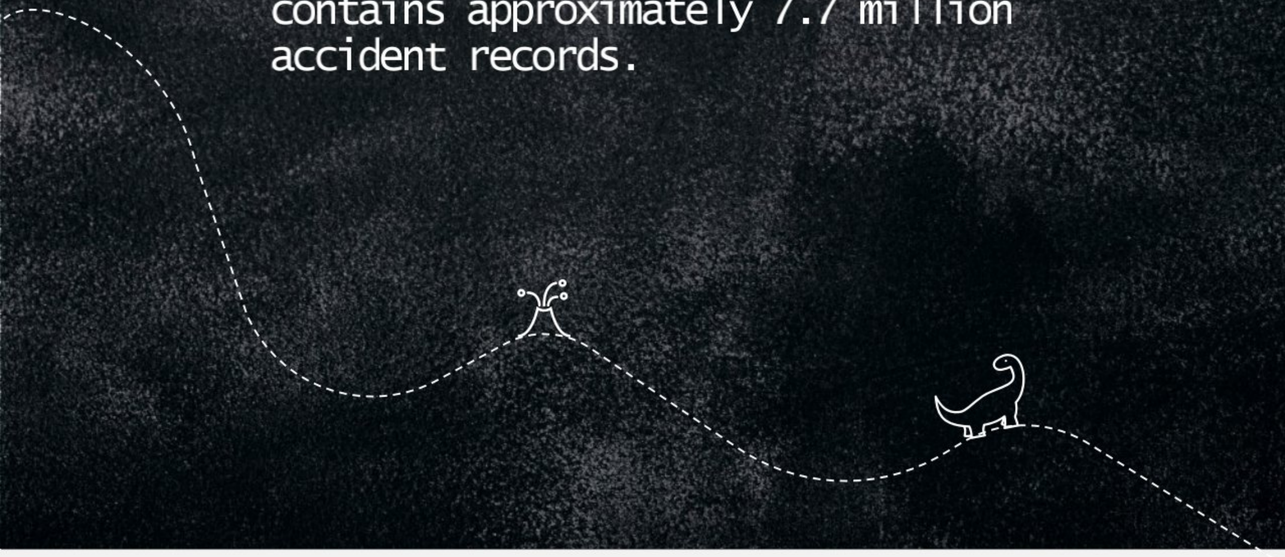
" ENHANCING TRAFFIC ACCIDENT CLASSIFICATIN USING MACHINE LEARNING"

Name : Ahmed Mohamed Salem
Email : ahmedsalem6686@gmail.com

Introduction:

This project is based on a countrywide car accident dataset covering 49 states in the USA. The accident data was collected from February 2016 to March 2023, using multiple APIs that provide streaming traffic incident data. These APIs broadcast traffic data captured by various entities, including federal and state departments of transportation, law enforcement agencies, traffic cameras, and traffic sensors within the road networks. The dataset currently contains approximately 7.7 million accident records.



About Data :

	ID	Source	Severity	Start_Time	End_Time	Start_Lat	Start_Lng	End_Lat	End_Lng	Distance(mi)	...	Roundabout	Station	Stop	Traffic_Calming	Traffic_Signal	Turning_Loop	Sunrise_Sunset	Civil_Twilight	Nautical_Tw
0	A-1	Source2	3	2016-02-08 05:46:00	2016-02-08 11:00:00	39.865147	-84.058723	NaN	NaN	0.010	...	False	False	False	False	False	False	Night	Night	
1	A-2	Source2	2	2016-02-08 06:07:59	2016-02-08 06:37:59	39.928059	-82.831184	NaN	NaN	0.010	...	False	False	False	False	False	False	Night	Night	
2	A-3	Source2	2	2016-02-08 06:49:27	2016-02-08 07:19:27	39.063148	-84.032608	NaN	NaN	0.010	...	False	False	False	False	True	False	Night	Night	
3	A-4	Source2	3	2016-02-08 07:23:34	2016-02-08 07:53:34	39.747753	-84.205582	NaN	NaN	0.010	...	False	False	False	False	False	False	Night	Day	
4	A-5	Source2	2	2016-02-08 07:39:07	2016-02-08 08:09:07	39.627781	-84.188354	NaN	NaN	0.010	...	False	False	False	False	True	False	Day	Day	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
7728389	A-7777757	Source1	2	2019-08-23 18:03:25	2019-08-23 18:32:01	34.002480	-117.379360	33.99888	-117.37094	0.543	...	False	False	False	False	False	False	Day	Day	
7728390	A-7777758	Source1	2	2019-08-23 19:11:30	2019-08-23 19:38:23	32.766960	-117.148060	32.76555	-117.15363	0.338	...	False	False	False	False	False	False	Day	Day	
7728391	A-7777759	Source1	2	2019-08-23 19:00:21	2019-08-23 19:28:49	33.775450	-117.847790	33.77740	-117.85727	0.561	...	False	False	False	False	False	False	Day	Day	
7728392	A-7777760	Source1	2	2019-08-23 19:00:21	2019-08-23 19:29:42	33.992460	-118.403020	33.98311	-118.39565	0.772	...	False	False	False	False	False	False	Day	Day	
7728393	A-7777761	Source1	2	2019-08-23 18:52:06	2019-08-23 19:21:31	34.133930	-117.230920	34.13736	-117.23934	0.537	...	False	False	False	False	False	False	Day	Day	

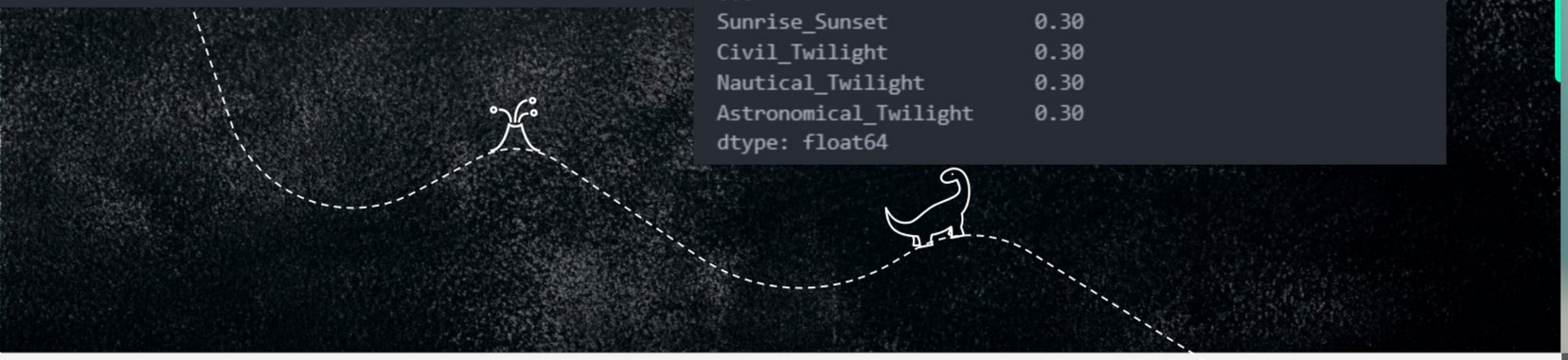
7728394 rows × 46 columns

About Data :

```
 #   Column                Dtype
---  ------                -----
 0   ID                    object
 1   Source                object
 2   Severity              int64
 3   Start_Time            object
 4   End_Time              object
 5   Start_Lat             float64
 6   Start_Lng             float64
 7   End_Lat               float64
 8   End_Lng               float64
 9   Distance(mi)          float64
 10  Description           object
 11  Street                object
 12  City                  object
 13  County                object
 14  State                 object
 15  Zipcode               object
 16  Country               object
 17  Timezone              object
 18  Airport_Code          object
 19  Weather_Timestamp     object
...
 44  Nautical_Twilight     object
 45  Astronomical_Twilight  object
dtypes: bool(13), float64(12), int64(1), object(20)
memory usage: 2.0+ GB
```

```
round(df.isna().sum()/ len(df) * 100, 2)
```

```
ID                      0.00
Source                  0.00
Severity                0.00
Start_Time              0.00
End_Time                0.00
Start_Lat               0.00
Start_Lng               0.00
End_Lat                44.03
End_Lng                44.03
Distance(mi)            0.00
Description             0.00
Street                  0.14
City                    0.00
County                  0.00
State                   0.00
Zipcode                 0.02
Country                 0.00
Timezone                0.10
Airport_Code            0.29
Weather_Timestamp       1.56
Temperature(F)          2.12
Wind_Chill(F)          25.87
Humidity(%)             2.25
Pressure(in)            1.82
Visibility(mi)          2.29
...
Sunrise_Sunset          0.30
Civil_Twilight          0.30
Nautical_Twilight       0.30
Astronomical_Twilight   0.30
dtype: float64
```



Column names and descriptions

ID: A unique identification number for each incident.

Source: Some source or the source by which the incident was reported.

Severity: The degree of severity of the accident.

Start_Time and End_Time: The time the incident started and the time it ended.

Start_Lat and Start_Lng: Latitude and longitude of the accident start location.

End_Lat and End_Lng: The latitude and longitude of the accident end location.

Distance(mi): The distance affected by the accident in miles.

Description: A brief description of the incident.

Street: The name of the street or road.

City, County, State, Zipcode, Country: Location, city, state, zip code and country information.

Timezone: The time zone.

Column names and descriptions

Airport_Code: Code of the nearby airport.

Weather_Timestamp: Time to read weather data.

Temperature(F), Wind_Chill(F), Humidity(%), Pressure(in), Visibility(mi), etc.: Weather data related to the incident, such as temperature, humidity, barometric pressure, and visibility.

Amenity, Bump, Crossing, Give_Way, etc.: Various factors related to roads and infrastructure.

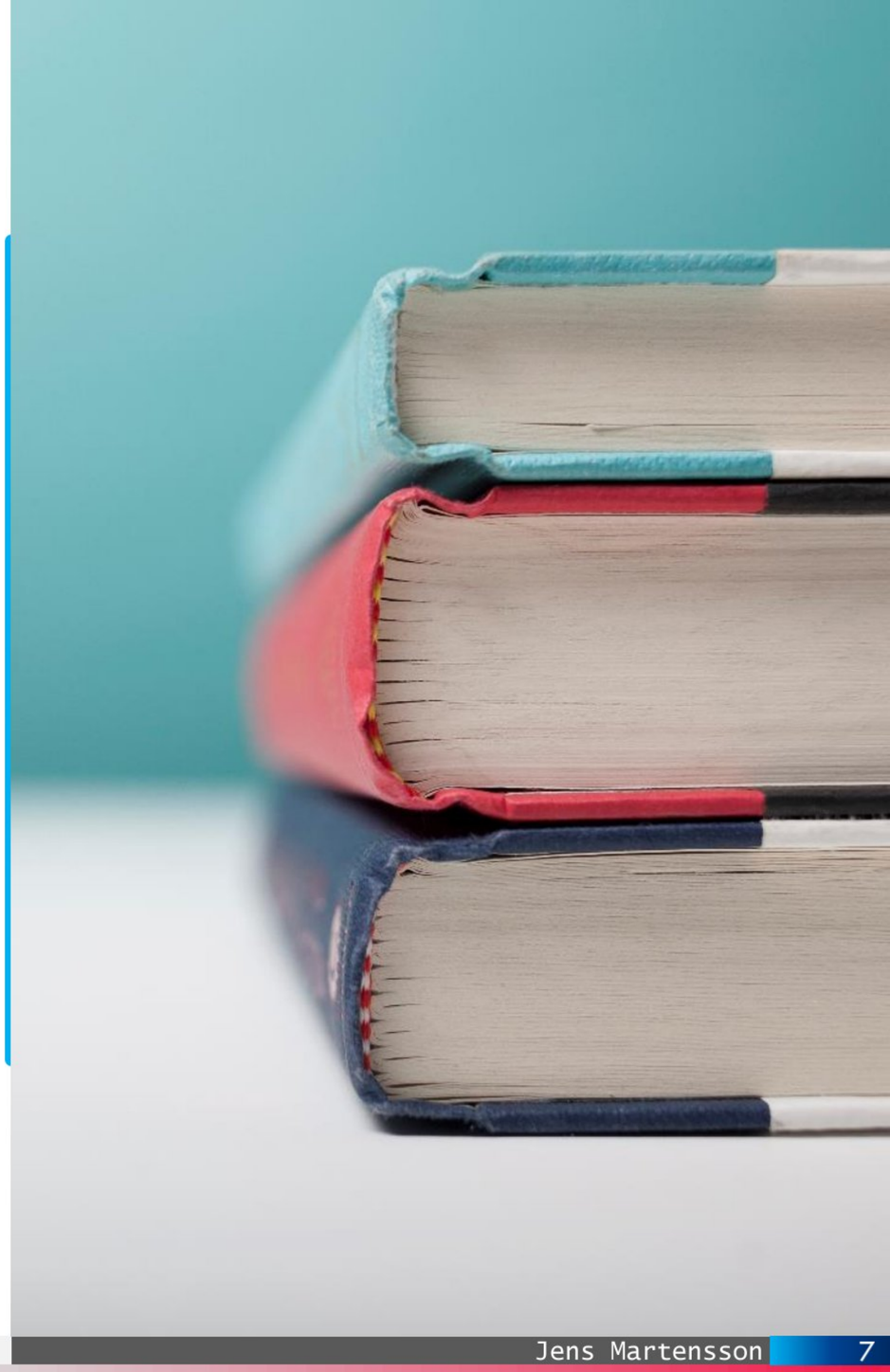
Sunrise_Sunset, Civil_Twilight, Nautical_Twilight, Astronomical_Twilight: Sunrise, dusk, sand and astronomical twilight

Solve NULL problems

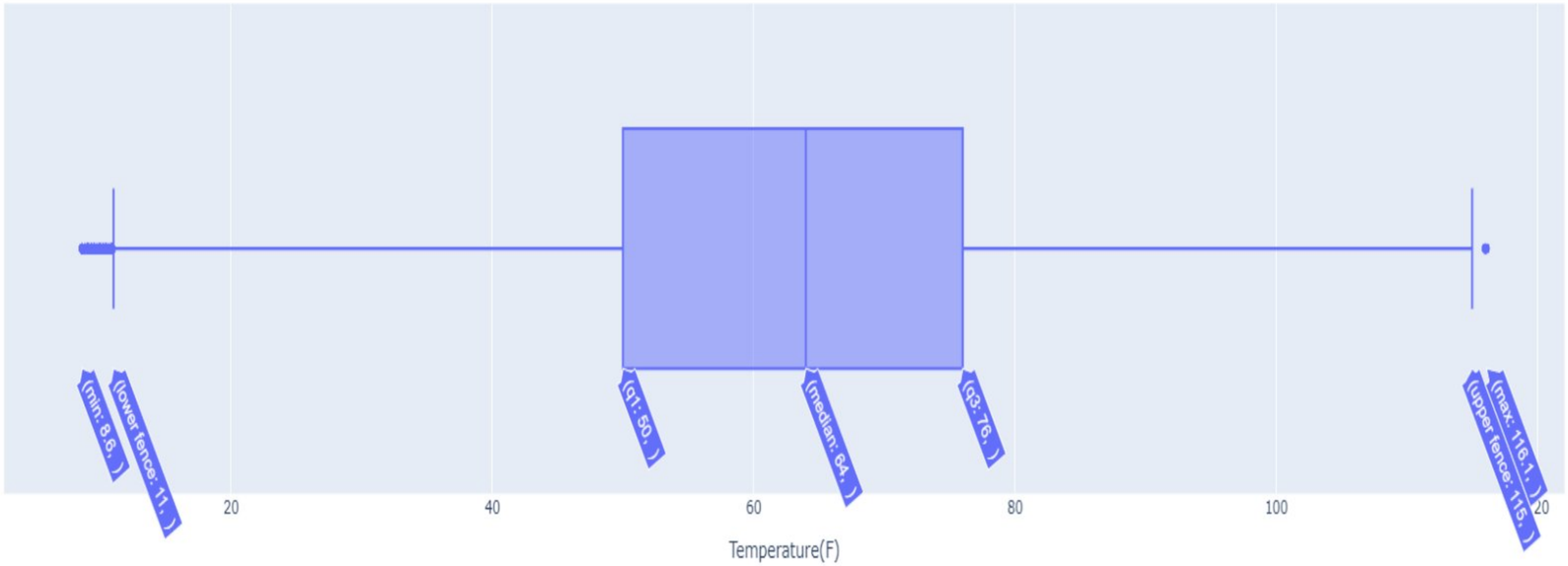
I addressed null values in the data by removing those with a null percentage exceeding 40%. For instance, when there were null values in street names, I extracted city names and identified the most frequently occurring street within each city. This method helped in dealing with missing data effectively by filling in the street names using the most common street associated with each city.

```
x=df[df["City"]=="San Diego"]
y=x["Street"].mode()
L=df[(df['Street'].isnull() ) & (df['City'] =='San Diego')].index
df['Street'].iloc[L] =y
```

```
x=df[df["City"]=="Charlotte"]
y=x["Street"].mode()
L=df[(df['Street'].isnull() ) & (df['City'] =='Charlotte')].index
df['Street'].iloc[L] =y
```

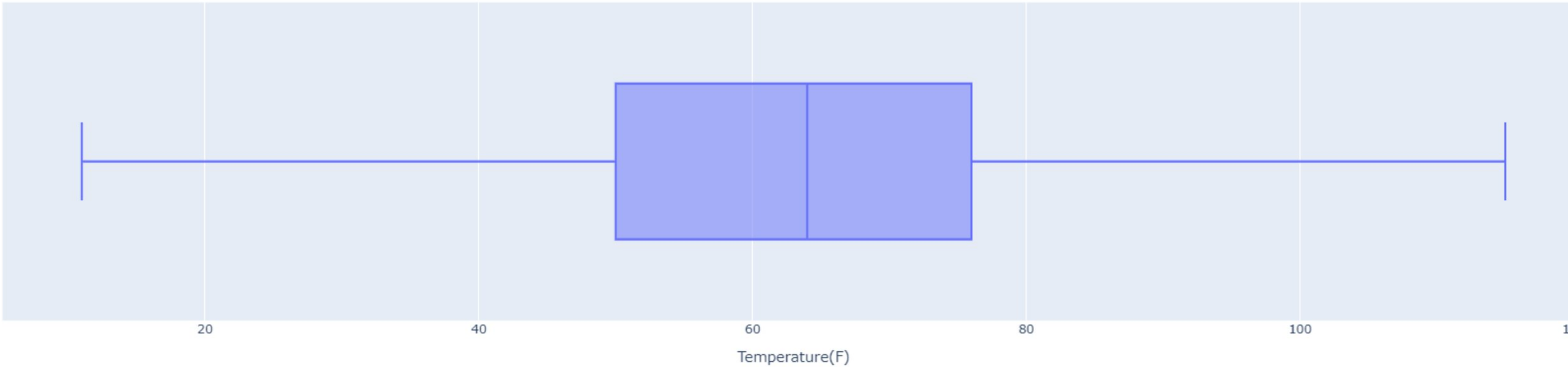


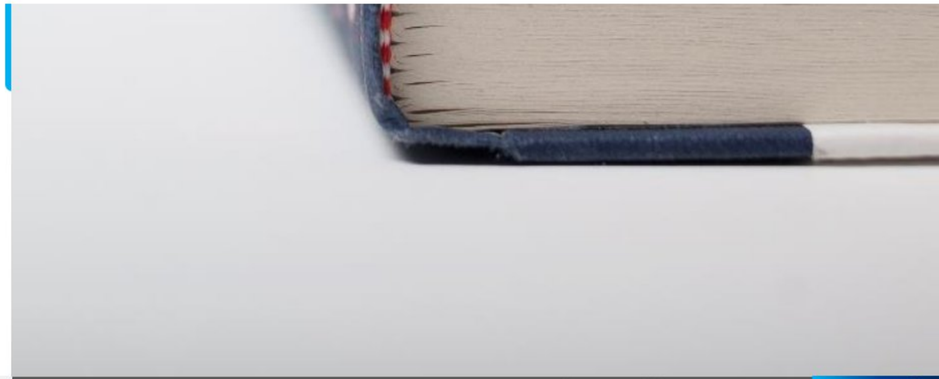
To handle outliers in a temperature



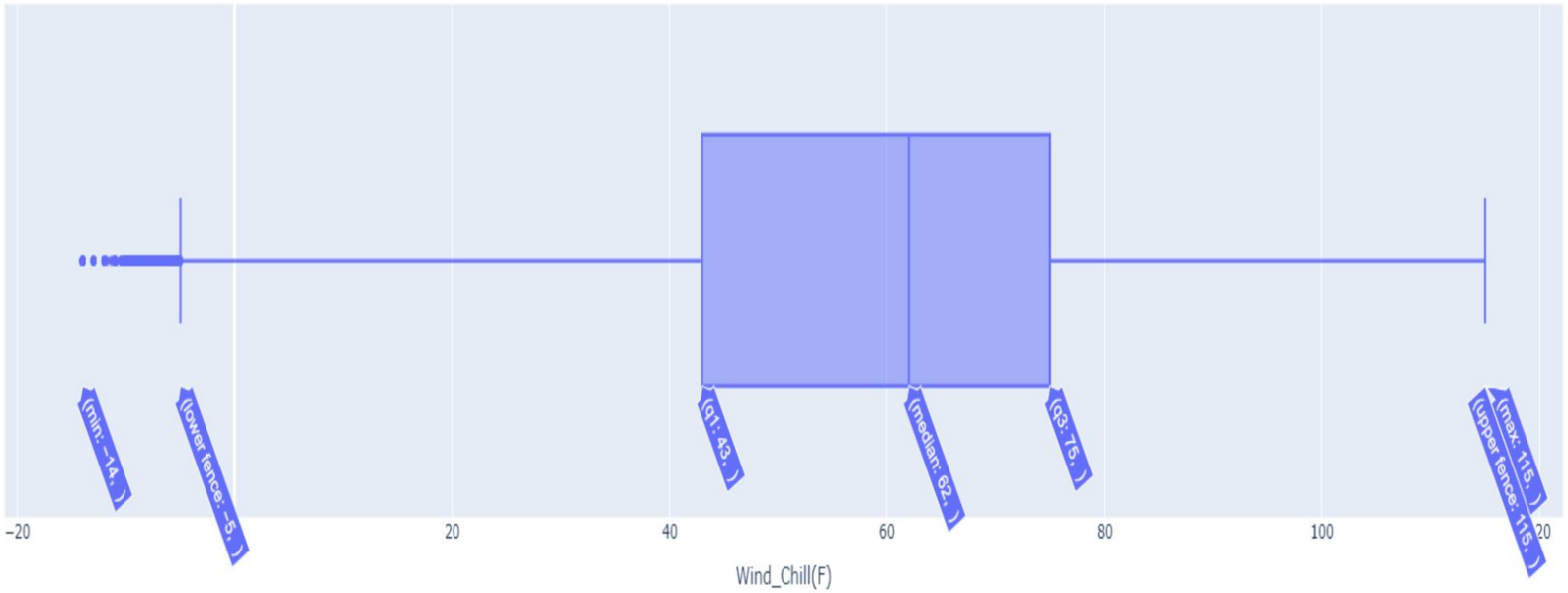

```
Q1 = df['Temperature(F)'].quantile(0.25)
Q3 = df['Temperature(F)'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
x=df[(df['Temperature(F)'] >= lower_bound) & (df['Temperature(F)'] <= upper_bound)]
df=x
px.box(df, x='Temperature(F)')
```

To handle outliers in a temperature



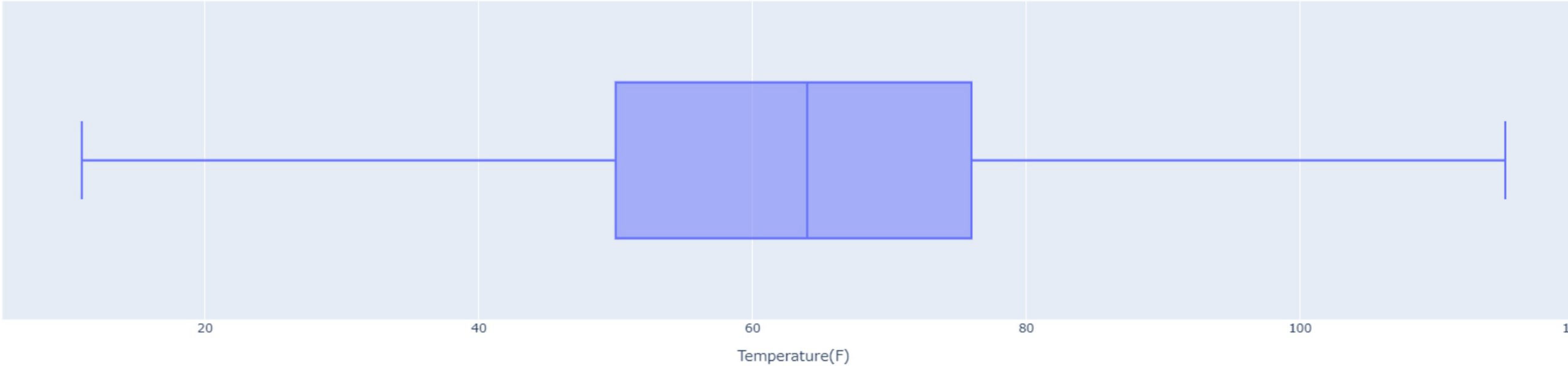


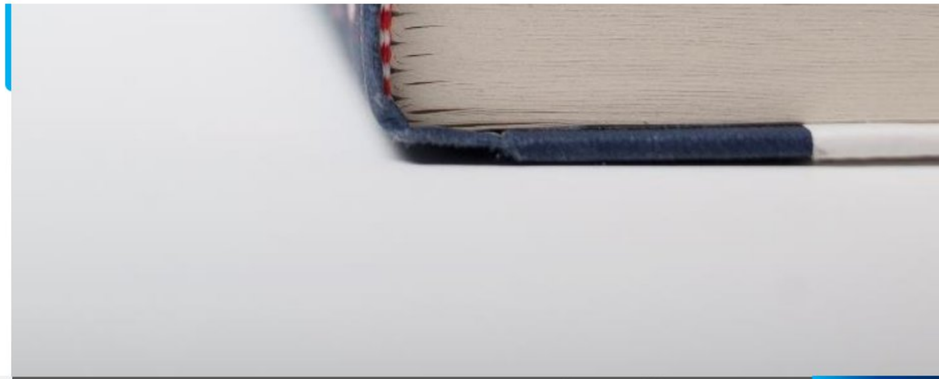
To handle outliers in a Wind_Chill



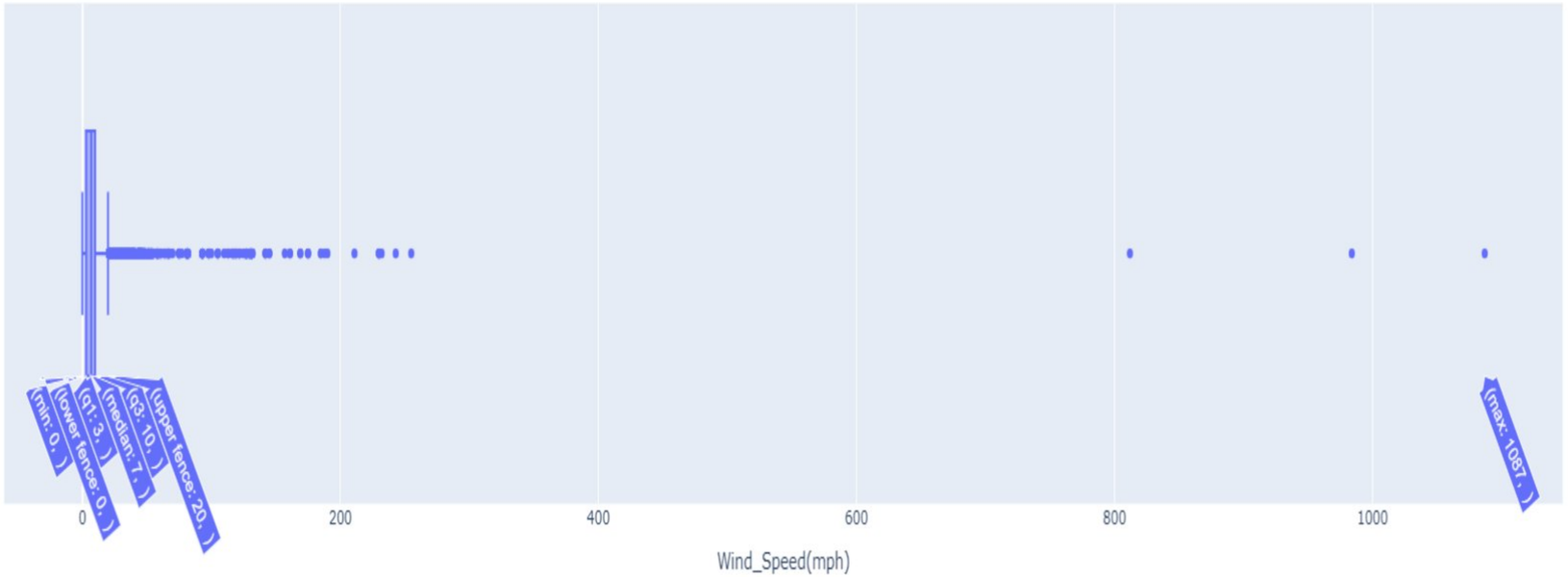

```
Q1 = df['Wind_Chill(F)'].quantile(0.25)
Q3 = df['Wind_Chill(F)'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
x=df[(df['Wind_Chill(F)'] >= lower_bound) & (df['Wind_Chill(F)'] <= upper_bound)]
df=x
px.box(df, x='Wind_Chill(F)')
```

To handle outliers in a Wind_Chill

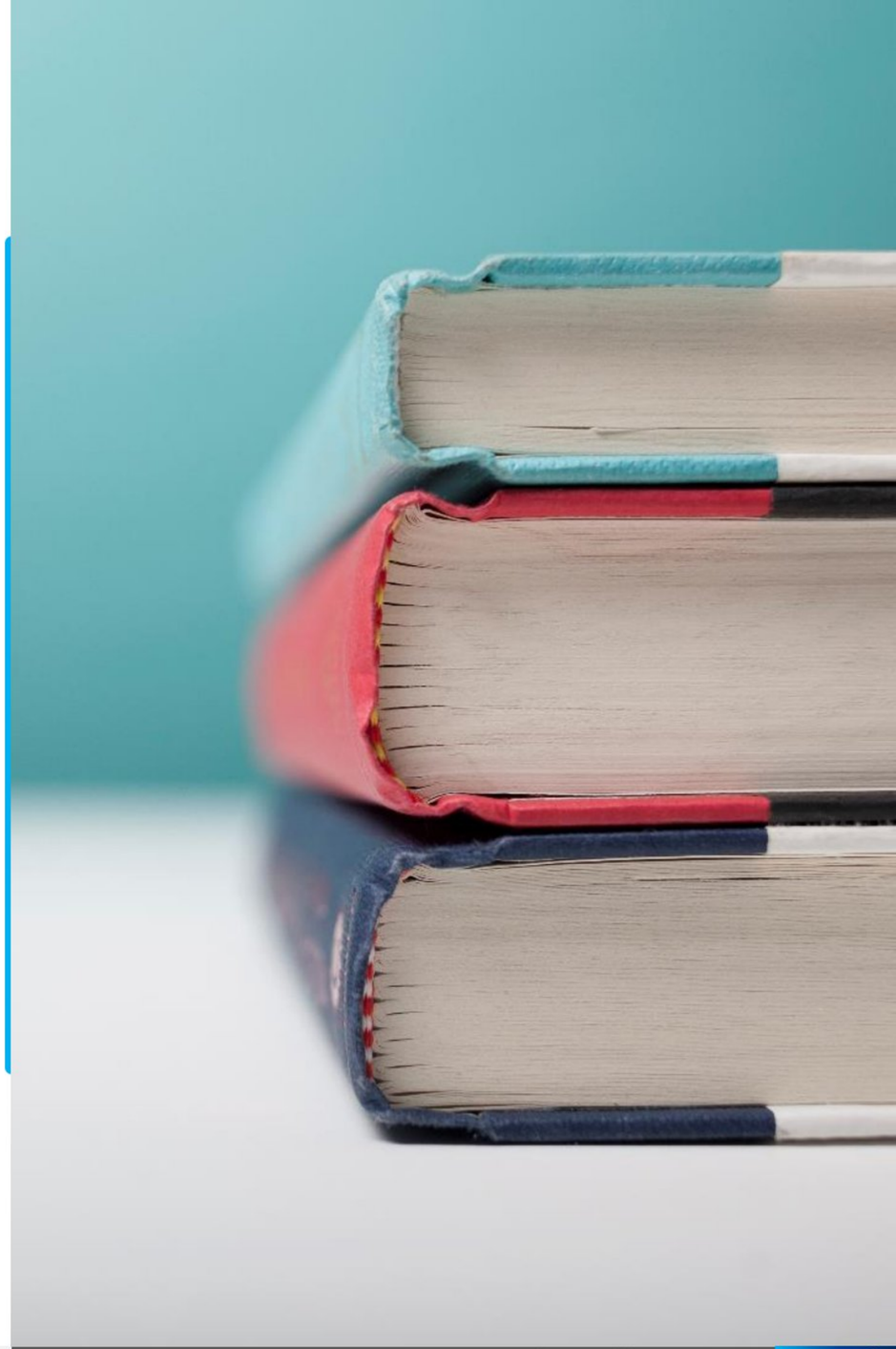




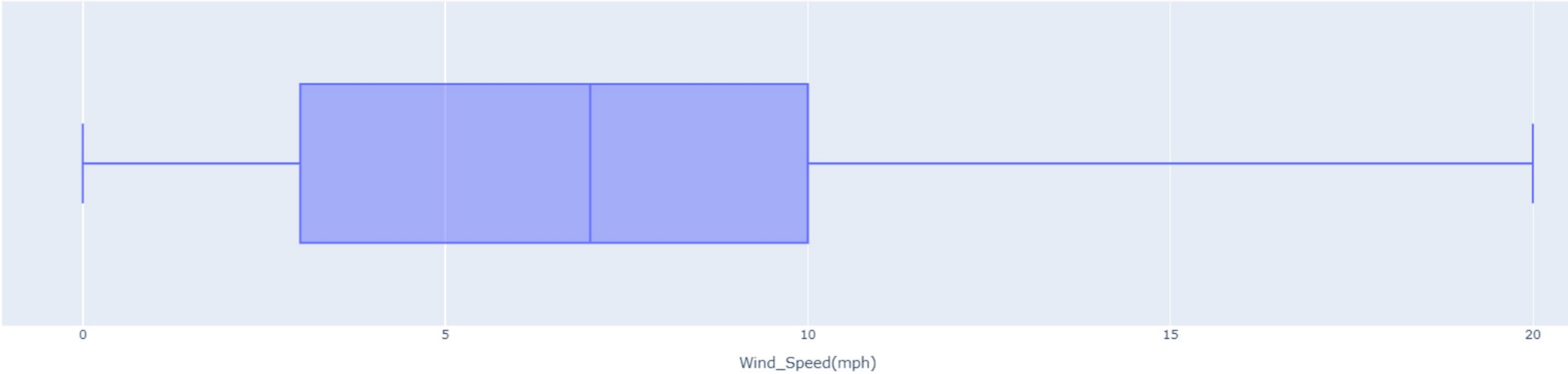
To handle outliers in a Wind_Speed




```
Q1 = df['Wind_Speed(mph)'].quantile(0.25)
Q3 = df['Wind_Speed(mph)'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
x=df[(df['Wind_Speed(mph)'] >= lower_bound) & (df['Wind_Speed(mph)'] <= upper_bound)]
df=x
px.box(df, x='Wind_Speed(mph)')
```



To handle outliers in a Wind_Speed



To handle outliers

Of course, these techniques have been used to handle outliers in more than one table in the data set. Among the tables that were dealt with using these standard techniques and methodology for handling outliers were the “Temperature”, “Humidity”, and “Pressure” tables.

Feature Enginnering

```
df['Day_of_accident']=df['Start_Date'].dt.day
df['Month_of_accident']=df['Start_Date'].dt.month
df['Year_of_accedent']=df['Start_Date'].dt.year
```

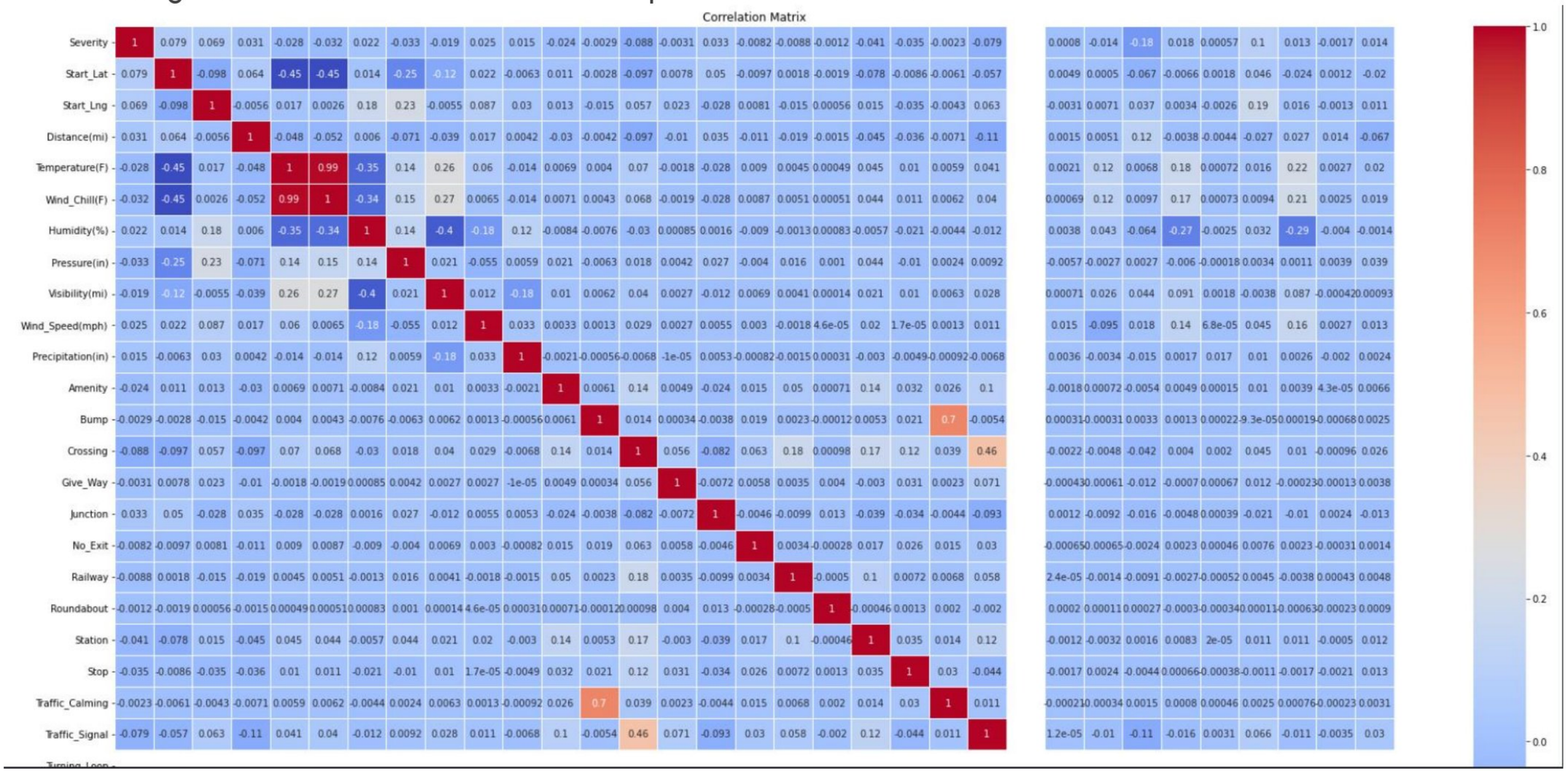
```
df["Hour_of_end_accident"]=df["End_Time"].dt.hour
df['Minutes_of_end_accident'] = df['End_Time'].dt.minute
df['Seconds_of_end_accident'] = df['End_Time'].dt.second
```

```
def map_months(x):
    if x in [12, 1, 2]:
        return 'Winter'
    elif x in [3, 4, 5]:
        return 'Spring'
    elif x in [6, 7, 8]:
        return 'Summer'
    elif x in [9, 10, 11]:
        return 'Autumn'

df['Season'] = df['Month_of_accident'].apply(map_months)
```

Analysis Data

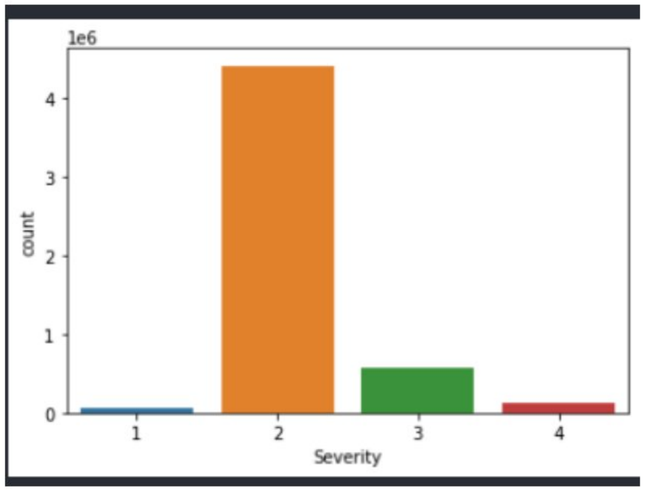
- Correlation Matrix is a table that shows the degrees of relationship between different variables in a data set. This matrix is used to understand how variables relate to each other. Values are between -1 and +1 and provide information about the strength and direction of the relationship between variables.

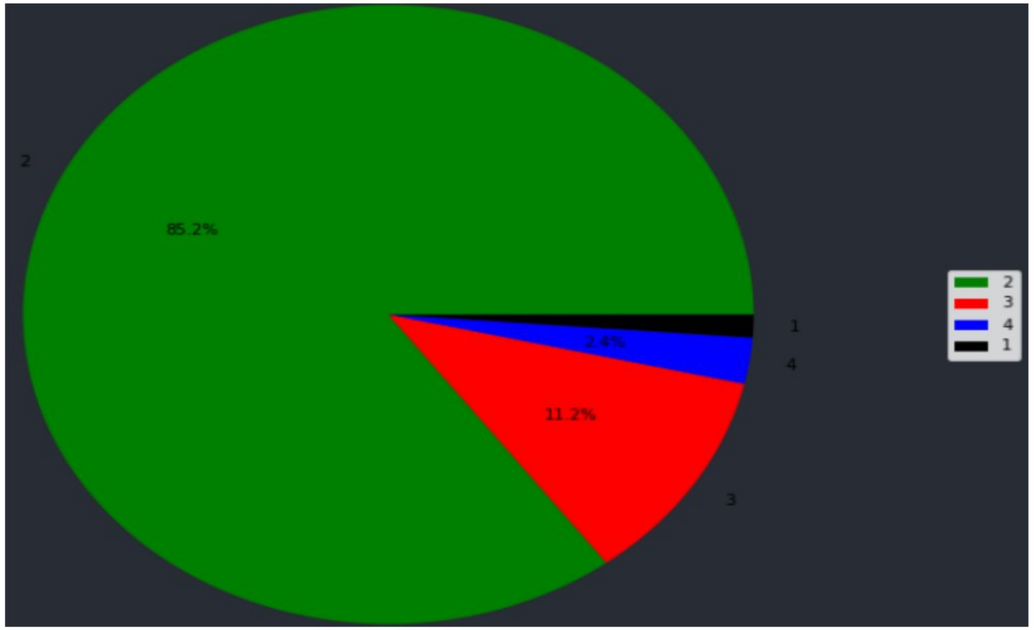


Analysis Data

We will then notice that there is not a very strong relationship between a certain table and another table

We will notice that most of the accidents in that period had a severity level of = 2





1: Represents accidents of less severity. These accidents may be minor and do not cause serious damage or serious injury.

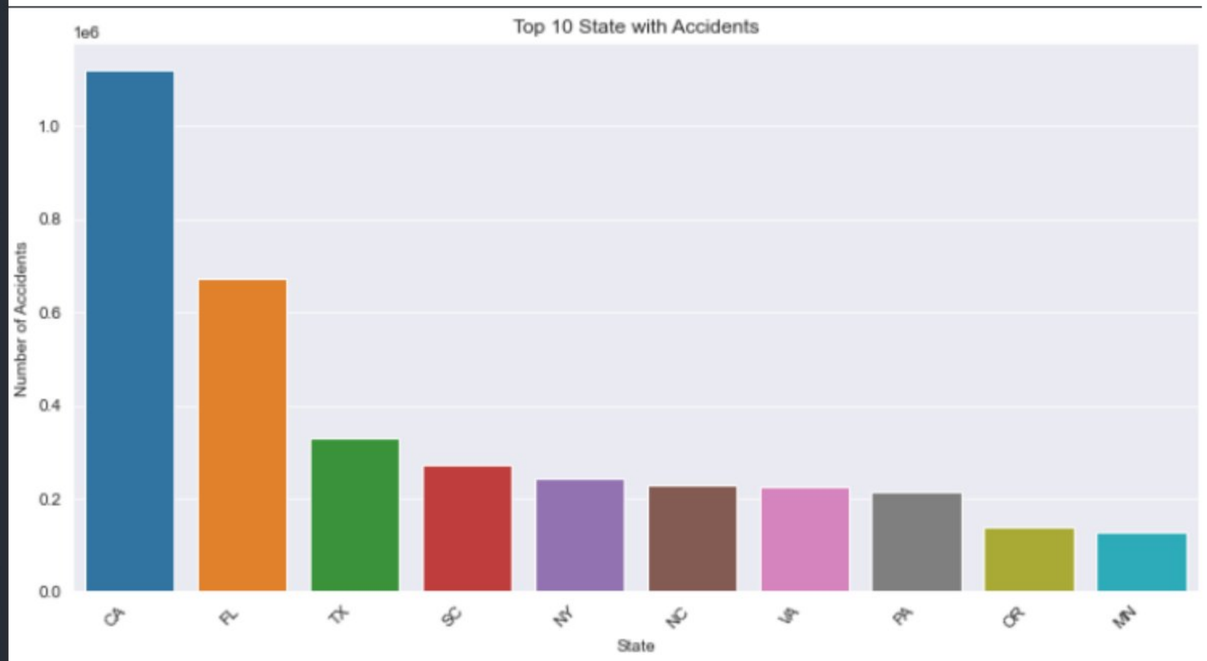
2: Represents incidents of moderate severity. These accidents may be more serious than those classified as 1, and may cause moderate injuries.

3: Represents incidents with a high degree of severity. These accidents are considered serious and may cause serious injuries and significant damage.

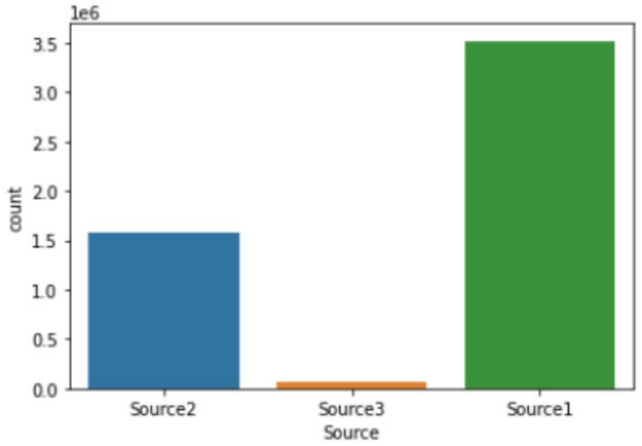
4: This value often indicates incidents with the highest degree of severity. These accidents can be very serious and cause serious injuries and significant damage.

Analysis Data

State	ID
CA	1121677
FL	672487
TX	330074
SC	273079
NY	244502
NC	227140
VA	225664
PA	214869
OR	138544
MN	126775
TN	125015
AZ	111476
LA	110609
IL	98438
GA	96332
NJ	96007
MI	95204
MD	80869
AL	77328
OH	72828
UT	65380
WA	60451
CO	58968
CT	52698
MO	50037
OK	48069
MA	37188
IN	34946
MT	23128
WI	21453



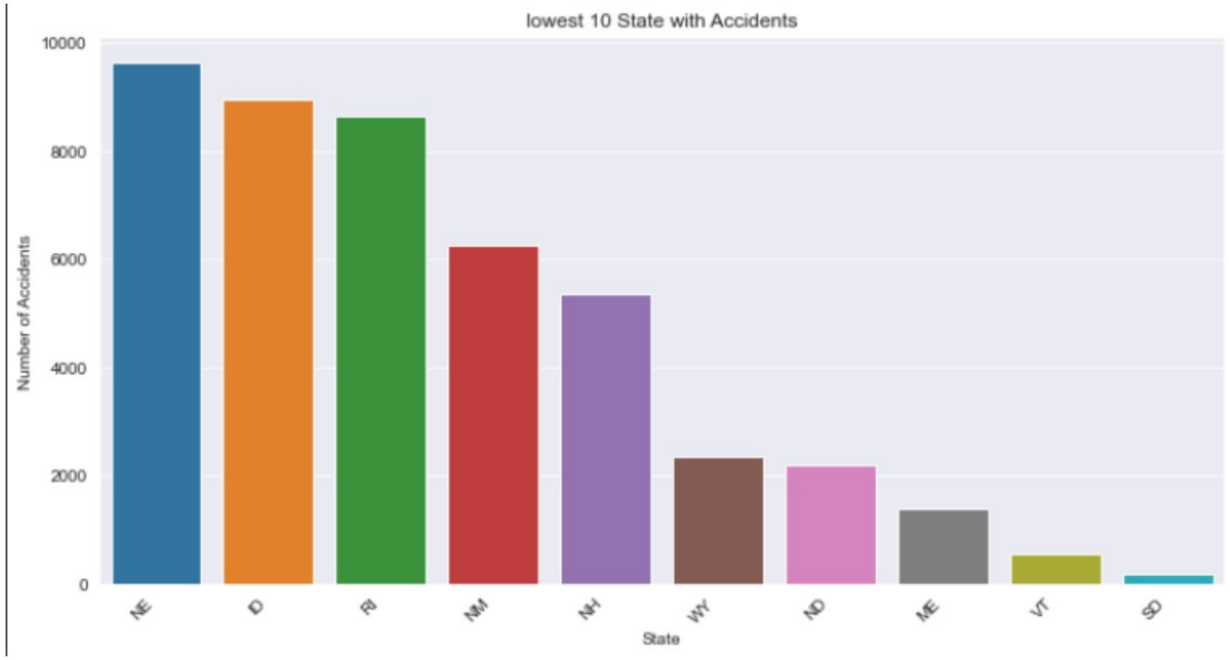
Most accident data was collected from source No. 1



California (CA) has the highest rate of road accidents, followed by Florida (FL) and Texas (TX). These results reflect the distribution of accidents across states and highlight the states with the highest numbers of road accidents.

Analysis Data

State	ID
NE	9620
ID	8956
RI	8651
NM	6259
NH	5346
WY	2351
ND	2188
ME	1405
VT	565
SD	198



Nebraska (NE) leads with the fewest accidents, followed by Idaho (ID), Rhode Island (RI), New Mexico (NM), and New Hampshire (NH), respectively.

Analysis Data

List of five cities showing the highest number of accidents

Miami recorded 158,905 accidents.

Los Angeles recorded 104,080 accidents.

Houston recorded 91,880 accidents.

Charlotte recorded 87,880 accidents.

Orlando recorded 8,690 accidents.

The five lowest cities with the lowest numbers of road accidents are:

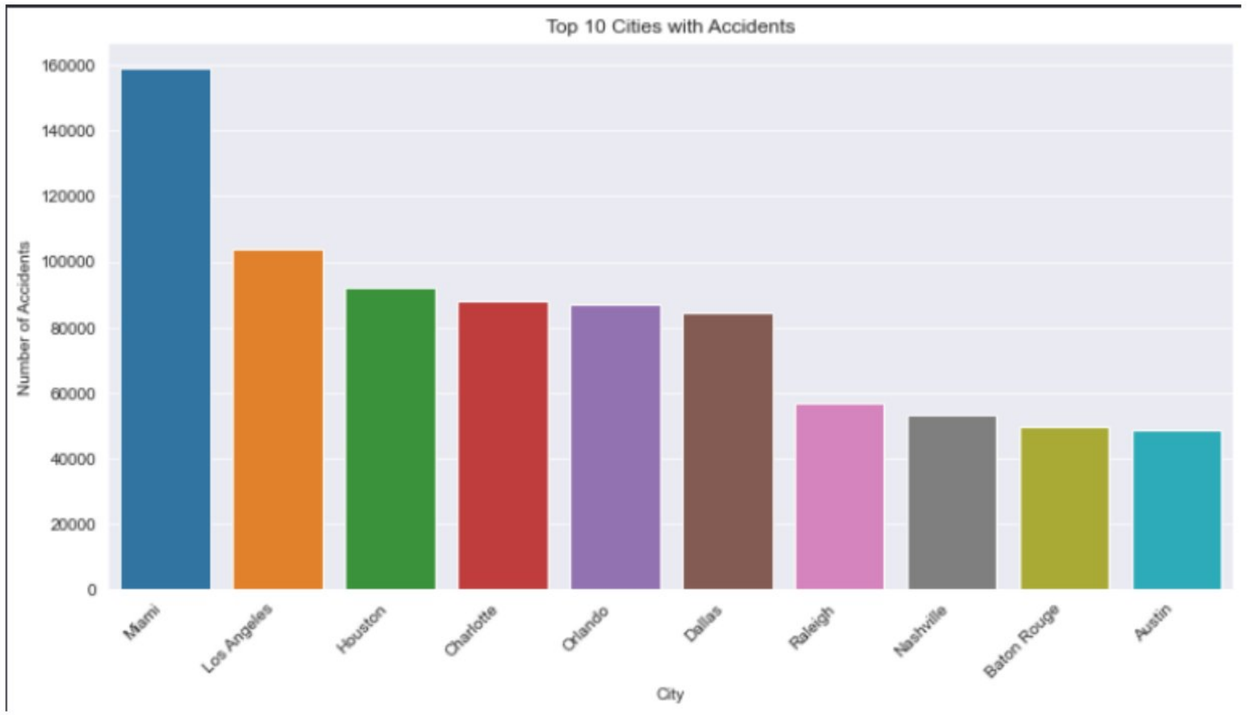
Only one incident was recorded in this city.

One incident occurred in the city of Valmora.

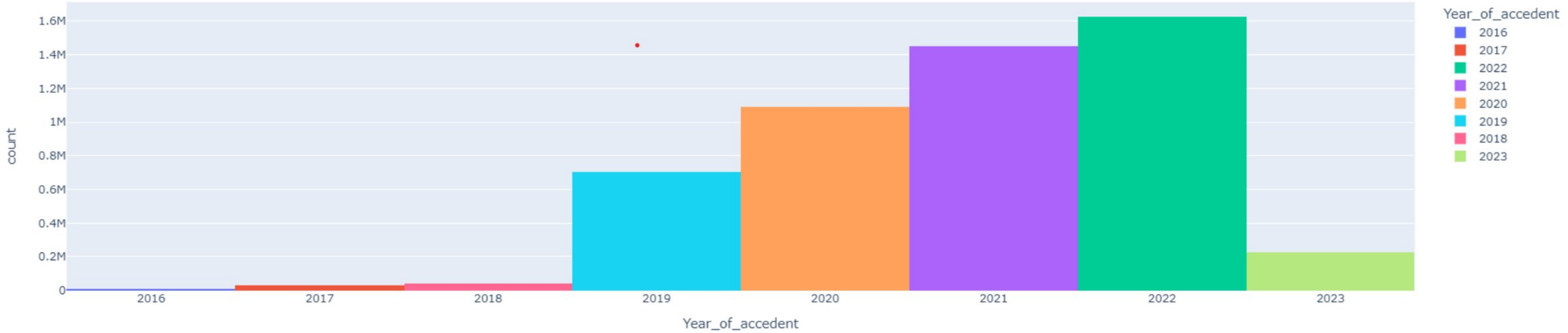
One incident was recorded in Van Alstyne.

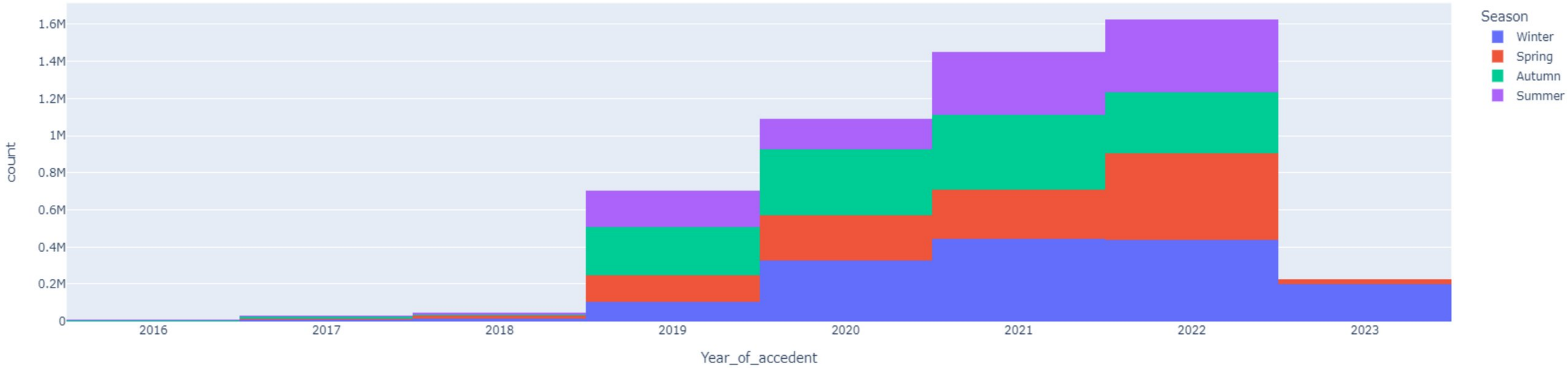
One incident was recorded in Sag Harbor.

The city of Harman recorded one incident

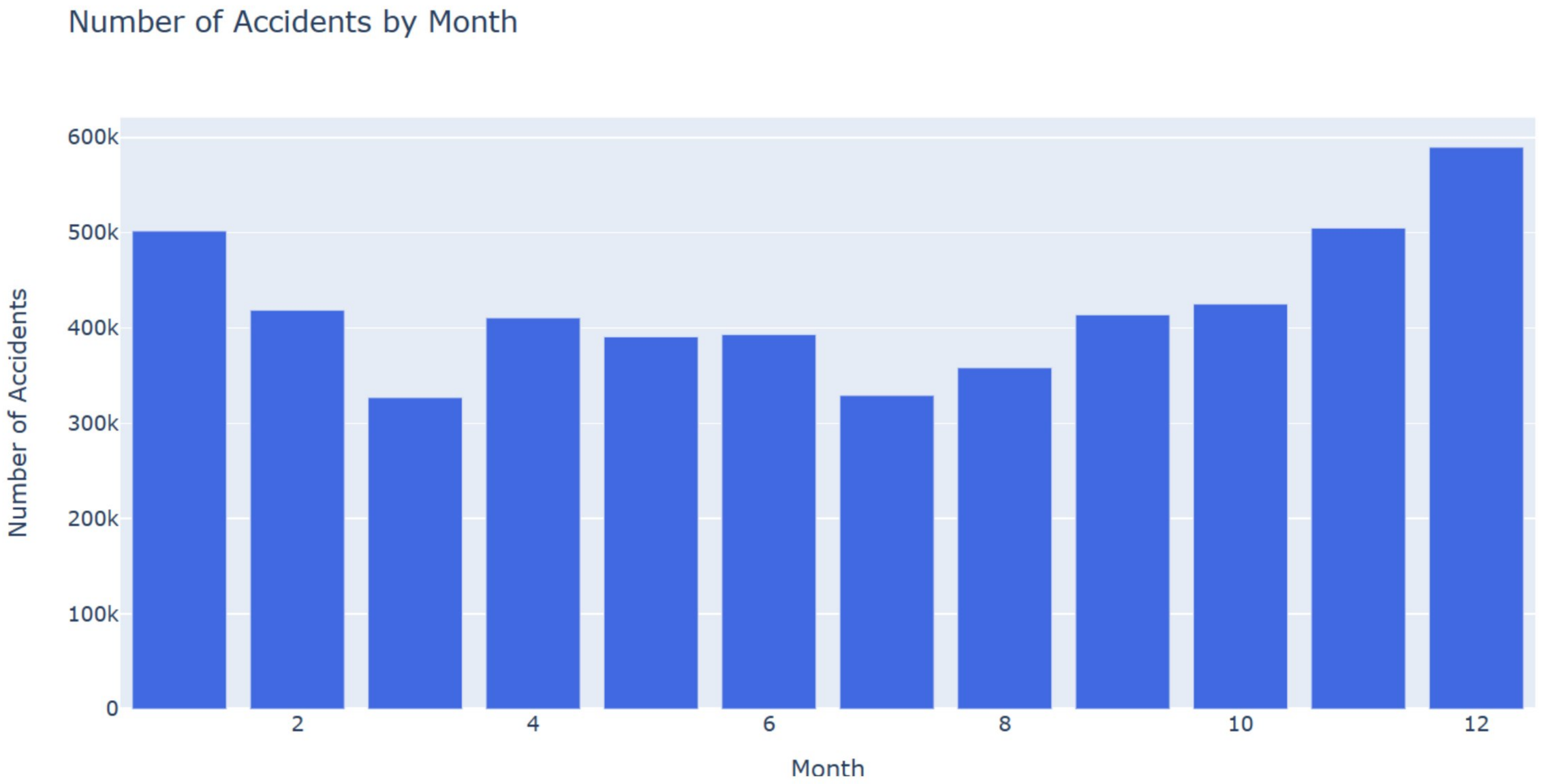


Analysis Data





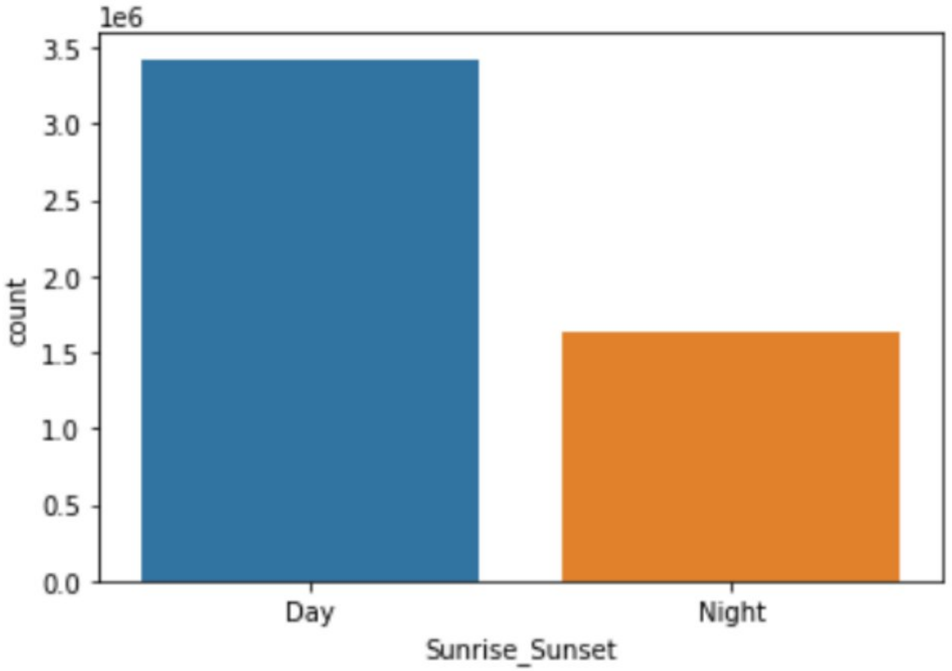
Analysis Data



Analysis Data

'Night': This value indicates that the incident occurred during the night period, i.e. after sunset and before sunrise.

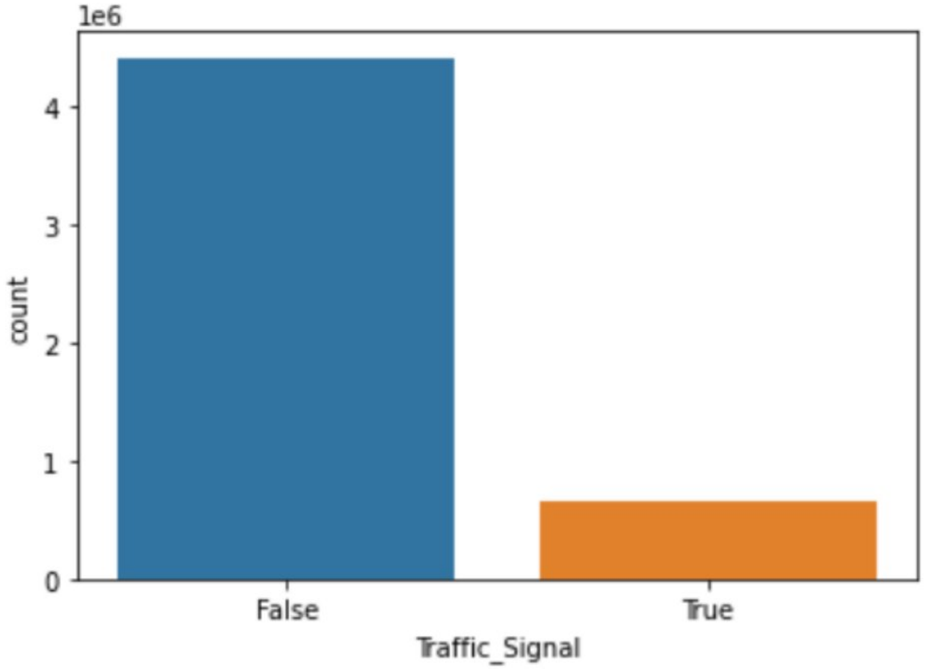
'Day': This value indicates that the incident occurred during the day, i.e. after sunrise and before sunset.



Analysis Data

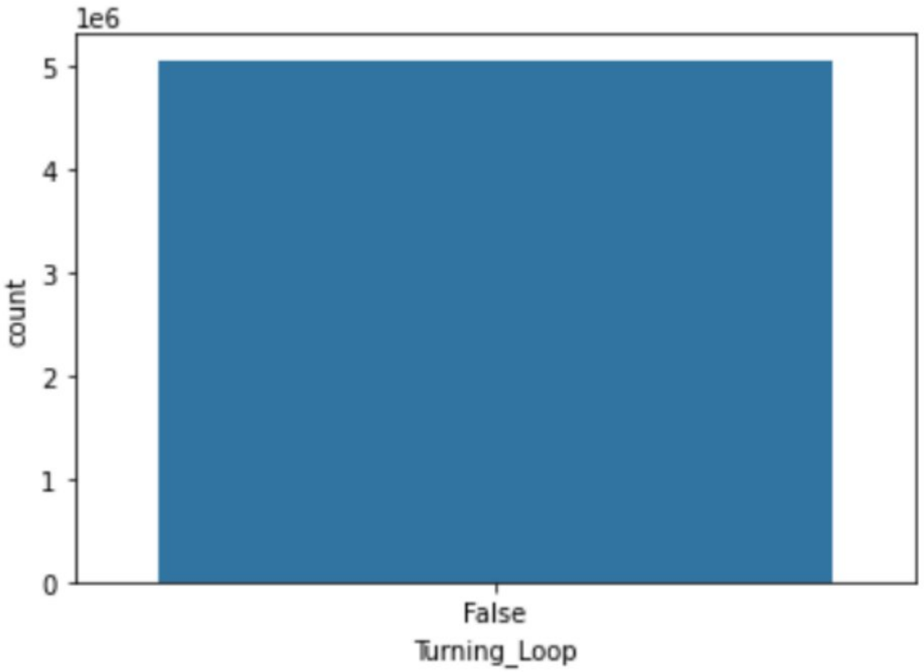
We see that most accidents occur in places where there are no traffic Signal

Therefore, it is important to place these signs in all places



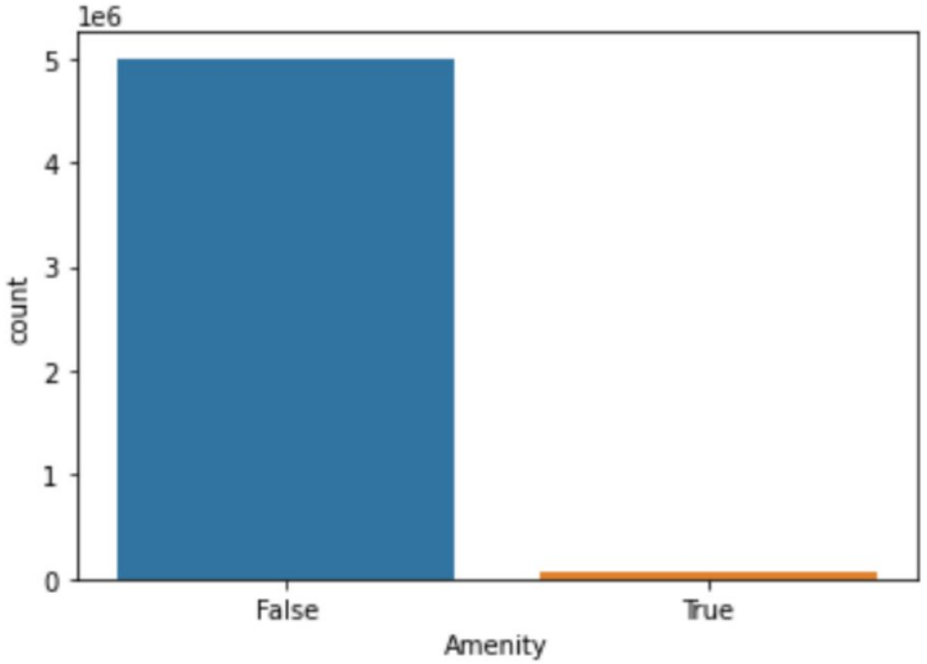
Analysis Data

All accidents occurred in places where there was no Turning Loop.



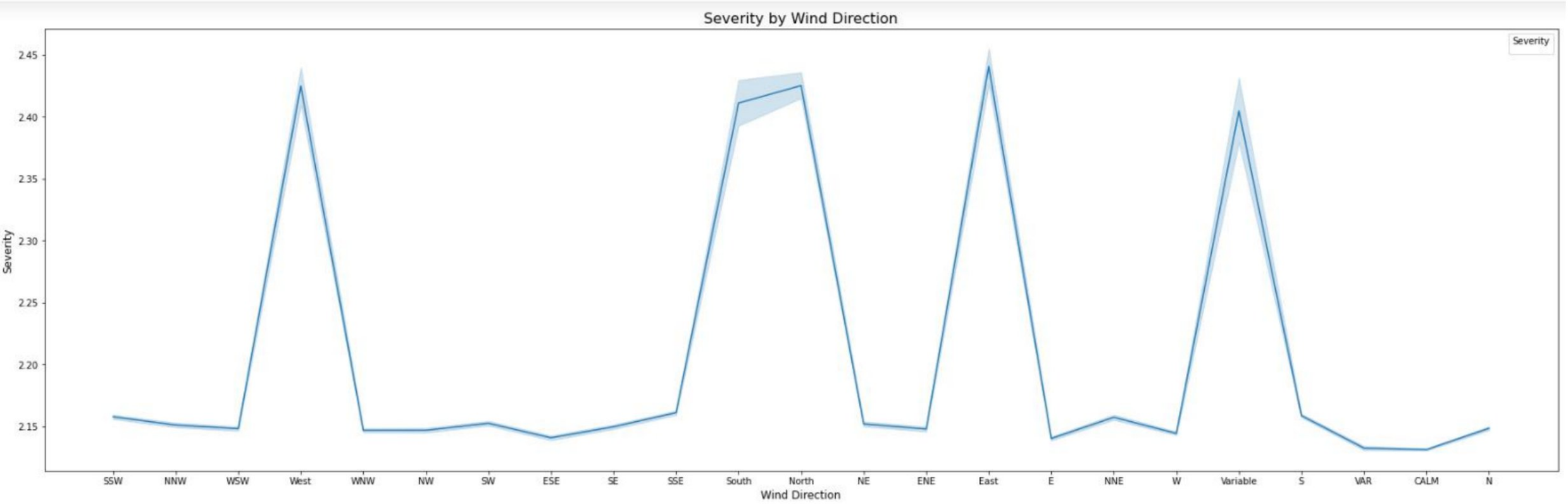
Analysis Data

Most accidents occurred in places where there was no safety permission sign on the road.



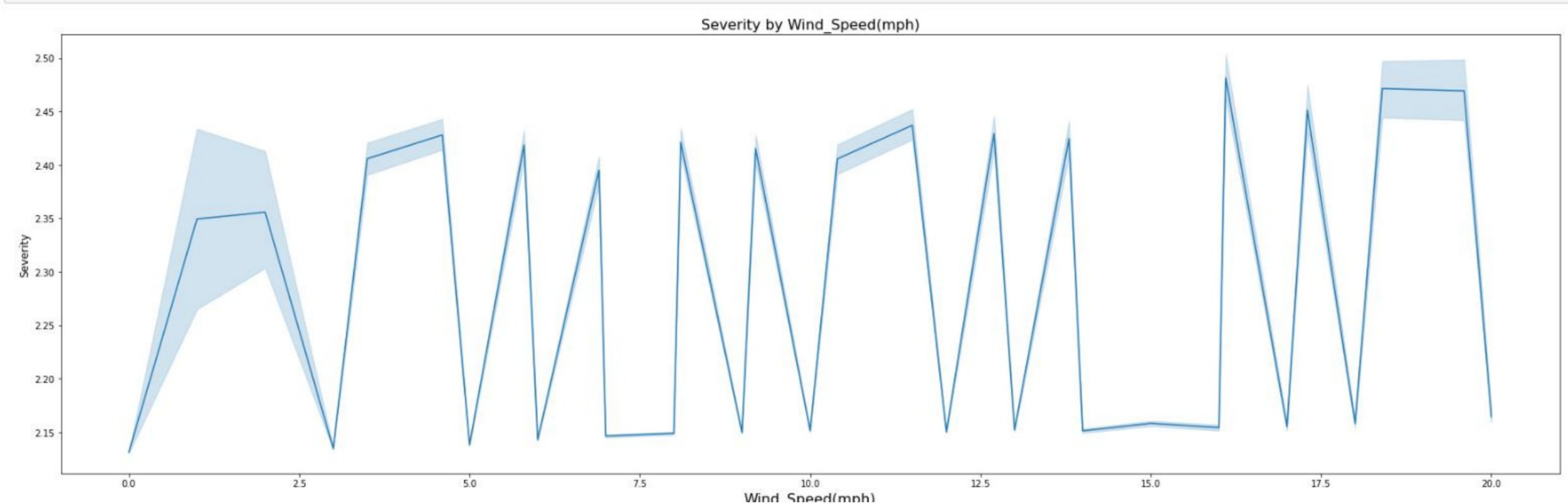
Analysis Data

Relationship between wind direction and degree of accident severity



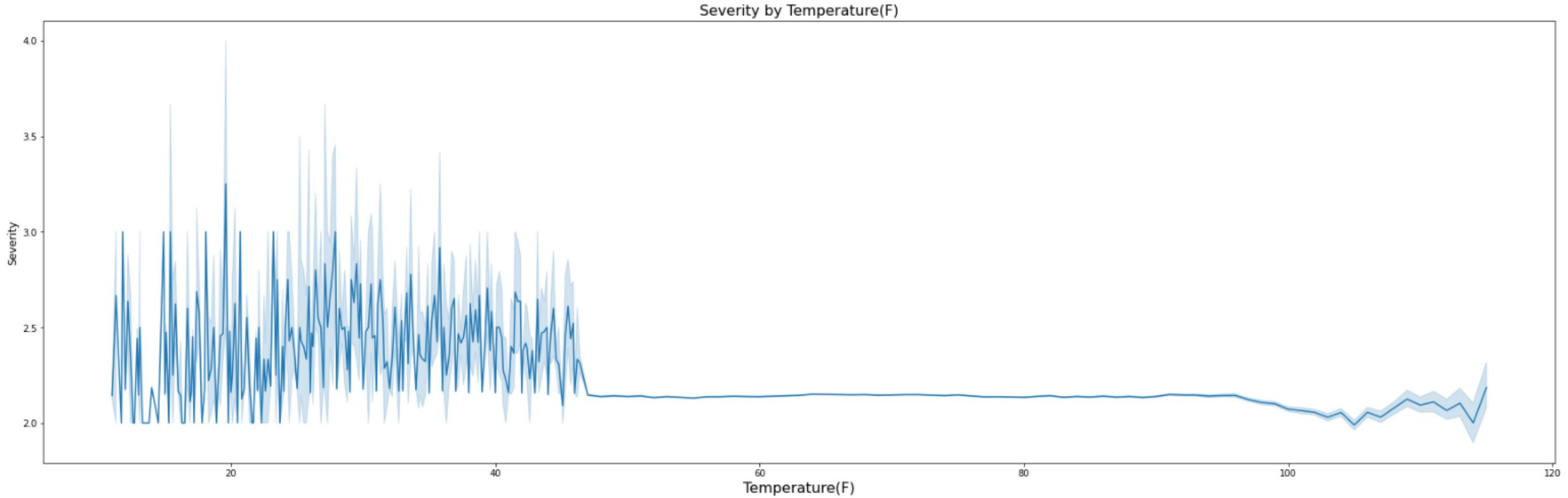
Analysis Data

Relationship between wind Speed(mph) and degree of accident severity



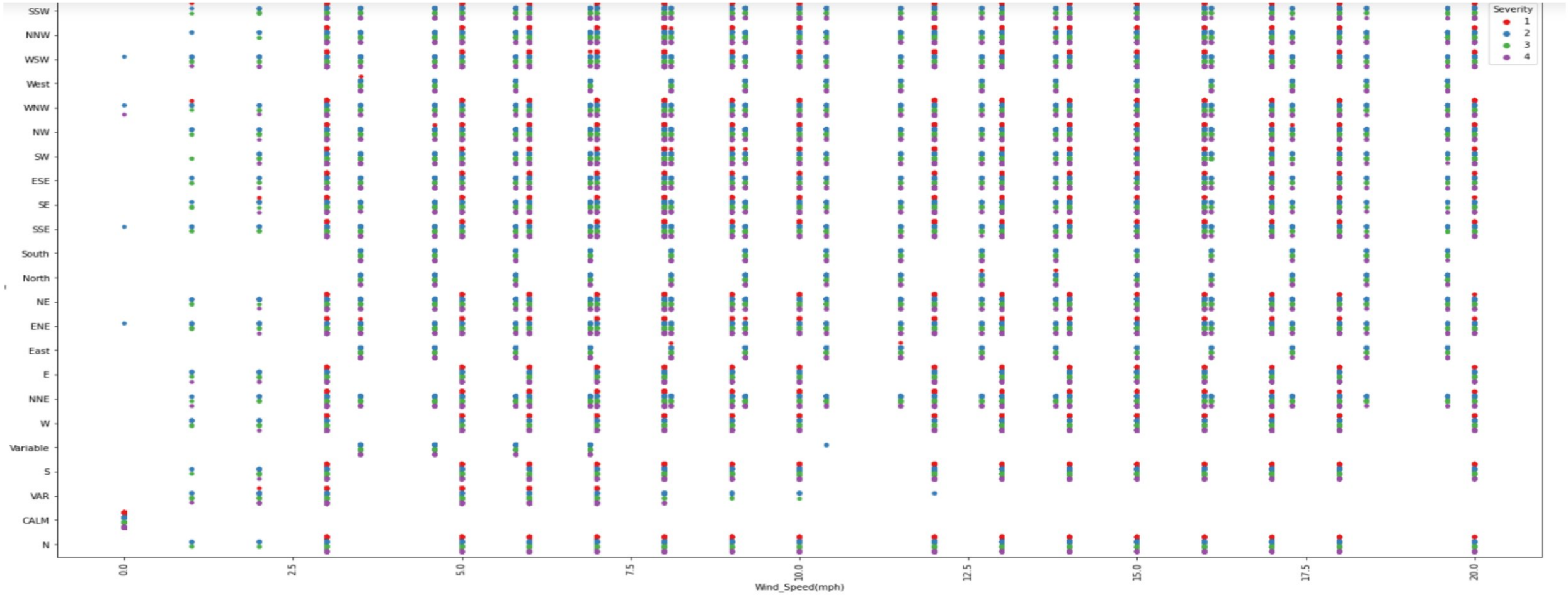
Analysis Data

Relationship between wind Temperature(F) and degree of accident severity



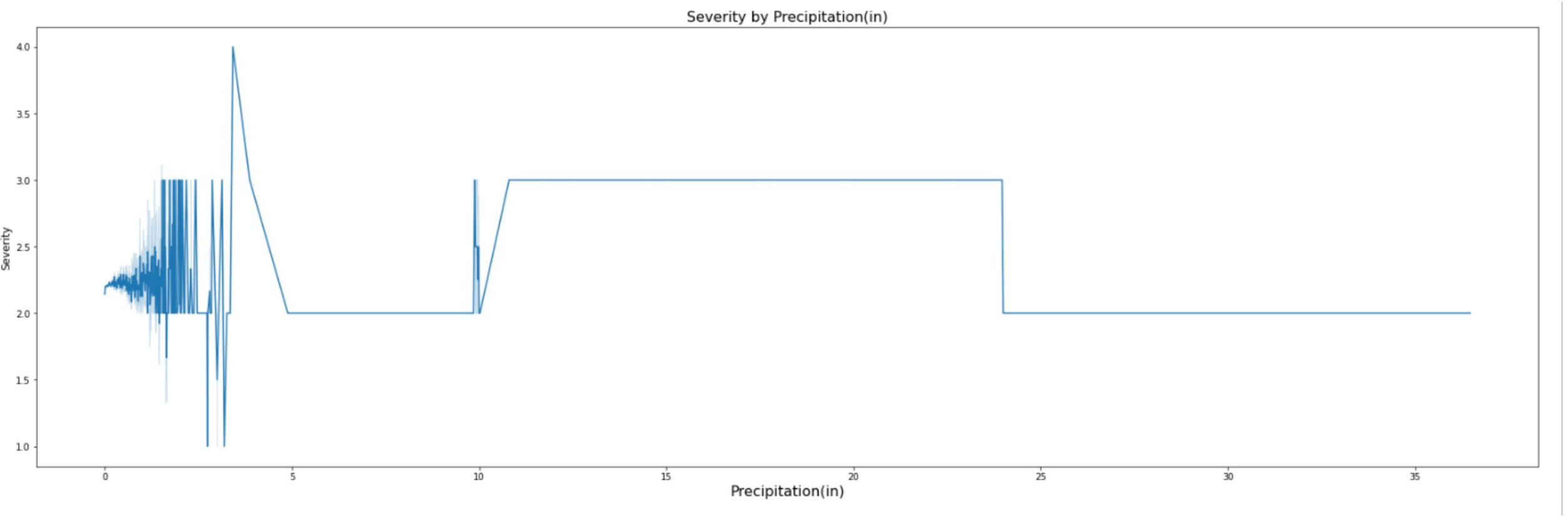
Analysis Data

A relationship between wind direction and wind speed and the extent of their impact on the Severity



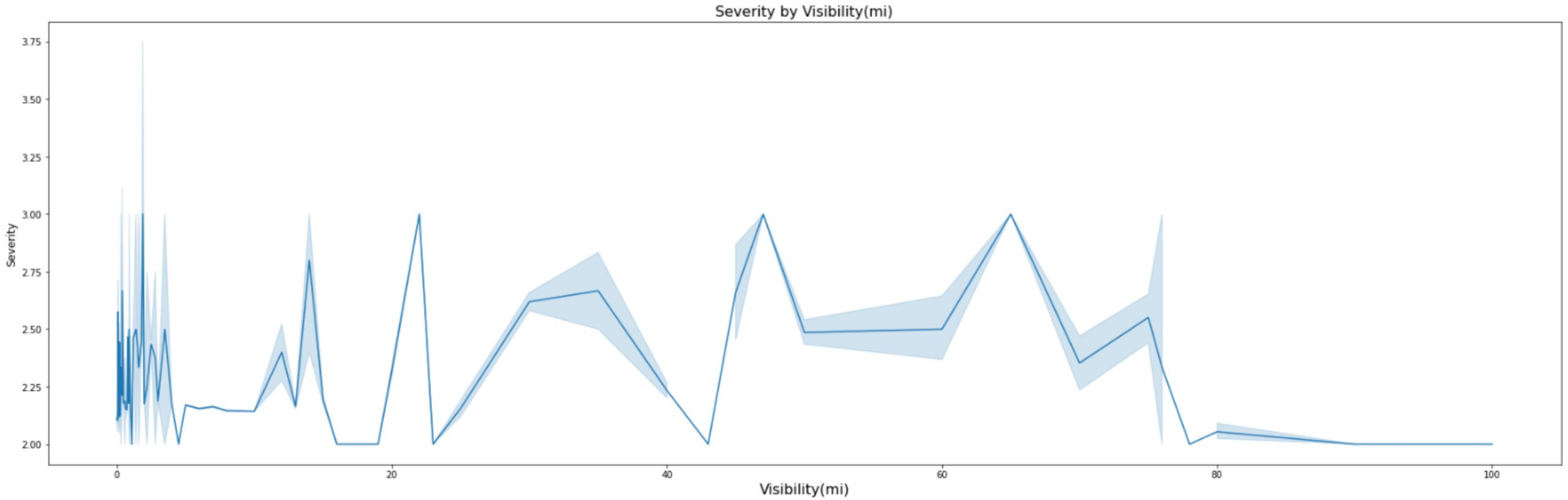
Analysis Data

Relationship between wind Precipitation(in) and degree of accident severity

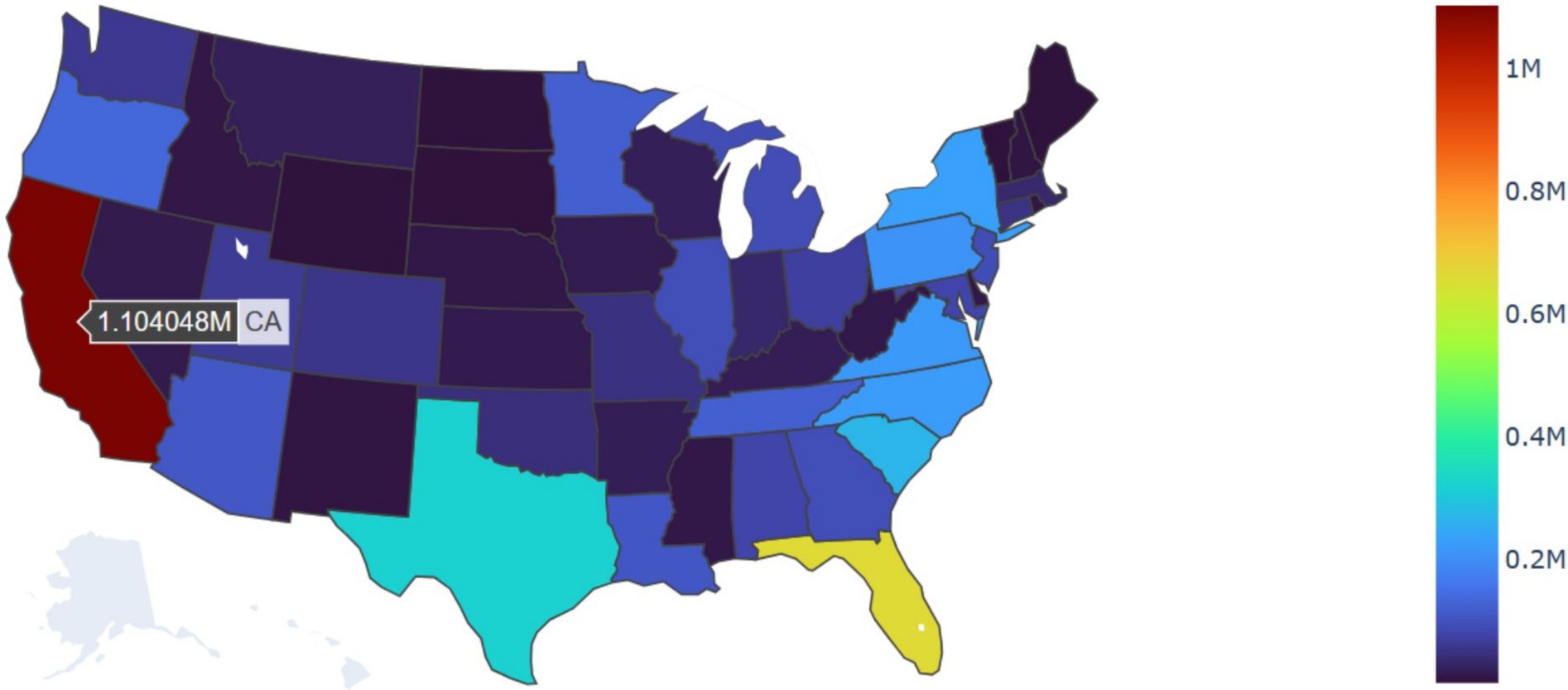


Analysis Data

Relationship between wind Visibility and degree of accident severity

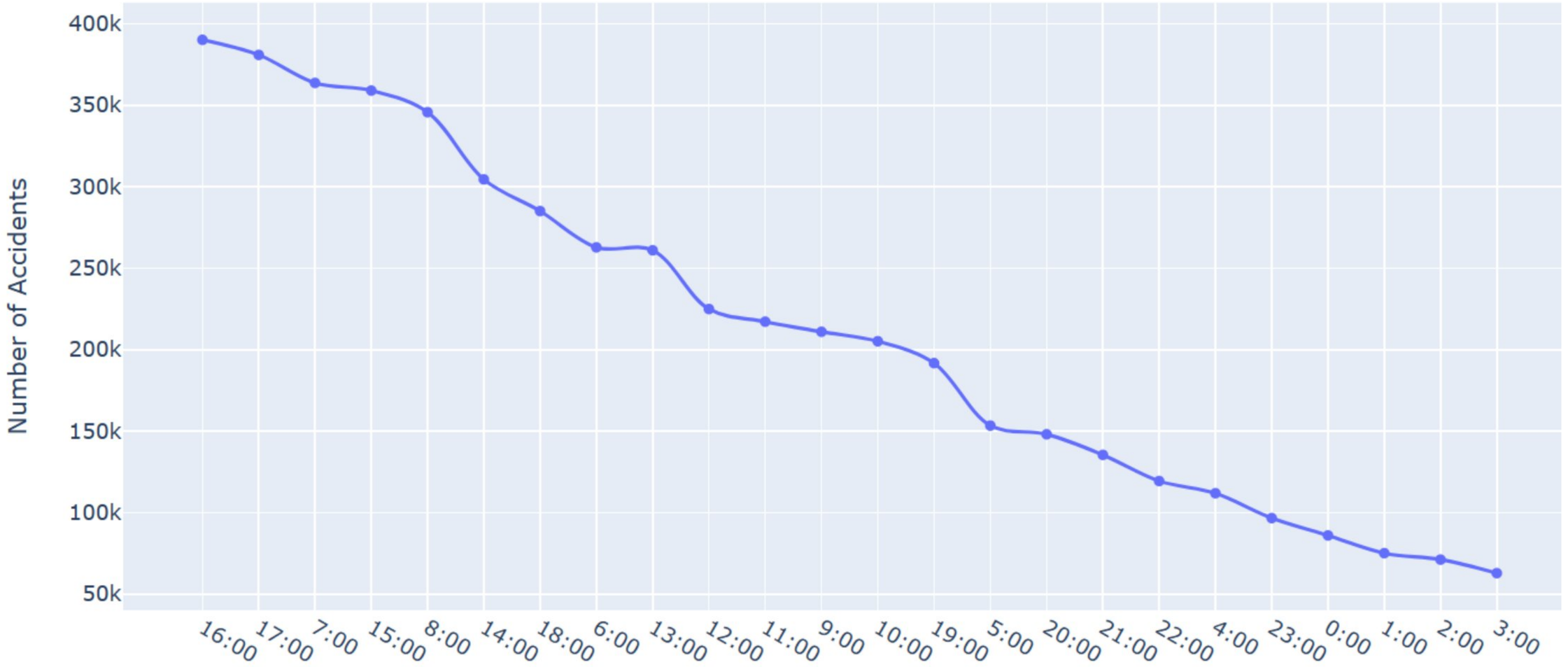


Analysis Data



Analysis Data

Distribution of accidents by hours



Analysis Data

"Observing that the severity of accidents isn't tied to any singular factor, but rather to a combination of various elements, it becomes apparent that the overall occurrence of accidents can be diminished through comprehensive measures. Implementing additional traffic signals across all areas, incorporating pedestrian alerts, removing road obstructions, and establishing safe pedestrian crossings are pivotal. The absence of these crucial elements significantly contributes to a high rate of accidents."

- Tighter enforcement of traffic laws: Effectively enforce and tighten traffic laws, and impose deterrent penalties on violators to improve compliance and road safety.

Thank you