MGS 616: Predictive Analytics EDA Report

School of Management
University at Buffalo, The State University of New York
Sreyashi Samanta - ssamanta@buffalo.edu

Title: Prediction and Analysis of Accident Severity

1. **Data Source Details**
   Dataset link: https://www.kaggle.com/datasets/sobhanmoosavi/us-accidents
   Dataset Name: US Accidents

2. **Description of the dataset**: This dataset on vehicle accidents in the USA includes data from all 49 states. Many APIs that offer streaming traffic incident (or event) data were used to gather the accident data. The US and state departments of transportation, law enforcement organizations, traffic cameras, and traffic sensors embedded in the road networks are only a few of the organizations whose traffic data is broadcast through these APIs. This dataset currently contains roughly 2.8 million accident records. For additional information on this dataset, go here.

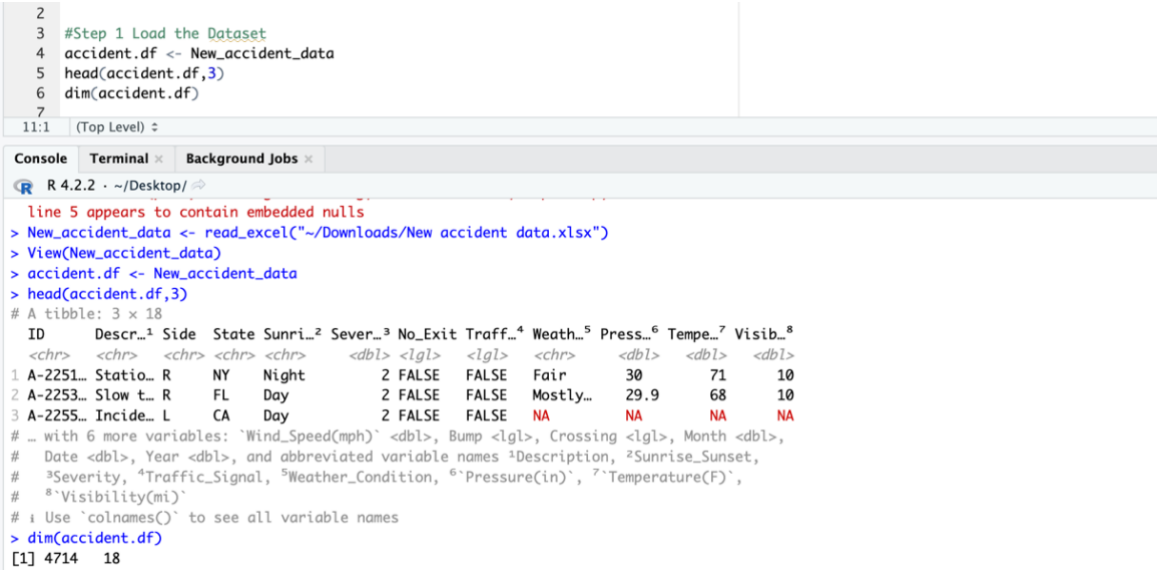
3. Identifying the response and the predictor variables:

   Response variable: Severity
   Predictor variables: Sunrise_Sunset, State, Traffic_Signal, Weather_Condition, Pressure, Temperature, Visibility, Wind_Speed, Bump, Crossing

4. **EDA**: Exploratory data analysis is the crucial process of doing rudimentary analyses on data to find patterns, identify anomalies, test hypotheses, and double-check assumptions with the use of summary statistics and visualization.

5. Load the dataset:

   Loaded the Excel file with 18 columns and 4714 rows.



```
#Step 1 Load the Dataset
accident.df <- New_accident_data
head(accident.df,3)
dim(accident.df)
```

```
Console   Terminal   Background Jobs
R 4.2.2 · ~/Desktop/
  line 5 appears to contain embedded nulls
> New_accident_data <- read_excel("~/Downloads/New accident data.xlsx")
> View(New_accident_data)
> accident.df <- New_accident_data
> head(accident.df,3)
# A tibble: 3 × 18
  ID      Descr…¹ Side  State Sunri…² Sever…³ No_Exit Traff…⁴ Weath…⁵ Press…⁶ Tempe…⁷ Visib…⁸
  <chr>   <chr>   <chr> <chr> <chr>      <dbl> <lgl>   <lgl>   <chr>     <dbl>   <dbl>   <dbl>
1 A-2251… Statio… R     NY    Night          2 FALSE   FALSE   Fair         30      71      10
2 A-2253… Slow t… R     FL    Day            2 FALSE   FALSE   Mostly…      29.9    68      10
3 A-2255… Incide… L     CA    Day            2 FALSE   FALSE   NA           NA      NA      NA
# … with 6 more variables: `Wind_Speed(mph)` <dbl>, Bump <lgl>, Crossing <lgl>, Month <dbl>,
#   Date <dbl>, Year <dbl>, and abbreviated variable names ¹Description, ²Sunrise_Sunset,
#   ³Severity, ⁴Traffic_Signal, ⁵Weather_Condition, ⁶`Pressure(in)`, ⁷`Temperature(F)`,
#   ⁸`Visibility(mi)`
# ℹ Use `colnames()` to see all variable names
> dim(accident.df)
[1] 4714   18
```

6. EDA Step 1: Checking the statistics of the columns using the summary function in R.
   For each of the **numeric** variables we can see the following information:
   Min: The minimum value.
   1st Qu: The value of the first quartile (25th percentile).
   Median: The median value.
   Mean: The mean value.
   3rd Qu: The value of the third quartile (75th percentile).
   Max: The maximum value.
   For the **categorical** variables in the dataset we see a frequency count of each value.

```
11  #Step 2 Summary of the Dataset
12  summary(accident.df)
13
8:1  (Top Level) ÷
```

Console | Terminal | Background Jobs

R 4.2.2 · ~/Desktop/

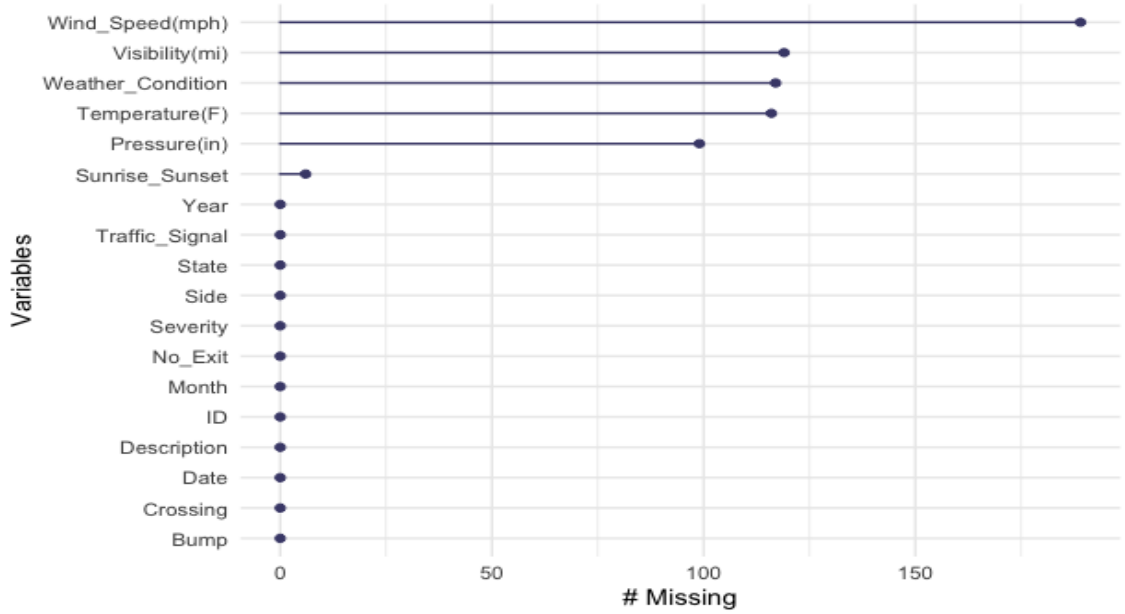
```
> #Step 2 Summary of the Dataset
> summary(accident.df)
      ID              Description            Side              State
 Length:4714        Length:4714        Length:4714        Length:4714
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character



 Sunrise_Sunset        Severity       No_Exit         Traffic_Signal  Weather_Condition
 Length:4714        Min.   :2.000   Mode :logical   Mode :logical   Length:4714
 Class :character   1st Qu.:2.000   FALSE:4710      FALSE:4374      Class :character
 Mode  :character   Median :2.000   TRUE :4         TRUE :340       Mode  :character
                    Mean   :2.096
                    3rd Qu.:2.000
                    Max.   :4.000

  Pressure(in)   Temperature(F)   Visibility(mi)   Wind_Speed(mph)     Bump
 Min.   :20.60   Min.   :-25.10   Min.   : 0.000   Min.   : 0.000   Mode :logical
 1st Qu.:29.21   1st Qu.: 53.00   1st Qu.:10.000   1st Qu.: 3.000   FALSE:4714
 Median :29.75   Median : 67.00   Median :10.000   Median : 7.000
 Mean   :29.37   Mean   : 64.43   Mean   : 9.209   Mean   : 7.118
 3rd Qu.:29.97   3rd Qu.: 78.00   3rd Qu.:10.000   3rd Qu.:10.000
 Max.   :30.68   Max.   :110.00   Max.   :50.000   Max.   :40.000
 NA's   :99      NA's   :116      NA's   :119      NA's   :189
  Crossing           Month             Date             Year
 Mode :logical   Min.   : 1.000   Min.   : 1.00   Min.   :2019
 FALSE:4412      1st Qu.: 6.000   1st Qu.: 8.00   1st Qu.:2019
 TRUE :302       Median : 9.000   Median :16.00   Median :2021
                 Mean   : 8.326   Mean   :16.02   Mean   :2020
                 3rd Qu.:11.000   3rd Qu.:24.00   3rd Qu.:2021
                 Max.   :12.000   Max.   :31.00   Max.   :2021
```

7. EDA Step 2: Checking the quality of data and finding the number of null and blank values

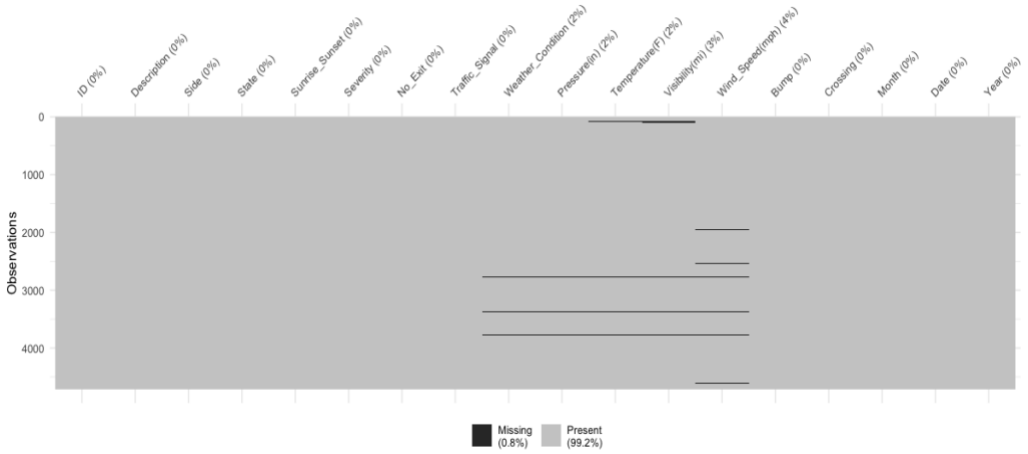
   1. Identifying the columns with missing values

```
> #Step 3 Identifying the columns with missing values
> colSums(is.na(accident.df)==TRUE|accident.df=='')
              ID      Description             Side            State    Sunrise_Sunset
               0                0                0                0                 6
        Severity          No_Exit    Traffic_Signal Weather_Condition      Pressure(in)
               0                0                0              117                99
   Temperature(F)    Visibility(mi)   Wind_Speed(mph)             Bump         Crossing
             116               119              189                0                0
           Month             Date             Year
               0                0                0
```

2. Plot the number of missing values in each column before cleaning the data



3. Percentage of missing values

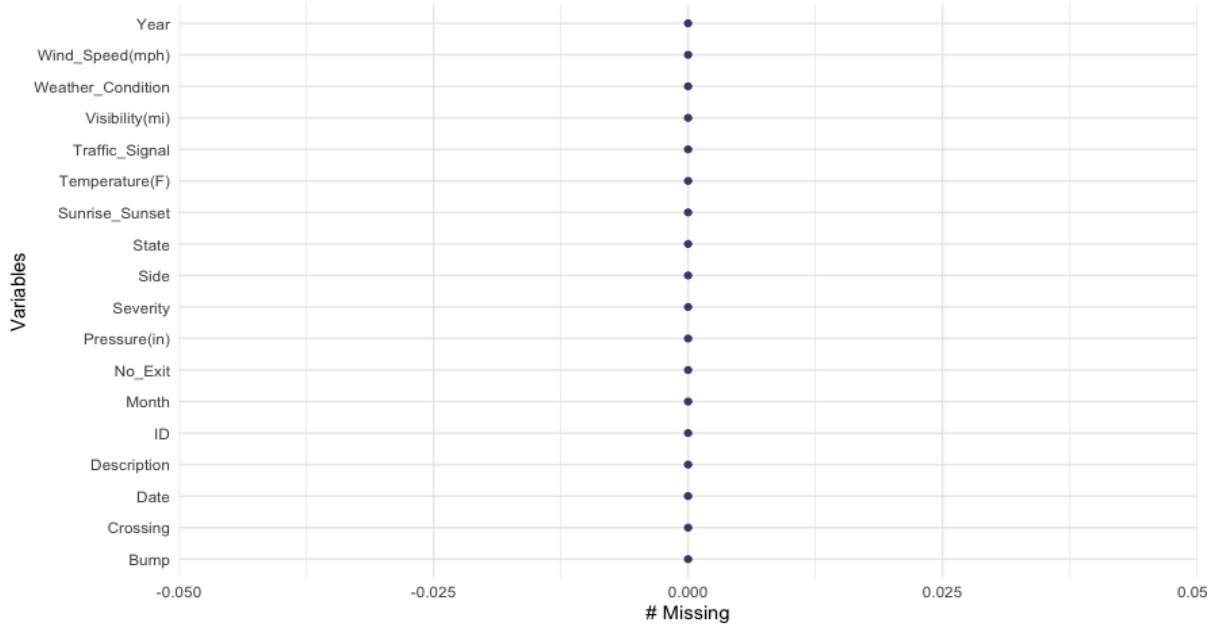


8. Data Cleaning and Preprocessing: Impute Missing Data by taking the either the mean or the mode of the numerical data and "Unknown" for the categorical data

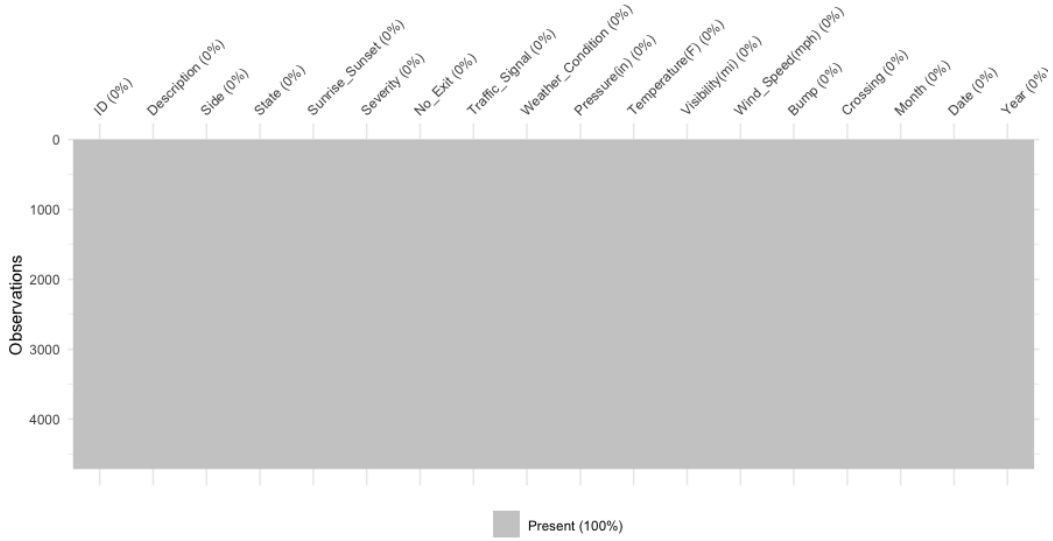
   Code in R for cleaning the data:

```r
#Step 6 Impute Missing Data by taking the either the mean or the mode of the numercial data and "Unknown" for the categorical data
#Replaced Weather_Condition Blank with Unknown
accident.df$Weather_Condition[is.na(accident.df$Weather_Condition)==TRUE]= "Unknown"

#Replaced Sunrise_Sunset Blank with Unknown
accident.df$Sunrise_Sunset[is.na(accident.df$Sunrise_Sunset)==TRUE]= "Unknown"

#Replaced Pressure Blank with the mean value
accident.df$`Pressure(in)`[is.na(accident.df$`Pressure(in)`)==TRUE] = round(mean(accident.df$`Pressure(in)`, na.rm = TRUE))

#Replaced Temperature Blank with the mean value
accident.df$`Temperature(F)`[is.na(accident.df$`Temperature(F)`)==TRUE] = round(mean(accident.df$`Temperature(F)` , na.rm = TRUE))

#Replaced Visibility Blank with the mode value
getmode <- function(a) {
  uniqage <- unique(a)
  uniqage[which.max(tabulate(match(a, uniqage)))]
}

accident.df$`Visibility(mi)`[is.na(accident.df$`Visibility(mi)`)==TRUE]= getmode(accident.df$`Visibility(mi)`)
print( getmode(accident.df$`Visibility(mi)`))

#Replaced Wind_Speed Blank with the mean value
accident.df$`Wind_Speed(mph)`[is.na(accident.df$`Wind_Speed(mph)`)==TRUE]= getmode(accident.df$`Wind_Speed(mph)`)
print( getmode(accident.df$`Wind_Speed(mph)`))
```

Plot post cleaning:

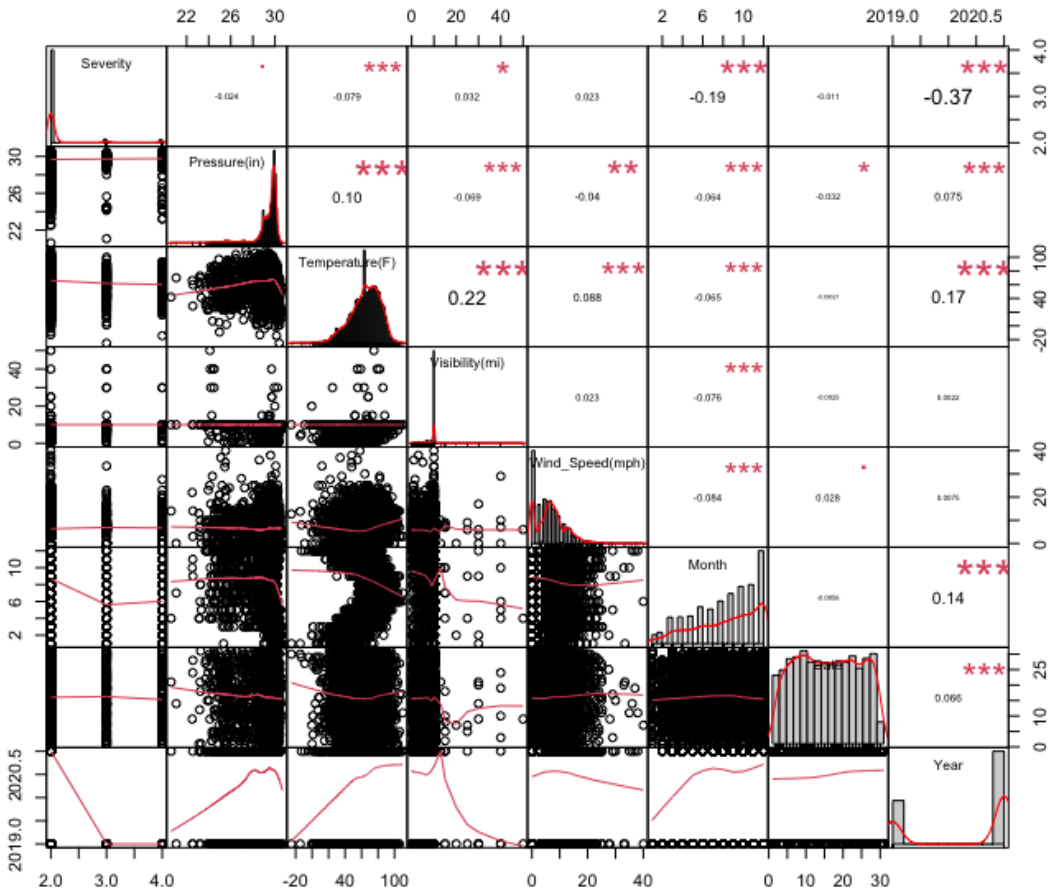


Percentage of missing values post cleaning:



9. EDA Step 3: Checking correlation between numerical columns using a Correlation Matrix

```
> res <- cor(my_data)
> round(res, 2)
                Severity Pressure(in) Temperature(F) Visibility(mi) Wind_Speed(mph) Month
Severity            1.00        -0.02          -0.08           0.03            0.02 -0.19
Pressure(in)       -0.02         1.00           0.10          -0.07           -0.04 -0.06
Temperature(F)     -0.08         0.10           1.00           0.22            0.09 -0.07
Visibility(mi)      0.03        -0.07           0.22           1.00            0.02 -0.08
Wind_Speed(mph)     0.02        -0.04           0.09           0.02            1.00 -0.08
Month              -0.19        -0.06          -0.07          -0.08           -0.08  1.00
Date               -0.01        -0.03           0.00           0.00            0.03 -0.01
Year               -0.37         0.07           0.17           0.00            0.01  0.14
                 Date  Year
Severity        -0.01 -0.37
Pressure(in)    -0.03  0.07
Temperature(F)   0.00  0.17
Visibility(mi)   0.00  0.00
Wind_Speed(mph)  0.03  0.01
Month           -0.01  0.14
Date             1.00  0.07
Year             0.07  1.00
```



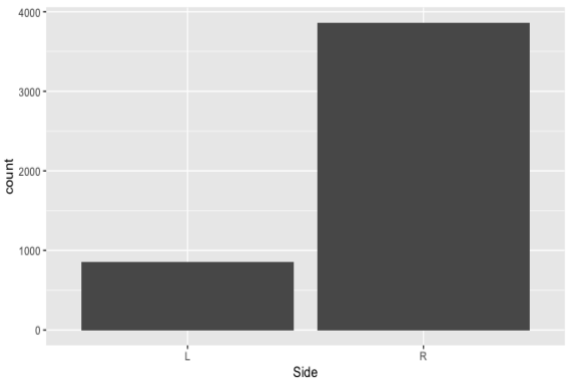
Observations from the correlation matrix:

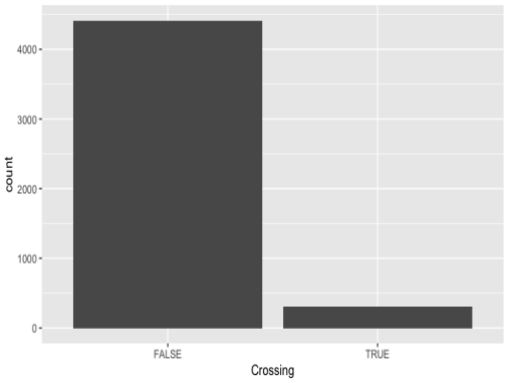
- The correlation between "Severity" and "Year" is -0.37, which suggests a moderate negative correlation between the two variables. This implies that accidents may have become less severe over time.
- There is a weak negative correlation between "Severity" and "Month", which suggests that accidents may be slightly less severe in certain months of the year.
- There is a weak positive correlation between "Temperature(F)" and "Visibility(mi)", which suggests that higher temperatures may be associated with better visibility.
- There is a weak negative correlation between "Month" and "Wind_Speed(mph)", which suggests that wind speeds may be slightly lower in certain months of the year.

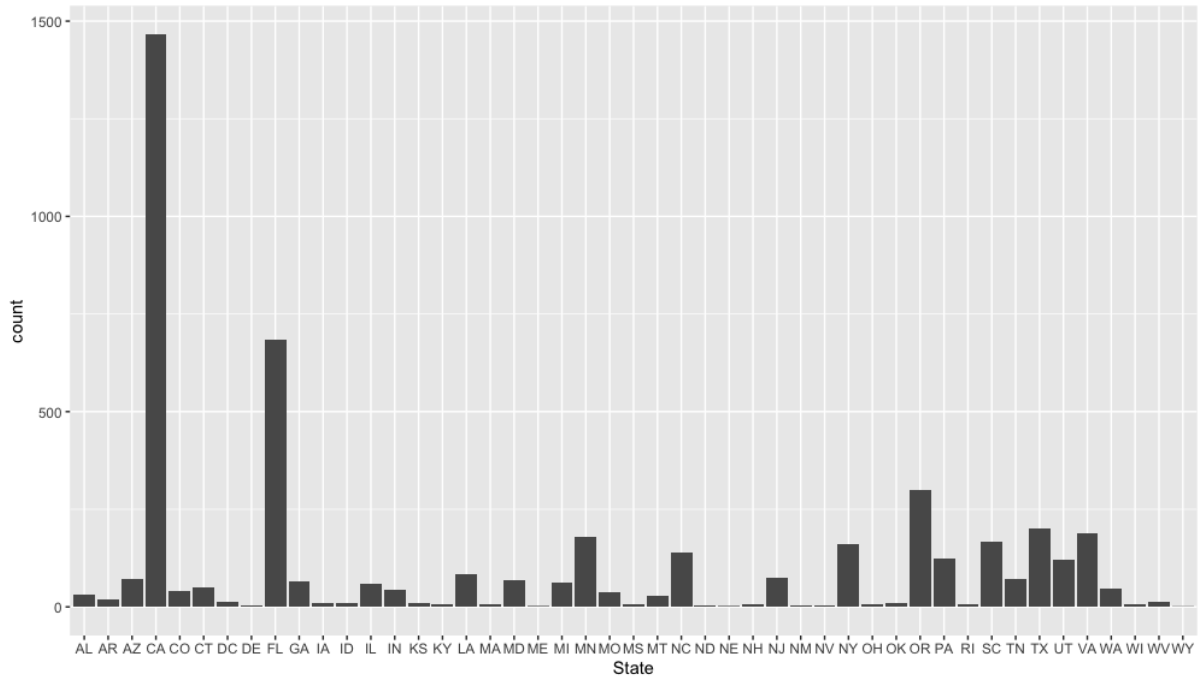
10. EDA Step 4: Histogram to look at the frequency

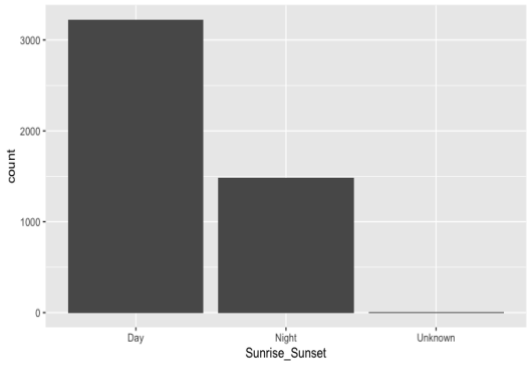
Code for creating histograms for the categorical columns:

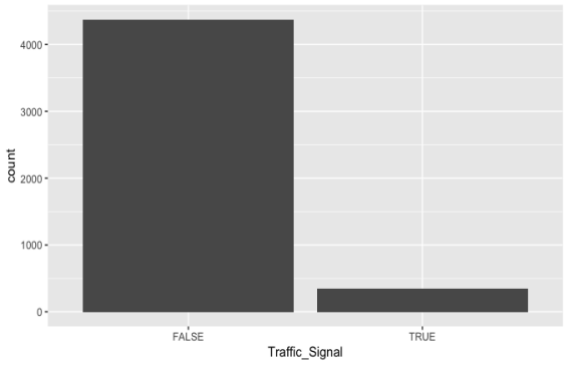
```r
categorical <- accident.df[, c(3,4,5,7,8,9,14,15)]

library(ggplot2)

ggplot(data.frame(accident.df), aes(x=Side)) + geom_bar()
ggplot(data.frame(accident.df), aes(x=State)) + geom_bar()
ggplot(data.frame(accident.df), aes(x=Sunrise_Sunset)) + geom_bar()
ggplot(data.frame(accident.df), aes(x=Traffic_Signal)) + geom_bar()

unique(accident.df$Weather_Condition)

library(dplyr)
library(ggplot2)

accident.df %>%
  count(Weather_Condition, sort = TRUE) %>%
  slice_head(n = 10) %>%
  ggplot(aes(x = Weather_Condition, y = n, fill = Weather_Condition)) +
  geom_col() +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 10)) +
  labs(x = "", y = "Count", fill = "") +
  theme(legend.position = "none")

ggplot(data.frame(accident.df), aes(x=Bump)) + geom_bar()
ggplot(data.frame(accident.df), aes(x=Crossing)) + geom_bar()
```

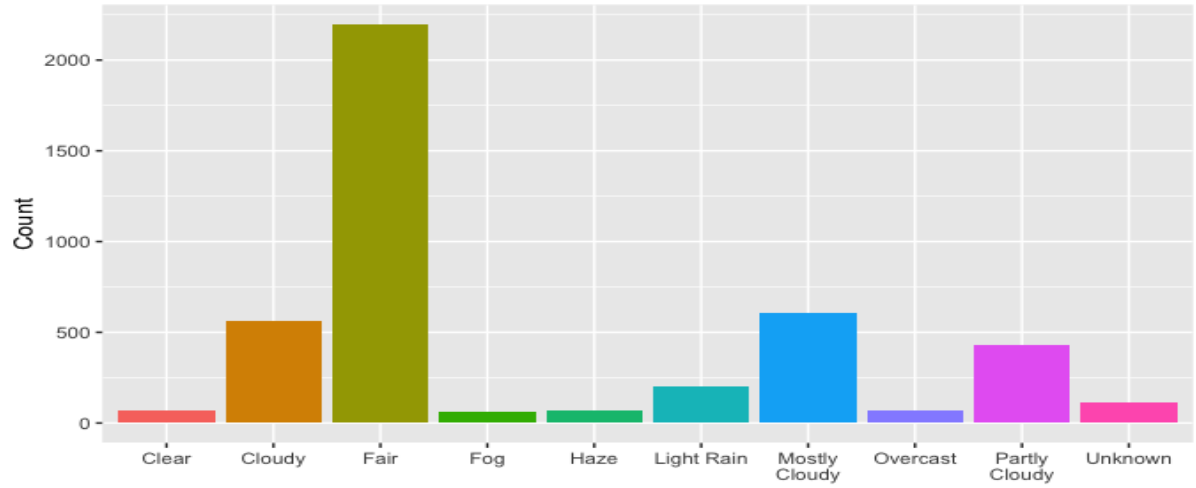












11. EDA Step 5: Exploring and Analyzing the relationship between variables with substantial correlation

    1. **Scatter Plot between Temperature and Visibility with color-coded data points based on severity level**:

Code:

```
#Scatter Plot to explore the relationship between Temperature and Visibility

library(ggplot2)

# create a subset of data with only relevant columns
sub_data <- accident.df[, c("Temperature(F)", "Visibility(mi)", "Severity")]

# plot the data with color-coded severity
ggplot(sub_data, aes(x = `Temperature(F)`, y = `Visibility(mi)`, color = Severity)) +
  geom_point() +
  xlab("Temperature (F)") +
  ylab("Visibility (mi)") +
  scale_color_gradient(low = "green", high = "red") +
  theme_classic()
```

Visualization:

Inference:

The plot suggests that there is a weak positive correlation between temperature and visibility for all severity levels. That is, as temperature increases, visibility tends to

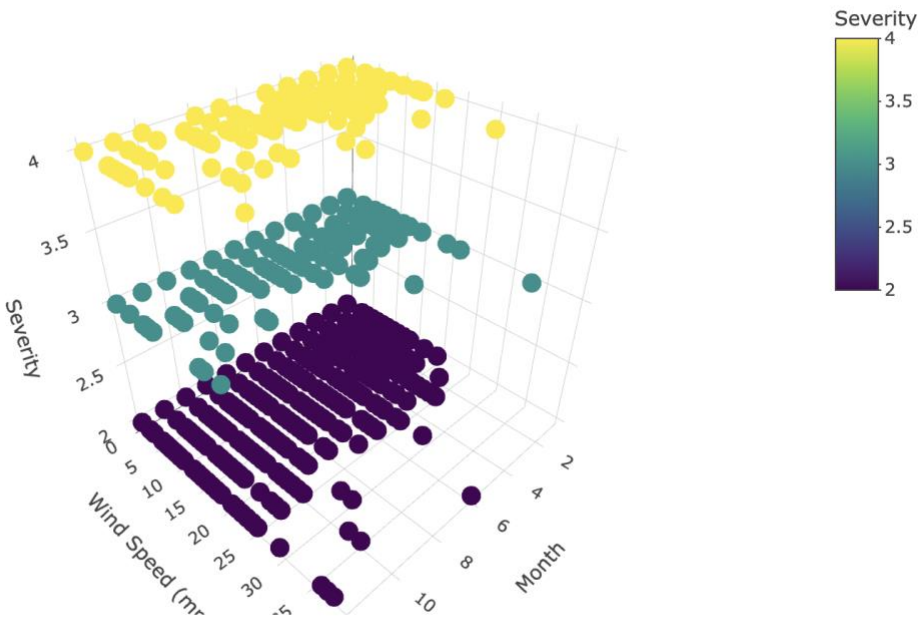
decrease. Additionally, the plot suggests that higher severity accidents tend to occur more frequently in the lower temperature and lower visibility ranges.

2. **3D Scatter plot between Month and Wind_Speed with Severity as the colour-coded metric**

   Code:

```
#3D Scatter Plot

library(plotly)
install.packages()

#Subset of the data with only the relevant variables
sub_data <- subset(accident.df, select = c("Month", "Wind_Speed(mph)", "Severity"))

#3D scatter plot
plot_ly(sub_data, x = ~Month, y = ~`Wind_Speed(mph)`, z = ~Severity,
        type = "scatter3d", mode = "markers", color = ~Severity) %>%
  layout(scene = list(xaxis = list(title = "Month"),
                      yaxis = list(title = "Wind Speed (mph)"),
                      zaxis = list(title = "Severity")))
```

   Visualization:



   Inference:

   Throughout the months of October, November, December, and January, accidents were most common.
   For all categories of wind speed, the severity of accidents is more commonly in the range of 2 to 4.

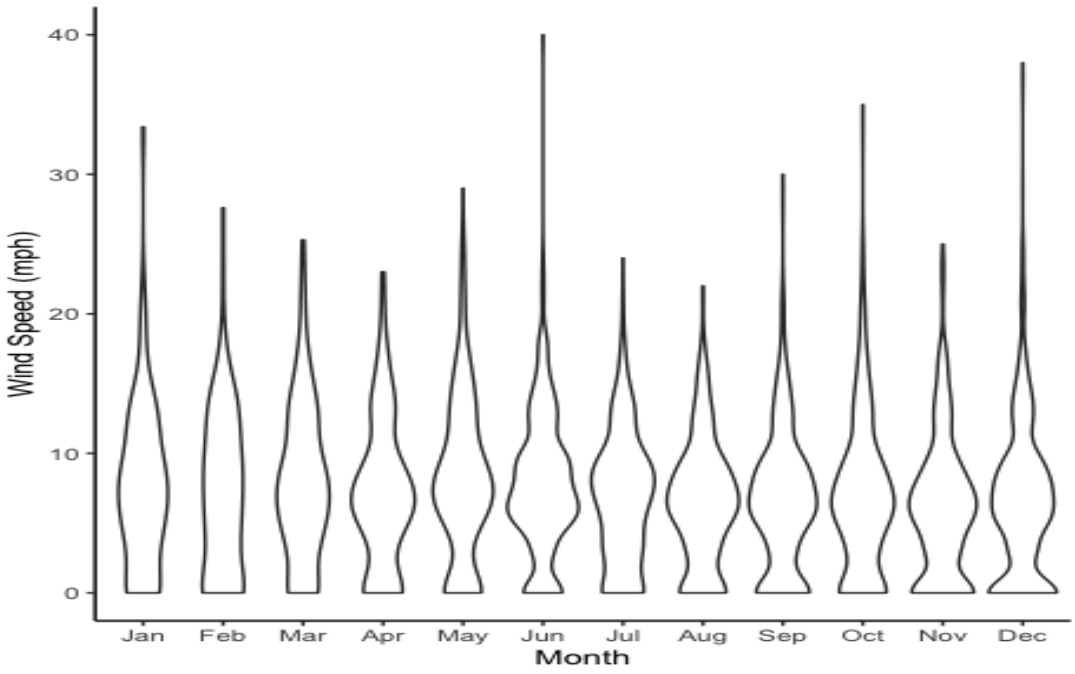
There are incidents in all wind speed categories for all severity levels, suggesting that wind speed doesn't significantly affect how serious accidents are.
The range of wind speeds from 0 to 20 mph sees a disproportionately larger number of accidents, but the range from 20 to 40 mph sees a decrease in the number of accidents.

3. **Violin Plot for the relationship exploration between Month and Wind_Speed**

Code:

```
library(ggplot2)

# Convert Month to a factor variable
sub_data$Month <- factor(sub_data$Month, levels = 1:12, labels = month.abb)


# Create a violin plot
ggplot(sub_data, aes(x = Month, y = `Wind_Speed(mph)`)) +
  geom_violin() +
  xlab("Month") +
  ylab("Wind Speed (mph)") +
  theme_classic()
```

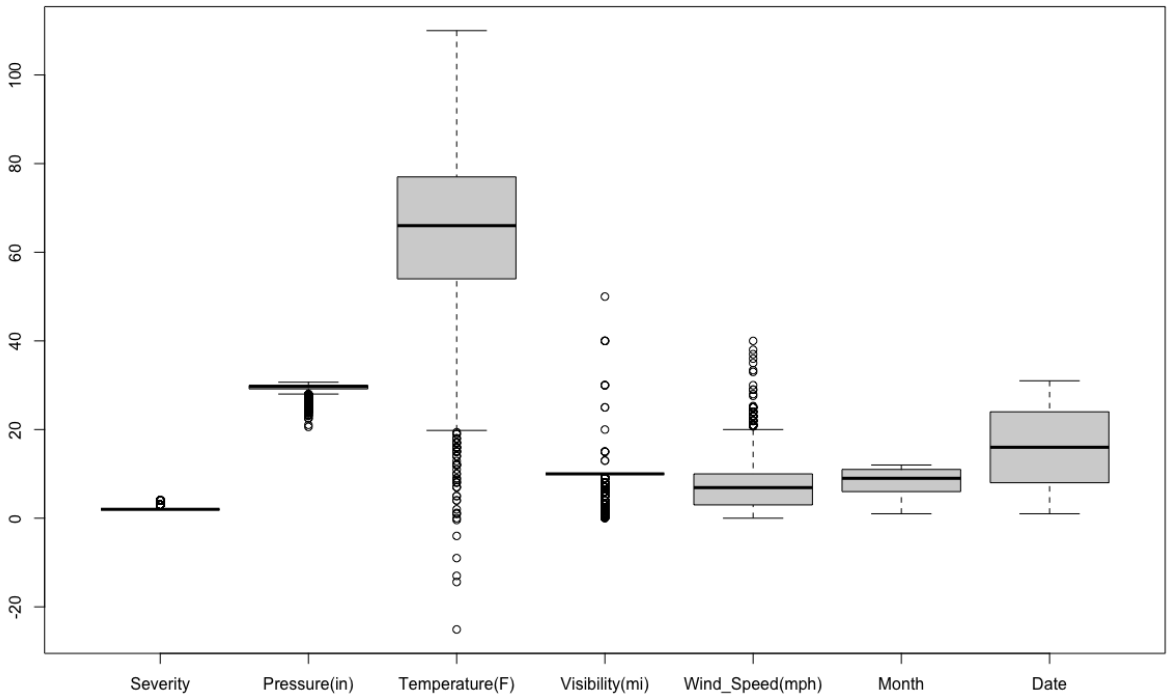
Visualization:



Inference:

The violin plot demonstrates that, on average, the winter months (December to February) have greater median wind speeds (mph) than the summer months (June - August). As comparison to less severe incidents, more severe accidents tend to occur under a larger range of wind speeds, as seen by the greater spread of the distribution for higher Severity levels.

12. EDA Step 6: Detecting Outliers in numerical data by using Box plot visualization



REFERENCES

1. https://towardsdatascience.com/exploratory-data-analysis-in-r-for-beginners-fe031add7072
2. https://rpubs.com/hossein_glm/traffic_accident_eda
3. https://www.dataquest.io/blog/statistical-learning-for-predictive-modeling-r/
4. https://www.statology.org/exploratory-data-analysis-in-r/
5. https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15
6. https://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-intro.html