Summary and Conclusion

Insights :

- No data from New York
- The number of accidents per city decreases exponentially
- Less than 5% of cities have more than 1000 in yearly accidents
- Over 1110 cities have reported just 1 accidents(need to investigate)
- A high percentage of accidents occurs between 12PM to 4PM.(Probably due to Miami Street Racing)
- There is some missing data in 2016 and 2017. So for proper analysis we can go for 2019 year.
- There is normal distribution in data of 2019 month but we can can see that there is slightly high number of accidents in winter season and this is due to conditions like poor visibility, snow- and ice-covered roads, and snow removal equipment, causing slowdowns or blocking travel.

↳ *21 cells hidden*

Ask and answer questions

1. Are there more accidents in warmer or colder areas?
2. Which 5 state has the highest number of accidents? How about per capita?
3. Does New York show up in the data? If yes, why is the count lower if this the most populated city.
4. Among the top 100 cities in number of accidents, which states do they belong to most frequently.
5. What time of the day are accidents most frequent in?
6. Which days of the week have the most accidents?
7. Which months have the most accidents?
8. What is the trend of accidents year over year?

[ ] ↳ *1 cell hidden*

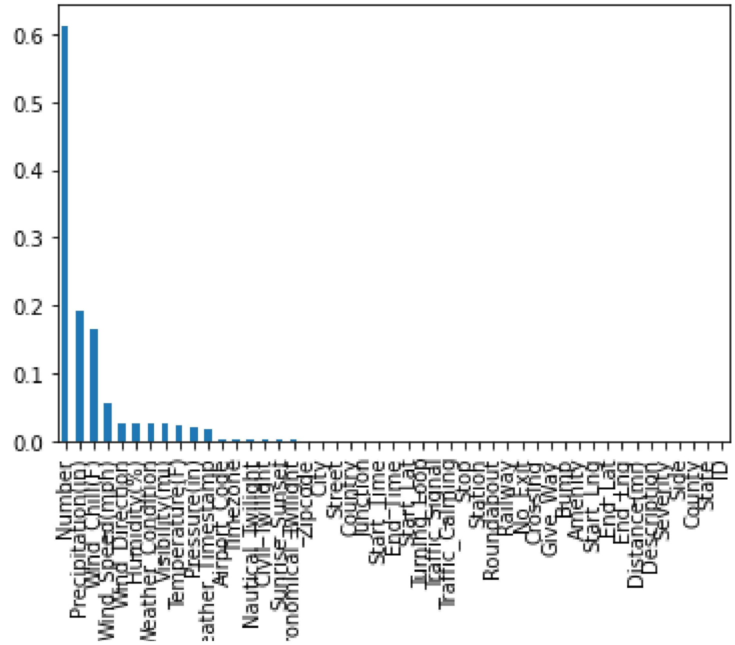
Percentage of missing value per column

```
missing_percentage = df.isna().sum().sort_values(ascending = False) / len(df)
missing_percentage
```

```
Number                   6.129003e-01
Precipitation(in)        1.931079e-01
Wind_Chill(F)            1.650568e-01
Wind_Speed(mph)          5.550967e-02
Wind_Direction           2.592834e-02
Humidity(%)              2.568830e-02
Weather_Condition        2.482514e-02
Visibility(mi)           2.479350e-02
Temperature(F)           2.434646e-02
Pressure(in)             2.080593e-02
Weather_Timestamp        1.783125e-02
Airport_Code             3.356011e-03
Timezone                 1.285961e-03
Nautical_Twilight        1.007612e-03
Civil_Twilight           1.007612e-03
Sunrise_Sunset           1.007612e-03
Astronomical_Twilight    1.007612e-03
Zipcode                  4.635647e-04
City                     4.814887e-05
Street                   7.029032e-07
Country                  0.000000e+00
Junction                 0.000000e+00
Start_Time               0.000000e+00
End_Time                 0.000000e+00
Start_Lat                0.000000e+00
Turning_Loop             0.000000e+00
Traffic_Signal           0.000000e+00
Traffic_Calming          0.000000e+00
Stop                     0.000000e+00
Station                  0.000000e+00
Roundabout               0.000000e+00
Railway                  0.000000e+00
No_Exit                  0.000000e+00
Crossing                 0.000000e+00
Give_Way                 0.000000e+00
Bump                     0.000000e+00
Amenity                  0.000000e+00
Start_Lng                0.000000e+00
End_Lat                  0.000000e+00
End_Lng                  0.000000e+00
Distance(mi)             0.000000e+00
Description              0.000000e+00
Severity                 0.000000e+00
Side                     0.000000e+00
County                   0.000000e+00
State                    0.000000e+00
ID                       0.000000e+00
dtype: float64
```

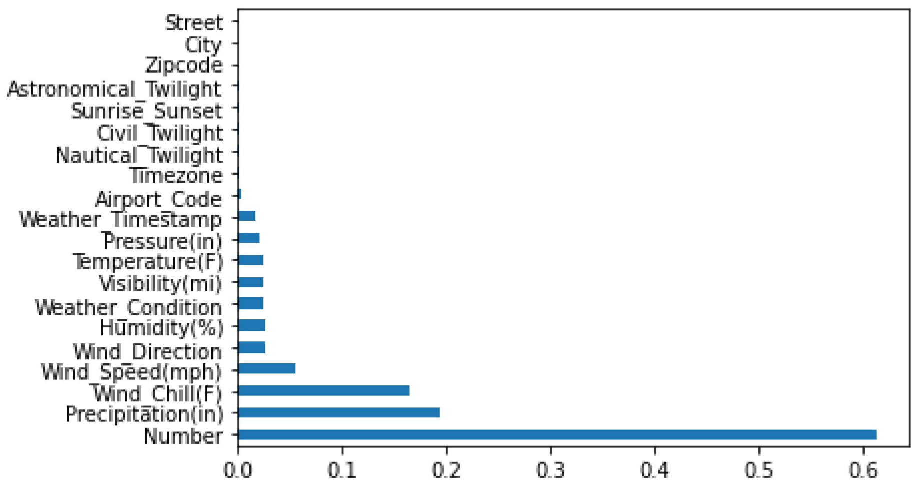
```
missing_percentage.plot(kind = 'bar')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7efd5c82bfd0>
```




```
missing_percentage[missing_percentage !=0].plot(kind = 'barh')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7efd77c65590>
```



Remove Columns that don't want to use

```
df.drop(['Number', 'Precipitation(in)'], axis =1)
```

	ID	Severity	Start_Time	End_Time	Start_Lat	Start_Lng	End_Lat
0	A-1	3	2016-02-08 00:37:08	2016-02-08 06:37:08	40.108910	-83.092860	40.112060
1	A-2	2	2016-02-08 05:56:20	2016-02-08 11:56:20	39.865420	-84.062800	39.865010
2	A-3	2	2016-02-08 06:15:39	2016-02-08 12:15:39	39.102660	-84.524680	39.102090
3	A-4	2	2016-02-08 06:51:45	2016-02-08 12:51:45	41.062130	-81.537840	41.062170
4	A-5	3	2016-02-08 07:53:43	2016-02-08 13:53:43	39.172393	-84.492792	39.170476
...	...	...	...	...	...	...	...
2845337	A-2845338	2	2019-08-23 18:03:25	2019-08-23 18:32:01	34.002480	-117.379360	33.998880
2845338	A-2845339	2	2019-08-23 19:11:30	2019-08-23 19:38:23	32.766960	-117.148060	32.765550
2845339	A-2845340	2	2019-08-23 19:00:21	2019-08-23 19:28:49	33.775450	-117.847790	33.777400
2845340	A-2845341	2	2019-08-23 19:00:21	2019-08-23 [illegible]	33.992460	-118.403020	33.983110

Exploratory Analysis and Visualization

Columns we will analyze:

1. City
2. Start Time
3. Start Lat, Start Lng
4. Temperature
5. Weather Condition

```
df.columns
```

```
Index(['ID', 'Severity', 'Start_Time', 'End_Time', 'Start_Lat', 'Start_Lng',
       'End_Lat', 'End_Lng', 'Distance(mi)', 'Description', 'Number', 'Street',
       'Side', 'City', 'County', 'State', 'Zipcode', 'Country', 'Timezone',
       'Airport_Code', 'Weather_Timestamp', 'Temperature(F)', 'Wind_Chill(F)',
       'Humidity(%)', 'Pressure(in)', 'Visibility(mi)', 'Wind_Direction',
       'Wind_Speed(mph)', 'Precipitation(in)', 'Weather_Condition', 'Amenity',
       'Bump', 'Crossing', 'Give_Way', 'Junction', 'No_Exit', 'Railway',
       'Roundabout', 'Station', 'Stop', 'Traffic_Calming', 'Traffic_Signal',
       'Turning_Loop', 'Sunrise_Sunset', 'Civil_Twilight', 'Nautical_Twilight',
       'Astronomical_Twilight'],
      dtype='object')
```

Cities

```
cities = df.City.unique()
len(cities)
```

```
11682
```

```
cities[:100]
```

```
array(['Dublin', 'Dayton', 'Cincinnati', 'Akron', 'Williamsburg',
       'Cleveland', 'Lima', 'Westerville', 'Jamestown', 'Freeport',
       'Columbus', 'Toledo', 'Roanoke', 'Ft Mitchell', 'Edinburgh',
       'Fairborn', 'Shelbyville', 'Greensburg', 'Saint Paul',
       'Parkersburg', 'Indianapolis', 'Dundee', 'Jeffersonville',
       'Pittsburgh', 'Lewis Center', 'Dunkirk', 'Redkey', 'Milton',
       'Willshire', 'Straughn', 'Cambridge Springs', 'Fremont',
       'Louisville', 'South Charleston', 'Edinboro', 'Buckhannon',
       'Lockbourne', 'Painesville', 'Washington', 'Dunbar', 'Angola',
       'Edon', 'Medina', 'De Mossville', 'New Albany', 'Charleston',
       'Fort Wayne', 'Burnsville', 'Bedford', 'Clarksville', 'Lakewood',
       'Richfield', 'Sewickley', 'Independence', 'Westlake', 'Erlanger',
       'Grove City', 'Monroe', 'West Middlesex', 'Gaston', 'Economy',
       'Fairmount', 'Hagerstown', 'Walton', 'Crittenden', 'Coraopolis',
       'Holland', 'Greenfield', 'Anderson', 'Englewood', 'Knightstown',
       'Bentleyville', 'Memphis', 'Henryville', 'Kendallville', 'Avilla',
       'Ohio City', 'Van Wert', 'Rocky River', 'Sturgis', 'West Chester',
       'Orient', 'Madison', 'Deputy', 'Keystone', 'Mercer', 'Bryant',
       'Pennville', 'Kimbolton', 'Thornville', 'Wexford', 'Fishers',
       'Noblesville', 'Macedonia', 'Youngstown', 'Fairdale', 'Sutton',
       'Mount Sterling', 'Northwood', 'Huntington'], dtype=object)
```

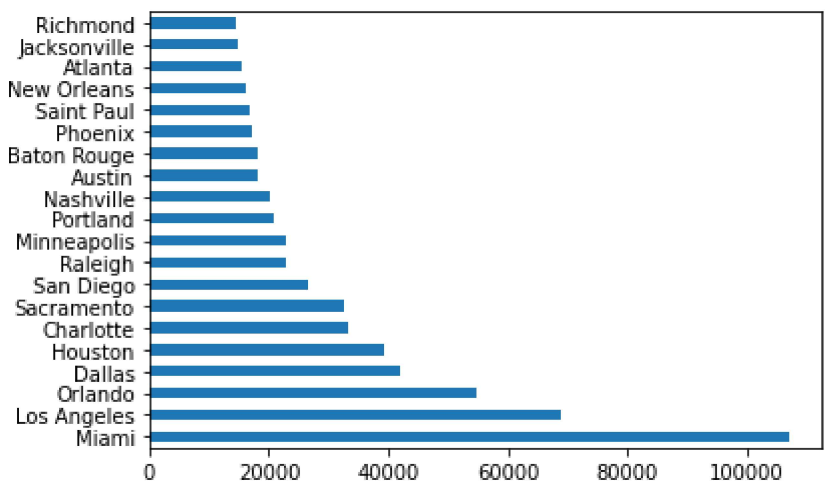
```
cities_by_accident = df.City.value_counts()
cities_by_accident
```

```
Miami                         106966
```

```
Los Angeles                   68956
Orlando                       54691
Dallas                        41979
Houston                       39448
                              ...
Ridgedale                         1
Sekiu                             1
Wooldridge                        1
Bullock                           1
American Fork-Pleasant Grove      1
Name: City, Length: 11681, dtype: int64
```

```
cities_by_accident[:20].plot(kind = 'barh')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7efd55839810>
```




```
cities_by_accident[:10]
```

```
Miami          106966
Los Angeles     68956
Orlando         54691
Dallas          41979
Houston         39448
Charlotte       33152
Sacramento      32559
San Diego       26627
Raleigh         22840
Minneapolis     22768
Name: City, dtype: int64
```

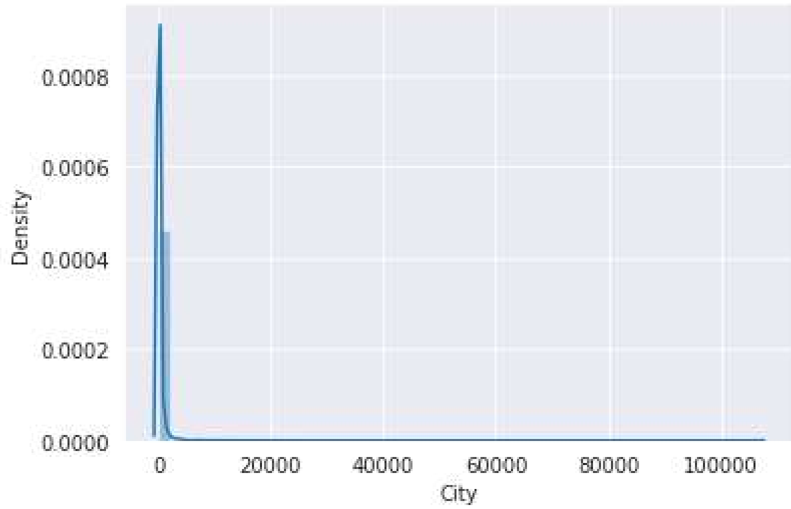
```
'New York' in df.City
```

```
False
```

```
import seaborn as sns
sns.set_style('darkgrid')
```

```
sns.distplot(cities_by_accident)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning:
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7efd46d53390>
```



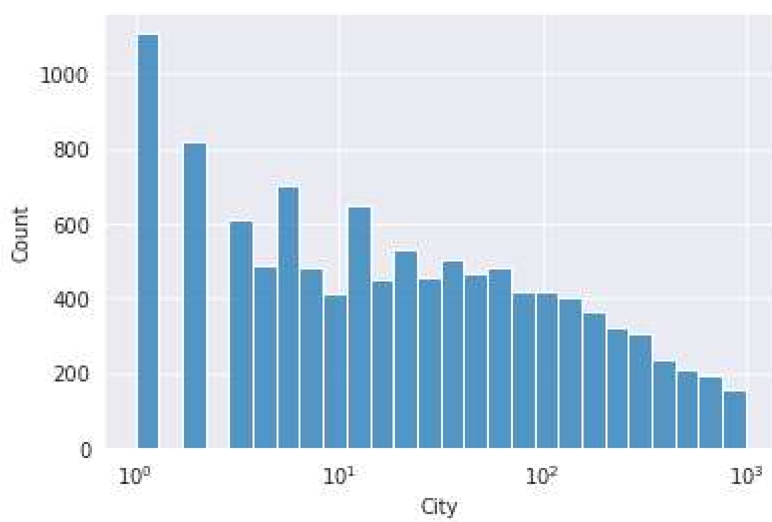

```
high_accident_cities = cities_by_accident[cities_by_accident >= 1000]
low_accident_cities = cities_by_accident[cities_by_accident < 1000]
```

```
(len(high_accident_cities) / len(cities)) * 100
```

```
4.245848313644924
```

```
sns.histplot(low_accident_cities, log_scale = True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7efd46d10590>
```




```
cities_by_accident[cities_by_accident == 1]
```

```
Carney                          1
Waverly Hall                    1
```

```
Center Sandwich               1
Glen Flora                    1
Sulphur Springs               1
                             ..
Ridgedale                     1
Sekiu                         1
Wooldridge                    1
Bullock                       1
American Fork-Pleasant Grove  1
Name: City, Length: 1110, dtype: int64
```

Start Time

```
df.Start_Time = pd.to_datetime(df.Start_Time)
```

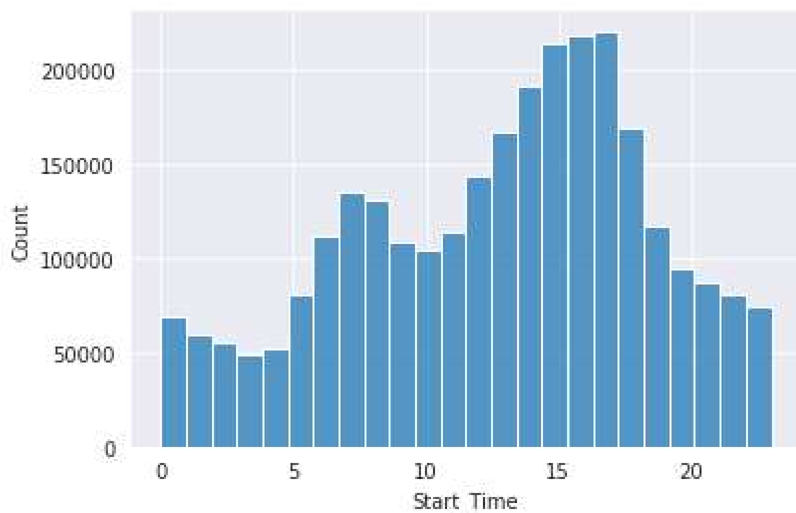
```
df.Start_Time[0]
```

```
Timestamp('2016-02-08 00:37:08')
```

- A high percentage of accidents occour between 2 PM to 5PM(Probably due to Street Racing in Miami as Race are starting from 1 PM)

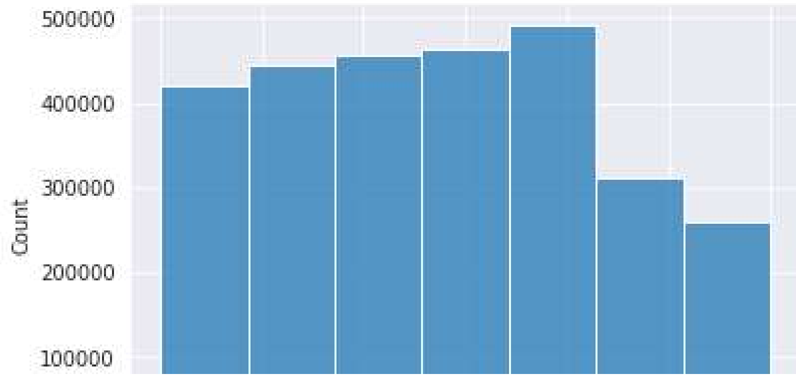
```
sns.histplot(df.Start_Time.dt.hour, bins=24)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7efd4220ad50>
```




```
sns.histplot(df.Start_Time.dt.dayofweek, bins=7)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7efd4210d3d0>



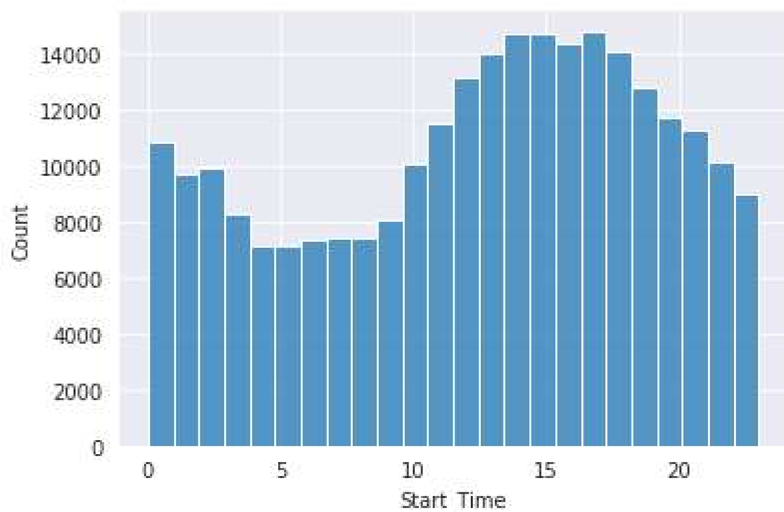
Is the distribution of accidents is same on weekends as compare to weekdays?

0 1 2 3 4 5 6

```
Sunday_Start_Time = df.Start_Time[df.Start_Time.dt.dayofweek == 6]
```

```
sns.histplot(Sunday_Start_Time.dt.hour, bins=24)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7efd42079510>




```
Monday_Start_Time = df.Start_Time[df.Start_Time.dt.dayofweek == 0]
```

```
sns.histplot(Monday_Start_Time.dt.hour, bins=24)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7efd42234e90>
```

Which month have high number of accidents?

```
month_2019 = df.Start_Time[df.Start_Time.dt.year ==2019]
```

```
sns.histplot(month_2019.dt.month, bins =12)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7efd41fb5b50>
```