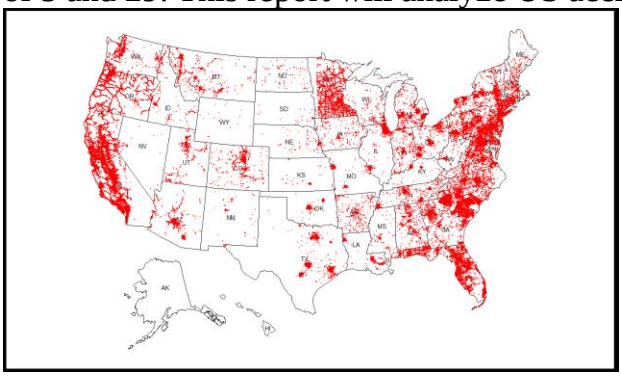
To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

US car accidents data was collected from various APIs that provide streaming traffic incident data from various entities such as the US and state departments of transportation, law enforcement agencies, traffic cameras, and traffic sensors within the road networks. Currently, there are about 1.5 million accident records in this dataset.

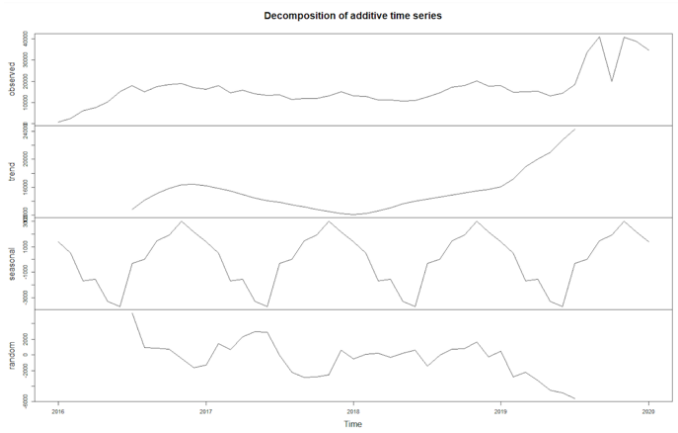
EXECUTIVE SUMMARY

Major Findings:

Reason for concern: The global report from the World Health Organization — which reviewed laws and crashes in 175 nations — explains that U.S.'s traffic fatality rate is 12.4 deaths per 100,000 — or about 50 percent higher than similar nations in Western Europe, plus Canada, Australia and Japan. About 1.3 million people are being killed globally by traffic crashes every year, a huge proportion of them pedestrians. Traffic deaths are now the leading cause of death globally for those between the ages of 5 and 29. This report will analyze US accidents on road and the factors affecting them.



The chart shows the accidents occurred through these across USA and the intensity can be clearly identified with the intensity of red points on the map. The chart was plotted using the dataset used for the project. The severity of the accidents was predicted using the regression models to provide appropriate recommendations.



Further Understanding: The dataset included 47 variables that were all utilized to understand the impact on severity of the accidents and forecast the trends for number of accidents accordingly.

Initial data concerns: Within each year, there is a significant change in trends for random, observed and trend. These are accounted for various models. The change in trend shows the outliers in the years 2019 to 2020 which could be because of the snowstorms, tornados, or other weather conditions. The hit of pandemic could also be part of the reason for outliers in data.

Final outcomes and analyses: The final output of this analysis is a forecast for number of accidents and severity for the years 2019 and 2020, applying assumptions of data from 2016 to 2018 several models were used to capture the aspects that affected the accidents in all states, counties, and streets. The model with least errors was selected to simulate the outcomes for the years forecasted.

Severity=1.583568-0.0328198*Temperature(F)-0.1880180*Civil_Twilight-0.2986898*Nautical_Twilight+0.9932627*Astronomical_twilight+0.0186887*Humidity+0.1526831*Pressure+0.0062393*wind_speed(mph)-0.4012353*Precipitation(in)

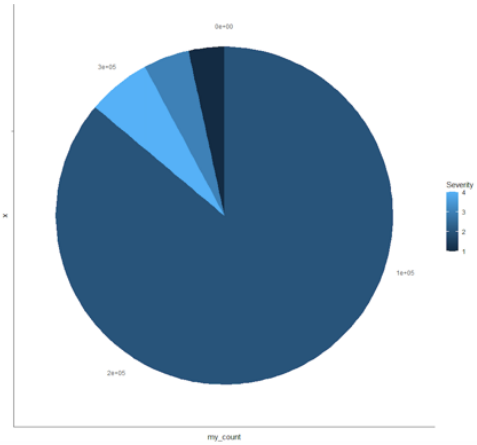
To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

State with greatest number of accidents:

- The data analysis of US data accidents shows that the greatest number of accidents occurred in the state of California which has a population of 39.1 million.The accidents that were recorded are 77247 from the year 2016 to the year 2020.
- The second greatest accidents were recorded in Florida for the same time with number 53173.

Severity of the accidents:

- The frequency of severity of all the accidents occurred recorded level 2 which is not too severe and damaging, the total number of level 2 accidents are 288919.
- The second highest severity of accidents recorded is of level 4 with the record of 20399. The severity level 1 is recorded the least of all the levels with 11483 records of accidents.



County with greatest number of accidents:

- The data analysis also reports that the county with highest number of accidents recorded is Miami-Dade in Florida. It had 973 accidents for the mentioned period.
- The second highest accidents of about 431 were recorded in Kendal Lakes, Florida.

Street with greatest number of accidents:

- The street with most number of accidents Highway 101(California, Oregon,Washington), reporting 1190 accidents from 2016 to 2020.
- The second highest accidents in streets were recorded in South Dixie highway, reporting 1147 accidents.

Recommendations:

- The response teams and hospitals must be given special provisions in the hours in which most accidents are occurred.
- Warning signs about speed limits are to be put in the accident prone streets
- The state with highest accidents must be provided with better resources and budget plans to avoid accidents and rescue the victims.
- Warnings are to be put depending on the weather conditions which cause accidents.
- A mandate of vehicles to have first aid kit should be passed.
- Online surveillance for prompt response from emergency services should be implemented.
- To have enough response teams to rescue in accident prone locations

To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

APPENDIX

```
library(psych)
install.packages("ISLR")
library(ISLR)
library("dlookr")
library(MASS)
library(dplyr)
library(car)
library(lmtest)
library(tidyverse)
library(broom)
library(ggplot2)
library(tidyverse)
library(caret)
library(lubridate)
library(readr)
library(ggpubr)
library(zoo)
library(tseries)
library(MLmetrics)
library(forecast)
library(usmap)
library(sp)
library(rgdal)
library(raster)
library(gstat)
library(gridExtra)
```

To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

EXPLORATORY DATA ANALYSIS:

*Library ISLR displays the datatype and content of the variables.

```
> library(ISLR)
> str(df)
'data.frame':   335552 obs. of  47 variables:
 $ ID                  : chr  "A-2716644" "A-2716645" "A-2716649" "A-2716652" ...
 $ Severity            : int  4 4 4 2 2 2 2 2 2 3 ...
 $ Start_Time          : chr  "2016-02-09 18:20:58" "2016-02-09 18:20:58" "2016-02-10 06:18:49" "2016-02-10 08:3
5:27" ...
 $ End_Time            : chr  "2016-02-10 00:20:58" "2016-02-10 00:20:58" "2016-02-10 12:18:49" "2016-02-10 14:3
5:27" ...
 $ Start_Lat           : num  40.5 40.4 40.7 41.8 41.5 ...
 $ Start_Lng           : num  -85.2 -85.1 -84.8 -80.1 -81.7 ...
 $ End_Lat             : num  40.4 40.5 40.7 41.8 41.5 ...
 $ End_Lng             : num  -85.1 -85.2 -84.8 -80.1 -81.7 ...
 $ Distance.mi.        : num  6.69 6.69 1.206 0.824 0.462 ...
 $ Description         : chr  "Closed between IN-26 and IN-67 - Road closed due to accident." "Closed between IN-
67 and IN-26 - Road closed due to accident." "Closed between Willshire and US-33/Rockford Rd - Road closed due to
 accident." "Between Irish Rd and Blystone Rd - Accident." ...
 $ Number              : num  9001 473 12998 25529 3937 ...
 $ Street              : chr  " W State Road 26" " N Meridian St" " State Route 49" " Highway 99" ...
 $ Side                : chr  "R" "R" "R" "L" ...
 $ City                : chr  "Dunkirk" "Redkey" "Willshire" "Cambridge Springs" ...
 $ County              : chr  "Jay" "Jay" "Van Wert" "Crawford" ...
 $ State               : chr  "IN" "IN" "OH" "PA" ...
 $ Zipcode             : chr  "47336" "47373-9430" "45898-9523" "16403" ...
 $ Country             : chr  "US" "US" "US" "US" ...
 $ Timezone            : chr  "US/Eastern" "US/Eastern" "US/Eastern" "US/Eastern" ...
 $ Airport_Code        : chr  "KMIE" "KMIE" "KFWA" "KGKJ" ...
 $ Weather_Timestamp   : chr  "2016-02-09 18:20:00" "2016-02-09 18:20:00" "2016-02-10 05:54:00" "2016-02-10 08:3
4:00" ...
 $ Temperature.F.      : num  19.9 19.9 17.1 21 24.1 16 16 16 17.1 6.1 ...
 $ Wind_Chill.F.       : num  7.3 7.3 0.6 9.9 7.1 5.3 5.3 -1.7 1.5 -12.2 ...
 $ Humidity...         : num  81 81 77 85 75 59 59 54 50 63 ...
 $ Pressure.in.        : num  29.9 29.9 29.9 29.7 29.8 ...
 $ Visibility.mi.      : num  2 2 2.5 1 2 10 10 10 10 10 ...
 $ Wind_Direction      : chr  "WNW" "WNW" "West" "WSW" ...
 $ Wind_Speed.mph.     : num  12.7 12.7 19.6 10.4 28.8 8.1 8.1 21.9 17.3 16.1 ...
 $ Precipitation.in.   : num  0 0 0.01 0 0 0 0 0 0 0 ...
 $ Weather_Condition   : chr  "Light Snow" "Light Snow" "Light Snow" "Light Snow" ...
 $ Amenity             : chr  "False" "False" "False" "False" ...
 $ Bump                : chr  "False" "False" "False" "False" ...
 $ Crossing            : chr  "False" "False" "False" "False" ...
 $ Give_Way            : chr  "False" "False" "False" "False" ...
 $ Junction            : chr  "False" "False" "False" "False" ...
 $ No_Exit             : chr  "False" "False" "False" "False" ...
 $ Railway             : chr  "False" "False" "False" "False" ...
```

To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

*Summary of the dataset shows all the required information for further analysis of the data.

```
> summary(df)
      ID                Severity       Start_Time          End_Time           Start_Lat       Start_Lng          End_Lat
 Length:335552      Min.   :1.000   Length:335552      Length:335552      Min.   :24.57   Min.   :-124.50   Min.   :24.57
 Class :character   1st Qu.:2.000   Class :character   Class :character   1st Qu.:33.18   1st Qu.:-118.28   1st Qu.:33.18
 Mode  :character   Median :2.000   Mode  :character   Mode  :character   Median :36.21   Median : -90.13   Median :36.21
                    Mean   :2.131                                         Mean   :36.41   Mean   : -97.13   Mean   :36.41
                    3rd Qu.:2.000                                         3rd Qu.:40.61   3rd Qu.: -80.42   3rd Qu.:40.61
                    Max.   :4.000                                         Max.   :48.99   Max.   : -67.48   Max.   :48.99
    End_Lng         Distance.mi.      Description            Number          Street              Side
 Min.   :-124.50   Min.   :  0.000   Length:335552      Min.   :      0   Length:335552      Length:335552
 1st Qu.:-118.28   1st Qu.:  0.000   Class :character   1st Qu.:   1200   Class :character   Class :character
 Median : -90.13   Median :  0.053   Mode  :character   Median :   3930   Mode  :character   Mode  :character
 Mean   : -97.13   Mean   :  0.262                      Mean   :   8769
 3rd Qu.: -80.42   3rd Qu.:  0.178                      3rd Qu.:  10019
 Max.   : -67.48   Max.   :112.968                      Max.   :9999997
     City              County             State              Zipcode            Country            Timezone
 Length:335552      Length:335552      Length:335552      Length:335552      Length:335552      Length:335552
 Class :character   Class :character   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character


 Airport_Code       Weather_Timestamp  Temperature.F.  Wind_Chill.F.     Humidity...      Pressure.in.   Visibility.mi.
 Length:335552      Length:335552      Min.   :-27.0   Min.   :-48.50   Min.   :  1.00   Min.   :19.37   Min.   :  0.000
 Class :character   Class :character   1st Qu.: 46.0   1st Qu.: 44.00   1st Qu.: 49.00   1st Qu.:29.18   1st Qu.: 10.000
 Mode  :character   Mode  :character   Median : 59.0   Median : 59.00   Median : 69.00   Median :29.73   Median : 10.000
                                       Mean   : 58.8   Mean   : 57.58   Mean   : 65.69   Mean   :29.33   Mean   :  9.121
                                       3rd Qu.: 73.0   3rd Qu.: 73.00   3rd Qu.: 86.00   3rd Qu.:29.97   3rd Qu.: 10.000
                                       Max.   :111.0   Max.   :111.00   Max.   :100.00   Max.   :58.04   Max.   :100.000
 Wind_Direction     Wind_Speed.mph.   Precipitation.in.  Weather_Condition    Amenity             Bump
 Length:335552      Min.   :  0.000   Min.   :0.000000   Length:335552      Length:335552      Length:335552
 Class :character   1st Qu.:  3.000   1st Qu.:0.000000   Class :character   Class :character   Class :character
 Mode  :character   Median :  7.000   Median :0.000000   Mode  :character   Mode  :character   Mode  :character
                    Mean   :  7.137   Mean   :0.003781
                    3rd Qu.: 10.000   3rd Qu.:0.000000
                    Max.   :211.000   Max.   :9.990000
   Crossing          Give_Way           Junction           No_Exit            Railway           Roundabout
 Length:335552      Length:335552      Length:335552      Length:335552      Length:335552      Length:335552
 Class :character   Class :character   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character
```

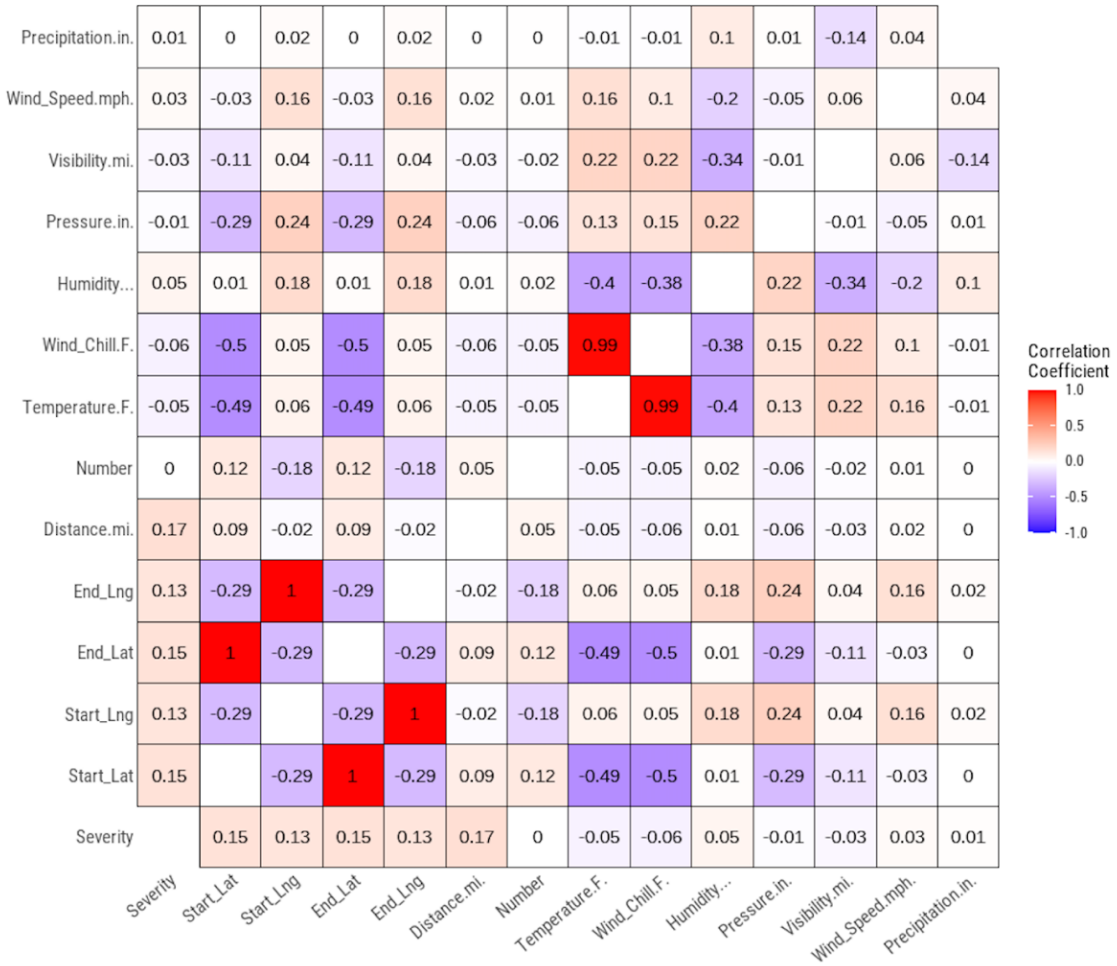
To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

Describe function shows if there are any missing values, number of values and unique values along with mean values.

```
> describe(df)
df

 47  Variables      335552  Observations
---------------------------------------------------------------------------------------------------------------------------
ID
       n  missing distinct
  335552        0   335552

lowest : A-2716644 A-2716645 A-2716649 A-2716652 A-2716669, highest: A-4239363 A-4239367 A-4239368 A-4239369 A-4239372
---------------------------------------------------------------------------------------------------------------------------
Severity
       n  missing distinct     Info     Mean      Gmd
  335552        0        4    0.361    2.131   0.3679

Value          1      2      3      4
Frequency   11483 288919  14751  20399
Proportion  0.034  0.861  0.044  0.061
---------------------------------------------------------------------------------------------------------------------------
Start_Time
       n  missing distinct
  335552        0   230252

lowest : 2016-02-09 18:20:58 2016-02-10 06:18:49 2016-02-10 08:35:27 2016-02-10 12:54:39 2016-02-11 07:20:03
highest: 2020-12-31 22:39:08 2020-12-31 22:41:39 2020-12-31 22:42:20 2020-12-31 22:49:31 2020-12-31 23:28:56
---------------------------------------------------------------------------------------------------------------------------
End_Time
       n  missing distinct
  335552        0   260345

lowest : 2016-02-10 00:20:58           2016-02-10 12:18:49           2016-02-10 14:35:27           2016-02-10 18:54:39           2016
-11 13:20:03
highest: 2020-12-31 23:58:11.000000000 2020-12-31 23:58:24           2020-12-31 23:58:30           2020-12-31 23:59:37           2020
-31 23:59:47.000000000
---------------------------------------------------------------------------------------------------------------------------
Start_Lat
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75      .90      .95
  335552        0   187880        1    36.41    6.538    25.89    28.08    33.18    36.21    40.61    44.21    45.41

lowest : 24.57200 24.57473 24.57476 24.59081 24.59456, highest: 48.98604 48.98978 48.99056 48.99333 48.99384
---------------------------------------------------------------------------------------------------------------------------
Start_Lng
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75      .90      .95
  335552        0   189005        1   -97.13    20.36  -122.81  -122.09  -118.28   -90.13   -80.42   -77.24   -75.41
```

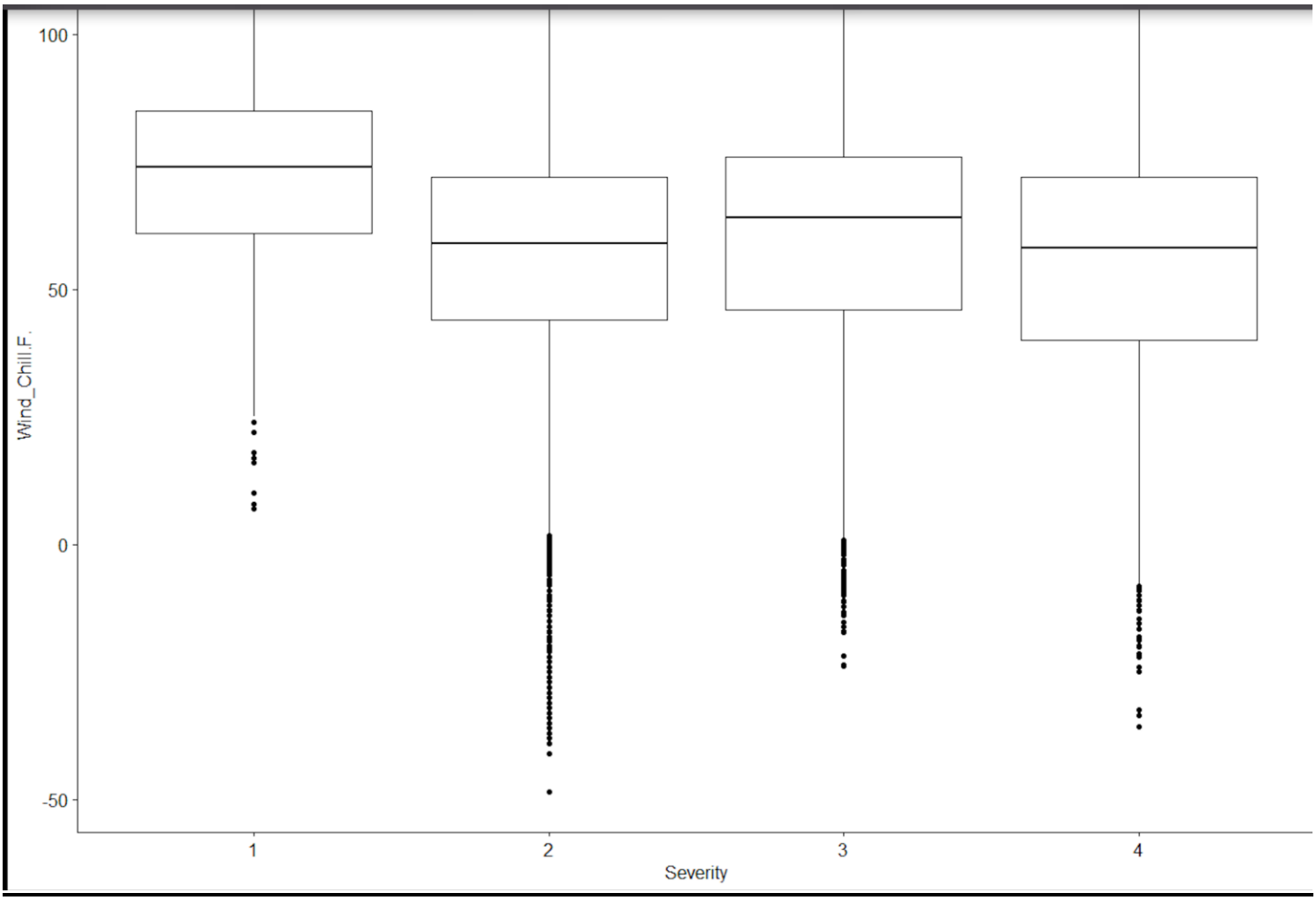
To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

*Below image shows the correlation among the variables. Correlation falls in range of -1 and 1. -1 being perfectly negative correlation and 1 being positive correlation. 0 shows no correlation among the variables.



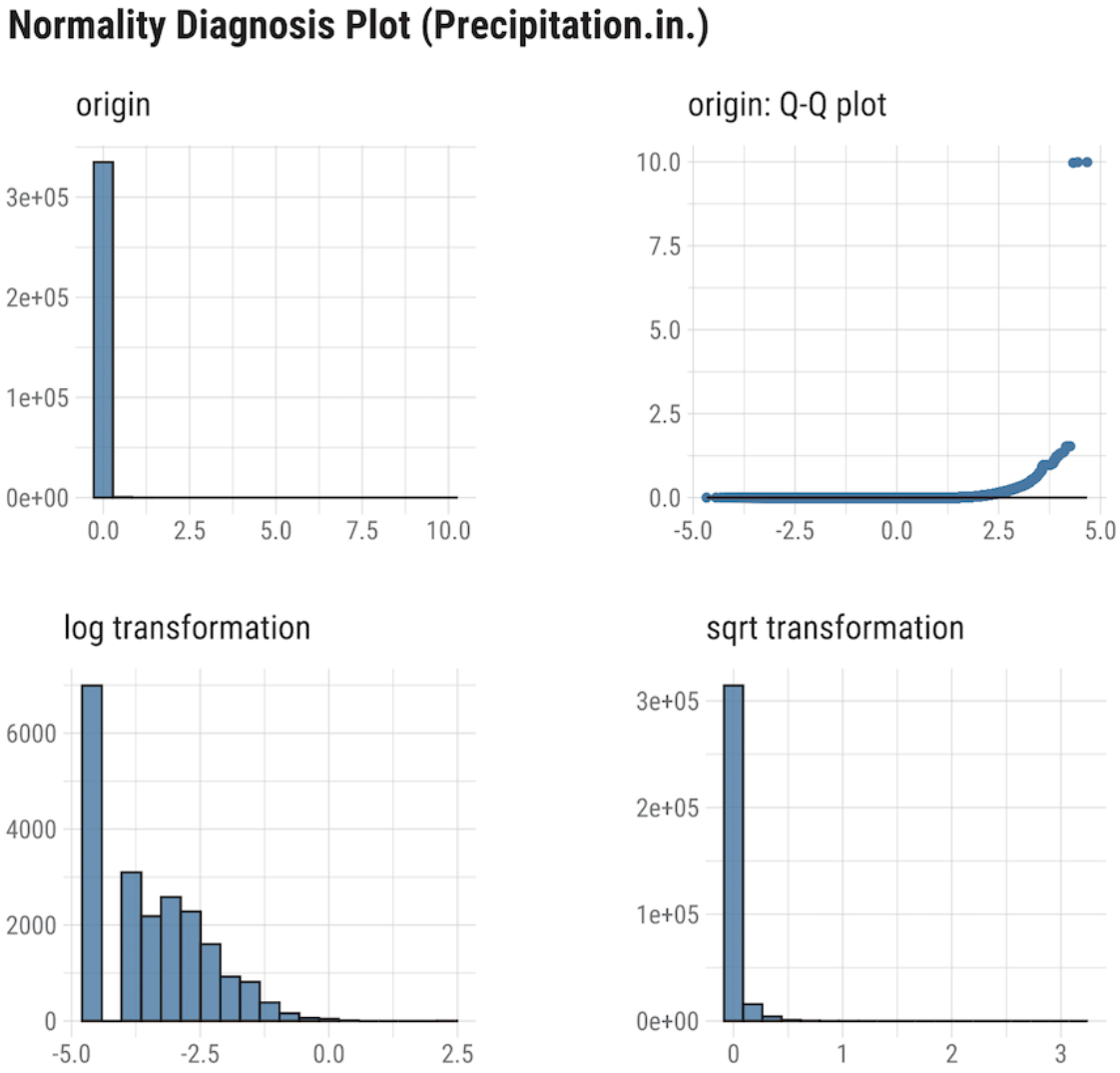
To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

*Below box plots show the outliers which are helpful in understanding the functioning of data.

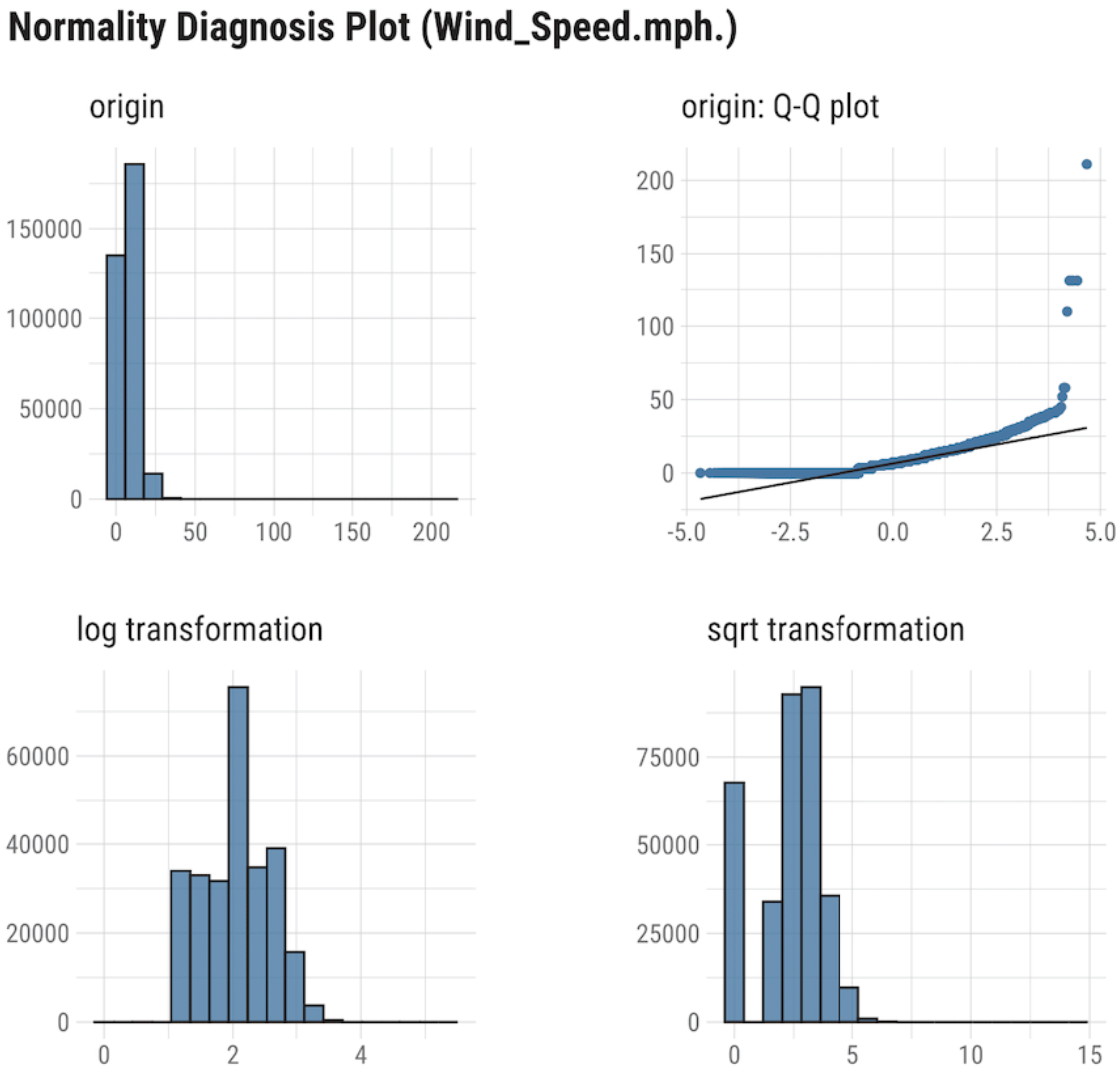


To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

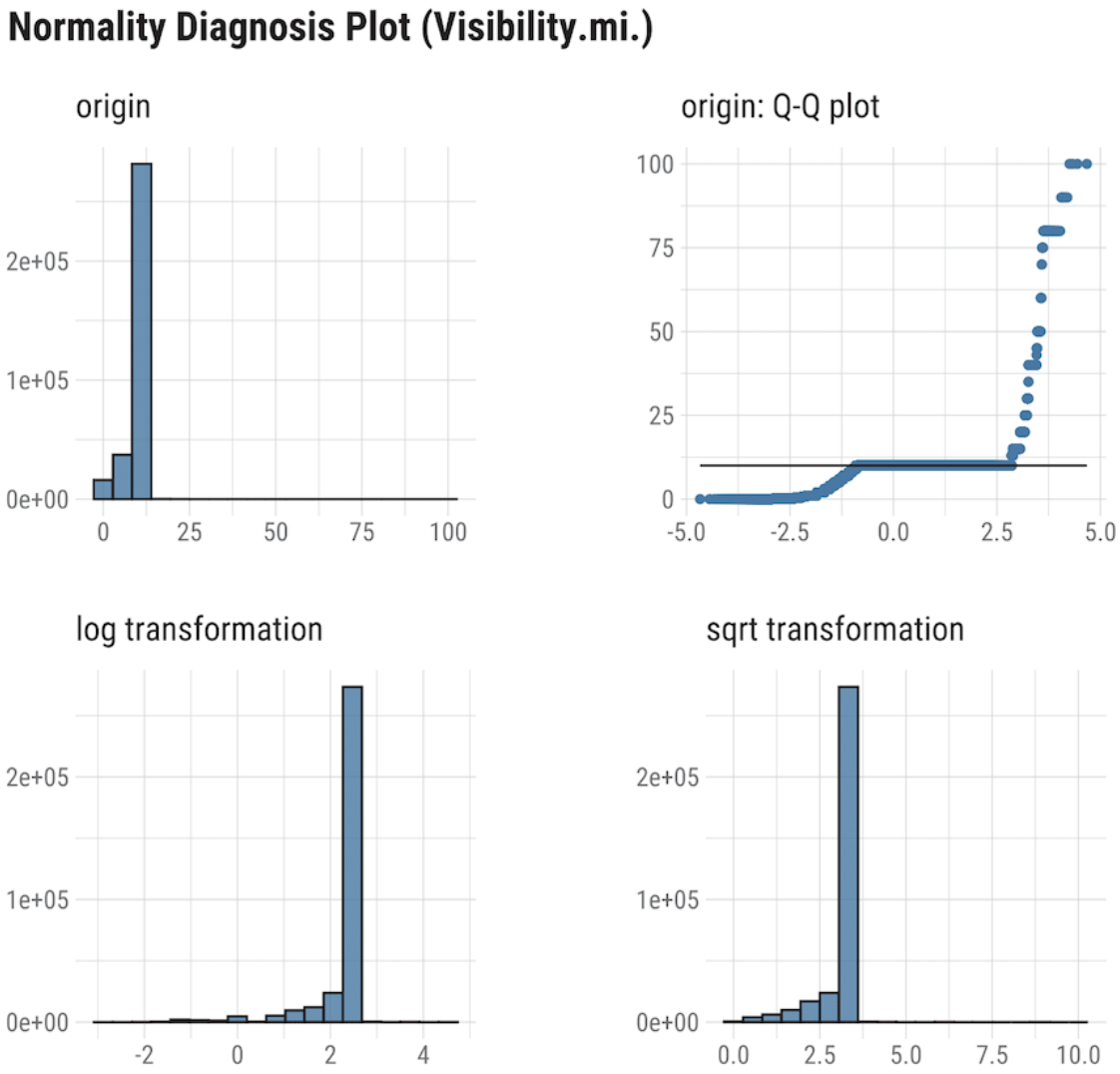
*To understand the normality of the variables, following graphs were plotted for each variable. Then the variable was applied with log transformation and sqrt transformation to identify further normality.



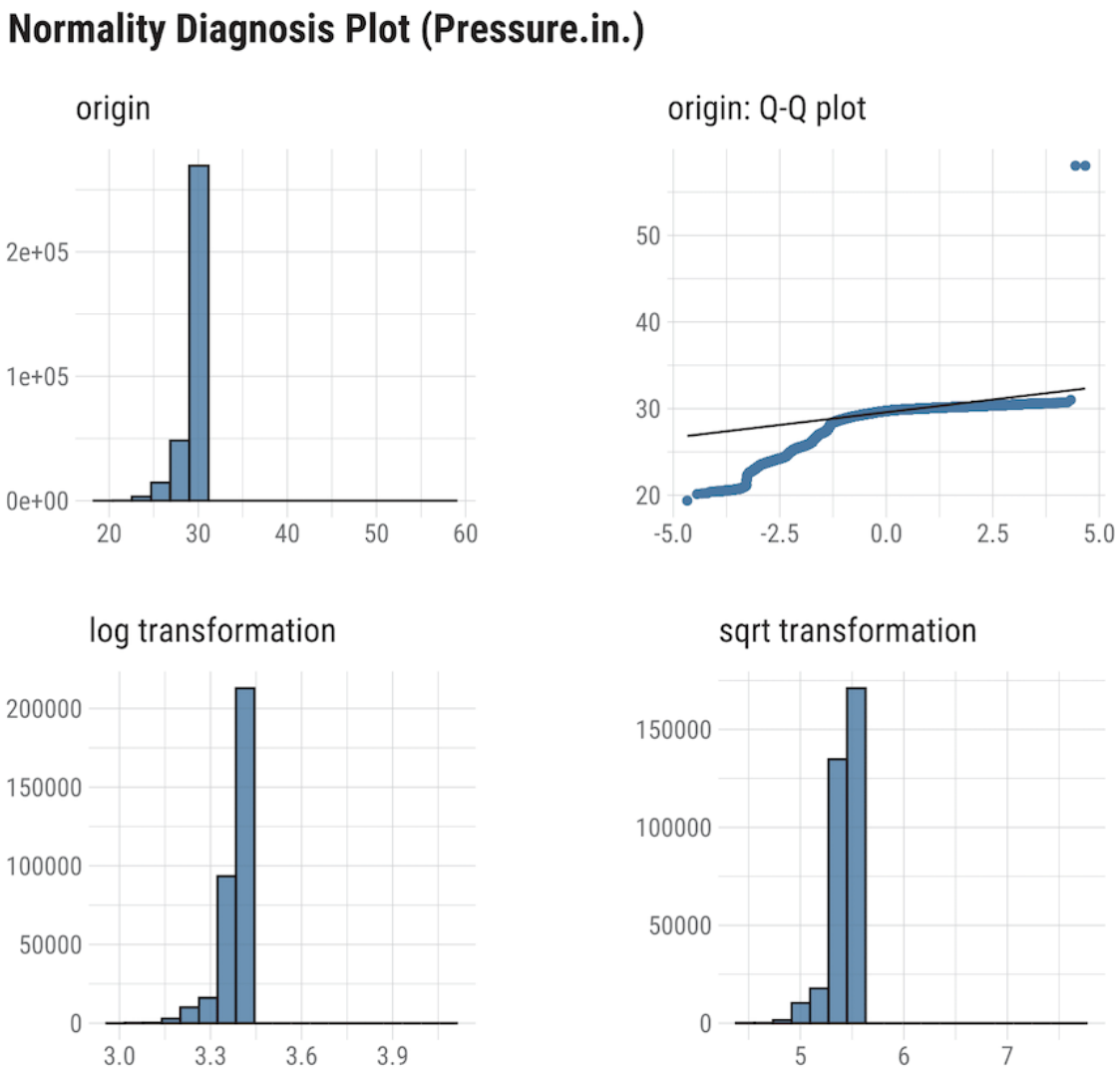
To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)



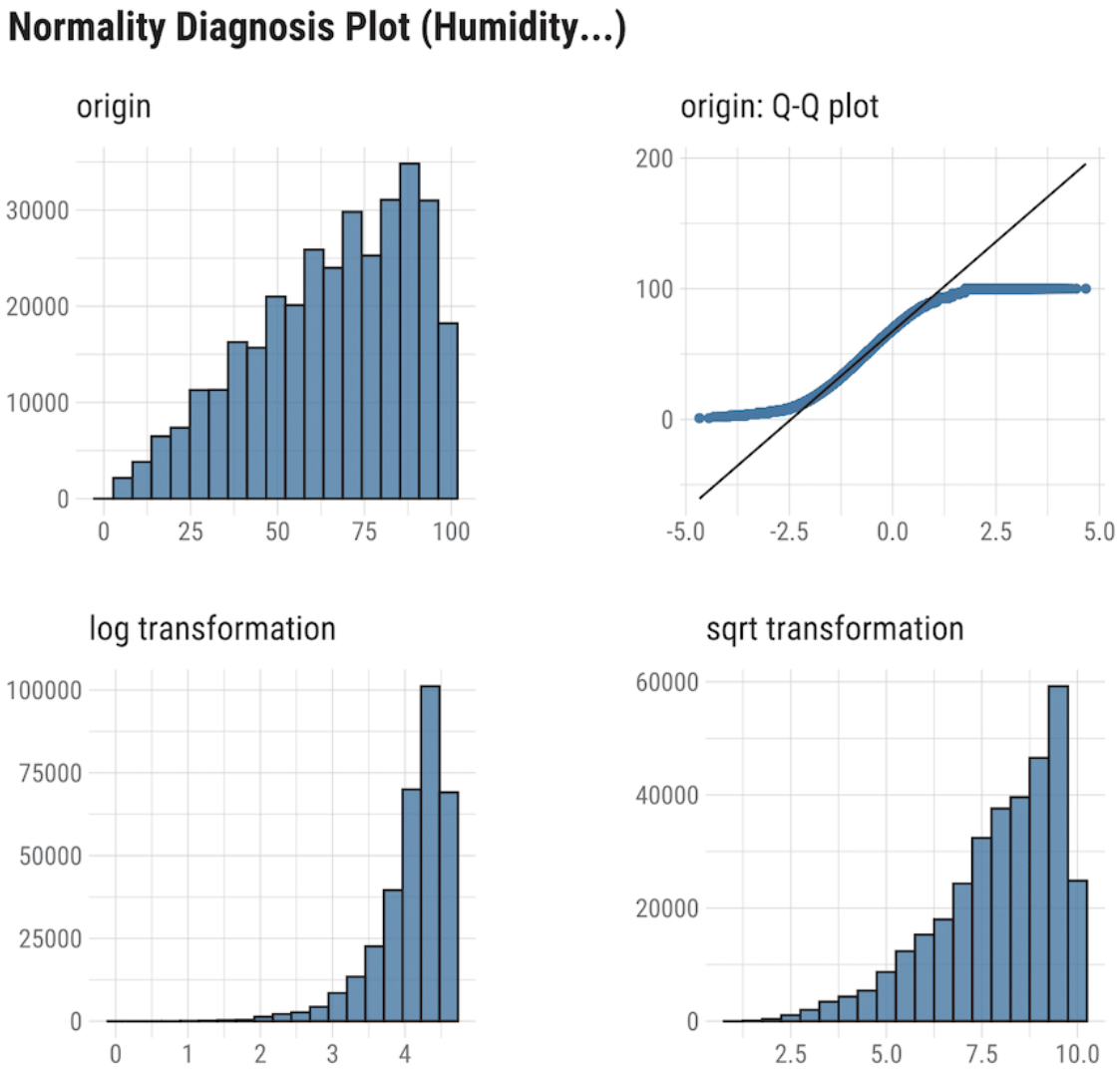
To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)



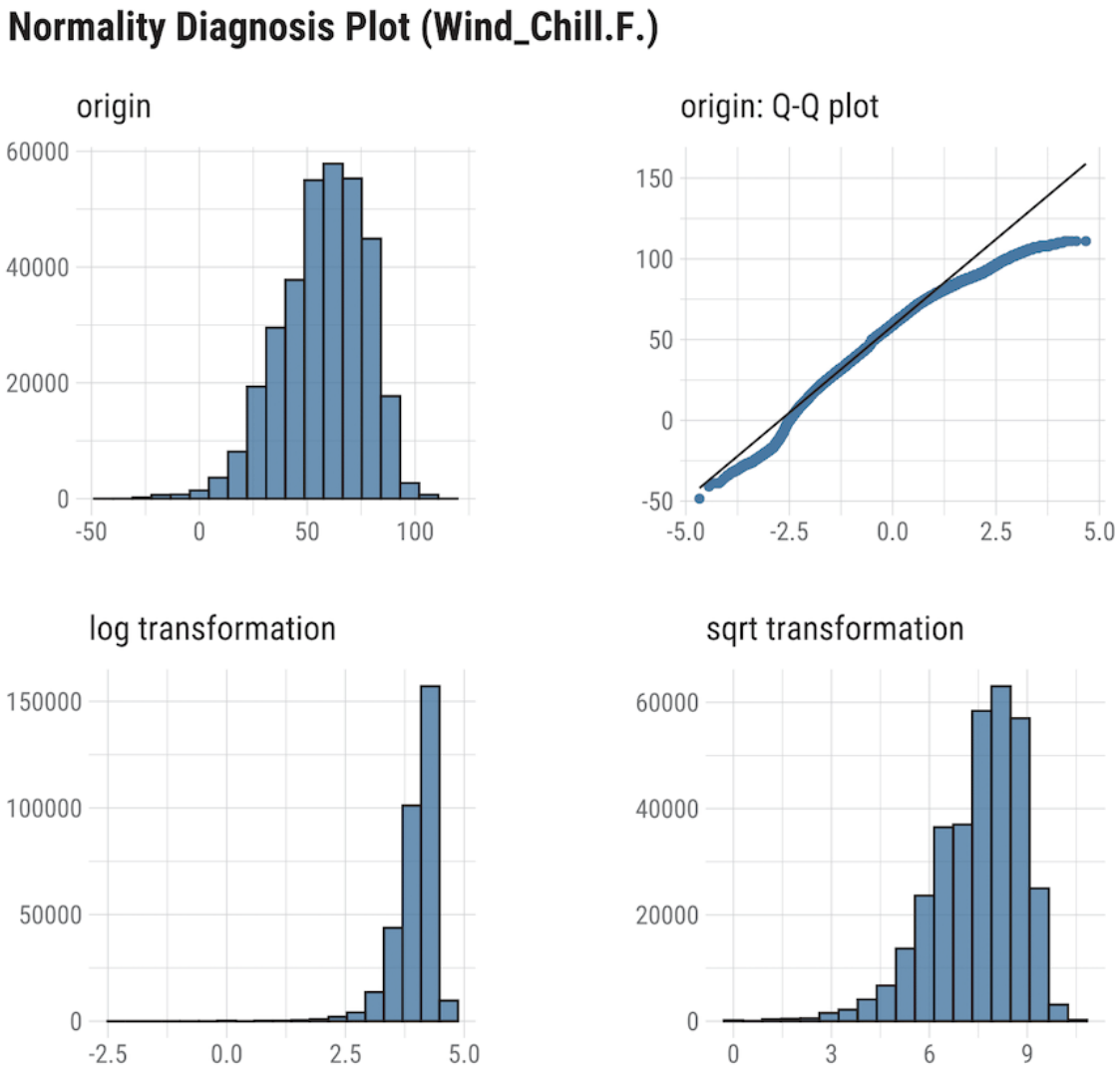
To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)



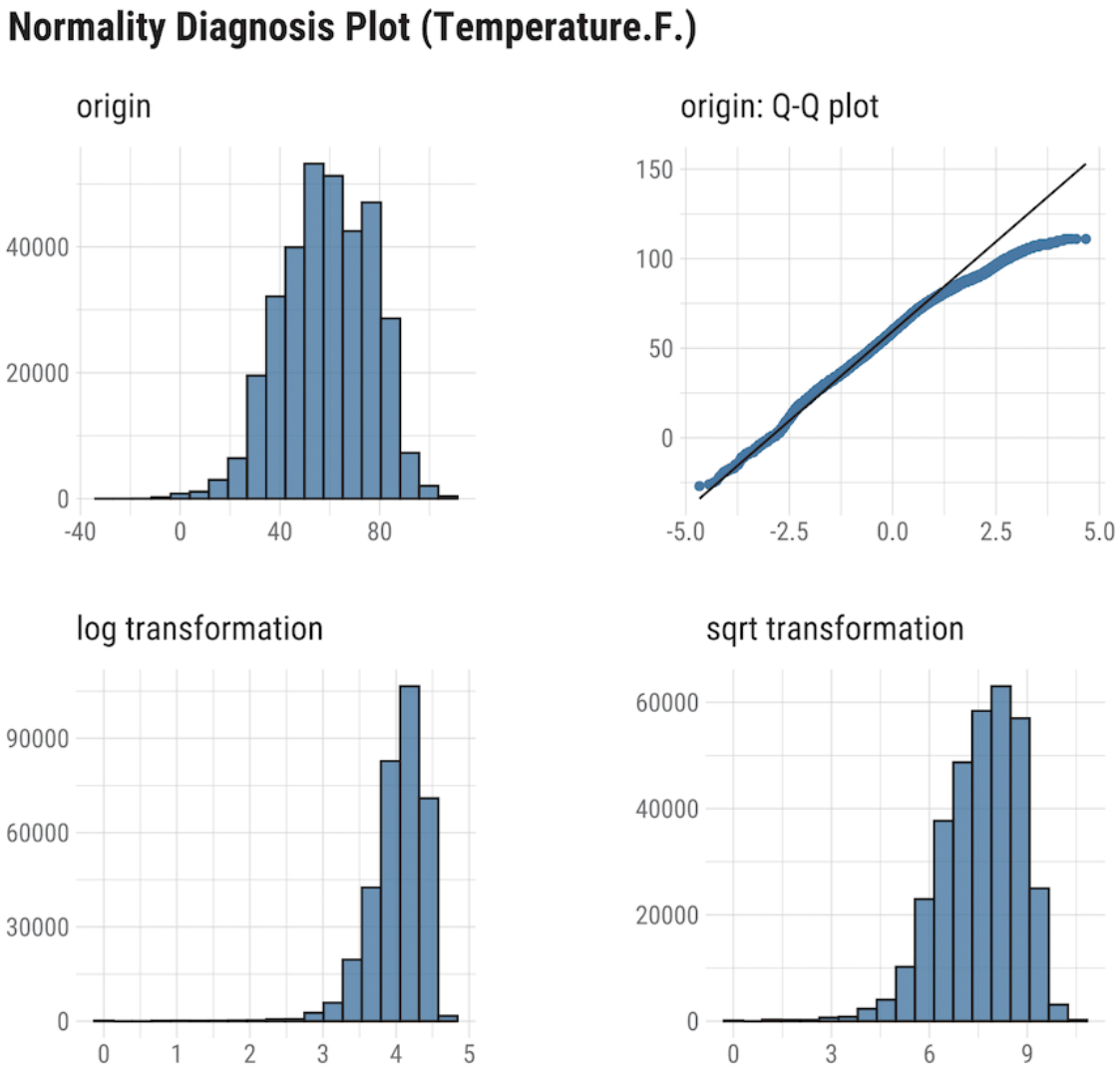
To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)



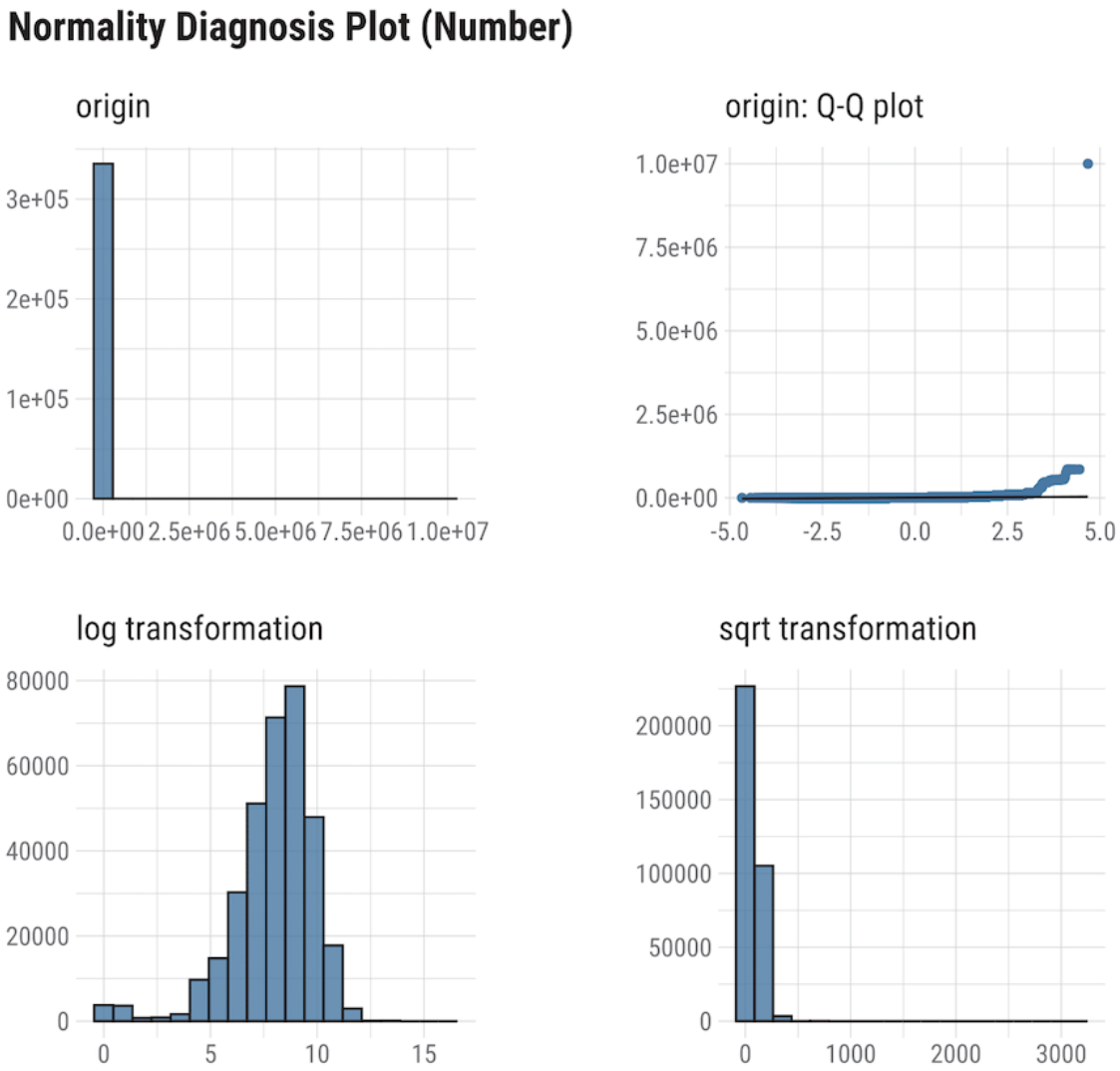
To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)



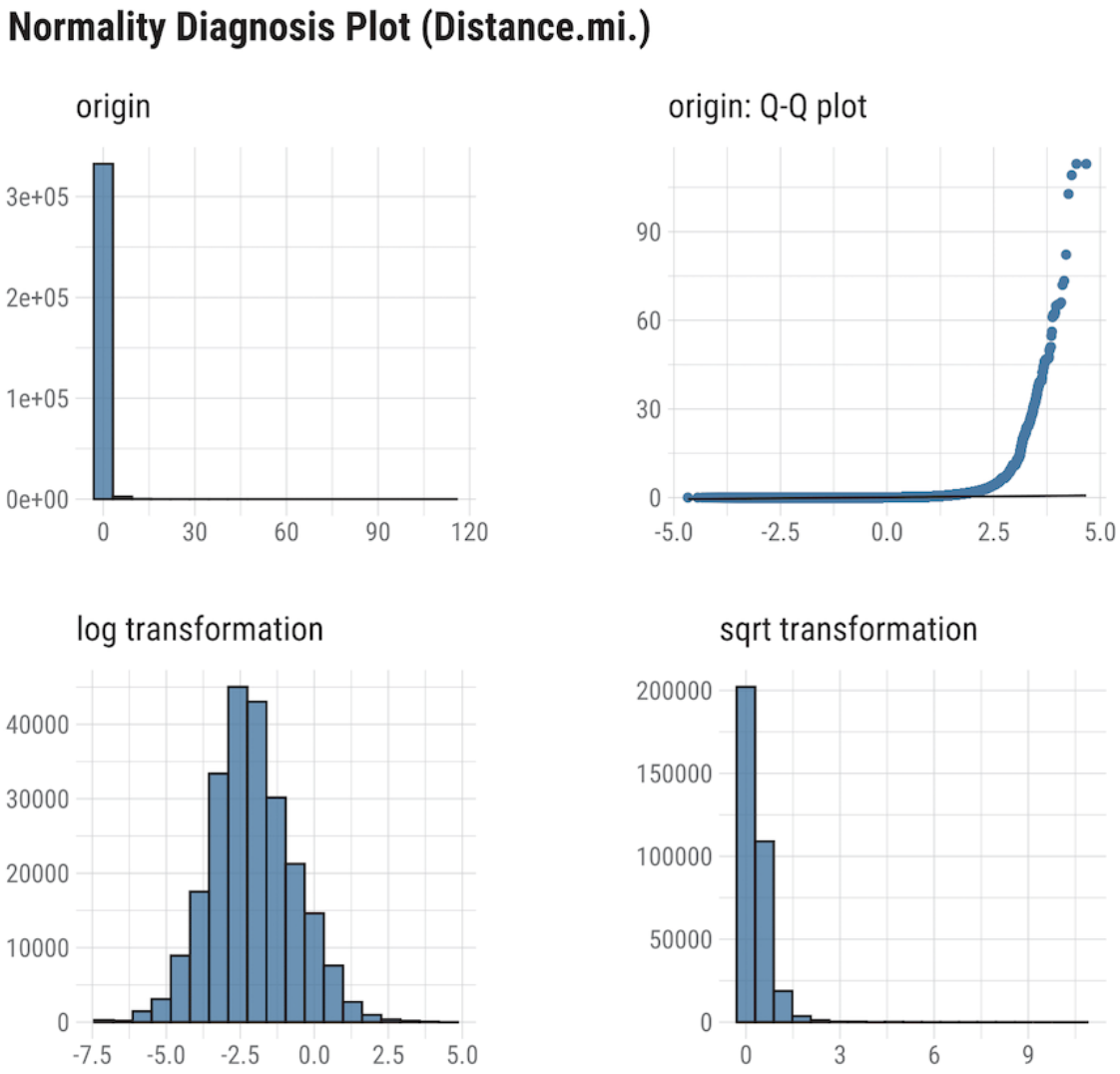
To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)



To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)



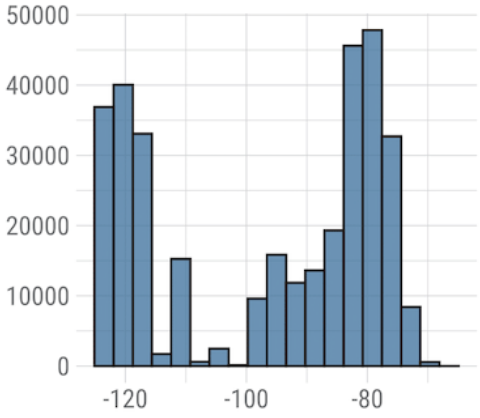
To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)



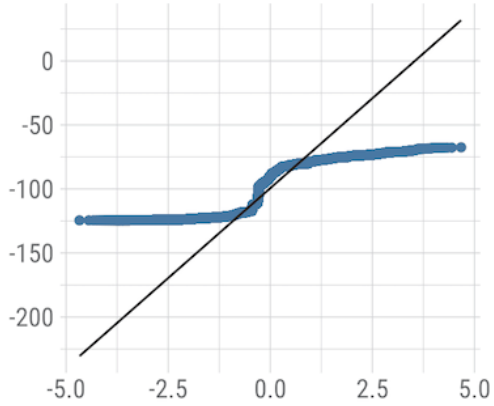
To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

Normality Diagnosis Plot (End_Lng)

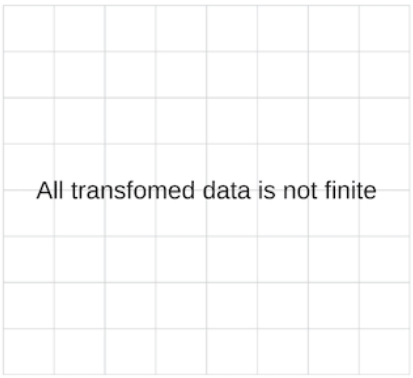
origin



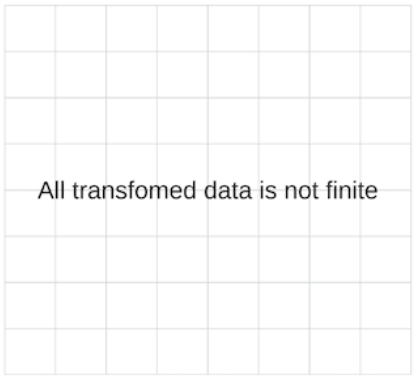
origin: Q-Q plot



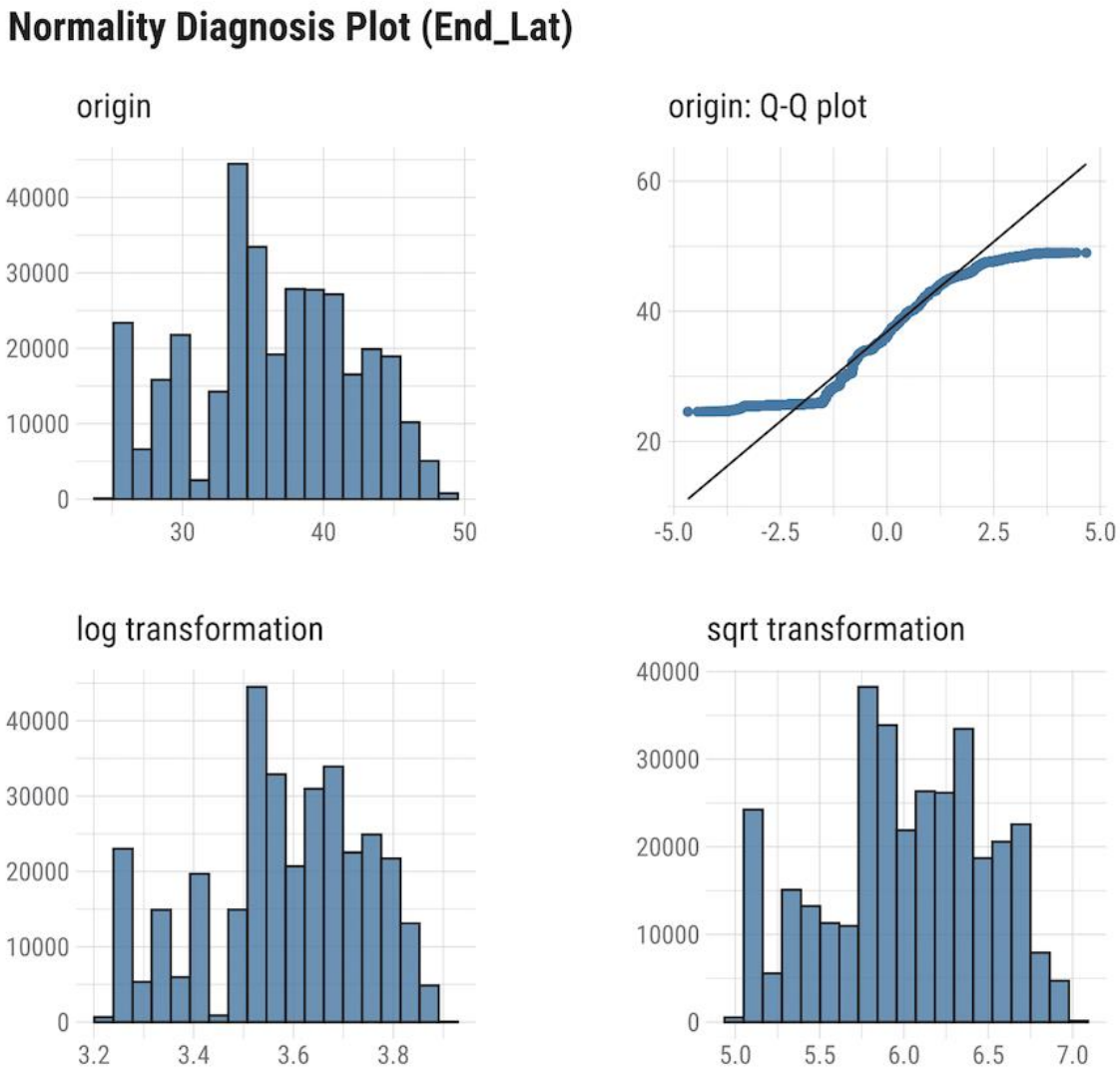
log transformation



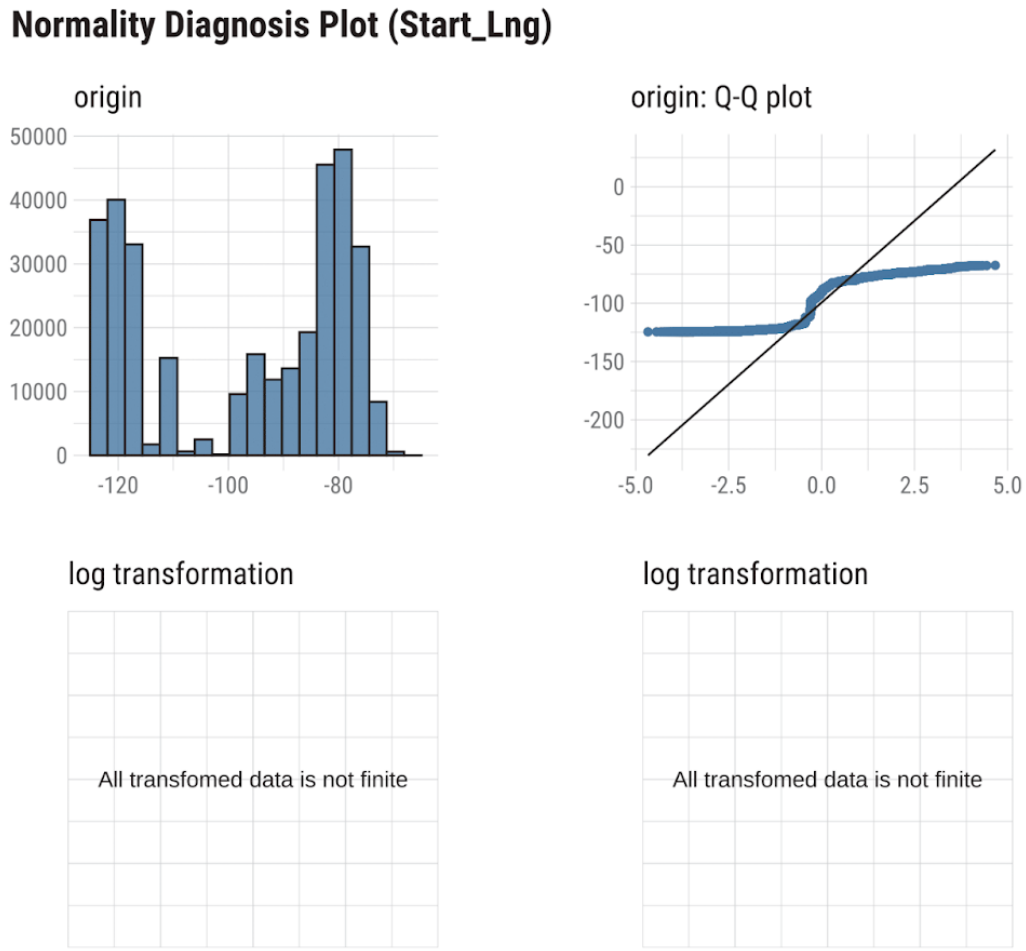
log transformation



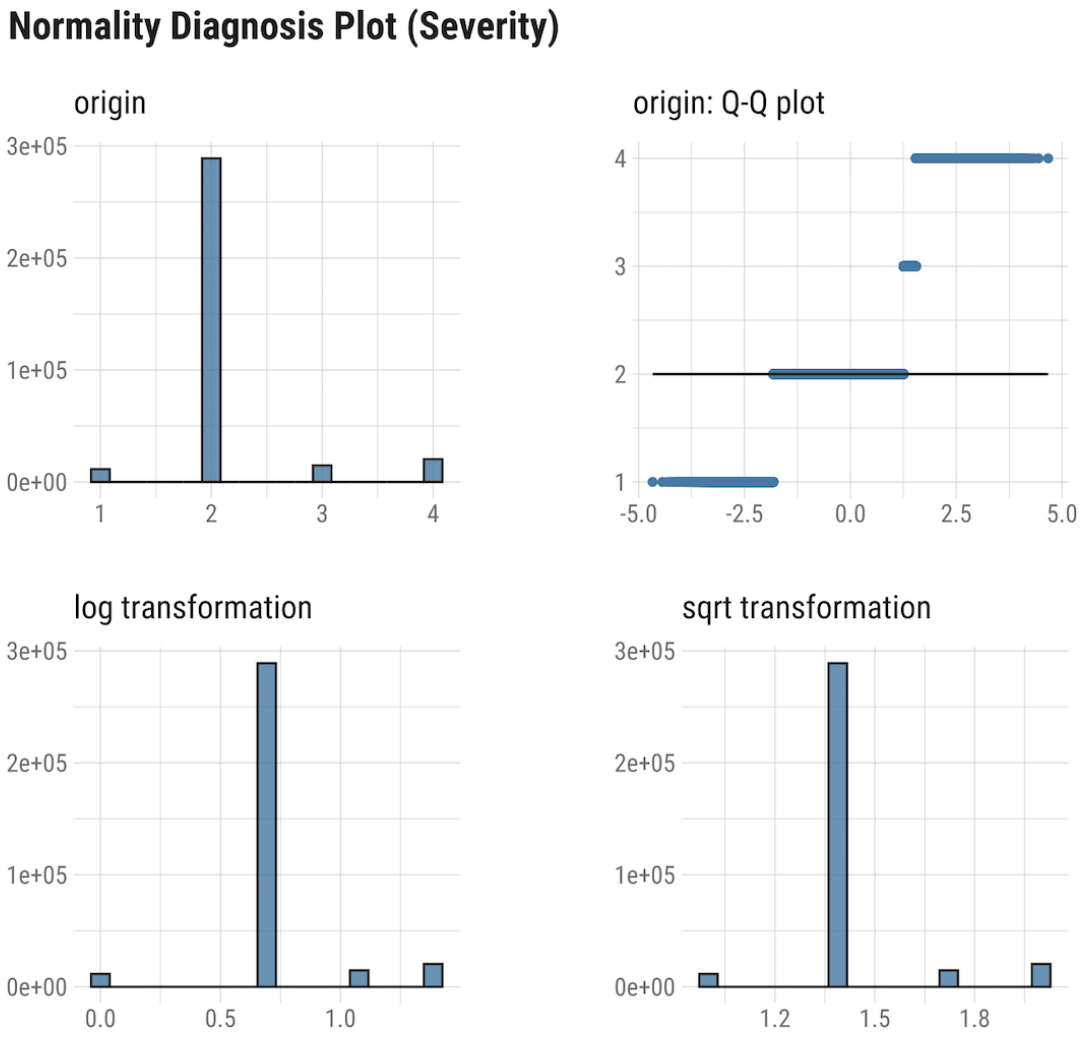
To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)



To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)



To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)



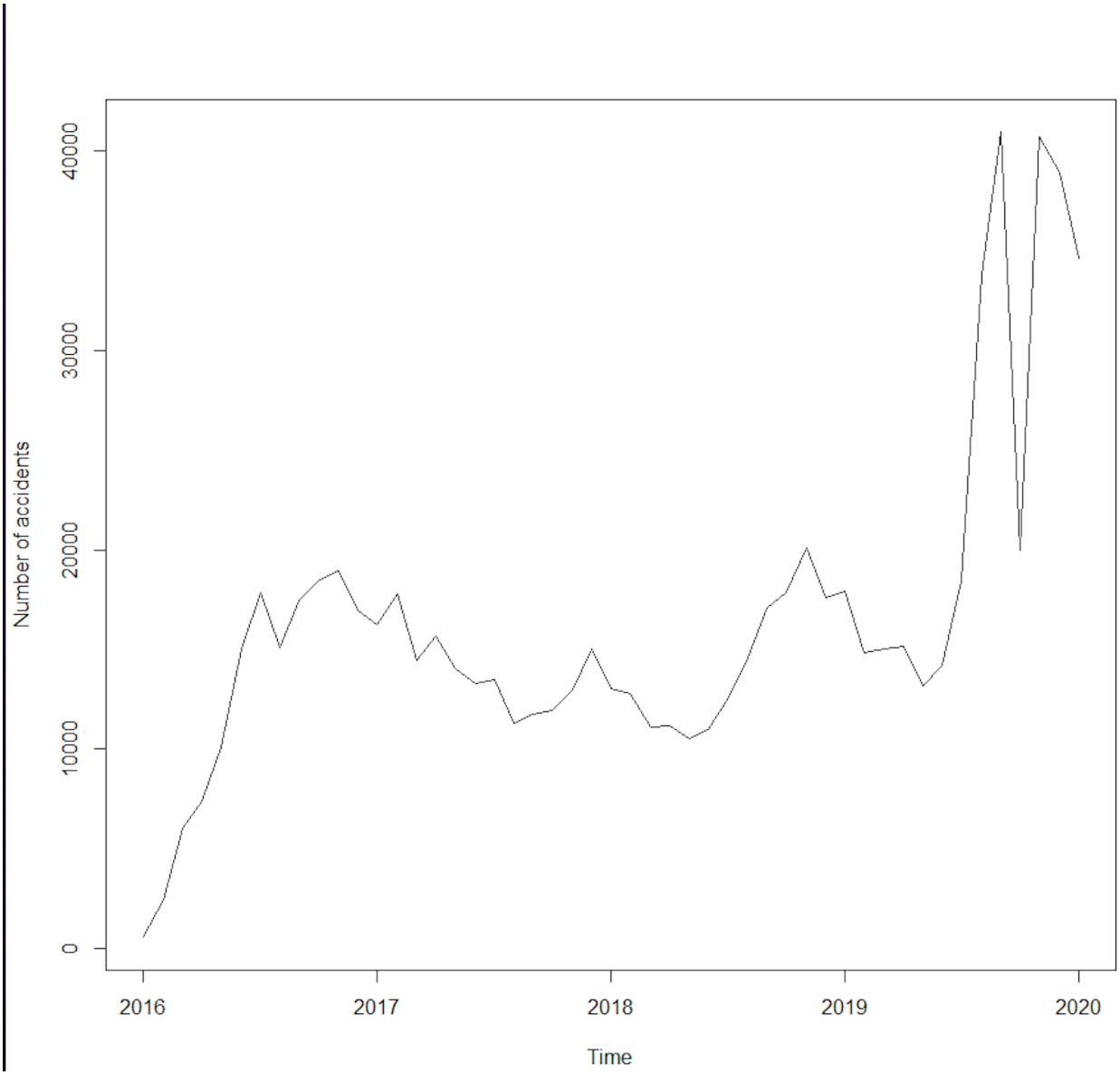
To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

INITIAL ANALYSIS

```
df_for_TSF = read.csv("US_Accidents.csv")
head(df_for_TSF)
df_for_TSF$MY = format(as.Date(df_for_TSF$Start_Time), "%m-%y")
count_accidents_MY = my_func(df_for_TSF,quo(MY))
install.packages("zoo")
library(zoo)
count_accidents_MY <- count_accidents_MY[order(as.yearmon(count_accidents_MY$MY, "%m-%Y")),]
head(count_accidents_MY,20)

summary(count_accidents_MY)
accidents.ts = ts(count_accidents_MY$my_count, start = 2016, end = 2020,frequency = 12)
plot(accidents.ts, xlab = "Time", ylab = "Number of accidents", )
plot(decompose(accidents.ts))
acf(accidents.ts)
pacf(accidents.ts)

install.packages("tseries")
library(tseries)
adf.test(accidents.ts)

seasonplot(accidents.ts, xlab = "Month", ylab = "Number of accidents", main="Seasonal plot",
year.labels.left = TRUE, col =1:20, pch = 19)

monthplot(accidents.ts, xlab = "Month", ylab = "Number of accidents", main="Seasonal Standard deviation",xaxt = "n")
axis(1, at = 1:12, labels = month.abb, cex = 0.8)

ntrain = length(accidents.ts) - 12
train.ts = window(accidents.ts, start =c(2016,1), end = c(2016,ntrain))
valid.ts = window(accidents.ts, start =c(2016, ntrain +1 ), end = c(2016,ntrain+ 12))

install.packages("MLmetrics")
library(MLmetrics)
```

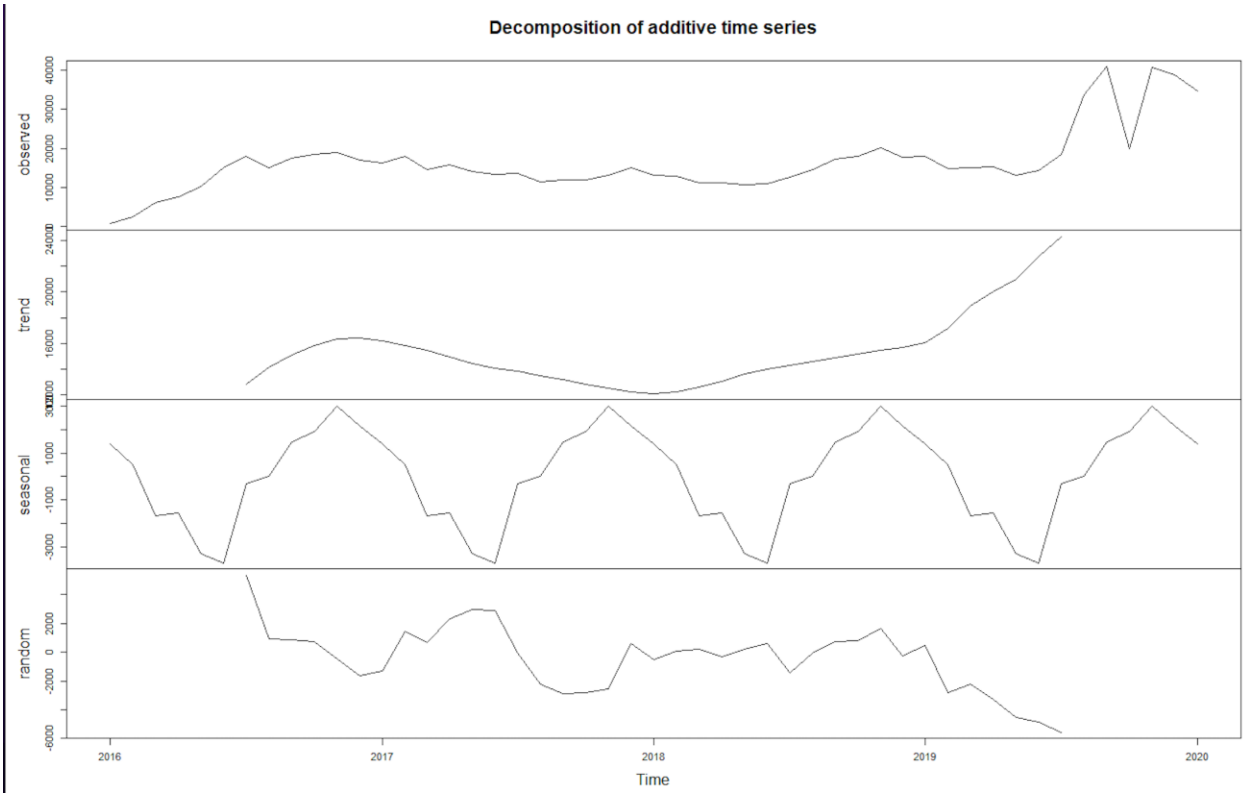
To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

*Following graph is used to visualize number of accidents occurring each year and we can clearly notice the spike in years 2019 and 2020.



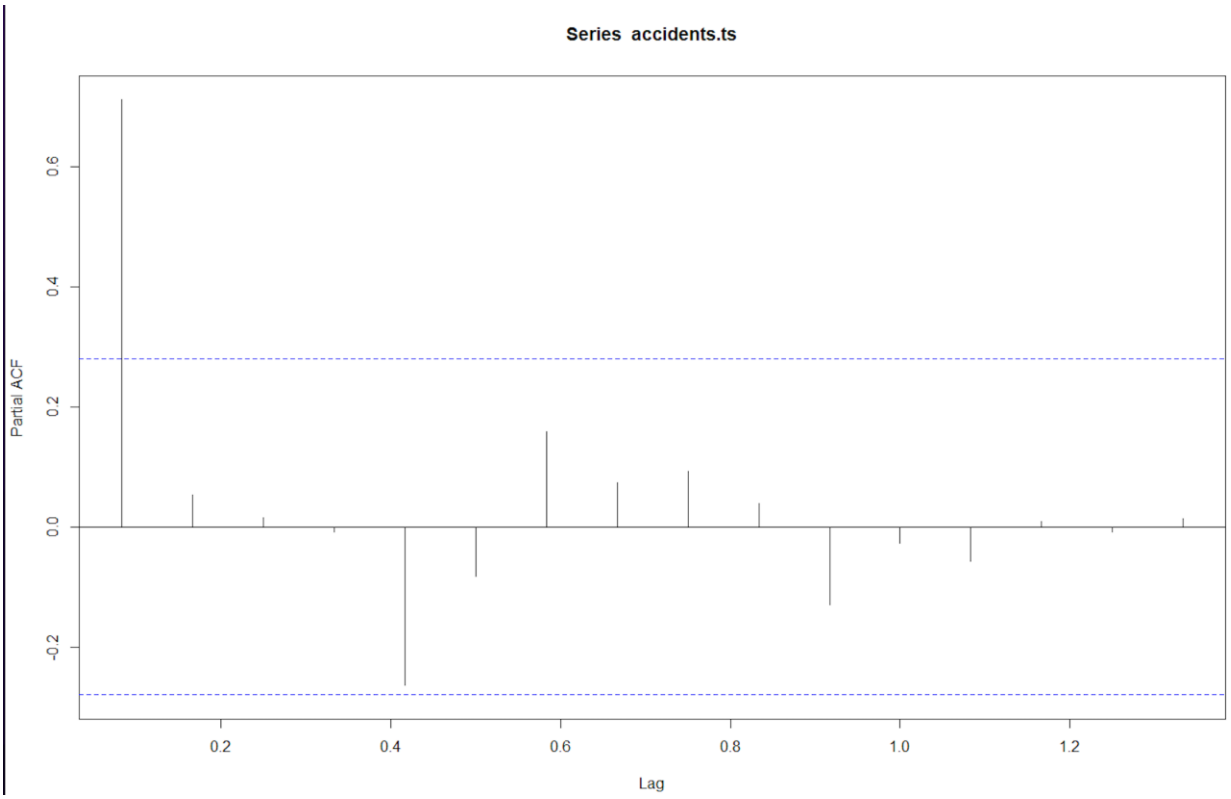
To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

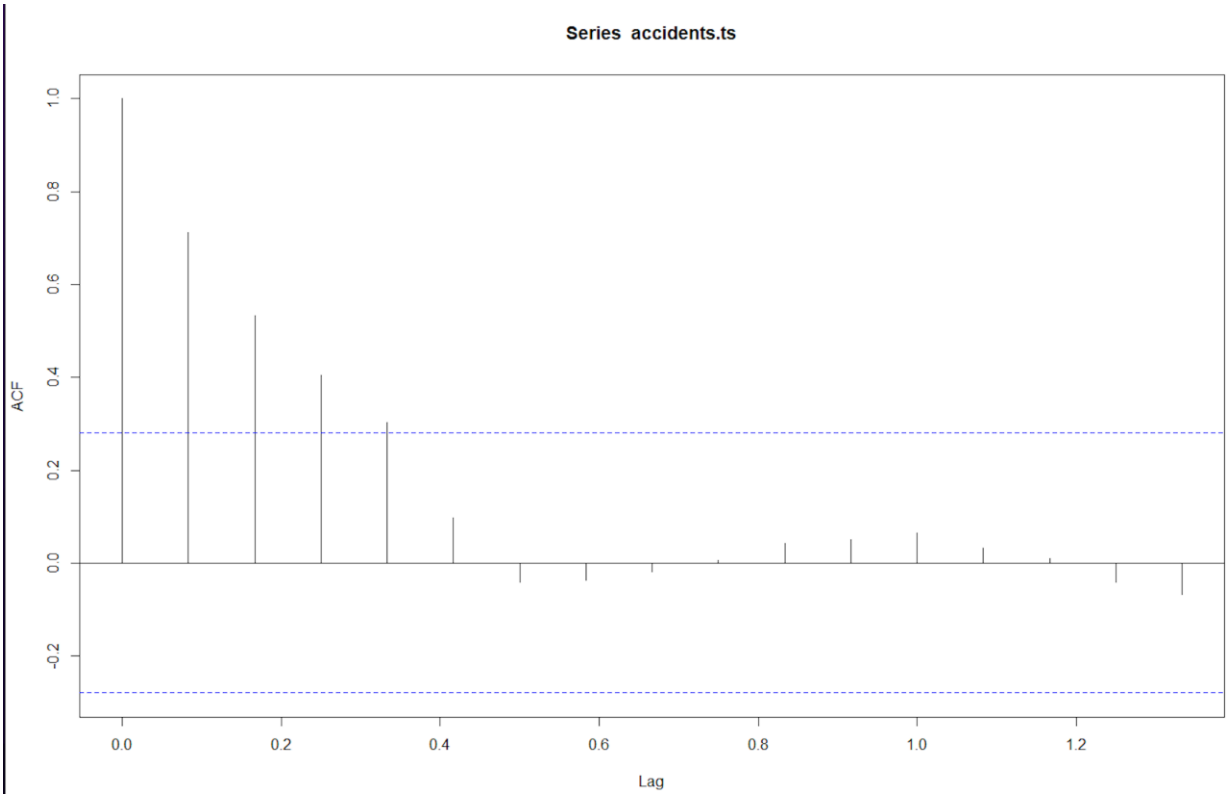
*In the below graph, we can observe clear outliers from 2019 to 2020 which changed the pattern of the trends.



To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

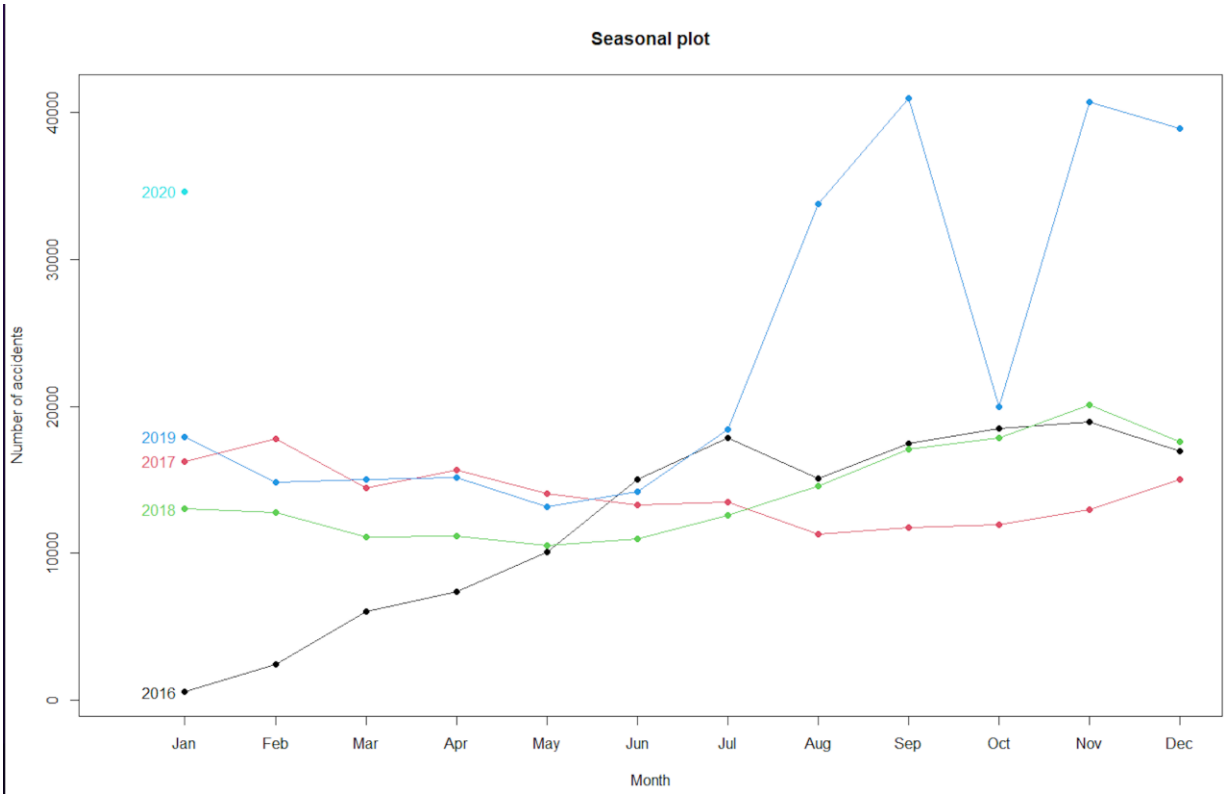
*Following graphs were plotted to visualize auto correlation functions.





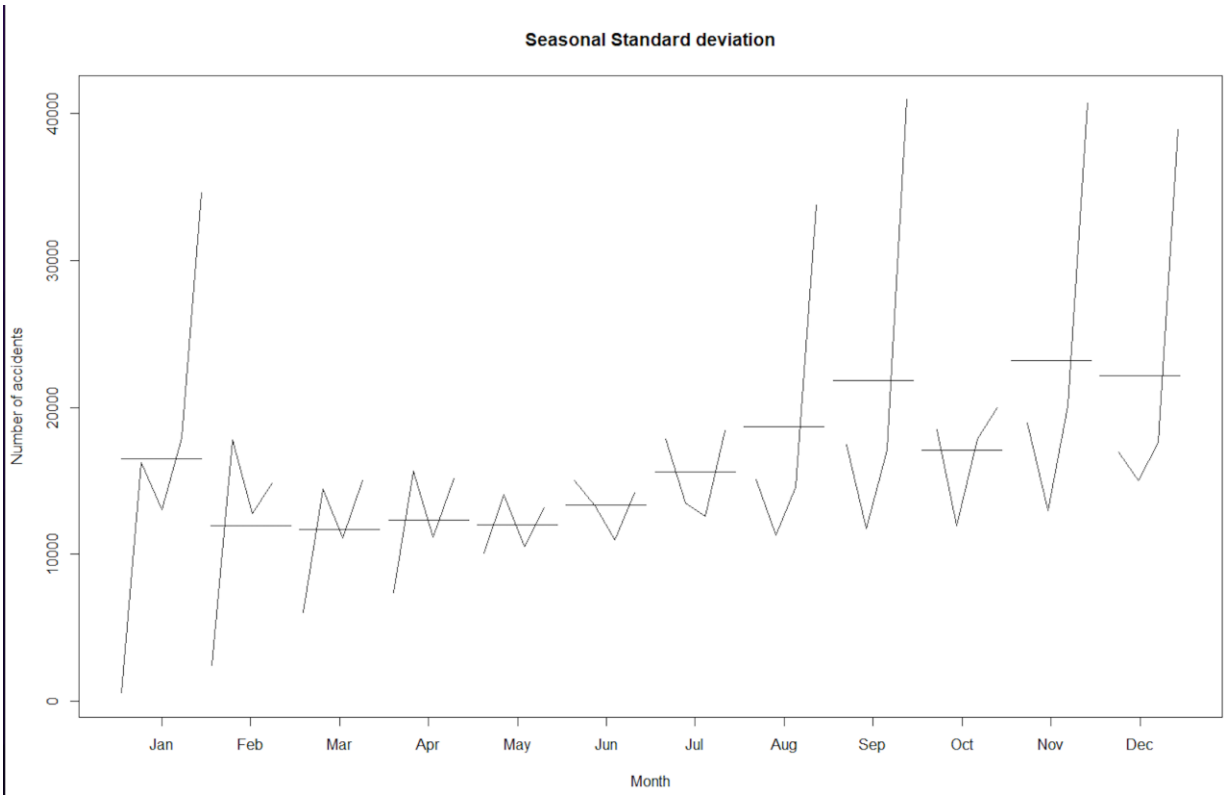
To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

*Below graphs shows number of accidents occurred in each month with respect to each year.



To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

*Below graph shows the number of accidents in each month and we can see that each month crossed mean number of accidents for each year.



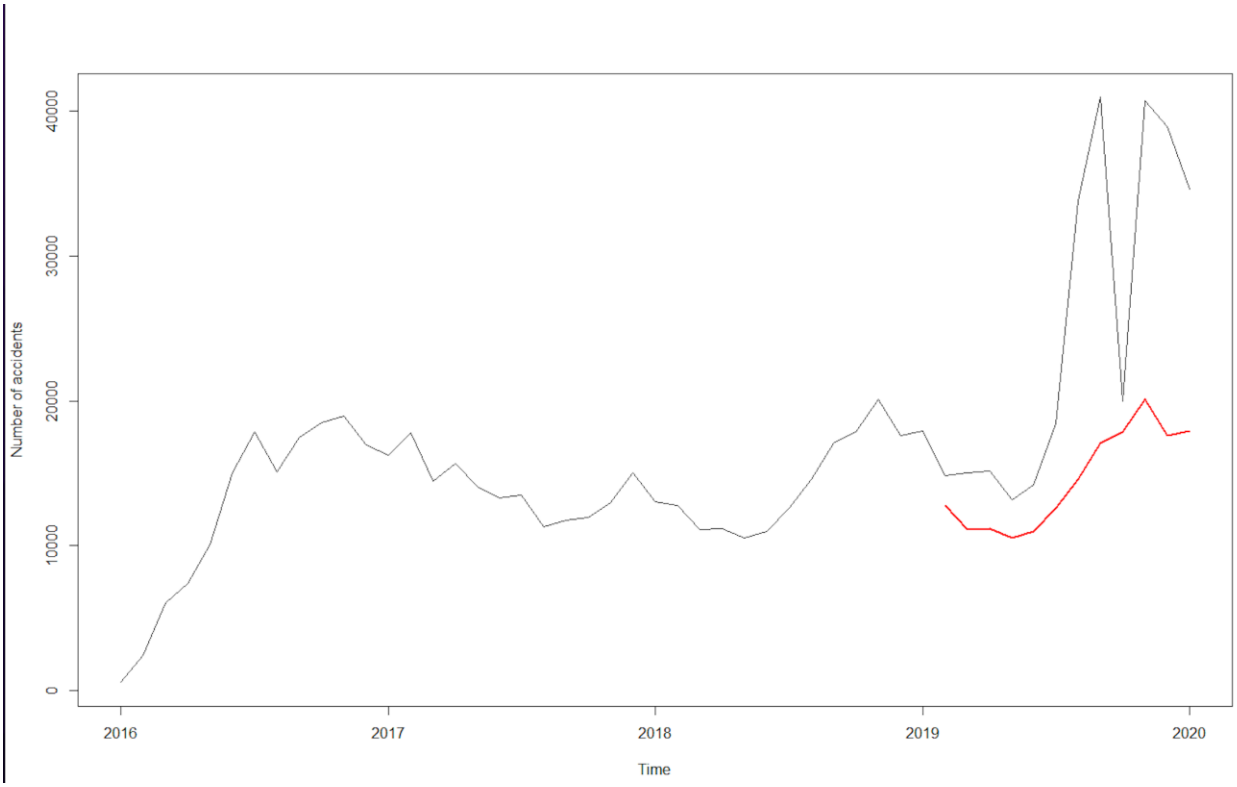
library(forecast)

Naive model

```
model_naive = snaive(train.ts, h = 12)
MAPE(model_naive$mean, 12)*100
accuracy(model_naive, valid.ts)
plot(accidents.ts, xlab = "Time", ylab = "Number of accidents", )
lines(model_naive$mean, col="red", lwd=2)
```

To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

Naive forecasting models are based exclusively on historical observation of sales or other variables, such as earning and cash flows. They do not attempt to explain the underlying causal relationships that produce the variable being forecast.



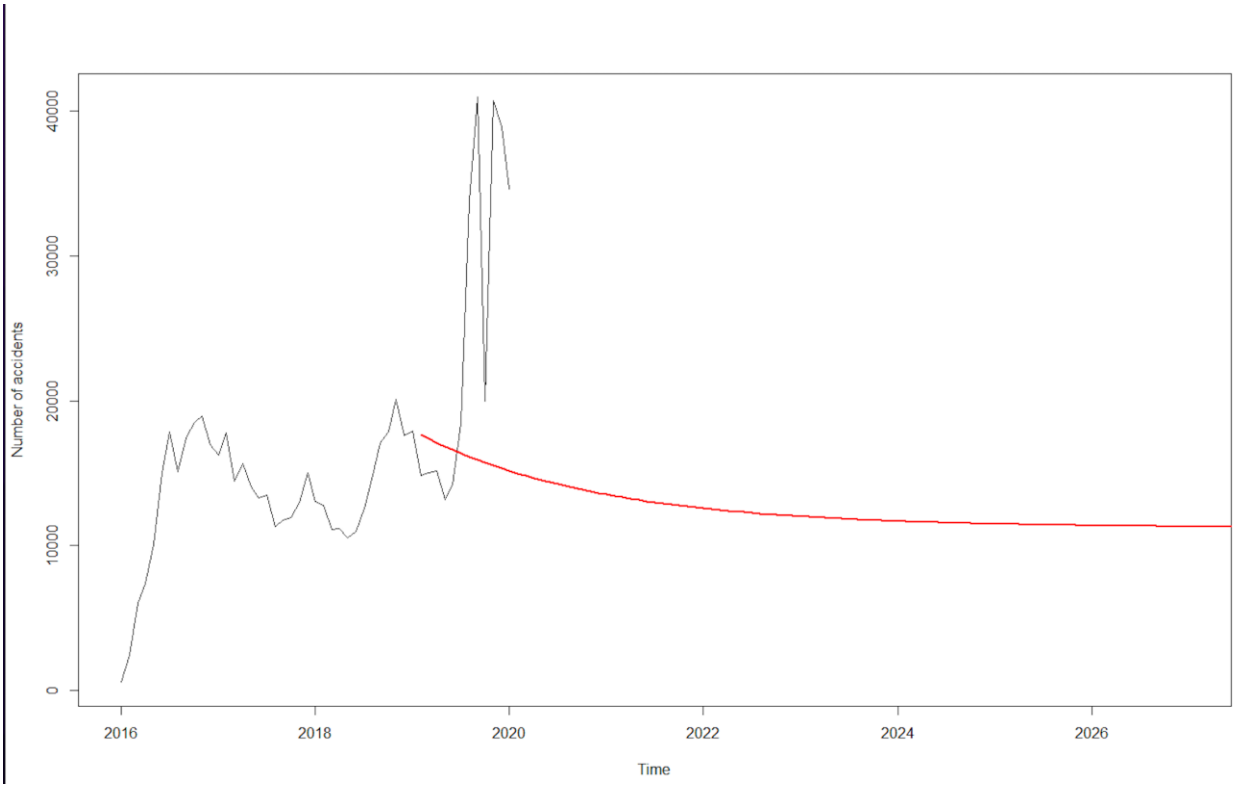
ARIMA model

ARIMA, short for 'Auto Regressive Integrated Moving Average' is actually a class of models that 'explains' a given time series based on its own past values, that is, its own lags and the lagged forecast errors, so that equation can be used to forecast future values.

```
model_arima = auto.arima(train.ts)
summary(model_arima)
model_arima_pred = forecast(model_arima, h= 12, level = 0)
accuracy(model_arima_pred, valid.ts)
Box.test(model_arima$residuals)
plot(model_arima_pred)

pred = predict(auto.arima(train.ts), n.ahead = 10*12)
plot(accidents.ts, xlab = "Time", ylab = "Number of accidents", xlim = c(2016,2027))
```

To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

```
lines(pred$se, col="red", lwd=2)

Forecast_val = ((forecast(accidents.ts, h=30)))
plot(accidents.ts, xlab = "Time", ylab = "Number of accidents", xlim = c(2016,2027))
lines(Forecast_val$mean, col="red", lwd=2)

Forecast_val2 = (forecast(auto.arima(train.ts), h=30))
plot(accidents.ts, xlab = "Time", ylab = "Number of accidents", xlim = c(2016,2027))
lines(Forecast_val2$mean, col="red", lwd=2)

highest_accident_zipcode=my_func(df, quo(Zipcode))
head(highest_accident_zipcode, 20)
ggplot(data=head(highest_accident_zipcode, 20), aes(x=Zipcode, y=my_count)) +
  geom_bar(stat="identity", fill="steelblue", width = 1)+
  geom_text(aes(label=my_count), vjust=1.6, color="white", size=3.5)+
  theme_minimal()

highest_accident_area = my_func(df, quo(Street))
head(highest_accident_area, 20)
ggplot(data=head(highest_accident_area, 20), aes(x=Street, y=my_count)) +
  geom_bar(stat="identity", fill="steelblue", width = 1)+
  geom_text(aes(label=my_count), vjust=1.6, color="white", size=3.5)+
  theme_minimal()
```



To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

KRIGING MODEL

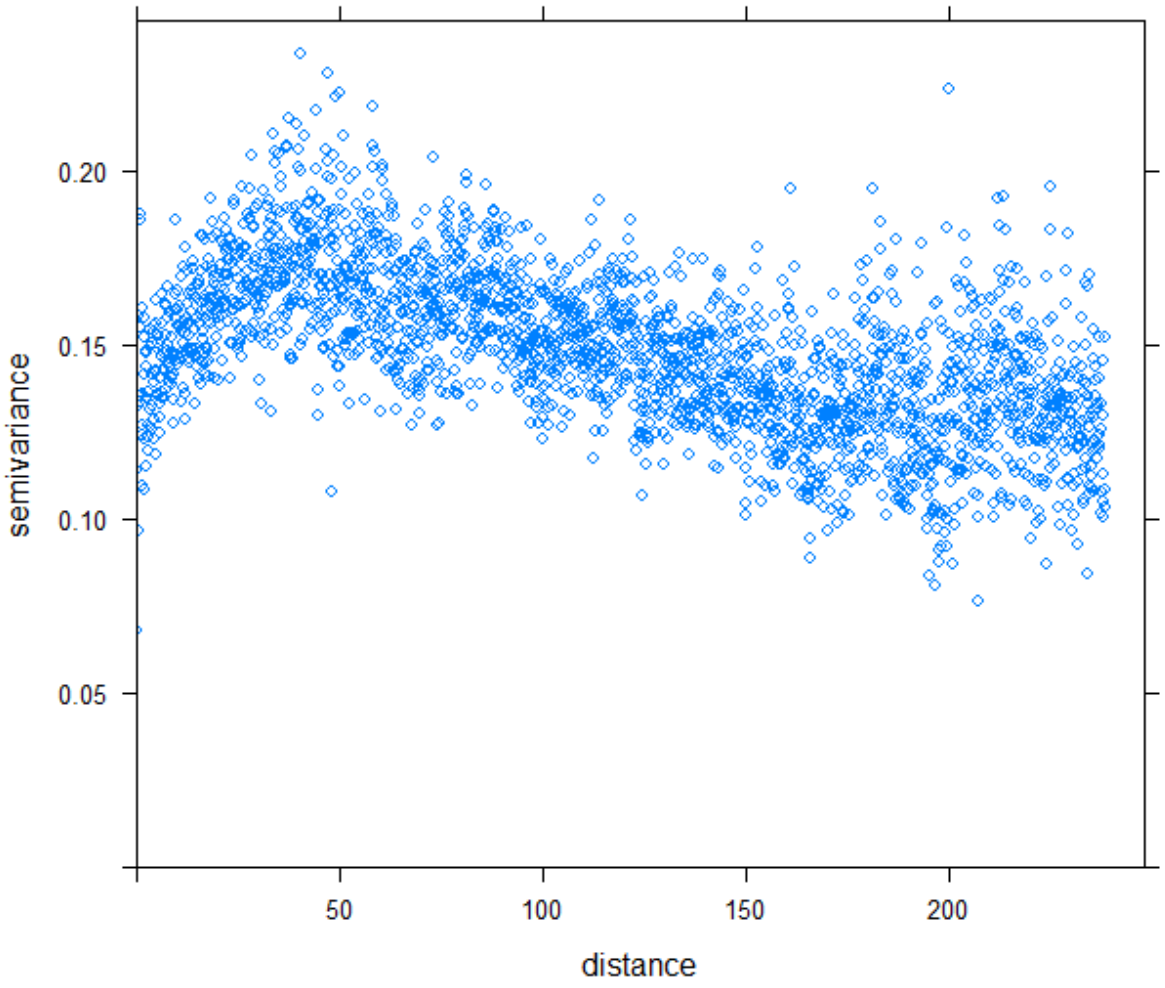
Kriging is an advanced geostatistical procedure that generates an estimated surface from a scattered set of points with z-values. Unlike other interpolation methods in the Interpolation toolset, to use the Kriging tool effectively involves an interactive investigation of the spatial behavior of the phenomenon represented by the z-values before you select the best estimation method for generating the output surface.

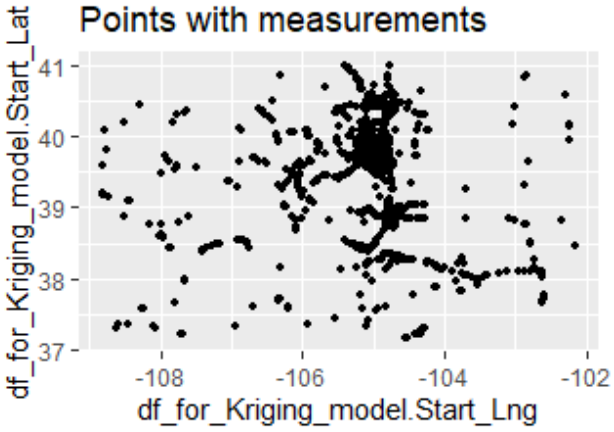
```
#Data Kriging model
#For Colorado
library(dplyr)
install.packages("raster")
library(sp)
library(rgdal)
library(raster)
library(gstat)
set.seed(100)
df_for_Kriging_model = data.frame(df[df$State == "CO",])
df_for_Kriging_model = sample_n(df_for_Kriging_model, 25)
#df_for_Kriging_model = (df_for_Kriging_model, 100)
df_for_Kriging_model = data.frame(df_for_Kriging_model$Start_Lng,df_for_Kriging_model$Start_Lat,df_for_Krigi
random_generator = sample(1:nrow(df_for_Kriging_model), round(nrow(df_for_Kriging_model)*.3), replace=F)
train_df = df_for_Kriging_model %>%
  filter(!df_for_Kriging_model.ID %in% random_generator) %>%
  slice(1:10000)
coordinates(train_df) = c("df_for_Kriging_model.Start_Lng", "df_for_Kriging_model.Start_Lat")
proj4string(train_df) = CRS("+proj=longlat +datum=WGS84")
lzn.vgm = variogram(log(df_for_Kriging_model.Severity) ~ df_for_Kriging_model.Start_Lng+ df_for_Kriging_mode
plot(lzn.vgm)
lzn.fit = fit.variogram(lzn.vgm, vgm(c("Gau", "Sph", "Mat", "Exp")), fit.kappa = TRUE)
```

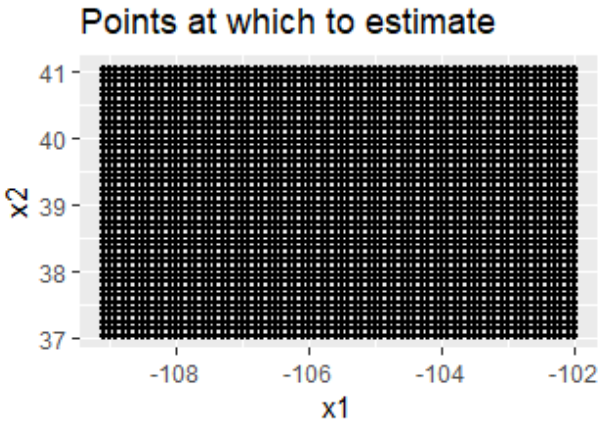
To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

```
# load some spatial data. Administrative Boundary
us = getData('GADM', country = 'US', level = 1)
us$NAME_1
colorado <- us[us$NAME_1 == "Colorado",]


# check the CRS to know which map units are used
proj4string(colorado)
# "+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0"

# Create a grid of points within the bbox of the SpatialPolygonsDataFrame
# colorado with decimal degrees as map units
grid <- makegrid(colorado, cellsize = 0.1) # cellsize in map units!

# grid is a data.frame. To change it to a spatial data set we have to
grid <- SpatialPoints(grid, proj4string = CRS(proj4string(colorado)))

plot(colorado)
plot(grid, pch = ".", add = T)

plot1 <- df_for_Kriging_model %>% as.data.frame %>%
  ggplot(aes(df_for_Kriging_model.Start_Lng, df_for_Kriging_model.Start_Lat)) + geom_point(size=1) + coord_
  ggtitle("Points with measurements")

# this is clearly gridded over the region of interest
plot2 <- grid %>% as.data.frame %>%
  ggplot(aes(x1, x2)) + geom_point(size=1) + coord_equal() +
  ggtitle("Points at which to estimate")
```

To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

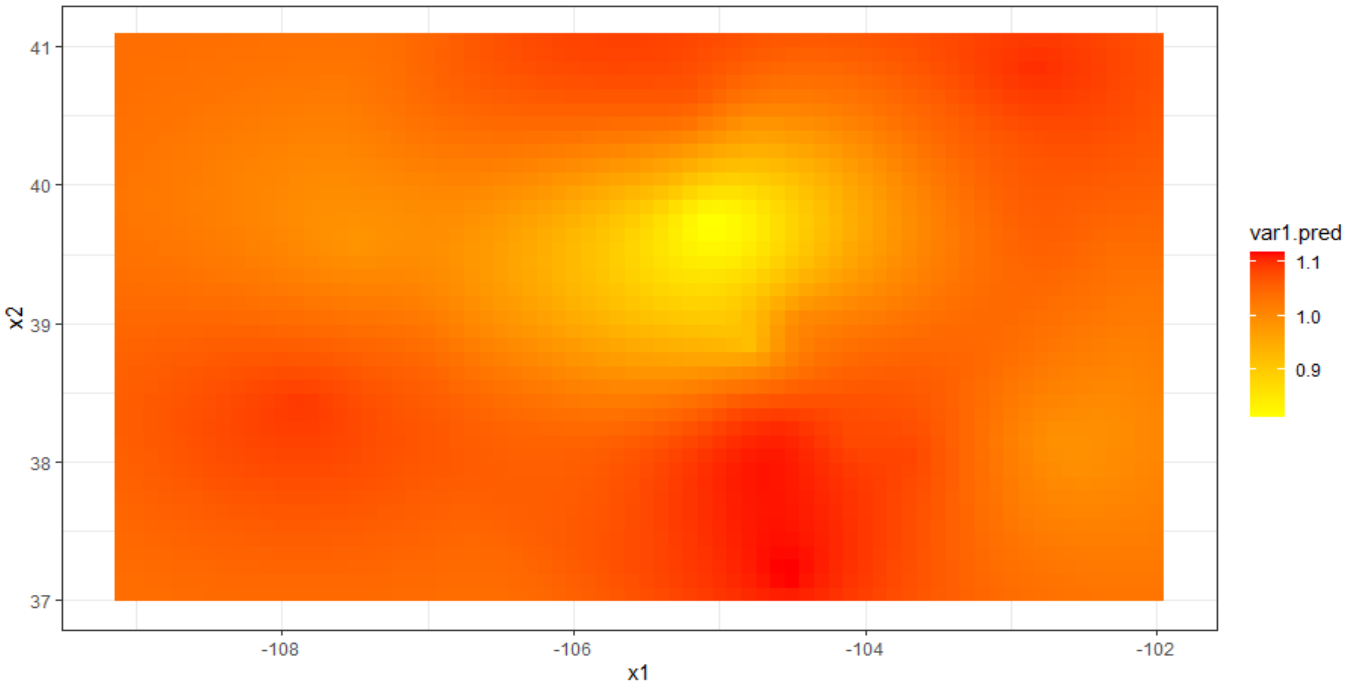






To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

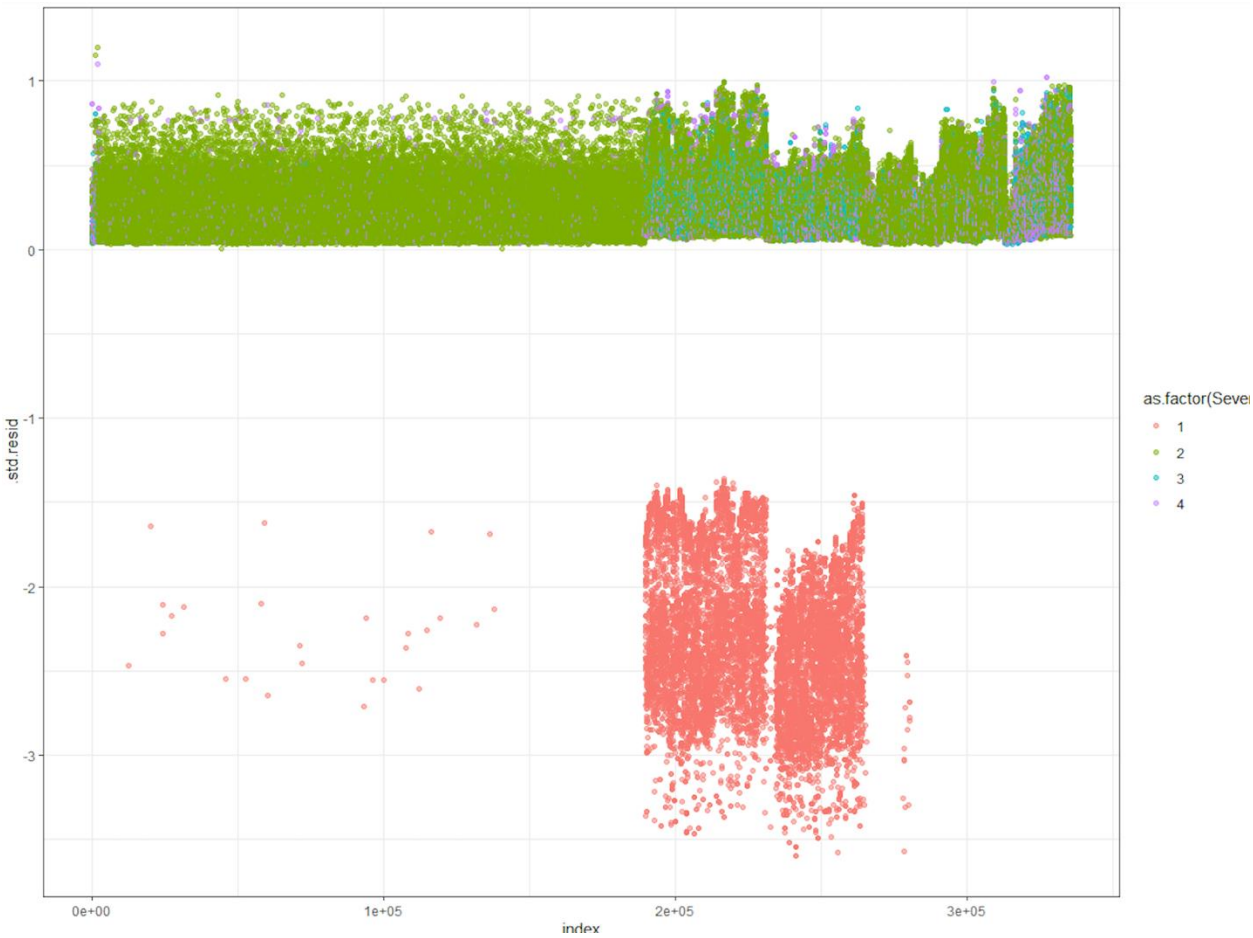
*The heat map below shows the frequency of accidents in Colorado. As the color intensifies, it represents occurrence of more accidents.




```
> model.data = augment(best_logistic_model) %>%
+   mutate(index = 1:n())
> model.data %>% top_n(3, .cooksd)
# A tibble: 3 x 17
  .rownames `as.factor(Severity)` Temperature.F. Civil_Twilight Nautical_Twilight Astronomical_Twilight Humidity... Pressure.in.
  <chr>     <fct>                          <dbl>          <int>             <int>                 <int>       <dbl>        <dbl>
1 145484    2                               41                0                 0                     1          53         30.4
2 167443    2                               28.4              1                 1                     1          55         30.4
3 184465    4                               42.8              1                 1                     1          87         30.4
#   with 9 more variables: Wind_Speed.mph. <dbl>, Precipitation.in. <dbl>, .fitted <dbl>, .resid <dbl>, .std.resid <dbl>,
```

To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

```
> probabilities = model_for_prediction %>% predict(test.data, type = "response")
> predicted.classes = ifelse(probabilities > 0.5, "1", "0")
> mean(predicted.classes==as.factor(test.data$Severity1))
[1] 0.8952764
> head(predicted.classes)
   53    70   617   660  4030  7737
  "0"   "0"   "0"   "0"   "0"   "0"
```



Regression Model:

Linear Regression

```
model=lm(Severity~Distance.mi.+Temperature.F.+ Humidity... +Pressure.in.
         +Visibility.mi.+Wind_Speed.mph.+ Precipitation.in.,data=df)
stepAIC(model,direction = "both",trace = FALSE)
summary(model)

install.packages("leaps")
library(leaps)
```

To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

```
regsubsets.out <-regsubsets(Severity~.,data =df,nbest = 1,      # 1 best model for each number of predictors
                    nvmax = NULL,    # NULL for no limit on number of variables
                    force.in = NULL, force.out = NULL,
                    method = "exhaustive")
summary_best_subset <- summary(regsubsets.out)
as.data.frame(summary_best_subset$outmat)
```

Logistic Regression

```
model = glm(as.factor(Severity) ~ Temperature.F.+ Sunrise_Sunset + Civil_Twilight  +
Nautical_Twilight + Astronomical_Twilight +
          Side_isRight + Humidity... + Pressure.in. + Visibility.mi.+ Wind_Speed.mph.+
Precipitation.in., data = df, family = binomial)

stepAIC(model,direction = "both",trace = FALSE)

#best model
best_logistic_model = glm(formula = as.factor(Severity) ~ Temperature.F. + Civil_Twilight +
                      Nautical_Twilight + Astronomical_Twilight + Humidity... +
                      Pressure.in. + Wind_Speed.mph. + Precipitation.in., family = binomial,
                    data = df)
summary(best_logistic_model)
plot(best_logistic_model)
car::vif(best_logistic_model)
anova(best_logistic_model,test = "Chisq")
library(car)
library(lmtest)
dwtest(best_logistic_model)

library(tidyverse)
library(broom)
theme_set(theme_classic())

probabilities = predict(best_logistic_model, type = "response")
predicted.classes = ifelse(probabilities > 0.5, "pos", "neg")
head(predicted.classes)
plot(best_logistic_model, which = 4, id.n = 3)

#
install.packages("broom")
library(broom)
library(ggplot2)
```

To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

```
model.data = augment(best_logistic_model) %>%
  mutate(index = 1:n())
model.data %>% top_n(3, .cooksd)
ggplot(model.data, aes(index, .std.resid)) +
  geom_point(aes(color = `as.factor(Severity)`), alpha = .5) +
  theme_bw()

car::vif(best_logistic_model)

df$Severity1   = as.integer(df$Severity >2)
model_isSevere = glm(as.factor(Severity1) ~ Temperature.F.+ Sunrise_Sunset + Civil_Twilight  +
Nautical_Twilight + Astronomical_Twilight +
                     Side_isRight + Humidity... + Pressure.in. + Visibility.mi.+ Wind_Speed.mph.+
Precipitation.in., data = df, family = binomial)

stepAIC(model_isSevere,direction = "both",trace = FALSE)

df2=df

library(tidyverse)
library(caret)
set.seed(12367)
training.samples = as.factor(df2$Severity1) %>%
  createDataPartition(p = 0.8, list = FALSE)
train.data  = df2[training.samples, ]
test.data = df2[-training.samples, ]

model_for_prediction = glm(formula = as.factor(Severity1) ~ Temperature.F. + Sunrise_Sunset +
                       Civil_Twilight + Astronomical_Twilight + Side_isRight + Humidity... +
                       Pressure.in. + Visibility.mi. + Wind_Speed.mph. + Precipitation.in.,
                     family = binomial, data = df2)
summary(model_for_prediction)

probabilities = model_for_prediction %>% predict(test.data, type = "response")
predicted.classes = ifelse(probabilities > 0.5, "1", "0")
head(predicted.classes)
# Model accuracy
mean(predicted.classes==as.factor(test.data$Severity1))

my_func <- function(df, group){
  df %>%
    group_by(!!group) %>%
    summarise(my_count = n()) %>%
    arrange(desc(my_count))
}
```

To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

```
my_group = quo(State)
highest_accident_state=my_func(df, my_group)

ggplot(data=highest_accident_state, aes(x=State, y=my_count)) +
  geom_bar(stat="identity", fill="steelblue", width = 1)+
  geom_text(aes(label=my_count), vjust=1.6, color="white", size=1.5)+
  theme_minimal()

install.packages("lubridate")
library(lubridate)
library(readr)
accidents_new = df %>%
  mutate(startHr=hour(Start_Time))
head(accidents_new)
accidents_count = accidents_new %>%
  count(startHr)
head(accidents_count)
accidents_count

ggplot(accidents_count, aes(startHr, n)) + geom_point() +geom_path()
```

The first model below is used to reach the best model by trying combination of various variables and stepAIC which shows the composition of best model.

```
> model = glm(as.factor(Severity) ~ Temperature.F.+ Sunrise_Sunset + Civil_Twilight  + Nautical_Twilight + Astrono
mical_Twilight +
+              Side_isRight + Humidity... + Pressure.in. + Visibility.mi.+ Wind_Speed.mph.+ Precipitation.in., da
ta = df, family = binomial)
> stepAIC(model,direction = "both",trace = FALSE)

Call:  glm(formula = as.factor(Severity) ~ Temperature.F. + Civil_Twilight +
    Nautical_Twilight + Astronomical_Twilight + Humidity... +
    Pressure.in. + Wind_Speed.mph. + Precipitation.in., family = binomial,
    data = df)

Coefficients:
          (Intercept)         Temperature.F.         Civil_Twilight      Nautical_Twilight
            1.158357              -0.032820              -0.188018              -0.298690
Astronomical_Twilight           Humidity...          Pressure.in.        Wind_Speed.mph.
           -0.993263               0.018689               0.152683               0.006239
    Precipitation.in.
           -0.401235

Degrees of Freedom: 335551 Total (i.e. Null);  335543 Residual
Null Deviance:      100100
Residual Deviance: 85670        AIC: 85690
>
```

To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

*Below output shows the best model which was identified using stepAIC function.We can observe the least AIC value in it.

```
> best_logistic_model = glm(formula = as.factor(Severity) ~ Temperature.F. + Civil_Twilight +
+                             Nautical_Twilight + Astronomical_Twilight + Humidity... +
+                             Pressure.in. + Wind_Speed.mph. + Precipitation.in., family = binomial,
+                           data = df)
> stepAIC(best_logistic_model,direction = "both",trace = FALSE)

Call:  glm(formula = as.factor(Severity) ~ Temperature.F. + Civil_Twilight +
    Nautical_Twilight + Astronomical_Twilight + Humidity... +
    Pressure.in. + Wind_Speed.mph. + Precipitation.in., family = binomial,
    data = df)

Coefficients:
          (Intercept)         Temperature.F.         Civil_Twilight      Nautical_Twilight
            1.158357              -0.032820              -0.188018              -0.298690
Astronomical_Twilight            Humidity...           Pressure.in.        Wind_Speed.mph.
           -0.993263               0.018689               0.152683               0.006239
    Precipitation.in.
           -0.401235

Degrees of Freedom: 335551 Total (i.e. Null);  335543 Residual
Null Deviance:      100100
Residual Deviance: 85670        AIC: 85690
```

The Durbin Watson (DW) statistic is a test for autocorrelation in the residuals from a statistical model or regression analysis. The Durbin-Watson statistic will always have a value ranging between 0 and 4. A value of 2.0 indicates there is no autocorrelation detected in the sample. Values from 0 to less than 2 point to positive autocorrelation and values from 2 to 4 means negative autocorrelation.

```
> dwtest(best_logistic_model)

        Durbin-Watson test

data:  best_logistic_model
DW = 1.1191, p-value < 2.2e-16
alternative hypothesis: true autocorrelation is greater than 0
```

To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

*Below output shows the details of our best model. We can observe all variables to be significant and least AIC value.

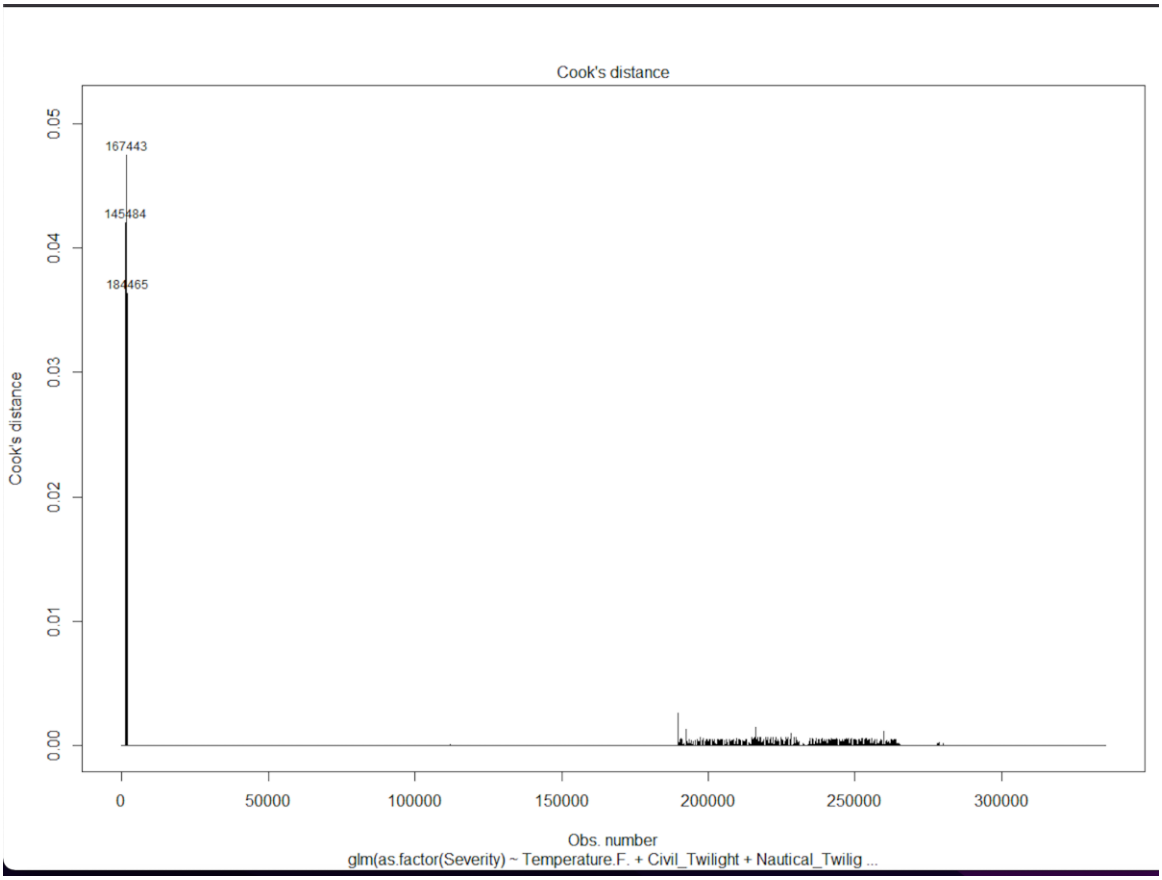
```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.5991   0.0983   0.1780   0.3018   1.0246

Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)            1.1583568  0.2064055   5.612    2e-08 ***
Temperature.F.        -0.0328198  0.0007383 -44.454  < 2e-16 ***
Civil_Twilight        -0.1880180  0.0583786  -3.221 0.001279 **
Nautical_Twilight     -0.2986898  0.0892893  -3.345 0.000822 ***
Astronomical_Twilight -0.9932627  0.0782458 -12.694  < 2e-16 ***
Humidity...            0.0186887  0.0005022  37.215  < 2e-16 ***
Pressure.in.           0.1526831  0.0073554  20.758  < 2e-16 ***
Wind_Speed.mph.        0.0062393  0.0018171   3.434 0.000595 ***
Precipitation.in.     -0.4012353  0.1155659  -3.472 0.000517 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 100077  on 335551  degrees of freedom
Residual deviance:  85670  on 335543  degrees of freedom
AIC: 85688

Number of Fisher Scoring iterations: 8

> AIC(best_logistic_model)
[1] 85687.65
```

*Anova test can identify significance of variables.

```
> anova(best_logistic_model,test = "Chisq")
Analysis of Deviance Table

Model: binomial, link: logit

Response: as.factor(Severity)

Terms added sequentially (first to last)


                     Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                                335551     100077
Temperature.F.        1   8814.5    335550      91262 < 2.2e-16 ***
Civil_Twilight        1   2234.1    335549      89028 < 2.2e-16 ***
Nautical_Twilight     1    305.9    335548      88722 < 2.2e-16 ***
Astronomical_Twilight 1    170.7    335547      88551 < 2.2e-16 ***
Humidity...           1   2466.9    335546      86085 < 2.2e-16 ***
Pressure.in.          1    396.2    335545      85688 < 2.2e-16 ***
Wind_Speed.mph.       1     11.7    335544      85677 0.0006264 ***
Precipitation.in.     1      7.0    335543      85670 0.0082885 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

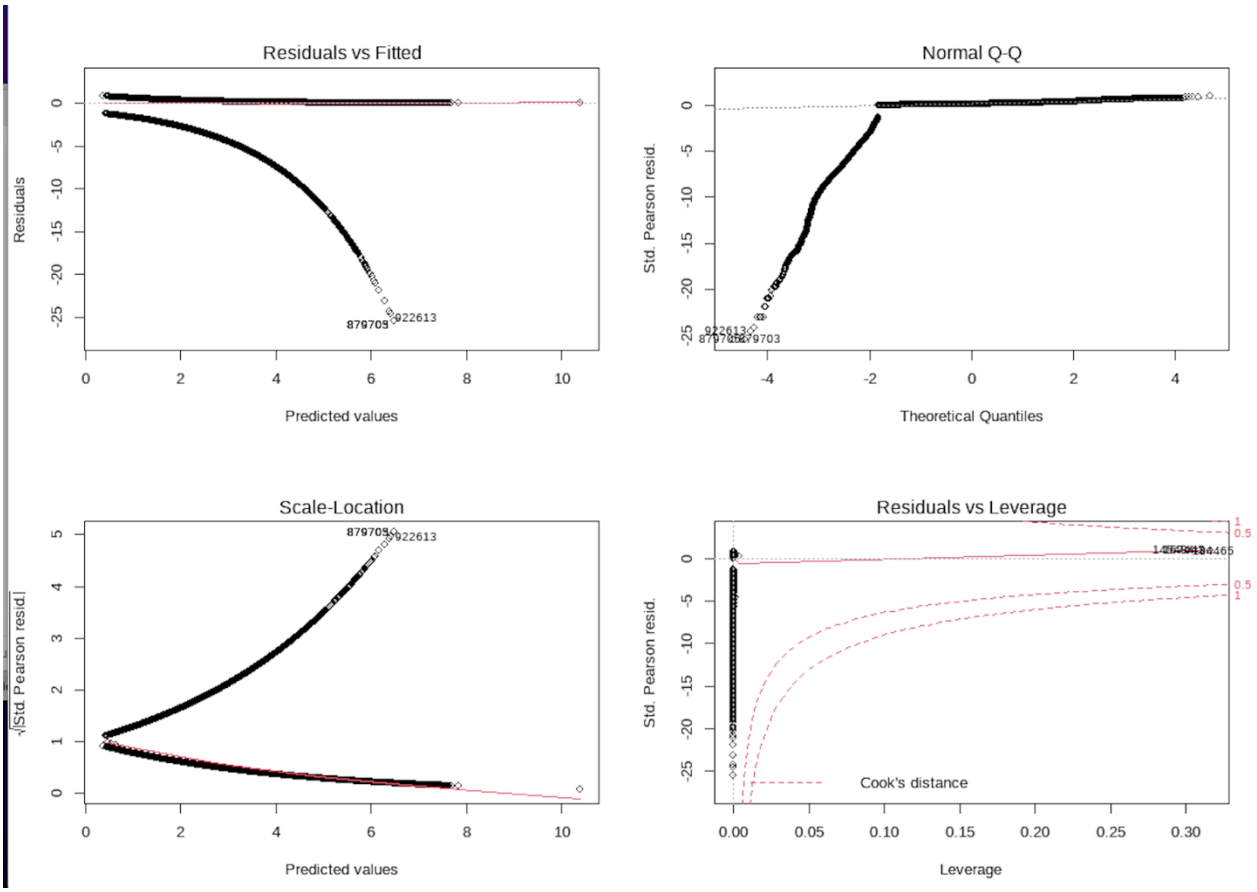
To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

*Below map shows the cooks distance which interprets most effective independent variables for the dependent variable.



To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

*Below plots show the assumption values for predicted and trained data.



*VIF function shows the multicollinearity.

```
> car::vif(best_logistic_model)
       Temperature.F.        Civil_Twilight     Nautical_Twilight Astronomical_Twilight          Humidity...
             1.332401              3.610137              6.472612              3.936925             1.491783
        Pressure.in.       Wind_Speed.mph.     Precipitation.in.
             1.206638              1.035459              1.002829
```

Count of Severity:

```
Count_severity = my_func(df, quo(Severity))
ggplot(data=Count_severity, aes(x=Severity, y=my_count)) +
  geom_bar(stat="identity", fill="steelblue", width = 1)+
  geom_text(aes(label=my_count), vjust=1.6, color="white", size=3.5)+
  theme_minimal()

bp= ggplot(Count_severity, aes(x="", y=my_count,fill=Severity))+
  geom_bar(width = 1, stat = "identity")
pie = bp + coord_polar("y", start=0)
pie
```

To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

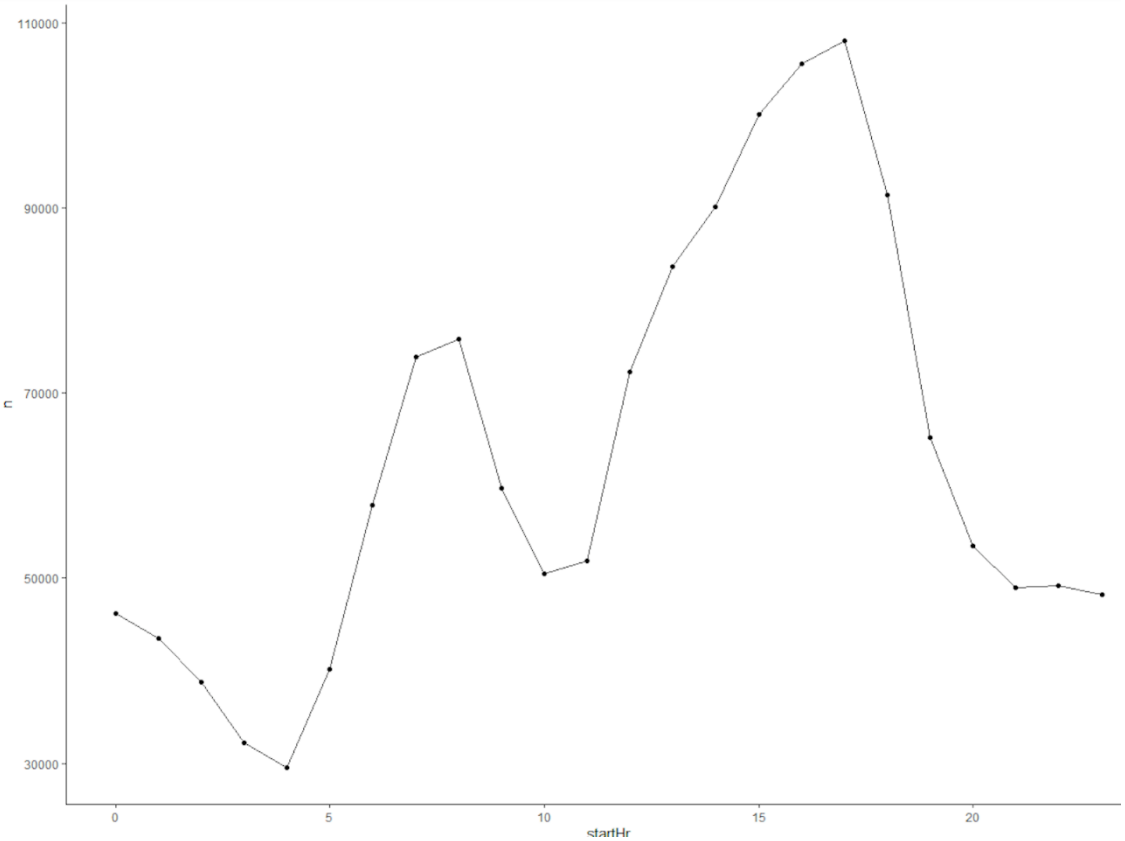
```
library(ggpubr)
ggboxplot(df, x = "Severity", y = "Wind_Chill.F.", width = 0.8)

Count_severity_city = my_func(df, quo(City))
Count_severity_city = head(Count_severity_city, 20)
ggplot(data=Count_severity_city, aes(x=City, y=my_count)) +
  geom_bar(stat="identity", fill="steelblue", width = 1)+
  geom_text(aes(label=my_count), vjust=2.6, color="white", size=2.5)+
  theme_minimal()
```

To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

Findings:

Number of accidents each hour:
Before Data cleaning:

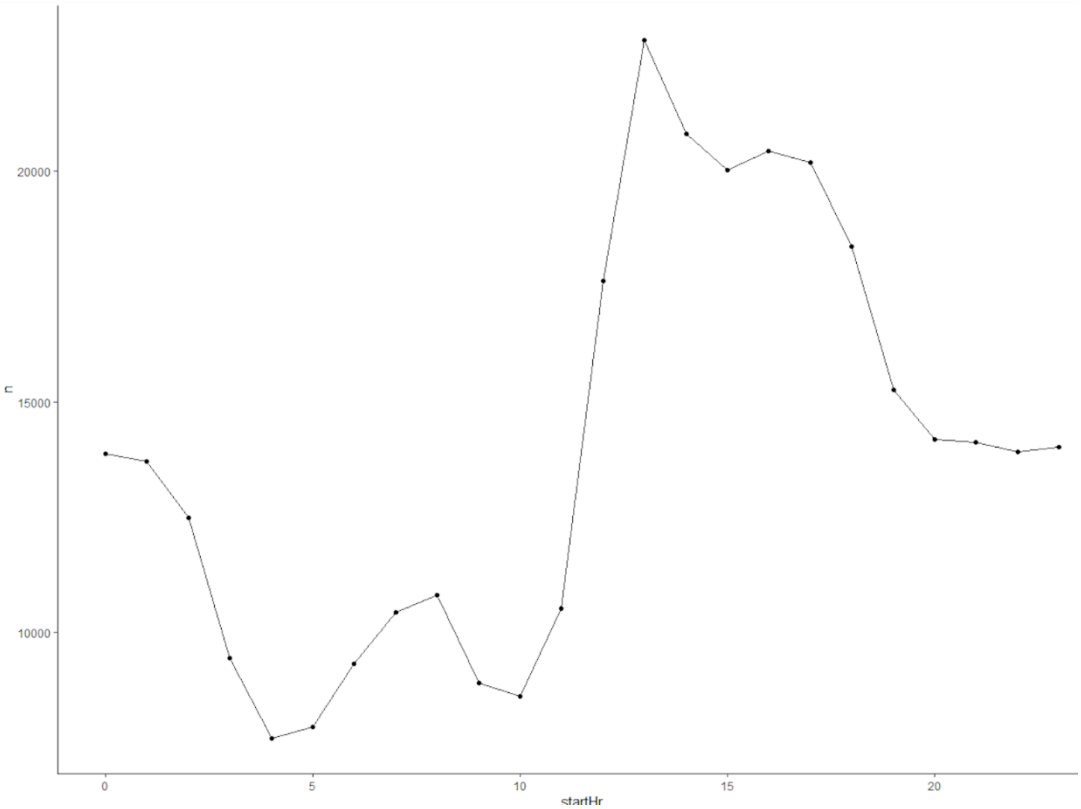
	startHr	n
1	0	46125
2	1	43434
3	2	38801
4	3	32175
5	4	29478
6	5	40105
7	6	57859
8	7	73938
9	8	75802
10	9	59744
11	10	50442
12	11	51884
13	12	72283
14	13	83700
15	14	90162
16	15	100074
17	16	105559
18	17	108011
19	18	91413
20	19	65190
21	20	53515
22	21	48988
23	22	49156
24	23	48226



To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

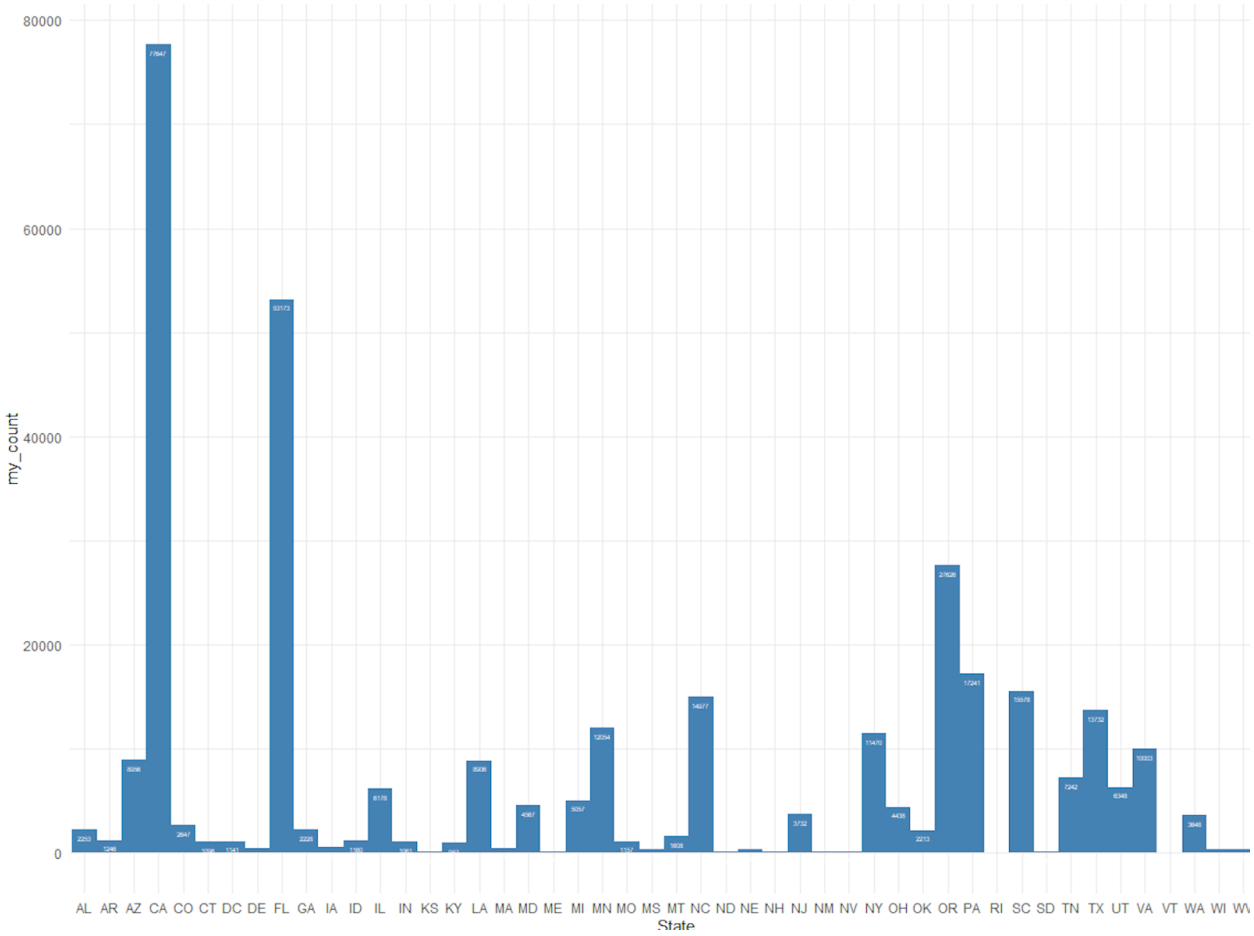
After Cleaning Data:

```
> accidents_count
   startHr     n
1        0 13873
2        1 13706
3        2 12478
4        3  9447
5        4  7690
6        5  7952
7        6  9307
8        7 10429
9        8 10800
10       9  8909
11      10  8609
12      11 10514
13      12 17628
14      13 22840
15      14 20825
16      15 20024
17      16 20447
18      17 20189
19      18 18365
20      19 15255
21      20 14195
22      21 14117
23      22 13924
24      23 14029
>
```



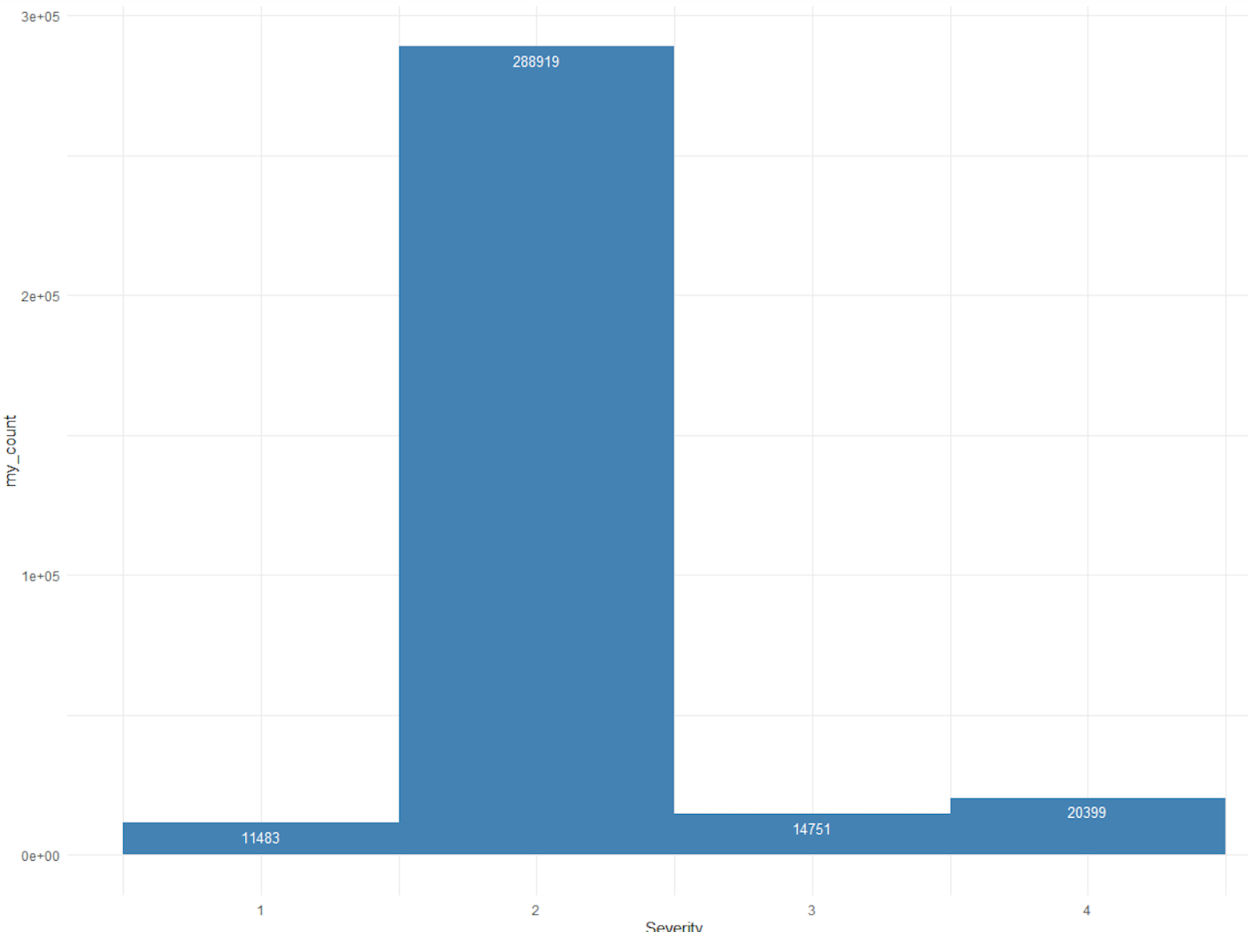
To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

Accidents according to state:



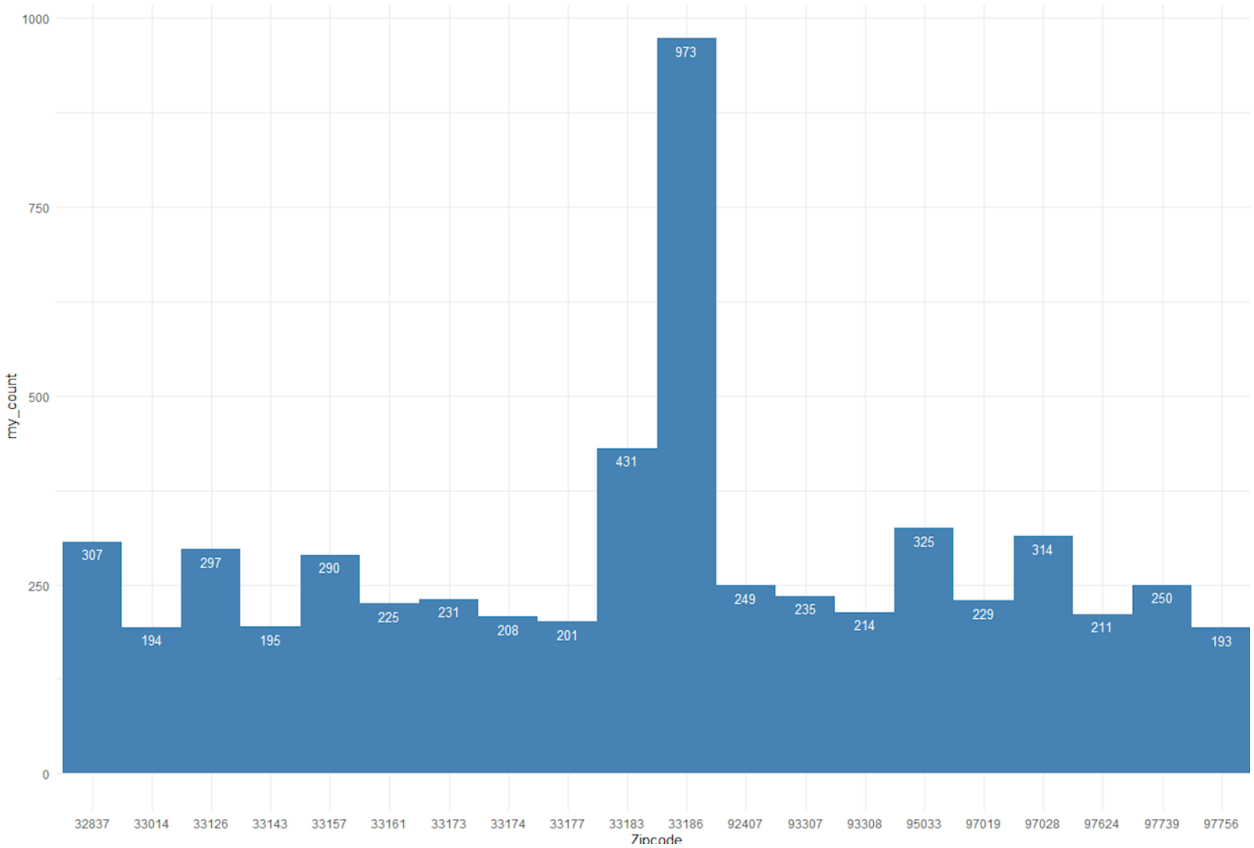
To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

Severity:



To: US Traffic Department & Accident Response Teams.
From: Aditya K Nagori
Subject: US Accident Analysis (2016-2020)

County:



Streets: