

Road Traffic Accidents

About Dataset

Context

This data set is collected from Addis Ababa Sub city police departments for Masters research work.

Content ¶

The data set has been prepared from manual records of road traffic accident of the year 2017-20. All the sensitive information have been excluded during data encoding and finally it has 32 features and 12316 instances of the accident.

Acknowledgements

Bedane, Tarikwa Tesfa (2020), "Road Traffic Accident Dataset of Addis Ababa City", Mendeley Data, V1, doi: 10.17632/xytv86278f.1

```
In [1]: # import libraries
import pandas as pd
import numpy as np
```

```
In [2]: #import data

df = pd.read_csv(r'C:\Users\16476\Documents\RTA Dataset.csv')
```

In [3]: *# show the data*

```
df.head()
```

Out[3]:

| | Time | Day_of_week | Age_band_of_driver | Sex_of_driver | Educational_level | Vehicle_driver_relation | Driving_experience | Type_of_vehic |
|---|----------|-------------|--------------------|---------------|--------------------|-------------------------|--------------------|---------------|
| 0 | 17:02:00 | Monday | 18-30 | Male | Above high school | Employee | 1-2yr | Automot |
| 1 | 17:02:00 | Monday | 31-50 | Male | Junior high school | Employee | Above 10yr | Public (>se |
| 2 | 17:02:00 | Monday | 18-30 | Male | Junior high school | Employee | 1-2yr | Lorry (41?10 |
| 3 | 1:06:00 | Sunday | 18-30 | Male | Junior high school | Employee | 5-10yr | Public (>se |
| 4 | 1:06:00 | Sunday | 18-30 | Male | Junior high school | Employee | 2-5yr | N |

5 rows × 32 columns



In [4]: *# shape of the data*

```
df.shape
```

Out[4]: (12316, 32)

Data Cleaning

Checking the null values

```
In [5]: # null values
df.isna().sum()
```

```
Out[5]: Time                                0
Day_of_week                                0
Age_band_of_driver                         0
Sex_of_driver                             0
Educational_level                          741
Vehicle_driver_relation                    579
Driving_experience                         829
Type_of_vehicle                           950
Owner_of_vehicle                          482
Service_year_of_vehicle                   3928
Defect_of_vehicle                         4427
Area_accident_occured                     239
Lanes_or_Medians                          385
Road_allignment                           142
Types_of_Junction                         887
Road_surface_type                         172
Road_surface_conditions                   0
Light_conditions                          0
Weather_conditions                        0
Type_of_collision                         155
Number_of_vehicles_involved               0
Number_of_casualties                     0
Vehicle_movement                          308
Casualty_class                            0
Sex_of_casualty                           0
Age_band_of_casualty                      0
Casualty_severity                         0
Work_of_casualty                          3198
Fitness_of_casualty                       2635
Pedestrian_movement                       0
Cause_of_accident                         0
Accident_severity                         0
dtype: int64
```

```
In [6]: # percentage of null values
round((df.isna().sum()/df.shape[0])*100,2)
```

```
Out[6]: Time                                0.00
Day_of_week                               0.00
Age_band_of_driver                        0.00
Sex_of_driver                            0.00
Educational_level                         6.02
Vehicle_driver_relation                   4.70
Driving_experience                        6.73
Type_of_vehicle                          7.71
Owner_of_vehicle                         3.91
Service_year_of_vehicle                  31.89
Defect_of_vehicle                        35.95
Area_accident_occured                    1.94
Lanes_or_Medians                         3.13
Road_allignment                          1.15
Types_of_Junction                        7.20
Road_surface_type                        1.40
Road_surface_conditions                  0.00
Light_conditions                         0.00
Weather_conditions                       0.00
Type_of_collision                        1.26
Number_of_vehicles_involved              0.00
Number_of_casualties                     0.00
Vehicle_movement                         2.50
Casualty_class                           0.00
Sex_of_casualty                          0.00
Age_band_of_casualty                     0.00
Casualty_severity                        0.00
Work_of_casualty                         25.97
Fitness_of_casualty                      21.39
Pedestrian_movement                      0.00
Cause_of_accident                        0.00
Accident_severity                        0.00
dtype: float64
```

Deleting the columns (having missing values > 2000)

There are quite a number of missing values in the dataset. Lets try to find out a way to clean the dataset.

Let us delete the columns that have missing values more than 2000.

For example: These columns `Service_year_of_vehicle` - 3928 `Defect_of_vehicle` - 4427

`Work_of_casuality` - 3198 `Fitness_of_casuality` - 2635 have missing values. So let us drop these columns.

```
In [7]: df.drop(columns = ['Service_year_of_vehicle', 'Defect_of_vehicle', 'Work_of_casuality', 'Fitness_of_casualit
y'], axis = 1, inplace = True)
```

```
In [8]: # Let us check if the columns have been dropped or not
df.columns
```

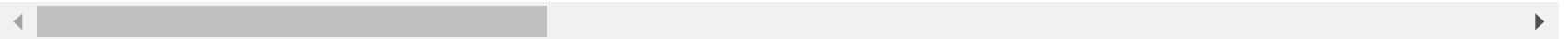
```
Out[8]: Index(['Time', 'Day_of_week', 'Age_band_of_driver', 'Sex_of_driver',
               'Educational_level', 'Vehicle_driver_relation', 'Driving_experience',
               'Type_of_vehicle', 'Owner_of_vehicle', 'Area_accident_occured',
               'Lanes_or_Medians', 'Road_allignment', 'Types_of_Junction',
               'Road_surface_type', 'Road_surface_conditions', 'Light_conditions',
               'Weather_conditions', 'Type_of_collision',
               'Number_of_vehicles_involved', 'Number_of_casualties',
               'Vehicle_movement', 'Casualty_class', 'Sex_of_casualty',
               'Age_band_of_casualty', 'Casualty_severity', 'Pedestrian_movement',
               'Cause_of_accident', 'Accident_severity'],
              dtype='object')
```

In [9]: `df.describe(include="all")`

Out[9]:

| | Time | Day_of_week | Age_band_of_driver | Sex_of_driver | Educational_level | Vehicle_driver_relation | Driving_experience | Type_o |
|---------------|----------|-------------|--------------------|---------------|--------------------|-------------------------|--------------------|--------|
| count | 12316 | 12316 | 12316 | 12316 | 11575 | 11737 | 11487 | |
| unique | 1074 | 7 | 5 | 3 | 7 | 4 | 7 | |
| top | 15:30:00 | Friday | 18-30 | Male | Junior high school | Employee | 5-10yr | Al |
| freq | 120 | 2041 | 4271 | 11437 | 7619 | 9627 | 3363 | |
| mean | NaN | NaN | NaN | NaN | NaN | NaN | NaN | |
| std | NaN | NaN | NaN | NaN | NaN | NaN | NaN | |
| min | NaN | NaN | NaN | NaN | NaN | NaN | NaN | |
| 25% | NaN | NaN | NaN | NaN | NaN | NaN | NaN | |
| 50% | NaN | NaN | NaN | NaN | NaN | NaN | NaN | |
| 75% | NaN | NaN | NaN | NaN | NaN | NaN | NaN | |
| max | NaN | NaN | NaN | NaN | NaN | NaN | NaN | |

11 rows × 28 columns



Listing the unique values and missing values for each column

In [10]: *#Let us study the unique values, missing values and the value counts of each variable*

```
for col in df:

    print('_'*50)
    print(f"Column: {col}")
    print('_'*50)
    print(df[col].value_counts())
    print("missing values:", df[col].isna().sum())
    print('\n')
```

Column: Time

| | |
|----------|-----|
| 15:30:00 | 120 |
| 17:10:00 | 110 |
| 18:30:00 | 103 |
| 11:30:00 | 99 |
| 17:00:00 | 98 |

...

| | |
|----------|---|
| 1:08:00 | 1 |
| 14:31:00 | 1 |
| 3:30:00 | 1 |
| 19:22:00 | 1 |
| 20:36:00 | 1 |

Name: Time, Length: 1074, dtype: int64

missing values: 0

Column: Day_of_week

| | |
|-----------|------|
| Friday | 2041 |
| Thursday | 1851 |
| Wednesday | 1840 |
| Tuesday | 1770 |
| Monday | 1681 |
| Saturday | 1666 |
| Sunday | 1467 |

Name: Day_of_week, dtype: int64

missing values: 0

Column: Age_band_of_driver

| | |
|----------|------|
| 18-30 | 4271 |
| 31-50 | 4087 |
| Over 51 | 1585 |
| Unknown | 1548 |
| Under 18 | 825 |

Name: Age_band_of_driver, dtype: int64

missing values: 0

Column: Sex_of_driver

| | |
|---------|-------|
| Male | 11437 |
| Female | 701 |
| Unknown | 178 |

Name: Sex_of_driver, dtype: int64
missing values: 0

Column: Educational_level

| | |
|--------------------|------|
| Junior high school | 7619 |
| Elementary school | 2163 |
| High school | 1110 |
| Above high school | 362 |
| Writing & reading | 176 |
| Unknown | 100 |
| Illiterate | 45 |

Name: Educational_level, dtype: int64
missing values: 741

Column: Vehicle_driver_relation

| | |
|----------|------|
| Employee | 9627 |
| Owner | 1973 |
| Other | 123 |
| Unknown | 14 |

Name: Vehicle_driver_relation, dtype: int64
missing values: 579

Column: Driving_experience

| | |
|------------|------|
| 5-10yr | 3363 |
| 2-5yr | 2613 |
| Above 10yr | 2262 |
| 1-2yr | 1756 |

Below 1yr 1342
 No Licence 118
 unknown 33
 Name: Driving_experience, dtype: int64
 missing values: 829

Column: Type_of_vehicle

| | |
|----------------------|------|
| Automobile | 3205 |
| Lorry (41?100Q) | 2186 |
| Other | 1208 |
| Pick up upto 10Q | 811 |
| Public (12 seats) | 711 |
| Stationwagen | 687 |
| Lorry (11?40Q) | 541 |
| Public (13?45 seats) | 532 |
| Public (> 45 seats) | 404 |
| Long lorry | 383 |
| Taxi | 265 |
| Motorcycle | 177 |
| Special vehicle | 84 |
| Ridden horse | 76 |
| Turbo | 46 |
| Bajaj | 29 |
| Bicycle | 21 |

Name: Type_of_vehicle, dtype: int64
 missing values: 950

Column: Owner_of_vehicle

| | |
|--------------|-------|
| Owner | 10459 |
| Governmental | 1041 |
| Organization | 312 |
| Other | 22 |

Name: Owner_of_vehicle, dtype: int64
 missing values: 482

Column: Area_accident_occured

| | |
|---------------------------------|------|
| Other | 3819 |
| Office areas | 3451 |
| Residential areas | 2060 |
| Church areas | 1060 |
| Industrial areas | 456 |
| School areas | 415 |
| Recreational areas | 327 |
| Outside rural areas | 218 |
| Hospital areas | 121 |
| Market areas | 63 |
| Rural village areas | 44 |
| Unknown | 22 |
| Rural village areasOffice areas | 20 |
| Recreational areas | 1 |

Name: Area_accident_occured, dtype: int64
missing values: 239

Column: Lanes_or_Medians

| | |
|--|------|
| Two-way (divided with broken lines road marking) | 4411 |
| Undivided Two way | 3796 |
| other | 1660 |
| Double carriageway (median) | 1020 |
| One way | 845 |
| Two-way (divided with solid lines road marking) | 142 |
| Unknown | 57 |

Name: Lanes_or_Medians, dtype: int64
missing values: 385

Column: Road_allignment

| | |
|---|-------|
| Tangent road with flat terrain | 10459 |
| Tangent road with mild grade and flat terrain | 501 |
| Steep grade downward with mountainous terrain | 429 |
| Tangent road with mountainous terrain and | 396 |
| Gentle horizontal curve | 163 |
| Escarpments | 113 |

| | |
|---|----|
| Sharp reverse curve | 57 |
| Tangent road with rolling terrain | 37 |
| Steep grade upward with mountainous terrain | 19 |

Name: Road_allignment, dtype: int64
missing values: 142

Column: Types_of_Junction

| | |
|-------------|------|
| Y Shape | 4543 |
| No junction | 3837 |
| Crossing | 2177 |
| Other | 445 |
| Unknown | 191 |
| O Shape | 164 |
| T Shape | 60 |
| X Shape | 12 |

Name: Types_of_Junction, dtype: int64
missing values: 887

Column: Road_surface_type

| | |
|----------------------------------|-------|
| Asphalt roads | 11296 |
| Earth roads | 358 |
| Gravel roads | 242 |
| Other | 167 |
| Asphalt roads with some distress | 81 |

Name: Road_surface_type, dtype: int64
missing values: 172

Column: Road_surface_conditions

| | |
|----------------------|------|
| Dry | 9340 |
| Wet or damp | 2904 |
| Snow | 70 |
| Flood over 3cm. deep | 2 |

Name: Road_surface_conditions, dtype: int64
missing values: 0

Column: Light_conditions

| | |
|-------------------------|------|
| Daylight | 8798 |
| Darkness - lights lit | 3286 |
| Darkness - no lighting | 192 |
| Darkness - lights unlit | 40 |

Name: Light_conditions, dtype: int64
missing values: 0

Column: Weather_conditions

| | |
|-------------------|-------|
| Normal | 10063 |
| Raining | 1331 |
| Other | 296 |
| Unknown | 292 |
| Cloudy | 125 |
| Windy | 98 |
| Snow | 61 |
| Raining and Windy | 40 |
| Fog or mist | 10 |

Name: Weather_conditions, dtype: int64
missing values: 0

Column: Type_of_collision

| | |
|---|------|
| Vehicle with vehicle collision | 8774 |
| Collision with roadside objects | 1786 |
| Collision with pedestrians | 896 |
| Rollover | 397 |
| Collision with animals | 171 |
| Collision with roadside-parked vehicles | 54 |
| Fall from vehicles | 34 |
| Other | 26 |
| Unknown | 14 |
| With Train | 9 |

Name: Type_of_collision, dtype: int64

missing values: 155

Column: Number_of_vehicles_involved

| | |
|---|------|
| 2 | 8340 |
| 1 | 1996 |
| 3 | 1568 |
| 4 | 363 |
| 6 | 42 |
| 7 | 7 |

Name: Number_of_vehicles_involved, dtype: int64

missing values: 0

Column: Number_of_casualties

| | |
|---|------|
| 1 | 8397 |
| 2 | 2290 |
| 3 | 909 |
| 4 | 394 |
| 5 | 207 |
| 6 | 89 |
| 7 | 22 |
| 8 | 8 |

Name: Number_of_casualties, dtype: int64

missing values: 0

Column: Vehicle_movement

| | |
|---------------------|------|
| Going straight | 8158 |
| Moving Backward | 985 |
| Other | 937 |
| Reversing | 563 |
| Turnover | 489 |
| Getting off | 339 |
| Entering a junction | 193 |
| Overtaking | 96 |
| Unknown | 88 |

```

Stopping          61
U-Turn            50
Waiting to go     39
Parked            10
Name: Vehicle_movement, dtype: int64
missing values: 308

```

Column: Casualty_class

```

Driver or rider   4944
na                4443
Pedestrian        1649
Passenger         1280
Name: Casualty_class, dtype: int64
missing values: 0

```

Column: Sex_of_casualty

```

Male             5253
na               4443
Female           2620
Name: Sex_of_casualty, dtype: int64
missing values: 0

```

Column: Age_band_of_casualty

```

na                4443
18-30             3145
31-50             2455
Under 18          1035
Over 51           994
5                 244
Name: Age_band_of_casualty, dtype: int64
missing values: 0

```

Column: Casualty_severity

3 7076

na 4443

2 771

1 26

Name: Casualty_severity, dtype: int64

missing values: 0

Column: Pedestrian_movement

Not a Pedestrian

11390

Crossing from nearside - masked by parked or stationery vehicle

337

Unknown or other

293

Crossing from driver's nearside

140

Crossing from offside - masked by parked or stationery vehicle

72

In carriageway, stationery - not crossing (standing or playing)

46

Walking along in carriageway, back to traffic

18

In carriageway, stationery - not crossing (standing or playing) - masked by parked or stationery vehicle

Walking along in carriageway, facing traffic

7

Name: Pedestrian_movement, dtype: int64

missing values: 0

Column: Cause_of_accident

No distancing 2263

Changing lane to the right 1808

Changing lane to the left 1473

Driving carelessly 1402

No priority to vehicle 1207

| | |
|--------------------------------------|------|
| Moving Backward | 1137 |
| No priority to pedestrian | 721 |
| Other | 456 |
| Overtaking | 430 |
| Driving under the influence of drugs | 340 |
| Driving to the left | 284 |
| Getting off the vehicle improperly | 197 |
| Driving at high speed | 174 |
| Overturning | 149 |
| Turnover | 78 |
| Overspeed | 61 |
| Overloading | 59 |
| Drunk driving | 27 |
| Improper parking | 25 |
| Unknown | 25 |

Name: Cause_of_accident, dtype: int64

missing values: 0

Column: Accident_severity

| | |
|----------------|-------|
| Slight Injury | 10415 |
| Serious Injury | 1743 |
| Fatal injury | 158 |

Name: Accident_severity, dtype: int64

missing values: 0

Dealing with Missing Values

As we know, the `Mean` of the numerical column data is used to replace null values when the data is normally distributed.

`Median` is used if the data comprised of outliers.

`Mode` is used when the data having more occurrences of a particular value or more frequent value.

Let us replace the nulls or missing values using mode i.e. with most frequent values in the cases where the proportion of top most frequent value of a variable is high comparable to other values.

Replacing missing values using mode

```
In [11]: cols = ('Educational_level', 'Vehicle_driver_relation',  
                'Type_of_vehicle', 'Owner_of_vehicle', 'Area_accident_occured', 'Road_allignment',  
                'Road_surface_type', 'Type_of_collision', 'Vehicle_movement')  
  
for cols in cols:  
    print(df[cols].mode())  
    df[cols] = df[cols].fillna(df[cols].mode()[0])  
    print("Null values in", df[cols].isna().sum())  
    print('/n')
```

```
0    Junior high school
dtype: object
Null values in 0
/n
0    Employee
dtype: object
Null values in 0
/n
0    Automobile
dtype: object
Null values in 0
/n
0    Owner
dtype: object
Null values in 0
/n
0    Other
dtype: object
Null values in 0
/n
0    Tangent road with flat terrain
dtype: object
Null values in 0
/n
0    Asphalt roads
dtype: object
Null values in 0
/n
0    Vehicle with vehicle collision
dtype: object
Null values in 0
/n
0    Going straight
dtype: object
Null values in 0
/n
```

```
In [12]: # validate
df.isna().sum()
```

```
Out[12]: Time                                0
Day_of_week                                0
Age_band_of_driver                         0
Sex_of_driver                             0
Educational_level                         0
Vehicle_driver_relation                    0
Driving_experience                         829
Type_of_vehicle                           0
Owner_of_vehicle                          0
Area_accident_occured                     0
Lanes_or_Medians                          385
Road_allignment                           0
Types_of_Junction                         887
Road_surface_type                         0
Road_surface_conditions                   0
Light_conditions                         0
Weather_conditions                       0
Type_of_collision                        0
Number_of_vehicles_involved               0
Number_of_casualties                     0
Vehicle_movement                         0
Casualty_class                           0
Sex_of_casualty                          0
Age_band_of_casualty                     0
Casualty_severity                        0
Pedestrian_movement                     0
Cause_of_accident                        0
Accident_severity                        0
dtype: int64
```

So in columns like `Driving_experience`, `Lanes_or_Medians` and `Types_of_Junction`, the above method i.e. replacing the missing values with mode could be done, however, in these columns the proportion of the top most frequent value of the variable is in proportion is not very high from second top value,

eg: In case of `Driving_experience`

Name: `Driving_experience`, dtype: object

5-10yr - 3363

2-5yr - 2613

Above 10yr - 2262

1-2yr - 1756

Below 1yr - 1342

No Licence - 118

unknown - 33

missing values: 829

we can see that there is not a huge difference in the no of times values `5-10yr` and `2-5yr` are repeated, , so if we would replace all 829 missing values with `5-10yr`, then it would just increase the difference, thus would eventually deviate our data. So a better way of dealing this would be either just removing the missing values all together or just replacing it with `Unknown`

```
In [13]: # replacing missing values with `Unknown`

cols = ['Driving_experience', 'Lanes_or_Medians', 'Types_of_Junction']

for col in cols:
    df[col] = df[col].fillna('Unknown')
```

```
In [14]: # validate
df.isna().sum()
```

```
Out[14]: Time                                0
Day_of_week                                0
Age_band_of_driver                         0
Sex_of_driver                             0
Educational_level                         0
Vehicle_driver_relation                   0
Driving_experience                        0
Type_of_vehicle                          0
Owner_of_vehicle                         0
Area_accident_occured                    0
Lanes_or_Medians                        0
Road_allignment                          0
Types_of_Junction                       0
Road_surface_type                       0
Road_surface_conditions                  0
Light_conditions                        0
Weather_conditions                      0
Type_of_collision                       0
Number_of_vehicles_involved              0
Number_of_casualties                    0
Vehicle_movement                        0
Casualty_class                          0
Sex_of_casualty                         0
Age_band_of_casualty                    0
Casualty_severity                       0
Pedestrian_movement                     0
Cause_of_accident                       0
Accident_severity                       0
dtype: int64
```

Checking for duplicates

```
In [15]: df.duplicated().sum()
```

```
Out[15]: 0
```

Age_band_of_casualty

```
In [16]: df['Age_band_of_casualty'].value_counts()
```

```
Out[16]: na          4443
18-30         3145
31-50         2455
Under 18      1035
Over 51        994
5             244
Name: Age_band_of_casualty, dtype: int64
```

We can see that there are some discrepancies in this column, we can replace 5 with Under 18 and na with Unknown

```
In [17]: df['Age_band_of_casualty'] = df['Age_band_of_casualty'].replace('5', 'Under 18').replace('na', 'Unknown')
# validate
df['Age_band_of_casualty'].value_counts()
```

```
Out[17]: Unknown      4443
18-30         3145
31-50         2455
Under 18      1279
Over 51        994
Name: Age_band_of_casualty, dtype: int64
```

Area_accident_occured

There are many discrepancies in this column Area_accident_occured

There are values like Rural village areasOffice areas which seems like it was misentered.

So we can replace Rural village areasOffice areas this with just Rural village areas .

We could have kept the column Rural village areasOffice areas as it is, as well, however, I chose to replace it with Rural village areas

Also remove the extra spacing in front of the values.

```
In [18]: df['Area_accident_occured'] = df['Area_accident_occured'].replace('Rural village areasOffice areas', 'Rural village areas')
```

```
In [19]: df['Area_accident_occured'] = df['Area_accident_occured'].replace(' Recreational areas', 'Recreational areas')
df['Area_accident_occured'] = df['Area_accident_occured'].replace(' Church areas', 'Church areas')
df['Area_accident_occured'] = df['Area_accident_occured'].replace(' Industrial areas', 'Industrial areas')
df['Area_accident_occured'] = df['Area_accident_occured'].replace(' Outside rural areas', 'Outside rural areas')
df['Area_accident_occured'] = df['Area_accident_occured'].replace(' Hospital areas', 'Hospital areas')
df['Area_accident_occured'] = df['Area_accident_occured'].replace(' Market areas', 'Market areas')
```



```
In [20]: #validate
df['Area_accident_occured'].value_counts()
```

```
Out[20]: Other                4058
Office areas                3451
Residential areas          2060
Church areas               1060
Industrial areas           456
School areas               415
Recreational areas        328
Outside rural areas        218
Hospital areas             121
Rural village areas         64
Market areas                63
Unknown                    22
Name: Area_accident_occured, dtype: int64
```

Type_of_vehicle

Putting all the values that contains Lorry (41?100Q) , Lorry (11?40Q) , Long Lorry and Pick up upto 10Q into one category of Lorry

also Public (12 seats) , Public (13?45 seats) and Public (> 45 seats) into Public

```
In [21]: # validate
df['Type_of_vehicle'].value_counts()
```

```
Out[21]: Automobile          4155
Lorry (41?100Q)             2186
Other                       1208
Pick up upto 10Q           811
Public (12 seats)          711
Stationwagen               687
Lorry (11?40Q)             541
Public (13?45 seats)       532
Public (> 45 seats)        404
Long lorry                 383
Taxi                       265
Motorcycle                 177
Special vehicle            84
Ridden horse              76
Turbo                      46
Bajaj                      29
Bicycle                    21
Name: Type_of_vehicle, dtype: int64
```

```
In [22]: df['Type_of_vehicle'] = df['Type_of_vehicle'].replace({'Lorry (41?100Q)': 'Lorry',
                                                                'Lorry (11?40Q)': 'Lorry',
                                                                'Long lorry': 'Lorry',
                                                                'Pick up upto 10Q': 'Lorry',
                                                                'Public (12 seats)': 'Public',
                                                                'Public (13?45 seats)': 'Public',
                                                                'Public (> 45 seats)': 'Public'
                                                                })
```

```
In [23]: #validate
df['Type_of_vehicle'].value_counts()
```

```
Out[23]: Automobile      4155
Lorry                  3921
Public                 1647
Other                  1208
Stationwagen           687
Taxi                   265
Motorcycle             177
Special vehicle         84
Ridden horse           76
Turbo                   46
Bajaj                   29
Bicycle                21
Name: Type_of_vehicle, dtype: int64
```

Driving_experience

Replacing unknown with Unknown

```
In [24]: df['Driving_experience'].value_counts()
```

```
Out[24]: 5-10yr      3363
2-5yr      2613
Above 10yr  2262
1-2yr      1756
Below 1yr   1342
Unknown     829
No Licence  118
unknown     33
Name: Driving_experience, dtype: int64
```

```
In [25]: df['Driving_experience']=df['Driving_experience'].replace('unknown','Unknown')
```

```
In [26]: df['Driving_experience'].value_counts()
```

```
Out[26]: 5-10yr      3363
         2-5yr      2613
         Above 10yr  2262
         1-2yr      1756
         Below 1yr   1342
         Unknown     862
         No Licence  118
         Name: Driving_experience, dtype: int64
```

Casualty_severity

Replacing na with Unknown

```
In [27]: df['Casualty_severity'].value_counts()
```

```
Out[27]: 3      7076
         na     4443
         2      771
         1       26
         Name: Casualty_severity, dtype: int64
```

```
In [28]: df['Casualty_severity'] = df['Casualty_severity'].replace('na', 'Unknown')
         df['Casualty_severity'].value_counts()
```

```
Out[28]: 3      7076
         Unknown  4443
         2      771
         1       26
         Name: Casualty_severity, dtype: int64
```

Time

Converting Time into Categorical Variable containing time buckets like

Early Morning , Morning , Afternoon , Evening and Night

```
In [29]: df['Time'].value_counts()
```

```
Out[29]: 15:30:00    120
         17:10:00    110
         18:30:00    103
         11:30:00     99
         17:00:00     98
         ...
         1:08:00      1
         14:31:00     1
         3:30:00      1
         19:22:00     1
         20:36:00     1
         Name: Time, Length: 1074, dtype: int64
```

```
In [30]: # converting into datetime format to extract hour of the day
         df['Time'] = pd.to_datetime(df['Time'])
```

```
In [31]: # Extracting the hour of the day
         df["hour"] = df['Time'].dt.hour
```

In [32]: `df['hour']`

Out[32]:

| | |
|-------|----|
| 0 | 17 |
| 1 | 17 |
| 2 | 17 |
| 3 | 1 |
| 4 | 1 |
| | .. |
| 12311 | 16 |
| 12312 | 18 |
| 12313 | 13 |
| 12314 | 13 |
| 12315 | 13 |

Name: hour, Length: 12316, dtype: int64

In [33]:

```
b = [0,4,8,12,16,20,24]
l = ['Late Night', 'Early Morning', 'Morning', 'Afternoon', 'Evening', 'Night']
df['Time_of_accident'] = pd.cut(df['hour'], bins=b, labels=l, include_lowest=True)
```

In [34]: `df['Time_of_accident'].value_counts()`

Out[34]:

| | |
|---------------|------|
| Evening | 3496 |
| Afternoon | 3206 |
| Morning | 2353 |
| Early Morning | 1650 |
| Night | 1012 |
| Late Night | 599 |

Name: Time_of_accident, dtype: int64

In [35]: *# Let us export this clean dataframe into a csv file*

```
#df.to_csv(r'C:\Users\16476\Downloads\RTA Dataset1.csv')
```

In []: