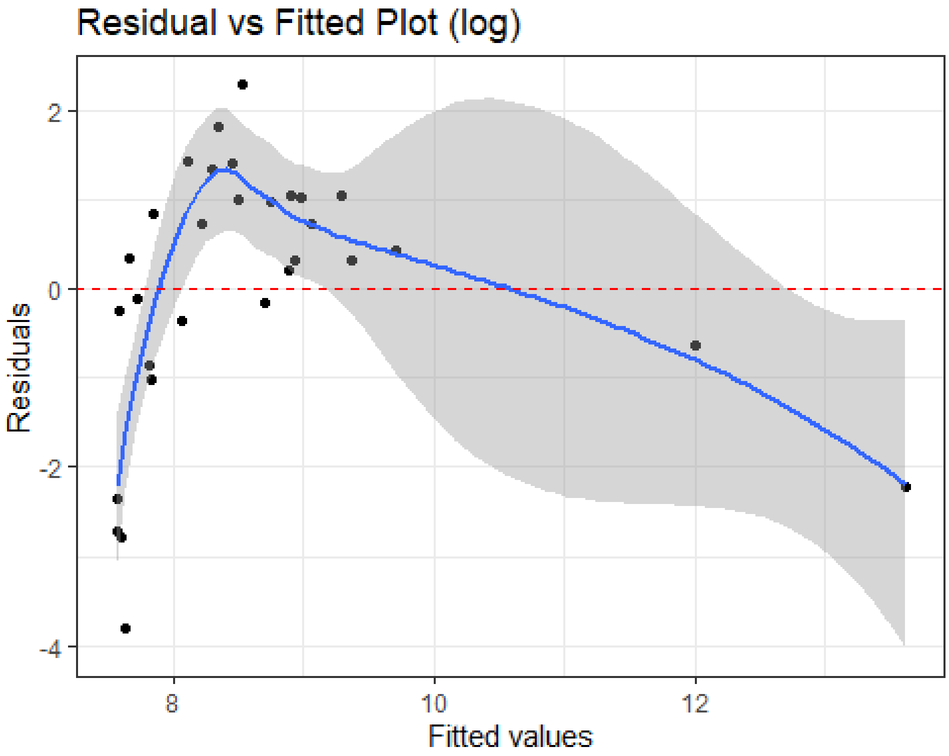
```
#Applying square root transformation

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
```

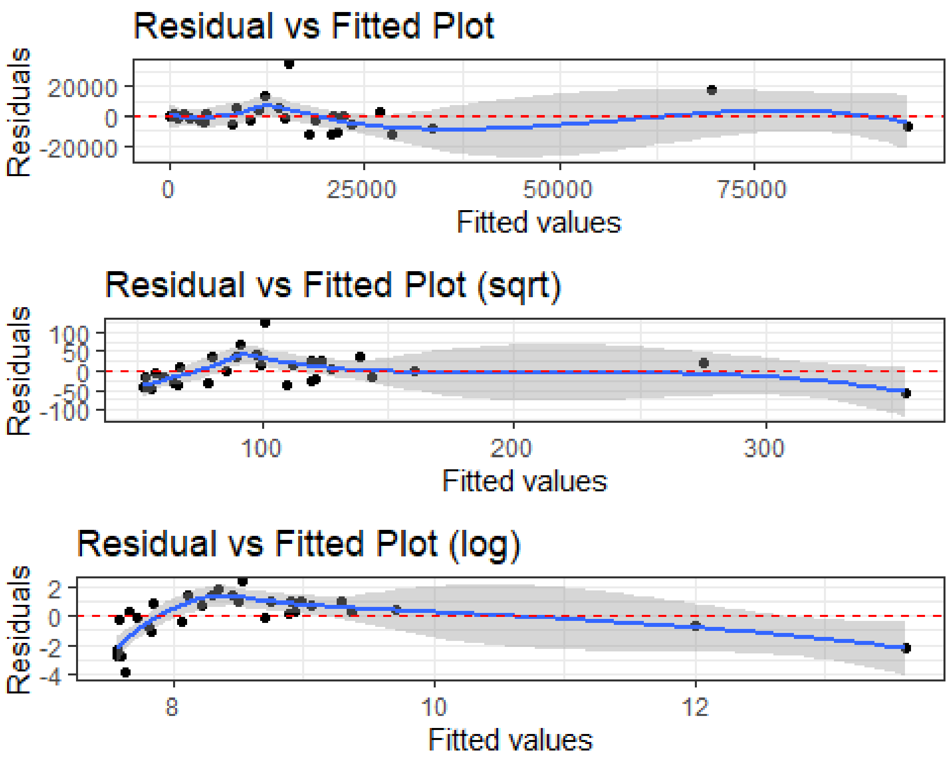


```
#Try with applying a log transformation
model3b <- lm(log(count) ~ Population, data = train)

 # 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 foun
d

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' n
ot 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. No
t all outliers are influential in linear regression analysis. Even though dat
a have extreme values, they might not be influential to determine a regressio
n line. That means, the results would not be much different if we either incl
ude 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 lanes merging that vehicles do while entering or exiting the express ways

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 effect 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.