**Capstone Project Data Analyst: Road Accident Analysis**

**Project description:** India has the highest number of road fatalities in the world. In 2016, for which global figures are available, India accounted for more than a third of global road accident deaths. The World Health Organization says such deaths are under-reported and estimated that in 2016, the figure for India was likely twice as big as that reported by the government.

In this project, perform a detailed statistical data analysis on Road Accidents of India over a period of 2003 to 2016 years using various Python Libraries.

With your analysis, you must answer to the below questions either in a statistical analysis or in a graphical representation using python code snippets.

import pandas as pd  
import matplotlib.pyplot as plt  
  
df = pd.read\_excel('roadAccStats13-16.xlsx') # Use pd.read\_excel to read excel files  
df1 = pd.read\_csv('Details\_of\_road\_accident\_deaths\_by\_situation\_state\_2014.csv')  
df2 = pd.read\_csv('Persons\_killed\_due\_to\_Non-use\_of\_Safety\_Device\_2016.csv')  
df3 = pd.read\_csv('accidentRate.csv')  
df4 = pd.read\_csv('laneAccidents.csv')  
df5 = pd.read\_csv('reasonOfAccident.csv')  
df6 = pd.read\_csv('typeOfVehicle.csv')  
df7 = pd.read\_excel('timeOfOccurence.xls')  
df8 = pd.read\_excel('location.xls')  
df9 = pd.read\_excel('accidents03-16.xls')

**1. The percentage of road accidents during all the years.**

# Load the dataset (Assuming df is the DataFrame)  
df = pd.read\_excel('roadAccStats13-16.xlsx') # Use pd.read\_excel to read excel files  
  
# Calculate the total accidents for each year  
total\_accidents\_per\_year = {  
    '2013': df['State/UT-Wise Total Number of Road Accidents during - 2013'].sum(),  
    '2014': df['State/UT-Wise Total Number of Road Accidents during - 2014'].sum(),  
    '2015': df['State/UT-Wise Total Number of Road Accidents during - 2015'].sum(),  
    '2016': df['State/UT-Wise Total Number of Road Accidents during - 2016'].sum(),  
}  
  
# Calculate the percentage of accidents for each state  
for year in total\_accidents\_per\_year:  
    column\_name = f'Percentage of Accidents - {year}'  
    df[column\_name] = (df[f'State/UT-Wise Total Number of Road Accidents during - {year}'] / total\_accidents\_per\_year[year]) \* 100  
  
# Display the results  
print(df[[ 'States/UTs'] + [f'Percentage of Accidents - {year}' for year in total\_accidents\_per\_year]])

**Output:**

States/UTs Percentage of Accidents - 2013 \

0 Andhra Pradesh 4.469080

1 Arunachal Pradesh 0.031656

2 Assam 0.741147

3 Bihar 1.048356

4 Chhattisgarh 1.403666

5 Goa 0.441337

6 Gujarat 2.609687

7 Haryana 1.077340

8 Himachal Pradesh 0.306387

9 Jammu & Kashmir 0.663650

10 Jharkhand 0.572382

11 Karnataka 4.524375

12 Kerala 3.619397

13 Madhya Pradesh 5.325031

14 Maharashtra 6.477092

15 Manipur 0.068965

16 Meghalaya 0.053959

17 Mizoram 0.011717

18 Nagaland 0.007297

19 Odisha 0.994910

20 Punjab 0.649878

21 Rajasthan 2.424786

22 Sikkim 0.025078

23 Tamil Nadu 6.807941

24 Telangana NaN

25 Tripura 0.084074

26 Uttarakhand 0.133306

27 Uttar Pradesh 3.146609

28 West Bengal 1.275911

29 Andaman & Nicobar Islands 0.020556

30 Chandigarh 0.042140

31 Dadra & Nagar Haveli 0.009353

32 Daman & Diu 0.006064

33 Delhi 0.777633

34 Lakshadweep 0.000103

35 Puducherry 0.149134

36 Total 50.000000

Percentage of Accidents - 2014 Percentage of Accidents - 2015 \

0 2.496935 2.418916

1 0.020944 0.028319

2 0.729873 0.693925

3 0.976298 0.952788

4 1.412035 1.440500

5 0.432060 0.432569

6 2.422558 2.311721

7 1.090723 1.114229

8 0.312423 0.300146

9 0.598794 0.581944

10 0.531365 0.514735

11 4.465979 4.388610

12 3.706784 3.890328

13 5.463016 5.479106

14 6.296179 6.362393

15 0.075909 0.066910

16 0.055374 0.060428

17 0.013486 0.006980

18 0.031161 0.005385

19 0.985697 1.051208

20 0.652942 0.668298

21 2.516142 2.400369

22 0.020740 0.021838

23 6.870658 6.886302

24 2.051287 2.119169

25 0.073151 0.064516

26 0.144054 0.151868

27 3.170617 3.229309

28 1.315386 1.317052

29 0.022272 0.025727

30 0.037699 0.041482

31 0.008888 0.006880

32 0.003984 0.006980

33 0.880977 0.806206

34 0.000102 0.000299

35 0.113506 0.152566

36 50.000000 50.000000

Percentage of Accidents - 2016

0 2.588983

1 0.025902

2 0.773429

3 0.855297

4 1.412664

5 0.447725

6 2.273890

7 1.168621

8 0.329552

9 0.572244

10 0.513053

11 4.619038

12 4.100680

13 5.614457

14 4.148324

15 0.055966

16 0.064496

17 0.008634

18 0.007802

19 1.095595

20 0.723184

21 2.399449

22 0.021845

23 7.430636

24 2.372923

25 0.057942

26 0.165504

27 3.704551

28 1.412664

29 0.024758

30 0.044523

31 0.007282

32 0.007386

33 0.767187

34 0.000104

35 0.183709

36 50.000000

**2. Mean Accidents per 1L population for each year.**

# Load the dataset (Assuming df is the DataFrame)

df = pd.read\_excel('roadAccStats13-16.xlsx') # Use pd.read\_excel to read excel files

# Calculate the mean accidents per lakh population for each year

mean\_accidents\_per\_lakh = pd.DataFrame()

mean\_accidents\_per\_lakh['Year'] = ['2013', '2014', '2015', '2016']

mean\_accidents\_per\_lakh['Mean Accidents Per Lakh - 2013'] = df['Total Number of Accidents Per Lakh Population - 2013'].mean()

mean\_accidents\_per\_lakh['Mean Accidents Per Lakh - 2014'] = df['Total Number of Accidents Per Lakh Population - 2014'].mean()

mean\_accidents\_per\_lakh['Mean Accidents Per Lakh - 2015'] = df['Total Number of Accidents Per Lakh Population - 2015'].mean()

mean\_accidents\_per\_lakh['Mean Accidents Per Lakh - 2016'] = df['Total Number of Accidents Per Lakh Population - 2016'].mean()

# Display the resulting DataFrame

print(mean\_accidents\_per\_lakh)

**Output:**

Year Mean Accidents Per Lakh - 2013 Mean Accidents Per Lakh - 2014 \

0 2013 41.911111 39.877778

1 2014 41.911111 39.877778

2 2015 41.911111 39.877778

3 2016 41.911111 39.877778

Mean Accidents Per Lakh - 2015 Mean Accidents Per Lakh - 2016

0 40.833333 39.636111

1 40.833333 39.636111

2 40.833333 39.636111

3 40.833333 39.636111

**3. The highest number of accident states and least number of accident states.**

# Identify the year with the maximum accidents for each state  
df['Total Accidents'] = df[['State/UT-Wise Total Number of Road Accidents during - 2013',  
                             'State/UT-Wise Total Number of Road Accidents during - 2014',  
                             'State/UT-Wise Total Number of Road Accidents during - 2015',  
                             'State/UT-Wise Total Number of Road Accidents during - 2016']].sum(axis=1)  
  
# State with highest number of accidents  
highest\_accidents\_state = df.loc[df['Total Accidents'].idxmax()]  
  
# State with least number of accidents  
least\_accidents\_state = df.loc[df['Total Accidents'].idxmin()]  
  
# Display the results  
print("State with Highest Number of Accidents:")  
print(highest\_accidents\_state[['States/UTs', 'Total Accidents']])  
  
print("\nState with Least Number of Accidents:")  
print(least\_accidents\_state[['States/UTs', 'Total Accidents']])

**Output:**

State with Highest Number of Accidents:

States/UTs Total

Total Accidents 1957951.0

Name: 36, dtype: object

State with Least Number of Accidents:

States/UTs Lakshadweep

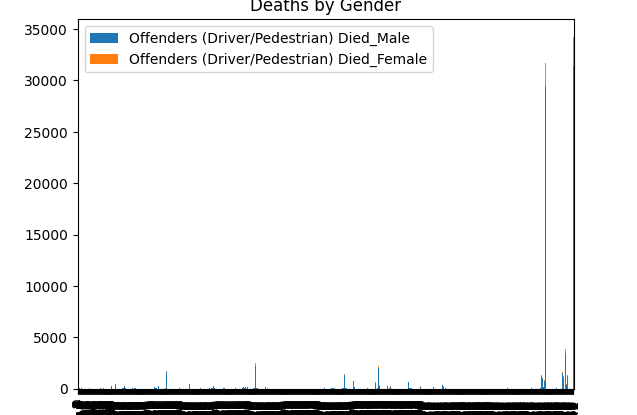
Total Accidents 6.0

Name: 34, dtype: object

**4. Offenders and victims who died according to gender as well the as the total deaths.**

import pandas as pd  
import matplotlib.pyplot as plt  
  
# Load the dataframe in this cell as well  
df1 = pd.read\_csv('Details\_of\_road\_accident\_deaths\_by\_situation\_state\_2014.csv')  
  
# Check for typos and correct the column names  
total\_deaths = df1[['Offenders (Driver/Pedestrian) Died\_Male', 'Offenders (Driver/Pedestrian) Died\_Female']].sum()  
  
df1[['Offenders (Driver/Pedestrian) Died\_Male', 'Offenders (Driver/Pedestrian) Died\_Female']].plot(kind='bar', stacked=True)  
plt.title('Deaths by Gender')  
plt.show()  
  
# To diagnose the issue further, print the column names  
# print(df1.columns)

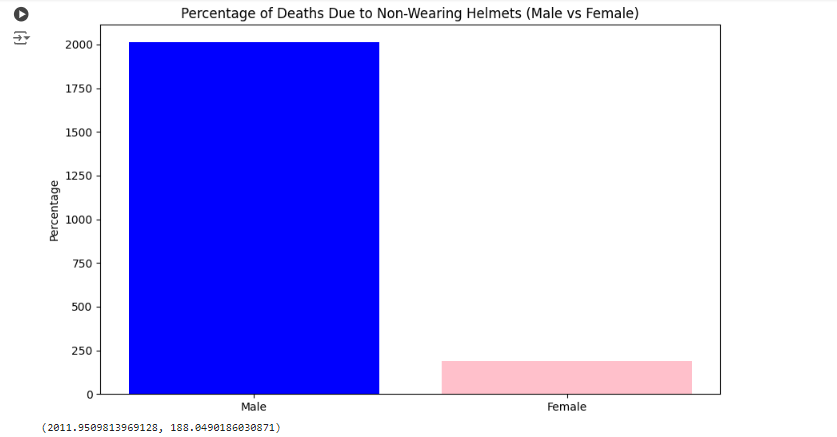
**Output:**

****

**5. Percentage of Deaths occurring due to non-wearing of helmets between male and female.**

import pandas as pd  
import matplotlib.pyplot as plt  
  
# Assuming df is the DataFrame with the necessary columns  
# Load the data (use your actual path or already loaded DataFrame)  
df2 = pd.read\_csv('Persons\_killed\_due\_to\_Non-use\_of\_Safety\_Device\_2016.csv')  
  
# Calculating the percentage of deaths  
df2['Male\_Death\_Percentage'] = (df2['Non-wearing of Helmet - Male'] / df2['Non-wearing of Helmet - Total']) \* 100  
df2['Female\_Death\_Percentage'] = (df2['Non-wearing of Helmet - Female'] / df2['Non-wearing of Helmet - Total']) \* 100  
  
# Plotting the results  
plt.figure(figsize=(10, 6))  
plt.bar(['Male', 'Female'], [df2['Male\_Death\_Percentage'].sum(), df2['Female\_Death\_Percentage'].sum()], color=['blue', 'pink'])  
plt.title('Percentage of Deaths Due to Non-Wearing Helmets (Male vs Female)')  
plt.ylabel('Percentage')  
plt.show()  
  
# Displaying the result  
male\_percentage = df2['Male\_Death\_Percentage'].sum()  
female\_percentage = df2['Female\_Death\_Percentage'].sum()  
  
male\_percentage, female\_percentage

**Output:**

****

**6. The number of accidents happening per state from the year 2003 to 2016.**

df9 = pd.read\_excel('accidents03-16.xls')

# Get the actual column names from the DataFrame

actual\_columns = df9.columns.tolist()

# Print the actual column names to inspect them

print(actual\_columns)

columns\_years = ['States/Uts', 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016]  # Assuming these are your desired columns

# Filter relevant columns using double brackets

df\_accidents = df9[columns\_years] # Changed to double brackets to select multiple columns

# Summing accidents from 2003 to 2016 for each state

df\_accidents['Total Accidents (2003-2016)'] = df\_accidents.iloc[:, 1:].sum(axis=1)

# Sorting by total accidents for a clearer visualization

df\_accidents\_sorted = df\_accidents.sort\_values(by='Total Accidents (2003-2016)', ascending=False)

# Plotting the total number of accidents per state from 2003 to 2016

plt.figure(figsize=(12, 8))

plt.bar(df\_accidents\_sorted['States/Uts'], df\_accidents\_sorted['Total Accidents (2003-2016)'], color='blue')

plt.title('Total Number of Road Accidents per State (2003-2016)')

plt.xlabel('States/UTs')

plt.ylabel('Total Accidents')

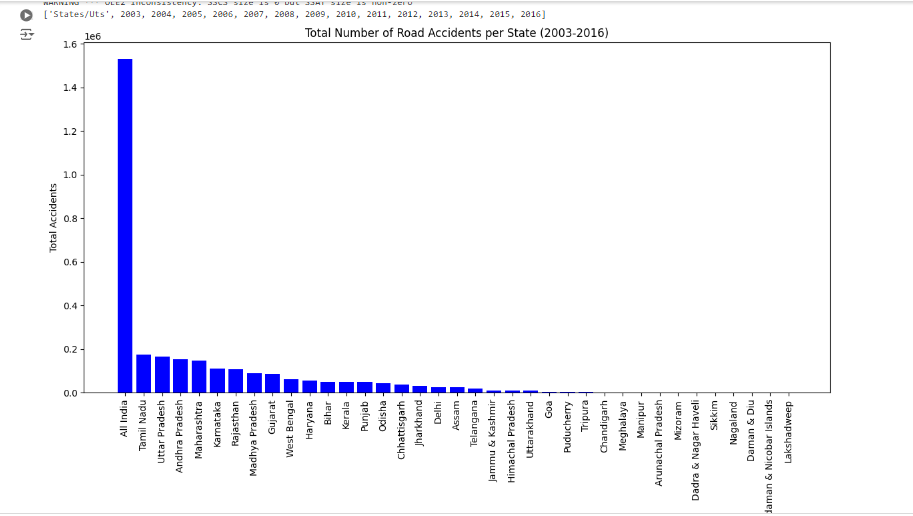
plt.xticks(rotation=90)

plt.tight\_layout()

# Display the plot

plt.show()

**Output:**



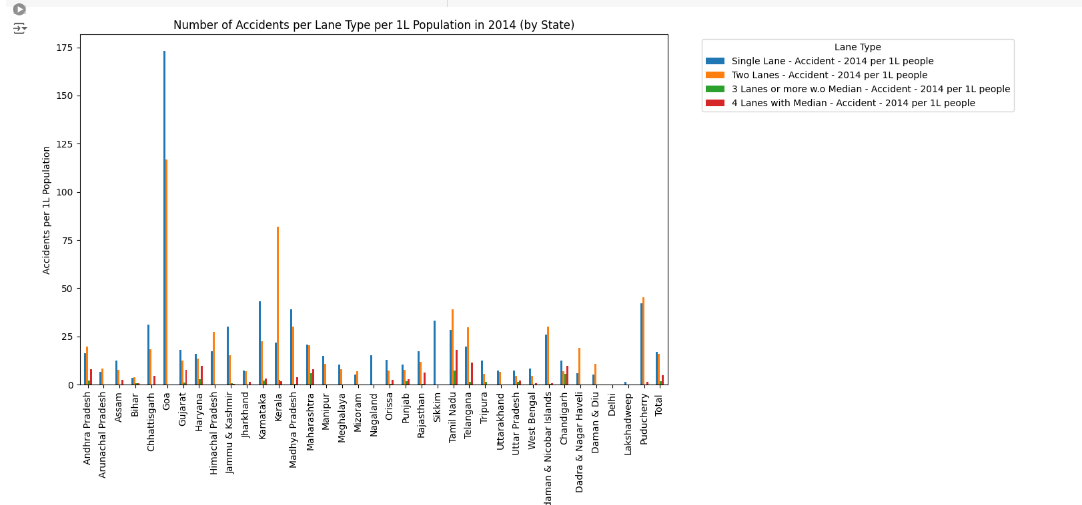
**7. Number of ACCIDENTS for 1,2,3,4 LANE per 1L population of resp. state.**

import pandas as pd  
import matplotlib.pyplot as plt  
  
# Load the dataset (Assuming df is already the DataFrame with relevant columns)  
df4 = pd.read\_csv('laneAccidents.csv')  
  
# Relevant columns: Accidents per 1L for each lane type  
columns = [  
    'Single Lane - Accident - 2014 per 1L people',  
    'Two Lanes - Accident - 2014 per 1L people',  
    '3 Lanes or more w.o Median - Accident - 2014 per 1L people',  
    '4 Lanes with Median - Accident - 2014 per 1L people'  
]  
  
# Create a new DataFrame with State/UT and Accident columns per lane type  
lane\_accidents = df4[['State/UT'] + columns]  
  
# Set State/UT as index for better readability in plot  
lane\_accidents.set\_index('State/UT', inplace=True)  
  
# Plotting the number of accidents per lane type per state  
lane\_accidents.plot(kind='bar', figsize=(15, 8))  
  
plt.title('Number of Accidents per Lane Type per 1L Population in 2014 (by State)')  
plt.xlabel('State/UT')  
plt.ylabel('Accidents per 1L Population')  
plt.xticks(rotation=90)  
plt.legend(title='Lane Type', bbox\_to\_anchor=(1.05, 1), loc='upper left')  
plt.tight\_layout()  
plt.show()  
  
# Displaying the DataFrame for verification  
lane\_accidents.head()

**Output:**

**A screenshot of a computer

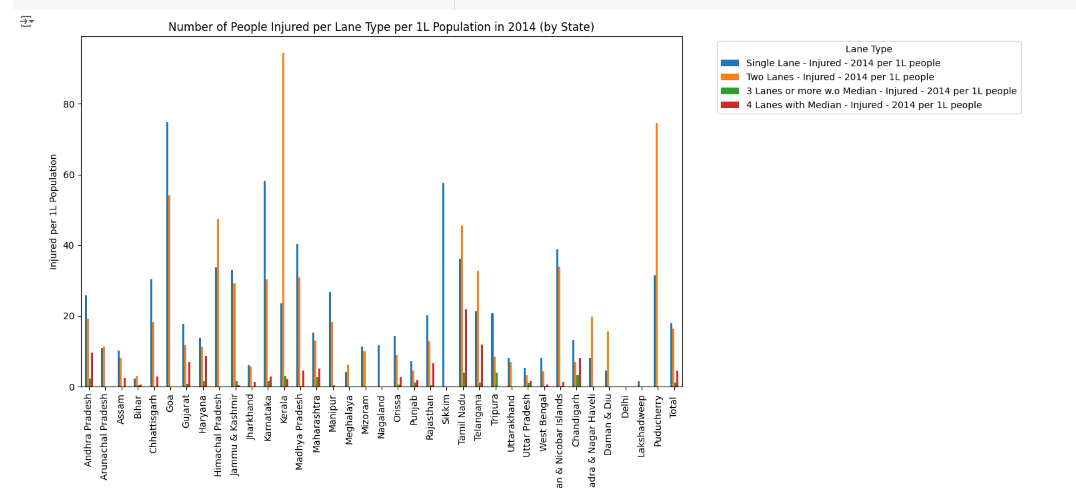
Description automatically generated**

****

**8. Number of people INJURED for 1,2,3,4 type of lane per 1L population of resp. State.**

# Relevant columns for injuries per 1L population  
injury\_columns = [  
    'Single Lane - Injured - 2014 per 1L people',  
    'Two Lanes - Injured - 2014 per 1L people',  
    '3 Lanes or more w.o Median - Injured - 2014 per 1L people',  
    '4 Lanes with Median - Injured - 2014 per 1L people'  
]  
  
# Create a DataFrame with State/UT and Injury columns per lane type  
lane\_injuries = df4[['State/UT'] + injury\_columns]  
  
# Set State/UT as index for better readability in plot  
lane\_injuries.set\_index('State/UT', inplace=True)  
  
# Plotting the number of injuries per lane type per state  
lane\_injuries.plot(kind='bar', figsize=(15, 8))  
  
plt.title('Number of People Injured per Lane Type per 1L Population in 2014 (by State)')  
plt.xlabel('State/UT')  
plt.ylabel('Injured per 1L Population')  
plt.xticks(rotation=90)  
plt.legend(title='Lane Type', bbox\_to\_anchor=(1.05, 1), loc='upper left')  
plt.tight\_layout()  
plt.show()  
  
# Displaying the DataFrame for verification  
lane\_injuries.head()

**Output:**

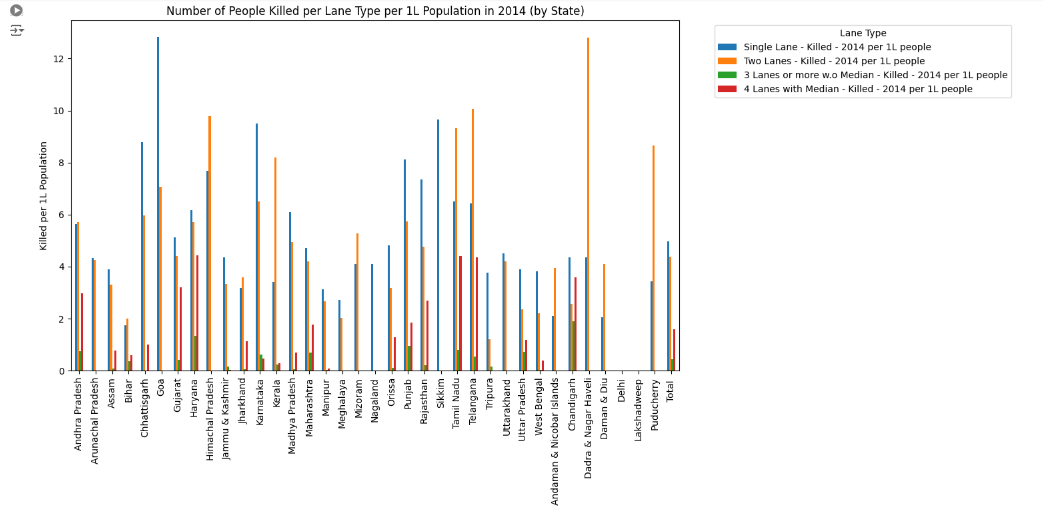


A white background with text

Description automatically generated with medium confidence

**9. Number of people KILLED for 1,2,3,4 LANES per 1L population of resp. States.**

# Relevant columns for killed per 1L population  
killed\_columns = [  
    'Single Lane - Killed - 2014 per 1L people',  
    'Two Lanes - Killed - 2014 per 1L people',  
    '3 Lanes or more w.o Median - Killed - 2014 per 1L people',  
    '4 Lanes with Median - Killed - 2014 per 1L people'  
]  
  
# Create a DataFrame with State/UT and Killed columns per lane type  
lane\_killed = df4[['State/UT'] + killed\_columns]  
  
# Set State/UT as index for better readability in plot  
lane\_killed.set\_index('State/UT', inplace=True)  
  
# Plotting the number of people killed per lane type per state  
lane\_killed.plot(kind='bar', figsize=(15, 8))  
  
plt.title('Number of People Killed per Lane Type per 1L Population in 2014 (by State)')  
plt.xlabel('State/UT')  
plt.ylabel('Killed per 1L Population')  
plt.xticks(rotation=90)  
plt.legend(title='Lane Type', bbox\_to\_anchor=(1.05, 1), loc='upper left')  
plt.tight\_layout()  
plt.show()  
  
# Displaying the DataFrame for verification  
lane\_killed.head()



A screenshot of a computer

Description automatically generated

**10. Number of Accidents, people KILLED, INJURED on SINGLE LANE per 1L population.**

# Select relevant columns  
single\_lane\_columns = [  
    'State/UT',  
    'Single Lane - Accident - 2014 per 1L people',  
    'Single Lane - Killed - 2014 per 1L people',  
    'Single Lane - Injured - 2014 per 1L people'  
]  
  
# Create a DataFrame with State/UT and relevant columns  
single\_lane\_data = df4[single\_lane\_columns]  
  
# Set State/UT as index for better readability in plot  
single\_lane\_data.set\_index('State/UT', inplace=True)  
  
# Plotting the accidents, killed, and injured per 1L population on Single Lanes  
single\_lane\_data.plot(kind='bar', figsize=(15, 8))  
  
plt.title('Accidents, People Killed, Injured on Single Lane per 1L Population in 2014 (by State)')  
plt.xlabel('State/UT')  
plt.ylabel('Per 1L Population')  
plt.xticks(rotation=90)  
plt.legend(title='Single Lane Metrics', bbox\_to\_anchor=(1.05, 1), loc='upper left')  
plt.tight\_layout()  
plt.show()  
  
# Displaying the DataFrame for verification  
single\_lane\_data.head()

**Output:**

**A screenshot of a computer screen

Description automatically generated**

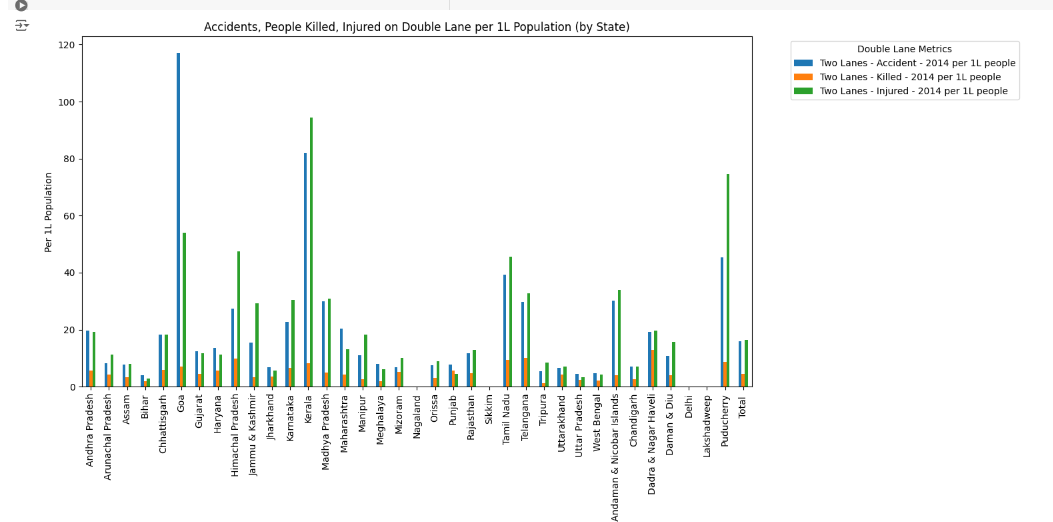
**A screenshot of a computer

Description automatically generated**

**11. Number of accidents, people INJURED, KILLED on DOUBLE LANE per 1L population.**

# Select relevant columns for double lane data  
double\_lane\_columns = [  
    'State/UT',  
    'Two Lanes - Accident - 2014 per 1L people',  
    'Two Lanes - Killed - 2014 per 1L people',  
    'Two Lanes - Injured - 2014 per 1L people'  
]  
  
# Create a DataFrame with State/UT and relevant columns for double lanes  
double\_lane\_data = df4[double\_lane\_columns]  
  
# Set State/UT as index for better readability in plot  
double\_lane\_data.set\_index('State/UT', inplace=True)  
  
# Plotting the accidents, killed, and injured per 1L population on Double Lanes  
double\_lane\_data.plot(kind='bar', figsize=(15, 8))  
  
plt.title('Accidents, People Killed, Injured on Double Lane per 1L Population (by State)')  
plt.xlabel('State/UT')  
plt.ylabel('Per 1L Population')  
plt.xticks(rotation=90)  
plt.legend(title='Double Lane Metrics', bbox\_to\_anchor=(1.05, 1), loc='upper left')  
plt.tight\_layout()  
plt.show()  
  
# Displaying the DataFrame for verification  
double\_lane\_data.head()

**Output:**

****

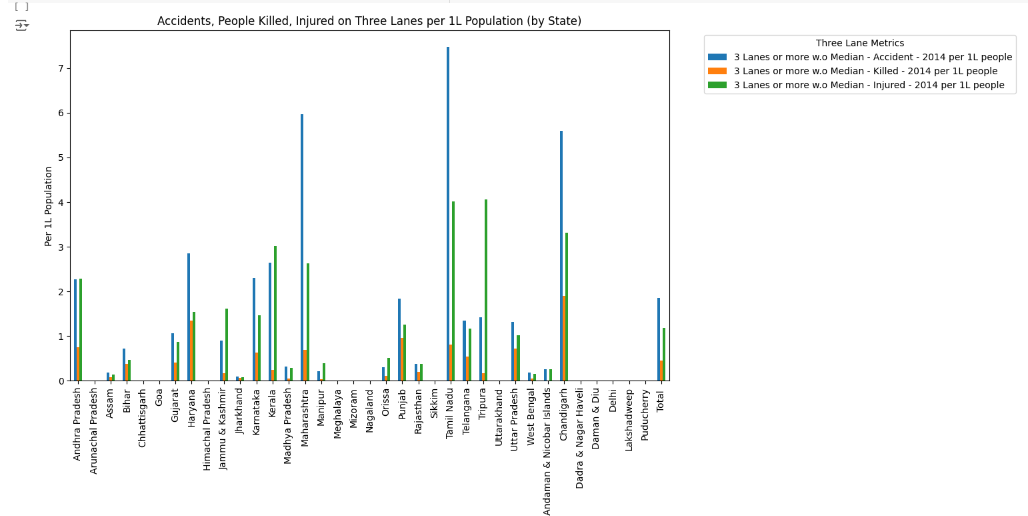
**A screenshot of a computer

Description automatically generated**

**12. Number of accidents, people INJURED, KILLED on THREE LANE per 1L population.**

# Select relevant columns for three lane data  
three\_lane\_columns = [  
    'State/UT',  
    '3 Lanes or more w.o Median - Accident - 2014 per 1L people',  
    '3 Lanes or more w.o Median - Killed - 2014 per 1L people',  
    '3 Lanes or more w.o Median - Injured - 2014 per 1L people'  
]  
  
# Create a DataFrame with State/UT and relevant columns for three lanes  
three\_lane\_data = df4[three\_lane\_columns]  
  
# Set State/UT as index for better readability in plot  
three\_lane\_data.set\_index('State/UT', inplace=True)  
  
# Plotting the accidents, killed, and injured per 1L population on Three Lanes  
three\_lane\_data.plot(kind='bar', figsize=(15, 8))  
  
plt.title('Accidents, People Killed, Injured on Three Lanes per 1L Population (by State)')  
plt.xlabel('State/UT')  
plt.ylabel('Per 1L Population')  
plt.xticks(rotation=90)  
plt.legend(title='Three Lane Metrics', bbox\_to\_anchor=(1.05, 1), loc='upper left')  
plt.tight\_layout()  
plt.show()  
  
# Displaying the DataFrame for verification  
three\_lane\_data.head()

**Output:**

****

**A screenshot of a computer

Description automatically generated**

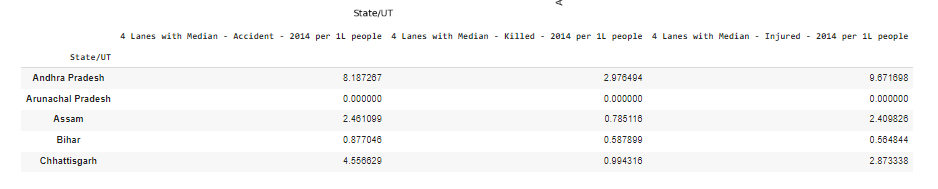
**13. Number of accidents, people INJURED, KILLED on FOUR LANE per 1L population**.

# Select relevant columns for four lane data  
four\_lane\_columns = [  
    'State/UT',  
    '4 Lanes with Median - Accident - 2014 per 1L people',  
    '4 Lanes with Median - Killed - 2014 per 1L people',  
    '4 Lanes with Median - Injured - 2014 per 1L people'  
]  
  
# Create a DataFrame with State/UT and relevant columns for four lanes  
four\_lane\_data = df4[four\_lane\_columns]  
  
# Set State/UT as index for better readability in plot  
four\_lane\_data.set\_index('State/UT', inplace=True)  
  
# Plotting the accidents, killed, and injured per 1L population on Four Lanes  
four\_lane\_data.plot(kind='bar', figsize=(15, 8))  
  
plt.title('Accidents, People Killed, Injured on Four Lanes per 1L Population (by State)')  
plt.xlabel('State/UT')  
plt.ylabel('Per 1L Population')  
plt.xticks(rotation=90)  
plt.legend(title='Four Lane Metrics', bbox\_to\_anchor=(1.05, 1), loc='upper left')  
plt.tight\_layout()  
plt.show()  
  
# Displaying the DataFrame for verification  
four\_lane\_data.head()

**Output:**

**A graph with text and numbers

Description automatically generated with medium confidence**

****

**14. Total Number of INJURED, KILLED, ROAD ACCIDENTS irrespective of lanes per 1L population of resp. State.**

# Create a DataFrame to calculate total injured, killed, and accidents per 1L population  
total\_data = pd.DataFrame()  
total\_data['State/UT'] = df4['State/UT']  
  
# Calculate total accidents  
total\_data['Total Accidents - 2014'] = (  
    df4['Single Lane - Accident - 2014'] +  
    df4['Two Lanes - Accident - 2014'] +  
    df4['3 Lanes or more w.o Median - Accident - 2014'] +  
    df4['4 Lanes with Median - Accident - 2014']  
)  
  
# Calculate total killed  
total\_data['Total Killed - 2014'] = (  
    df4['Single Lane - Killed - 2014'] +  
    df4['Two Lanes - Killed - 2014'] +  
    df4['3 Lanes or more w.o Median - Killed - 2014'] +  
    df4['4 Lanes with Median - Killed - 2014']  
)  
  
# Calculate total injured  
total\_data['Total Injured - 2014'] = (  
    df4['Single Lane - Injured - 2014'] +  
    df4['Two Lanes - Injured - 2014'] +  
    df4['3 Lanes or more w.o Median - Injured - 2014'] +  
    df4['4 Lanes with Median - Injured - 2014']  
)  
  
# Calculate per 1L population  
total\_data['Total Accidents per 1L'] = total\_data['Total Accidents - 2014'] / (df4['Population'] / 100000)  
total\_data['Total Killed per 1L'] = total\_data['Total Killed - 2014'] / (df4['Population'] / 100000)  
total\_data['Total Injured per 1L'] = total\_data['Total Injured - 2014'] / (df4['Population'] / 100000)  
  
# Display the total data DataFrame  
print(total\_data[['State/UT', 'Total Accidents per 1L', 'Total Killed per 1L', 'Total Injured per 1L']])

**Output:**

State/UT Total Accidents per 1L Total Killed per 1L \

0 Andhra Pradesh 46.631741 15.088536

1 Arunachal Pradesh 14.815061 8.599962

2 Assam 22.893344 8.081889

3 Bihar 9.179683 4.719525

4 Chhattisgarh 54.104102 15.744642

5 Goa 289.946488 19.882828

6 Gujarat 39.232496 13.161880

7 Haryana 42.111970 17.683398

8 Himachal Pradesh 44.547375 17.466417

9 Jammu & Kashmir 46.733585 7.909865

10 Jharkhand 15.717773 7.966501

11 Karnataka 71.548879 17.107700

12 Kerala 108.609034 12.120555

13 Madhya Pradesh 73.625705 11.798673

14 Maharashtra 54.840815 11.393171

15 Manipur 26.017283 5.882777

16 Meghalaya 18.268294 4.752453

17 Mizoram 12.030558 9.387481

18 Nagaland 15.415703 4.094006

19 Orissa 22.985538 9.365273

20 Punjab 23.036161 16.656251

21 Rajasthan 35.927880 15.009824

22 Sikkim 33.247240 9.662991

23 Tamil Nadu 93.212430 21.054228

24 Telangana 62.247945 21.410714

25 Tripura 19.488736 5.117154

26 Uttarakhand 13.979369 8.704884

27 Uttar Pradesh 15.531573 8.151148

28 West Bengal 14.105552 6.436514

29 Andaman & Nicobar Islands 57.280842 6.043392

30 Chandigarh 34.961391 12.411767

31 Dadra & Nagar Haveli 25.312110 17.165684

32 Daman & Diu 16.033086 6.166571

33 Delhi NaN NaN

34 Lakshadweep 1.551037 0.000000

35 Puducherry 89.025789 12.099815

36 Total 39.704259 11.396906

Total Injured per 1L

0 57.108619

1 22.258726

2 20.826406

3 6.378516

4 51.504788

5 128.827016

6 37.215610

7 35.280017

8 81.228307

9 64.132097

10 13.204748

11 93.020253

12 123.019592

13 76.190873

14 36.000214

15 45.346408

16 10.482360

17 21.326898

18 11.624957

19 26.413833

20 14.875643

21 40.049053

22 57.650386

23 107.731392

24 67.078222

25 33.343159

26 15.179017

27 11.178989

28 13.166643

29 74.359992

30 31.740016

31 27.930604

32 20.144133

33 NaN

34 1.551037

35 106.013608

36 40.070117

**15. Number of people KILLED for each different REASON per 1L population of that state.**

# Convert 'Population' column to numeric type  
df5['Population'] = df5['Population'].str.replace(',','').astype(int)  
  
# Create a DataFrame to calculate total killed by different reasons per 1L population  
killed\_data = pd.DataFrame()  
killed\_data['States/UTs'] = df5['States/UTs']  
  
# Calculate total killed by different reasons  
killed\_data['Killed - Fault of Driver'] = df5['Fault of Driver-Number of Persons-Killed - 2014']  
killed\_data['Killed - Fault of Driver of Other Vehicles'] = df5['Fault of Driver of other vehicles-Number of Persons-Killed - 2014']  
killed\_data['Killed - Fault of Pedestrian'] = df5['Fault of Pedestrian-Number of Persons-Killed - 2014']  
killed\_data['Killed - Defect in Condition of Motor Vehicle'] = df5['Defect in Condition of Motor Vehicle-Number of Persons-Killed - 2014']  
killed\_data['Killed - Defect in Road Condition'] = df5['Defect in Road Condition-Number of Persons-Killed - 2014']  
killed\_data['Killed - Weather Condition'] = df5['Weather Condition-Number of Persons-Killed - 2014']  
killed\_data['Killed - Fault of Passenger'] = df5['Fault of Passenger-Number of Persons-Killed - 2014']  
killed\_data['Killed - Poor Light'] = df5['Poor light-Number of Persons-Killed - 2014']  
killed\_data['Killed - Falling of Boulders'] = df5['Falling of boulders-Number of Persons-Killed - 2014']  
killed\_data['Killed - Other Causes'] = df5['Other causes/causes not known-Number of Persons-Killed - 2014']  
  
# Calculate per 1L population for each reason  
killed\_data['Killed per 1L - Fault of Driver'] = killed\_data['Killed - Fault of Driver'] / (df5['Population'] / 100000)  
killed\_data['Killed per 1L - Fault of Driver of Other Vehicles'] = killed\_data['Killed - Fault of Driver of Other Vehicles'] / (df5['Population'] / 100000)  
killed\_data['Killed per 1L - Fault of Pedestrian'] = killed\_data['Killed - Fault of Pedestrian'] / (df5['Population'] / 100000)  
killed\_data['Killed per 1L - Defect in Condition of Motor Vehicle'] = killed\_data['Killed - Defect in Condition of Motor Vehicle'] / (df5['Population'] / 100000)  
killed\_data['Killed per 1L - Defect in Road Condition'] = killed\_data['Killed - Defect in Road Condition'] / (df5['Population'] / 100000)  
killed\_data['Killed per 1L - Weather Condition'] = killed\_data['Killed - Weather Condition'] / (df5['Population'] / 100000)  
killed\_data['Killed per 1L - Fault of Passenger'] = killed\_data['Killed - Fault of Passenger'] / (df5['Population'] / 100000)  
killed\_data['Killed per 1L - Poor Light'] = killed\_data['Killed - Poor Light'] / (df5['Population'] / 100000)  
killed\_data['Killed per 1L - Falling of Boulders'] = killed\_data['Killed - Falling of Boulders'] / (df5['Population'] / 100000)  
killed\_data['Killed per 1L - Other Causes'] = killed\_data['Killed - Other Causes'] / (df5['Population'] / 100000)  
  
# Display the killed data DataFrame  
print(killed\_data[['States/UTs',  
                   'Killed per 1L - Fault of Driver',  
                   'Killed per 1L - Fault of Driver of Other Vehicles',  
                   'Killed per 1L - Fault of Pedestrian',  
                   'Killed per 1L - Defect in Condition of Motor Vehicle']])

**Output:**

States/UTs Killed per 1L - Fault of Driver \

0 Andhra Pradesh 12.865705

1 Arunachal Pradesh 1.373103

2 Assam 7.783865

3 Bihar 2.541800

4 Chhattisgarh 9.622161

5 Goa 16.934685

6 Gujarat 10.977885

7 Haryana 12.567323

8 Himachal Pradesh 16.723475

9 Jammu & Kashmir 4.473220

10 Jharkhand 4.104506

11 Karnataka 13.925785

12 Kerala 12.120555

13 Madhya Pradesh 9.280320

14 Maharashtra 9.889269

15 Manipur 0.000000

16 Meghalaya 0.404464

17 Mizoram 8.567215

18 Nagaland 3.133684

19 Odisha 8.993616

20 Punjab 9.951939

21 Rajasthan 14.410248

22 Sikkim 5.240944

23 Tamil Nadu 19.855287

24 Telangana 16.608340

25 Tripura 4.001179

26 Uttarakhand 6.315502

27 Uttar Pradesh 2.025400

28 West Bengal 2.936146

29 Andaman & Nicobar Is. 6.043392

30 Chandigarh 12.411767

31 Dadra & Nagar Haveli 17.165684

32 Daman & Diu 6.166571

33 Delhi 5.098898

34 Lakshadweep 0.000000

35 Puducherry 12.099815

36 Total 8.496311

Killed per 1L - Fault of Driver of Other Vehicles \

0 0.143101

1 1.011760

2 0.003205

3 0.243997

4 0.990401

5 0.000000

6 0.281272

7 0.828355

8 0.043702

9 0.494367

10 0.633561

11 1.060638

12 0.000000

13 0.476408

14 0.472528

15 0.000000

16 0.505580

17 0.820265

18 0.657063

19 0.038119

20 0.702871

21 0.000000

22 1.801575

23 0.047126

24 0.018602

25 0.190532

26 0.803070

27 0.752706

28 0.673780

29 0.000000

30 0.000000

31 0.000000

32 0.000000

33 0.077437

34 0.000000

35 0.000000

36 0.437294

Killed per 1L - Fault of Pedestrian \

0 0.198433

1 0.000000

2 0.169841

3 0.138329

4 0.160500

5 1.988283

6 0.670089

7 0.552236

8 0.000000

9 0.279078

10 0.585059

11 0.126033

12 0.000000

13 0.013769

14 0.098777

15 0.000000

16 0.370759

17 0.000000

18 0.000000

19 0.109591

20 0.407305

21 0.000000

22 0.000000

23 0.381166

24 0.155015

25 0.326627

26 0.000000

27 0.250735

28 0.224593

29 0.000000

30 0.000000

31 0.000000

32 0.000000

33 0.000000

34 0.000000

35 0.000000

36 0.211008

Killed per 1L - Defect in Condition of Motor Vehicle

0 0.049608

1 0.578149

2 0.000000

3 0.290107

4 0.896450

5 0.000000

6 0.099273

7 0.422066

8 0.538997

9 0.302999

10 0.472897

11 0.184957

12 0.000000

13 0.509454

14 0.087209

15 0.945446

16 0.741517

17 0.000000

18 0.303260

19 0.021442

20 0.385678

21 0.052518

22 0.000000

23 0.271667

24 0.430943

25 0.108876

26 0.158631

27 0.543010

28 0.383452

29 0.000000

30 0.000000

31 0.000000

32 0.000000

33 0.041697

34 0.000000

35 0.000000

36 0.293016

**16. Number of people INJURED for each reason per 1L people of that state.**

# Create a DataFrame to calculate total injured by different reasons  
injured\_data = pd.DataFrame()  
injured\_data['States/UTs'] = df5['States/UTs']  
  
# Calculate total injured by different reasons  
injured\_data['Injured - Fault of Driver'] = df5['Fault of Driver-Number of Persons-Injured - 2014']  
injured\_data['Injured - Fault of Driver of Other Vehicles'] = df5['Fault of Driver of other vehicles-Number of Persons-Injured - 2014']  
injured\_data['Injured - Fault of Pedestrian'] = df5['Fault of Pedestrian-Number of Persons-Injured - 2014']  
injured\_data['Injured - Defect in Condition of Motor Vehicle'] = df5['Defect in Condition of Motor Vehicle-Number of Persons-Injured - 2014']  
injured\_data['Injured - Defect in Road Condition'] = df5['Defect in Road Condition-Number of Persons-Injured - 2014']  
injured\_data['Injured - Weather Condition'] = df5['Weather Condition-Number of Persons-Injured - 2014']  
injured\_data['Injured - Fault of Passenger'] = df5['Fault of Passenger-Number of Persons-Injured - 2014']  
injured\_data['Injured - Poor Light'] = df5['Poor light-Number of Persons-Injured - 2014']  
injured\_data['Injured - Falling of Boulders'] = df5['Falling of boulders-Number of Persons-Injured - 2014']  
injured\_data['Injured - Other Causes'] = df5['Other causes/causes not known-Number of Persons-Injured - 2014']  
  
# Calculate per 1L population for each reason  
injured\_data['Injured per 1L - Fault of Driver'] = injured\_data['Injured - Fault of Driver'] / (df5['Population'] / 100000)  
injured\_data['Injured per 1L - Fault of Driver of Other Vehicles'] = injured\_data['Injured - Fault of Driver of Other Vehicles'] / (df5['Population'] / 100000)  
injured\_data['Injured per 1L - Fault of Pedestrian'] = injured\_data['Injured - Fault of Pedestrian'] / (df5['Population'] / 100000)  
injured\_data['Injured per 1L - Defect in Condition of Motor Vehicle'] = injured\_data['Injured - Defect in Condition of Motor Vehicle'] / (df5['Population'] / 100000)  
injured\_data['Injured per 1L - Defect in Road Condition'] = injured\_data['Injured - Defect in Road Condition'] / (df5['Population'] / 100000)  
injured\_data['Injured per 1L - Weather Condition'] = injured\_data['Injured - Weather Condition'] / (df5['Population'] / 100000)  
injured\_data['Injured per 1L - Fault of Passenger'] = injured\_data['Injured - Fault of Passenger'] / (df5['Population'] / 100000)  
injured\_data['Injured per 1L - Poor Light'] = injured\_data['Injured - Poor Light'] / (df5['Population'] / 100000)  
injured\_data['Injured per 1L - Falling of Boulders'] = injured\_data['Injured - Falling of Boulders'] / (df5['Population'] / 100000)  
injured\_data['Injured per 1L - Other Causes'] = injured\_data['Injured - Other Causes'] / (df5['Population'] / 100000)  
  
# Display the injured data DataFrame  
print(injured\_data[['States/UTs',  
                     'Injured per 1L - Fault of Driver',  
                     'Injured per 1L - Fault of Driver of Other Vehicles',  
                     'Injured per 1L - Fault of Pedestrian',  
                     'Injured per 1L - Defect in Condition of Motor Vehicle',  
                     'Injured per 1L - Defect in Road Condition',  
                     'Injured per 1L - Weather Condition',  
                     'Injured per 1L - Fault of Passenger',  
                     'Injured per 1L - Poor Light',  
                     'Injured per 1L - Falling of Boulders',  
                     'Injured per 1L - Other Causes']])

**Output:**

States/UTs Injured per 1L - Fault of Driver \

0 Andhra Pradesh 50.155834

1 Arunachal Pradesh 2.168058

2 Assam 20.127813

3 Bihar 3.241131

4 Chhattisgarh 34.096428

5 Goa 113.126438

6 Gujarat 30.205316

7 Haryana 30.108717

8 Himachal Pradesh 77.586435

9 Jammu & Kashmir 32.596297

10 Jharkhand 7.175307

11 Karnataka 74.694784

12 Kerala 123.019592

13 Madhya Pradesh 60.707610

14 Maharashtra 30.517645

15 Manipur 0.000000

16 Meghalaya 0.269643

17 Mizoram 20.233211

18 Nagaland 7.935297

19 Odisha 25.560929

20 Punjab 9.288716

21 Rajasthan 38.084019

22 Sikkim 29.644091

23 Tamil Nadu 103.506686

24 Telangana 44.278571

25 Tripura 26.429557

26 Uttarakhand 9.329494

27 Uttar Pradesh 2.796624

28 West Bengal 6.241501

29 Andaman & Nicobar Is. 74.359992

30 Chandigarh 31.740016

31 Dadra & Nagar Haveli 27.930604

32 Daman & Diu 20.144133

33 Delhi NaN

34 Lakshadweep 1.551037

35 Puducherry 106.013608

36 Total 32.484485

Injured per 1L - Fault of Driver of Other Vehicles \

0 0.810904

1 3.252087

2 0.003205

3 0.379445

4 2.603229

5 0.000000

6 0.853744

7 0.796798

8 0.437025

9 7.471314

10 0.682063

11 7.347538

12 0.000000

13 3.084261

14 1.945284

15 0.000000

16 0.572991

17 1.093687

18 2.577708

19 0.147710

20 0.717289

21 0.000000

22 3.766929

23 0.119201

24 0.124012

25 1.388164

26 1.328536

27 0.942885

28 1.350846

29 0.000000

30 0.000000

31 0.000000

32 0.000000

33 NaN

34 0.000000

35 0.000000

36 1.333603

Injured per 1L - Fault of Pedestrian \

0 0.305281

1 0.000000

2 0.413388

3 0.223824

4 0.544134

5 5.896287

6 2.169104

7 0.761297

8 0.000000

9 1.060496

10 0.733597

11 1.419095

12 0.000000

13 0.136313

14 0.238489

15 0.000000

16 0.606696

17 0.000000

18 0.000000

19 0.185828

20 0.079298

21 0.000000

22 0.000000

23 1.137954

24 1.382737

25 0.680473

26 0.000000

27 0.369347

28 0.353871

29 0.000000

30 0.000000

31 0.000000

32 0.000000

33 NaN

34 0.000000

35 0.000000

36 0.522854

Injured per 1L - Defect in Condition of Motor Vehicle \

0 0.162181

1 1.589909

2 0.000000

3 0.341020

4 1.914254

5 0.891299

6 0.271345

7 0.378676

8 2.272528

9 1.538915

10 0.666906

11 1.325798

12 0.000000

13 2.326964

14 0.404897

15 7.738653

16 1.381919

17 0.000000

18 1.111952

19 0.083385

20 0.306380

21 0.110871

22 0.000000

23 0.712434

24 1.987296

25 0.871005

26 0.178460

27 0.739194

28 0.591612

29 0.000000

30 0.000000

31 0.000000

32 0.000000

33 NaN

34 0.000000

35 0.000000

36 0.697854

Injured per 1L - Defect in Road Condition \

0 0.843340

1 2.095789

2 0.000000

3 0.270895

4 1.491474

5 0.000000

6 0.177036

7 0.220895

8 0.917752

9 1.020628

10 0.700252

11 0.299532

12 0.000000

13 1.840918

14 0.288322

15 4.482116

16 0.438169

17 0.000000

18 0.000000

19 0.054796

20 0.623573

21 0.297600

22 0.000000

23 0.228700

24 3.797875

25 0.000000

26 0.000000

27 0.793244

28 0.456856

29 0.000000

30 0.000000

31 0.000000

32 0.000000

33 NaN

34 0.000000

35 0.000000

36 0.619067

Injured per 1L - Weather Condition Injured per 1L - Fault of Passenger \

0 0.133561 0.253765

1 1.806715 1.951252

2 0.000000 0.000000

3 0.293950 0.298753

4 0.806414 1.190048

5 0.000000 0.000000

6 0.398745 0.226672

7 0.268229 0.315564

8 0.000000 0.000000

9 0.287052 0.000000

10 0.697220 0.139444

11 0.502494 0.473032

12 0.000000 0.000000

13 0.616852 0.691205

14 0.081869 0.061402

15 4.447099 3.501653

16 0.471875 0.539285

17 0.000000 0.000000

18 0.000000 0.000000

19 0.021442 0.016677

20 0.598342 0.036045

21 0.037929 0.004376

22 2.620472 1.637795

23 0.077619 1.675745

24 2.852282 0.542554

25 0.000000 0.000000

26 0.000000 0.694011

27 0.593056 1.212638

28 1.017791 0.197204

29 0.000000 0.000000

30 0.000000 0.000000

31 0.000000 0.000000

32 0.000000 0.000000

33 NaN NaN

34 0.000000 0.000000

35 0.000000 0.000000

36 0.452325 0.503859

Injured per 1L - Poor Light Injured per 1L - Falling of Boulders \

0 0.591483 0.022896

1 2.384863 0.000000

2 0.000000 0.000000

3 0.219982 0.055716

4 1.706779 0.148756

5 0.000000 0.000000

6 0.248181 0.000000

7 0.323453 0.220895

8 0.000000 0.000000

9 0.000000 0.000000

10 0.227354 0.072753

11 0.530319 0.065471

12 0.000000 0.000000

13 0.503946 0.023407

14 0.126363 0.056953

15 2.206041 0.035017

16 0.438169 0.134821

17 0.000000 0.000000

18 0.000000 0.000000

19 0.023824 0.021442

20 0.245104 0.162201

21 0.142965 0.001459

22 4.749606 0.000000

23 0.008316 0.000000

24 1.171916 3.267723

25 0.299408 0.000000

26 0.000000 0.019829

27 0.688646 0.231717

28 0.213637 0.109558

29 0.000000 0.000000

30 0.000000 0.000000

31 0.000000 0.000000

32 0.000000 0.000000

33 NaN NaN

34 0.000000 0.000000

35 0.000000 0.000000

36 0.362884 0.164182

Injured per 1L - Other Causes

0 3.085251

1 3.902504

2 0.112159

3 0.715662

4 5.562689

5 5.690603

6 1.542033

7 1.242532

8 0.000000

9 13.563185

10 1.915840

11 5.049489

12 0.000000

13 5.643921

14 2.115252

15 22.340547

16 5.493970

17 0.000000

18 0.000000

19 0.069090

20 2.292442

21 1.321693

22 15.231494

23 0.013861

24 6.888880

25 3.538458

26 3.628687

27 1.620020

28 2.194440

29 0.000000

30 0.000000

31 0.000000

32 0.000000

33 NaN

34 0.000000

35 0.000000

36 2.273518

**17. Number of ACCIDENTS for each reason per 1L people of that state.**

# Create a DataFrame to calculate total accidents by different reasons  
accidents\_data = pd.DataFrame()  
accidents\_data['States/UTs'] = df5['States/UTs']  
  
# Calculate total accidents by different reasons  
accidents\_data['Accidents - Fault of Driver'] = df5['Fault of Driver-Total No. of Road Accidents - 2014']  
accidents\_data['Accidents - Fault of Driver of Other Vehicles'] = df5['Fault of Driver of other vehicles-Total No. of Road Accidents - 2014']  
accidents\_data['Accidents - Fault of Pedestrian'] = df5['Fault of Pedestrian-Total No. of Road Accidents - 2014']  
accidents\_data['Accidents - Defect in Condition of Motor Vehicle'] = df5['Defect in Condition of Motor Vehicle-Total No. of Road Accidents - 2014']  
accidents\_data['Accidents - Defect in Road Condition'] = df5['Defect in Road Condition-Total No. of Road Accidents - 2014']  
accidents\_data['Accidents - Weather Condition'] = df5['Weather Condition-Total No. of Road Accidents - 2014']  
accidents\_data['Accidents - Fault of Passenger'] = df5['Fault of Passenger-Total No. of Road Accidents - 2014']  
accidents\_data['Accidents - Poor Light'] = df5['Poor light-Total No. of Road Accidents - 2014']  
accidents\_data['Accidents - Falling of Boulders'] = df5['Falling of boulders-Total No. of Road Accidents - 2014']  
accidents\_data['Accidents - Other Causes'] = df5['Other causes/causes not known-Total No. of Road Accidents - 2014']  
  
# Calculate per 1L population for each reason  
accidents\_data['Accidents per 1L - Fault of Driver'] = accidents\_data['Accidents - Fault of Driver'] / (df5['Population'] / 100000)  
accidents\_data['Accidents per 1L - Fault of Driver of Other Vehicles'] = accidents\_data['Accidents - Fault of Driver of Other Vehicles'] / (df5['Population'] / 100000)  
accidents\_data['Accidents per 1L - Fault of Pedestrian'] = accidents\_data['Accidents - Fault of Pedestrian'] / (df5['Population'] / 100000)  
accidents\_data['Accidents per 1L - Defect in Condition of Motor Vehicle'] = accidents\_data['Accidents - Defect in Condition of Motor Vehicle'] / (df5['Population'] / 100000)  
accidents\_data['Accidents per 1L - Defect in Road Condition'] = accidents\_data['Accidents - Defect in Road Condition'] / (df5['Population'] / 100000)  
accidents\_data['Accidents per 1L - Weather Condition'] = accidents\_data['Accidents - Weather Condition'] / (df5['Population'] / 100000)  
accidents\_data['Accidents per 1L - Fault of Passenger'] = accidents\_data['Accidents - Fault of Passenger'] / (df5['Population'] / 100000)  
accidents\_data['Accidents per 1L - Poor Light'] = accidents\_data['Accidents - Poor Light'] / (df5['Population'] / 100000)  
accidents\_data['Accidents per 1L - Falling of Boulders'] = accidents\_data['Accidents - Falling of Boulders'] / (df5['Population'] / 100000)  
accidents\_data['Accidents per 1L - Other Causes'] = accidents\_data['Accidents - Other Causes'] / (df5['Population'] / 100000)  
  
# Display the accidents data DataFrame  
print(accidents\_data[['States/UTs',  
                      'Accidents per 1L - Fault of Driver',  
                      'Accidents per 1L - Fault of Driver of Other Vehicles',  
                      'Accidents per 1L - Fault of Pedestrian',  
                      'Accidents per 1L - Defect in Condition of Motor Vehicle',  
                      'Accidents per 1L - Defect in Road Condition',  
                      'Accidents per 1L - Weather Condition',  
                      'Accidents per 1L - Fault of Passenger',  
                      'Accidents per 1L - Poor Light',  
                      'Accidents per 1L - Falling of Boulders',  
                      'Accidents per 1L - Other Causes']])

**Output:**

States/UTs Accidents per 1L - Fault of Driver \

0 Andhra Pradesh 40.753165

1 Arunachal Pradesh 2.168058

2 Assam 22.095410

3 Bihar 4.810784

4 Chhattisgarh 35.654451

5 Goa 260.190807

6 Gujarat 31.345295

7 Haryana 32.376831

8 Himachal Pradesh 42.770142

9 Jammu & Kashmir 28.816785

10 Jharkhand 8.975955

11 Karnataka 57.956998

12 Kerala 108.609034

13 Madhya Pradesh 56.831631

14 Maharashtra 42.925283

15 Manipur 0.000000

16 Meghalaya 0.842633

17 Mizoram 10.390027

18 Nagaland 10.664634

19 Odisha 22.039720

20 Punjab 13.873601

21 Rajasthan 34.482187

22 Sikkim 20.144879

23 Tamil Nadu 89.174842

24 Telangana 50.671402

25 Tripura 15.297025

26 Uttarakhand 10.301110

27 Uttar Pradesh 3.834098

28 West Bengal 6.520874

29 Andaman & Nicobar Is. 57.280842

30 Chandigarh 34.961391

31 Dadra & Nagar Haveli 25.312110

32 Daman & Diu 16.033086

33 Delhi NaN

34 Lakshadweep 1.551037

35 Puducherry 89.025789

36 Total 31.299537

Accidents per 1L - Fault of Driver of Other Vehicles \

0 0.549507

1 1.445372

2 0.006409

3 0.584057

4 2.842021

5 0.000000

6 0.744544

7 1.798713

8 0.320485

9 3.029988

10 1.085239

11 4.998748

12 0.000000

13 3.049838

14 6.469449

15 0.000000

16 1.348214

17 1.640531

18 3.689660

19 0.131033

20 1.056109

21 0.000000

22 2.456693

23 0.158011

24 0.055806

25 0.626035

26 1.447509

27 1.369785

28 1.637887

29 0.000000

30 0.000000

31 0.000000

32 0.000000

33 NaN

34 0.000000

35 0.000000

36 1.724071

Accidents per 1L - Fault of Pedestrian \

0 0.446474

1 0.000000

2 0.461456

3 0.274737

4 0.571536

5 11.998259

6 2.501667

7 1.372702

8 0.000000

9 1.778125

10 0.924575

11 0.548324

12 0.000000

13 0.209289

14 0.297221

15 0.000000

16 1.011160

17 0.000000

18 0.000000

19 0.214417

20 0.443350

21 0.000000

22 0.000000

23 1.307053

24 1.454044

25 0.680473

26 0.000000

27 0.486456

28 0.377974

29 0.000000

30 0.000000

31 0.000000

32 0.000000

33 NaN

34 0.000000

35 0.000000

36 0.593878

Accidents per 1L - Defect in Condition of Motor Vehicle \

0 0.127837

1 1.228566

2 0.000000

3 0.519695

4 2.630631

5 1.165545

6 0.266381

7 0.650850

8 1.077994

9 2.719016

10 0.863947

11 1.024629

12 0.000000

13 2.291165

14 0.442272

15 4.622182

16 2.527900

17 0.000000

18 1.061409

19 0.066708

20 0.609155

21 0.107953

22 0.000000

23 0.719364

24 1.078907

25 0.544378

26 0.347006

27 1.030467

28 0.645295

29 0.000000

30 0.000000

31 0.000000

32 0.000000

33 NaN

34 0.000000

35 0.000000

36 0.759463

Accidents per 1L - Defect in Road Condition \

0 0.543783

1 1.084029

2 0.000000

3 0.428436

4 1.722437

5 0.068561

6 0.335872

7 0.556181

8 0.364187

9 0.901023

10 0.676001

11 0.291348

12 0.000000

13 1.991000

14 0.339045

15 2.836339

16 1.011160

17 0.000000

18 0.000000

19 0.061943

20 0.684849

21 0.248000

22 0.000000

23 0.163555

24 1.531551

25 0.000000

26 0.009914

27 1.360777

28 0.690213

29 0.000000

30 0.000000

31 0.000000

32 0.000000

33 NaN

34 0.000000

35 0.000000

36 0.690091

Accidents per 1L - Weather Condition \

0 0.114481

1 1.662178

2 0.000000

3 0.392894

4 0.829902

5 0.068561

6 0.320981

7 0.500957

8 0.000000

9 0.159473

10 0.615373

11 1.111379

12 0.000000

13 0.742150

14 0.090768

15 3.851818

16 0.943749

17 0.000000

18 0.000000

19 0.026207

20 1.120990

21 0.033553

22 1.637795

23 0.049898

24 1.047903

25 0.000000

26 0.000000

27 0.791743

28 1.033129

29 0.000000

30 0.000000

31 0.000000

32 0.000000

33 NaN

34 0.000000

35 0.000000

36 0.492297

Accidents per 1L - Fault of Passenger Accidents per 1L - Poor Light \

0 0.129745 0.364430

1 0.794954 2.457132

2 0.000000 0.000000

3 0.377524 0.316044

4 1.119584 1.377950

5 0.000000 0.205684

6 0.221709 0.254799

7 0.607460 0.520680

8 0.000000 0.000000

9 0.000000 0.000000

10 0.163695 0.239480

11 0.225877 0.404286

12 0.000000 0.000000

13 0.722874 0.574168

14 0.084539 0.133482

15 1.645777 1.050496

16 1.112276 1.348214

17 0.000000 0.000000

18 0.000000 0.000000

19 0.030971 0.033354

20 0.100925 0.403701

21 0.005835 0.096282

22 0.655118 1.801575

23 1.449817 0.029107

24 0.192219 0.458845

25 0.000000 0.081657

26 0.505637 0.000000

27 1.687083 0.738193

28 0.223498 0.221307

29 0.000000 0.000000

30 0.000000 0.000000

31 0.000000 0.000000

32 0.000000 0.000000

33 NaN NaN

34 0.000000 0.000000

35 0.000000 0.000000

36 0.555062 0.347688

Accidents per 1L - Falling of Boulders Accidents per 1L - Other Causes

0 0.011448 2.919254

1 0.000000 2.457132

2 0.000000 0.137796

3 0.078771 0.922195

4 0.101780 6.075506

5 0.000000 9.804291

6 0.004964 2.197232

7 0.512791 2.070886

8 0.000000 0.000000

9 0.000000 7.016815

10 0.072753 1.912809

11 0.068745 3.903737

12 0.000000 0.000000

13 0.097760 5.966116

14 0.063182 3.834506

15 0.035017 11.555455

16 0.404464 7.482585

17 0.000000 0.000000

18 0.000000 0.000000

19 0.019059 0.090532

20 0.183828 3.712603

21 0.004376 0.904470

22 0.000000 6.551180

23 0.000000 0.038810

24 1.751673 3.469242

25 0.000000 2.123075

26 0.009914 1.358279

27 0.232718 2.552895

28 0.112844 2.264557

29 0.000000 0.000000

30 0.000000 0.000000

31 0.000000 0.000000

32 0.000000 0.000000

33 NaN NaN

34 0.000000 0.000000

35 0.000000 0.000000

36 0.137424 2.433239

**18. Total number of ROAD ACCIDENTS, INJURIES, DEATHS due to FAULT OF THE DRIVER per 1L population of that state.**

# Create a DataFrame to calculate total accidents, injuries, and deaths due to fault of the driver  
driver\_fault\_data = pd.DataFrame()  
driver\_fault\_data['States/UTs'] = df5['States/UTs']  
  
# Calculate total accidents, injuries, and deaths  
driver\_fault\_data['Total Accidents - Fault of Driver'] = df5['Fault of Driver-Total No. of Road Accidents - 2014']  
driver\_fault\_data['Total Deaths - Fault of Driver'] = df5['Fault of Driver-Number of Persons-Killed - 2014']  
driver\_fault\_data['Total Injuries - Fault of Driver'] = df5['Fault of Driver-Number of Persons-Injured - 2014']  
  
# Calculate per 1L population for each  
driver\_fault\_data['Accidents per 1L - Fault of Driver'] = driver\_fault\_data['Total Accidents - Fault of Driver'] / (df5['Population'] / 100000)  
driver\_fault\_data['Deaths per 1L - Fault of Driver'] = driver\_fault\_data['Total Deaths - Fault of Driver'] / (df5['Population'] / 100000)  
driver\_fault\_data['Injuries per 1L - Fault of Driver'] = driver\_fault\_data['Total Injuries - Fault of Driver'] / (df5['Population'] / 100000)  
  
# Display the resulting DataFrame  
print(driver\_fault\_data[['States/UTs',  
                         'Accidents per 1L - Fault of Driver',  
                         'Deaths per 1L - Fault of Driver',  
                         'Injuries per 1L - Fault of Driver']])

**Output:**

States/UTs Accidents per 1L - Fault of Driver \

0 Andhra Pradesh 40.753165

1 Arunachal Pradesh 2.168058

2 Assam 22.095410

3 Bihar 4.810784

4 Chhattisgarh 35.654451

5 Goa 260.190807

6 Gujarat 31.345295

7 Haryana 32.376831

8 Himachal Pradesh 42.770142

9 Jammu & Kashmir 28.816785

10 Jharkhand 8.975955

11 Karnataka 57.956998

12 Kerala 108.609034

13 Madhya Pradesh 56.831631

14 Maharashtra 42.925283

15 Manipur 0.000000

16 Meghalaya 0.842633

17 Mizoram 10.390027

18 Nagaland 10.664634

19 Odisha 22.039720

20 Punjab 13.873601

21 Rajasthan 34.482187

22 Sikkim 20.144879

23 Tamil Nadu 89.174842

24 Telangana 50.671402

25 Tripura 15.297025

26 Uttarakhand 10.301110

27 Uttar Pradesh 3.834098

28 West Bengal 6.520874

29 Andaman & Nicobar Is. 57.280842

30 Chandigarh 34.961391

31 Dadra & Nagar Haveli 25.312110

32 Daman & Diu 16.033086

33 Delhi NaN

34 Lakshadweep 1.551037

35 Puducherry 89.025789

36 Total 31.299537

Deaths per 1L - Fault of Driver Injuries per 1L - Fault of Driver

0 12.865705 50.155834

1 1.373103 2.168058

2 7.783865 20.127813

3 2.541800 3.241131

4 9.622161 34.096428

5 16.934685 113.126438

6 10.977885 30.205316

7 12.567323 30.108717

8 16.723475 77.586435

9 4.473220 32.596297

10 4.104506 7.175307

11 13.925785 74.694784

12 12.120555 123.019592

13 9.280320 60.707610

14 9.889269 30.517645

15 0.000000 0.000000

16 0.404464 0.269643

17 8.567215 20.233211

18 3.133684 7.935297

19 8.993616 25.560929

20 9.951939 9.288716

21 14.410248 38.084019

22 5.240944 29.644091

23 19.855287 103.506686

24 16.608340 44.278571

25 4.001179 26.429557

26 6.315502 9.329494

27 2.025400 2.796624

28 2.936146 6.241501

29 6.043392 74.359992

30 12.411767 31.740016

31 17.165684 27.930604

32 6.166571 20.144133

33 5.098898 NaN

34 0.000000 1.551037

35 12.099815 106.013608

36 8.496311 32.484485

**19. Total number of ROAD ACCIDENTS, INJURIES, DEATHS due to the FAULT Of DRIVER'S FROM OTHER VEHICLES per 1L people of that state.**

# Create a DataFrame to calculate total accidents, injuries, and deaths due to the fault of drivers from other vehicles  
other\_vehicle\_fault\_data = pd.DataFrame()  
other\_vehicle\_fault\_data['States/UTs'] = df5['States/UTs']  
  
# Calculate total accidents, injuries, and deaths  
other\_vehicle\_fault\_data['Total Accidents - Fault of Other Vehicles'] = df5['Fault of Driver of other vehicles-Total No. of Road Accidents - 2014']  
other\_vehicle\_fault\_data['Total Deaths - Fault of Other Vehicles'] = df5['Fault of Driver of other vehicles-Number of Persons-Killed - 2014']  
other\_vehicle\_fault\_data['Total Injuries - Fault of Other Vehicles'] = df5['Fault of Driver of other vehicles-Number of Persons-Injured - 2014']  
  
# Calculate per 1L population for each  
other\_vehicle\_fault\_data['Accidents per 1L - Fault of Other Vehicles'] = other\_vehicle\_fault\_data['Total Accidents - Fault of Other Vehicles'] / (df5['Population'] / 100000)  
other\_vehicle\_fault\_data['Deaths per 1L - Fault of Other Vehicles'] = other\_vehicle\_fault\_data['Total Deaths - Fault of Other Vehicles'] / (df5['Population'] / 100000)  
other\_vehicle\_fault\_data['Injuries per 1L - Fault of Other Vehicles'] = other\_vehicle\_fault\_data['Total Injuries - Fault of Other Vehicles'] / (df5['Population'] / 100000)  
  
# Display the resulting DataFrame  
print(other\_vehicle\_fault\_data[['States/UTs',  
                                 'Accidents per 1L - Fault of Other Vehicles',  
                                 'Deaths per 1L - Fault of Other Vehicles',  
                                 'Injuries per 1L - Fault of Other Vehicles']])

**Output:**

States/UTs Accidents per 1L - Fault of Other Vehicles \

0 Andhra Pradesh 0.549507

1 Arunachal Pradesh 1.445372

2 Assam 0.006409

3 Bihar 0.584057

4 Chhattisgarh 2.842021

5 Goa 0.000000

6 Gujarat 0.744544

7 Haryana 1.798713

8 Himachal Pradesh 0.320485

9 Jammu & Kashmir 3.029988

10 Jharkhand 1.085239

11 Karnataka 4.998748

12 Kerala 0.000000

13 Madhya Pradesh 3.049838

14 Maharashtra 6.469449

15 Manipur 0.000000

16 Meghalaya 1.348214

17 Mizoram 1.640531

18 Nagaland 3.689660

19 Odisha 0.131033

20 Punjab 1.056109

21 Rajasthan 0.000000

22 Sikkim 2.456693

23 Tamil Nadu 0.158011

24 Telangana 0.055806

25 Tripura 0.626035

26 Uttarakhand 1.447509

27 Uttar Pradesh 1.369785

28 West Bengal 1.637887

29 Andaman & Nicobar Is. 0.000000

30 Chandigarh 0.000000

31 Dadra & Nagar Haveli 0.000000

32 Daman & Diu 0.000000

33 Delhi NaN

34 Lakshadweep 0.000000

35 Puducherry 0.000000

36 Total 1.724071

Deaths per 1L - Fault of Other Vehicles \

0 0.143101

1 1.011760

2 0.003205

3 0.243997

4 0.990401

5 0.000000

6 0.281272

7 0.828355

8 0.043702

9 0.494367

10 0.633561

11 1.060638

12 0.000000

13 0.476408

14 0.472528

15 0.000000

16 0.505580

17 0.820265

18 0.657063

19 0.038119

20 0.702871

21 0.000000

22 1.801575

23 0.047126

24 0.018602

25 0.190532

26 0.803070

27 0.752706

28 0.673780

29 0.000000

30 0.000000

31 0.000000

32 0.000000

33 0.077437

34 0.000000

35 0.000000

36 0.437294

Injuries per 1L - Fault of Other Vehicles

0 0.810904

1 3.252087

2 0.003205

3 0.379445

4 2.603229

5 0.000000

6 0.853744

7 0.796798

8 0.437025

9 7.471314

10 0.682063

11 7.347538

12 0.000000

13 3.084261

14 1.945284

15 0.000000

16 0.572991

17 1.093687

18 2.577708

19 0.147710

20 0.717289

21 0.000000

22 3.766929

23 0.119201

24 0.124012

25 1.388164

26 1.328536

27 0.942885

28 1.350846

29 0.000000

30 0.000000

31 0.000000

32 0.000000

33 NaN

34 0.000000

35 0.000000

36 1.333603

**20. Total number of ROAD ACCIDENTS, INJURIES, DEATHS due to the FAULT OF PEDESTRIANS per 1L people of that state**

# Create a DataFrame to calculate total accidents, injuries, and deaths due to the fault of pedestrians  
pedestrian\_fault\_data = pd.DataFrame()  
pedestrian\_fault\_data['States/UTs'] = df5['States/UTs']  
  
# Calculate total accidents, injuries, and deaths  
pedestrian\_fault\_data['Total Accidents - Fault of Pedestrians'] = df5['Fault of Pedestrian-Total No. of Road Accidents - 2014']  
pedestrian\_fault\_data['Total Deaths - Fault of Pedestrians'] = df5['Fault of Pedestrian-Number of Persons-Killed - 2014']  
pedestrian\_fault\_data['Total Injuries - Fault of Pedestrians'] = df5['Fault of Pedestrian-Number of Persons-Injured - 2014']  
  
# Calculate per 1L population for each  
pedestrian\_fault\_data['Accidents per 1L - Fault of Pedestrians'] = pedestrian\_fault\_data['Total Accidents - Fault of Pedestrians'] / (df5['Population'] / 100000)  
pedestrian\_fault\_data['Deaths per 1L - Fault of Pedestrians'] = pedestrian\_fault\_data['Total Deaths - Fault of Pedestrians'] / (df5['Population'] / 100000)  
pedestrian\_fault\_data['Injuries per 1L - Fault of Pedestrians'] = pedestrian\_fault\_data['Total Injuries - Fault of Pedestrians'] / (df5['Population'] / 100000)  
  
# Display the resulting DataFrame  
print(pedestrian\_fault\_data[['States/UTs',  
                               'Accidents per 1L - Fault of Pedestrians',  
                               'Deaths per 1L - Fault of Pedestrians',  
                               'Injuries per 1L - Fault of Pedestrians']])

**Output:**

States/UTs Accidents per 1L - Fault of Pedestrians \

0 Andhra Pradesh 0.446474

1 Arunachal Pradesh 0.000000

2 Assam 0.461456

3 Bihar 0.274737

4 Chhattisgarh 0.571536

5 Goa 11.998259

6 Gujarat 2.501667

7 Haryana 1.372702

8 Himachal Pradesh 0.000000

9 Jammu & Kashmir 1.778125

10 Jharkhand 0.924575

11 Karnataka 0.548324

12 Kerala 0.000000

13 Madhya Pradesh 0.209289

14 Maharashtra 0.297221

15 Manipur 0.000000

16 Meghalaya 1.011160

17 Mizoram 0.000000

18 Nagaland 0.000000

19 Odisha 0.214417

20 Punjab 0.443350

21 Rajasthan 0.000000

22 Sikkim 0.000000

23 Tamil Nadu 1.307053

24 Telangana 1.454044

25 Tripura 0.680473

26 Uttarakhand 0.000000

27 Uttar Pradesh 0.486456

28 West Bengal 0.377974

29 Andaman & Nicobar Is. 0.000000

30 Chandigarh 0.000000

31 Dadra & Nagar Haveli 0.000000

32 Daman & Diu 0.000000

33 Delhi NaN

34 Lakshadweep 0.000000

35 Puducherry 0.000000

36 Total 0.593878

Deaths per 1L - Fault of Pedestrians \

0 0.198433

1 0.000000

2 0.169841

3 0.138329

4 0.160500

5 1.988283

6 0.670089

7 0.552236

8 0.000000

9 0.279078

10 0.585059

11 0.126033

12 0.000000

13 0.013769

14 0.098777

15 0.000000

16 0.370759

17 0.000000

18 0.000000

19 0.109591

20 0.407305

21 0.000000

22 0.000000

23 0.381166

24 0.155015

25 0.326627

26 0.000000

27 0.250735

28 0.224593

29 0.000000

30 0.000000

31 0.000000

32 0.000000

33 0.000000

34 0.000000

35 0.000000

36 0.211008

Injuries per 1L - Fault of Pedestrians

0 0.305281

1 0.000000

2 0.413388

3 0.223824

4 0.544134

5 5.896287

6 2.169104

7 0.761297

8 0.000000

9 1.060496

10 0.733597

11 1.419095

12 0.000000

13 0.136313

14 0.238489

15 0.000000

16 0.606696

17 0.000000

18 0.000000

19 0.185828

20 0.079298

21 0.000000

22 0.000000

23 1.137954

24 1.382737

25 0.680473

26 0.000000

27 0.369347

28 0.353871

29 0.000000

30 0.000000

31 0.000000

32 0.000000

33 NaN

34 0.000000

35 0.000000

36 0.522854

**21. Total number of ROAD ACCIDENTS, INJURIES, DEATHS due to the DEFECTS IN THE VEHICLE per 1L people of that state.**

# Load the dataset (Assuming df is the DataFrame)  
df6 = pd.read\_csv('typeOfVehicle.csv')  
  
# Create a DataFrame to calculate total accidents, injuries, and deaths due to defects in the vehicle  
vehicle\_defect\_data = pd.DataFrame()  
vehicle\_defect\_data['States/UTs'] = df6['States/UTs']  
  
# Calculate total accidents, injuries, and deaths due to defects in vehicles  
vehicle\_defect\_data['Total Accidents - Defects in Vehicles'] = (  
    df6['Two-Wheelers - Number of Road Accidents - Total - 2014'] +  
    df6['Auto-Rickshaws - Number of Road Accidents - Total - 2014'] +  
    df6['Cars, Jeeps,Taxis - Number of Road Accidents - Total - 2014'] +  
    df6['Buses - Number of Road Accidents - Total - 2014'] +  
    df6['Trucks, Tempos,MAVs,Tractors - Number of Road Accidents - Total - 2014'] +  
    df6['Other Motor Vehicles - Number of Road Accidents - Total - 2014'] +  
    df6['Other Vehicles/Objects - Number of Road Accidents - Total - 2014']  
)  
  
vehicle\_defect\_data['Total Deaths - Defects in Vehicles'] = (  
    df6['Two-Wheelers - Number of Persons - Killed - 2014'] +  
    df6['Auto-Rickshaws - Number of Persons - Killed - 2014'] +  
    df6['Cars, Jeeps,Taxis - Number of Persons - Killed - 2014'] +  
    df6['Buses - Number of Persons - Killed - 2014'] +  
    df6['Trucks, Tempos,MAVs,Tractors - Number of Persons - Killed - 2014'] +  
    df6['Other Motor Vehicles - Number of Persons - Killed - 2014'] +  
    df6['Other Vehicles/Objects - Number of Persons - Killed - 2014']  
)  
  
vehicle\_defect\_data['Total Injuries - Defects in Vehicles'] = (  
    df6['Two-Wheelers - Number of Persons - Injured - 2014'] +  
    df6['Auto-Rickshaws - Number of Persons - Injured - 2014'] +  
    df6['Cars, Jeeps,Taxis - Number of Persons - Injured - 2014'] +  
    df6['Buses - Number of Persons - Injured - 2014'] +  
    df6['Trucks, Tempos,MAVs,Tractors - Number of Persons - Injured - 2014'] +  
    df6['Other Motor Vehicles - Number of Persons - Injured - 2014'] +  
    df6['Other Vehicles/Objects - Number of Persons - Injured - 2014']  
)  
  
# Calculate per 1L population for each  
vehicle\_defect\_data['Accidents per 1L - Defects in Vehicles'] = vehicle\_defect\_data['Total Accidents - Defects in Vehicles'] / (df6['Population'] / 100000)  
vehicle\_defect\_data['Deaths per 1L - Defects in Vehicles'] = vehicle\_defect\_data['Total Deaths - Defects in Vehicles'] / (df6['Population'] / 100000)  
vehicle\_defect\_data['Injuries per 1L - Defects in Vehicles'] = vehicle\_defect\_data['Total Injuries - Defects in Vehicles'] / (df6['Population'] / 100000)  
  
# Display the resulting DataFrame  
print(vehicle\_defect\_data[['States/UTs',  
                            'Accidents per 1L - Defects in Vehicles',  
                            'Deaths per 1L - Defects in Vehicles',  
                            'Injuries per 1L - Defects in Vehicles']])

**Output:**

States/UTs Accidents per 1L - Defects in Vehicles \

0 Andhra Pradesh 46.631741

1 Arunachal Pradesh 14.815061

2 Assam 22.893344

3 Bihar 9.179683

4 Chhattisgarh 54.104102

5 Goa 289.946488

6 Gujarat 39.232496

7 Haryana 42.111970

8 Himachal Pradesh 44.547375

9 Jammu & Kashmir 46.733585

10 Jharkhand 15.766275

11 Karnataka 71.548879

12 Kerala 108.609034

13 Madhya Pradesh 73.625705

14 Maharashtra 54.840815

15 Manipur 26.017283

16 Meghalaya 18.268294

17 Mizoram 12.030558

18 Nagaland 15.415703

19 Odisha 22.985538

20 Punjab 23.036161

21 Rajasthan 35.927880

22 Sikkim 33.247240

23 Tamil Nadu 93.212430

24 Telangana 62.247945

25 Tripura 19.488736

26 Uttarakhand 13.979369

27 Uttar Pradesh 15.531573

28 West Bengal 14.105552

29 Andaman & Nicobar Islands 57.280842

30 Chandigarh 34.961391

31 Dadra & Nagar Haveli 25.312110

32 Daman & Diu 16.033086

33 Delhi 51.364250

34 Lakshadweep 1.551037

35 Puducherry 89.025789

36 Total 40.417722

Deaths per 1L - Defects in Vehicles Injuries per 1L - Defects in Vehicles

0 15.088536 57.108619

1 8.599962 22.258726

2 8.081889 20.826406

3 4.719525 6.378516

4 15.744642 51.504788

5 19.882828 128.827016

6 13.161880 37.215610

7 17.683398 35.280017

8 17.466417 81.228307

9 7.909865 64.132097

10 7.966501 13.204748

11 17.107700 93.020253

12 12.120555 123.019592

13 11.798673 76.190873

14 11.393171 36.000214

15 5.882777 45.346408

16 4.752453 10.482360

17 9.387481 21.326898

18 4.094006 11.624957

19 9.365273 26.413833

20 16.656251 14.875643

21 15.009824 40.049053

22 9.662991 57.650386

23 21.054228 107.731392

24 21.410714 67.078222

25 5.117154 33.343159

26 8.704884 15.179017

27 8.151148 11.178989

28 6.436514 13.166643

29 6.043392 74.359992

30 12.411767 31.740016

31 17.165684 27.930604

32 6.166571 20.144133

33 9.953573 49.338987

34 0.000000 1.551037

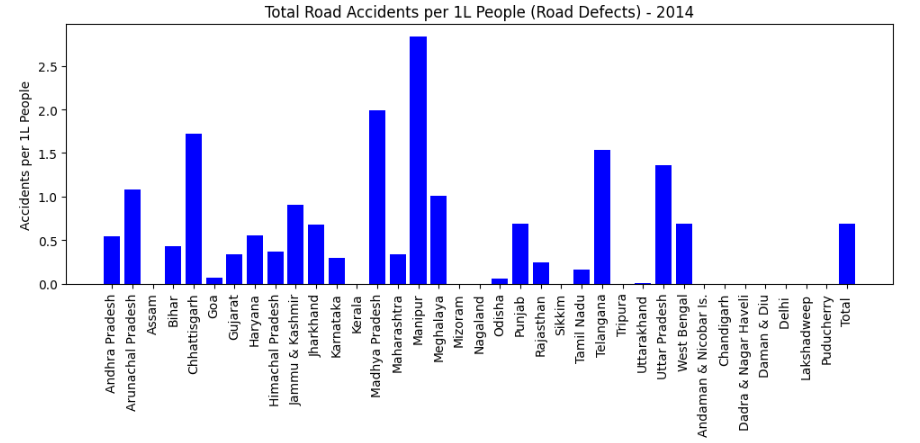
35 12.099815 106.013608

36 11.534907 40.754179

**22. Total number of ROAD ACCIDENTS, INJURIES, DEATHS due to DEFECTS IN THE ROAD CONDITION per 1L people of that state.**

# Filter relevant columns  
columns\_defects = ['States/UTs',  
                   'Defect in Road Condition-Total No. of Road Accidents - 2014',  
                   'Defect in Road Condition-Number of Persons-Killed - 2014',  
                   'Defect in Road Condition-Number of Persons-Injured - 2014',  
                   'Population']  
  
df\_defects = df5[columns\_defects]  
  
# Convert 'Population' and 'Defect in Road Condition-Total No. of Road Accidents - 2014' columns to numeric  
# errors='coerce' will replace non-numeric values with NaN  
df\_defects['Population'] = pd.to\_numeric(df\_defects['Population'], errors='coerce')  
df\_defects['Defect in Road Condition-Total No. of Road Accidents - 2014'] = pd.to\_numeric(df\_defects['Defect in Road Condition-Total No. of Road Accidents - 2014'], errors='coerce')  
  
  
# Calculate per 1L people for each metric (assuming population is in absolute numbers)  
df\_defects['Accidents per 1L'] = (df\_defects['Defect in Road Condition-Total No. of Road Accidents - 2014'] / df\_defects['Population']) \* 100000  
df\_defects['Deaths per 1L'] = (df\_defects['Defect in Road Condition-Number of Persons-Killed - 2014'] / df\_defects['Population']) \* 100000  
df\_defects['Injuries per 1L'] = (df\_defects['Defect in Road Condition-Number of Persons-Injured - 2014'] / df\_defects['Population']) \* 100000  
  
  
# Plotting the data for accidents, deaths, and injuries per 1L people  
states = df\_defects['States/UTs']  
  
# Create subplots to compare Accidents, Deaths, Injuries  
fig, ax = plt.subplots(3, 1, figsize=(10, 15))  
  
# Plotting Total Road Accidents per 1L people due to road defects  
ax[0].bar(states, df\_defects['Accidents per 1L'], color='blue')  
ax[0].set\_title('Total Road Accidents per 1L People (Road Defects) - 2014')  
ax[0].set\_ylabel('Accidents per 1L People')  
ax[0].tick\_params(axis='x', rotation=90)  
  
# Plotting Number of Persons Killed per 1L people due to road defects  
ax[1].bar(states, df\_defects['Deaths per 1L'], color='red')  
ax[1].set\_title('Number of Persons Killed per 1L People (Road Defects) - 2014')  
ax[1].set\_ylabel('Deaths per 1L People')  
ax[1].tick\_params(axis='x', rotation=90)  
  
# Plotting Number of Persons Injured per 1L people due to road defects  
ax[2].bar(states, df\_defects['Injuries per 1L'], color='green')  
ax[2].set\_title('Number of Persons Injured per 1L People (Road Defects) - 2014')  
ax[2].set\_ylabel('Injuries per 1L People')  
ax[2].tick\_params(axis='x', rotation=90)  
  
# Layout adjustment  
plt.tight\_layout()  
  
# Display the plot  
plt.show()

**Output:**



A graph of people killed by people

Description automatically generated with medium confidence

A graph with green bars

Description automatically generated with medium confidence

**23. Total number of ROAD ACCIDENTS, INJURIES, DEATHS due to WEATHER CONDITION per 1L people of that state.**

# List of relevant columns for weather condition  
columns\_weather = ['States/UTs',  
                   'Weather Condition-Total No. of Road Accidents - 2014',  
                   'Weather Condition-Total No. of Road Accidents - 2014 per 1L people',  
                   'Weather Condition-Number of Persons-Killed - 2014',  
                   'Weather Condition-Number of Persons-Killed - 2014 per 1L people',  
                   'Weather Condition-Number of Persons-Injured - 2014',  
                   'Weather Condition-Number of Persons-Injured - 2014 per 1L people']  
  
# Filter the dataset for relevant columns  
df\_weather = df5[columns\_weather]  
  
# Convert numerical columns to appropriate types  
df\_weather.iloc[:, 1:] = df\_weather.iloc[:, 1:].apply(pd.to\_numeric, errors='coerce')  
  
# Display the results (States/UTs with road accidents, injuries, and deaths per 1L people due to weather conditions)  
print(df\_weather)

**Output:**

States/UTs \

0 Andhra Pradesh

1 Arunachal Pradesh

2 Assam

3 Bihar

4 Chhattisgarh

5 Goa

6 Gujarat

7 Haryana

8 Himachal Pradesh

9 Jammu & Kashmir

10 Jharkhand

11 Karnataka

12 Kerala

13 Madhya Pradesh

14 Maharashtra

15 Manipur

16 Meghalaya

17 Mizoram

18 Nagaland

19 Odisha

20 Punjab

21 Rajasthan

22 Sikkim

23 Tamil Nadu

24 Telangana

25 Tripura

26 Uttarakhand

27 Uttar Pradesh

28 West Bengal

29 Andaman & Nicobar Is.

30 Chandigarh

31 Dadra & Nagar Haveli

32 Daman & Diu

33 Delhi

34 Lakshadweep

35 Puducherry

36 Total

Weather Condition-Total No. of Road Accidents - 2014 \

0 60.0

1 23.0

2 0.0

3 409.0

4 212.0

5 1.0

6 194.0

7 127.0

8 0.0

9 20.0

10 203.0

11 679.0

12 0.0

13 539.0

14 102.0

15 110.0

16 28.0

17 0.0

18 0.0

19 11.0

20 311.0

21 23.0

22 10.0

23 36.0

24 338.0

25 0.0

26 0.0

27 1582.0

28 943.0

29 0.0

30 0.0

31 0.0

32 0.0

33 NaN

34 0.0

35 0.0

36 5961.0

Weather Condition-Total No. of Road Accidents - 2014 per 1L people \

0 0.114481

1 1.662178

2 0.000000

3 0.392894

4 0.829902

5 0.068561

6 0.320981

7 0.500957

8 0.000000

9 0.159473

10 0.615373

11 1.111379

12 0.000000

13 0.742150

14 0.090768

15 3.851818

16 0.943749

17 0.000000

18 0.000000

19 0.026207

20 1.120990

21 0.033553

22 1.637795

23 0.049898

24 1.047903

25 0.000000

26 0.000000

27 0.791743

28 1.033129

29 0.000000

30 0.000000

31 0.000000

32 0.000000

33 NaN

34 0.000000

35 0.000000

36 0.492297

Weather Condition-Number of Persons-Killed - 2014 \

0 19

1 13

2 0

3 210

4 60

5 0

6 43

7 75

8 0

9 2

10 94

11 202

12 0

13 83

14 41

15 15

16 9

17 0

18 0

19 11

20 232

21 11

22 2

23 13

24 183

25 0

26 0

27 838

28 398

29 0

30 0

31 0

32 0

33 0

34 0

35 0

36 2554

Weather Condition-Number of Persons-Killed - 2014 per 1L people \

0 0.036252

1 0.939492

2 0.000000

3 0.201730

4 0.234878

5 0.000000

6 0.071145

7 0.295841

8 0.000000

9 0.015947

10 0.284951

11 0.330631

12 0.000000

13 0.114283

14 0.036485

15 0.525248

16 0.303348

17 0.000000

18 0.000000

19 0.026207

20 0.836237

21 0.016047

22 0.327559

23 0.018019

24 0.567356

25 0.000000

26 0.000000

27 0.419394

28 0.436040

29 0.000000

30 0.000000

31 0.000000

32 0.000000

33 0.000000

34 0.000000

35 0.000000

36 0.210925

Weather Condition-Number of Persons-Injured - 2014 \

0 70.0

1 25.0

2 0.0

3 306.0

4 206.0

5 0.0

6 241.0

7 68.0

8 0.0

9 36.0

10 230.0

11 307.0

12 0.0

13 448.0

14 92.0

15 127.0

16 14.0

17 0.0

18 0.0

19 9.0

20 166.0

21 26.0

22 16.0

23 56.0

24 920.0

25 0.0

26 0.0

27 1185.0

28 929.0

29 0.0

30 0.0

31 0.0

32 0.0

33 NaN

34 0.0

35 0.0

36 5477.0

Weather Condition-Number of Persons-Injured - 2014 per 1L people

0 0.133561

1 1.806715

2 0.000000

3 0.293950

4 0.806414

5 0.000000

6 0.398745

7 0.268229

8 0.000000

9 0.287052

10 0.697220

11 0.502494

12 0.000000

13 0.616852

14 0.081869

15 4.447099

16 0.471875

17 0.000000

18 0.000000

19 0.021442

20 0.598342

21 0.037929

22 2.620472

23 0.077619

24 2.852282

25 0.000000

26 0.000000

27 0.593056

28 1.017791

29 0.000000

30 0.000000

31 0.000000

32 0.000000

33 NaN

34 0.000000

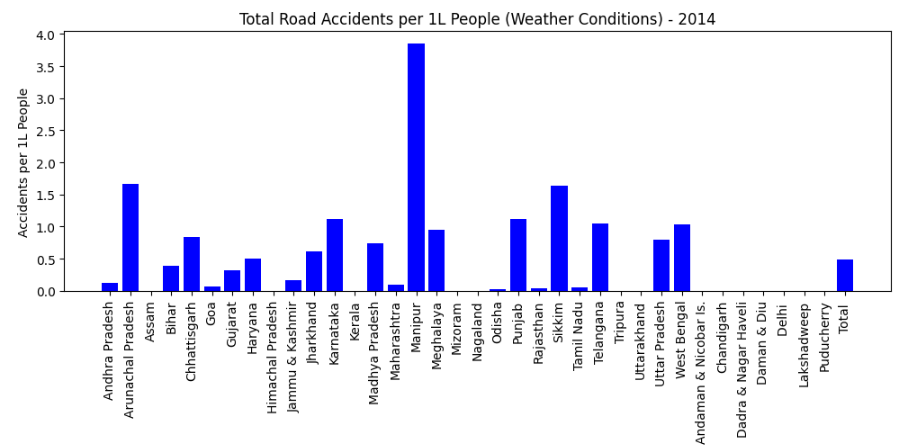
35 0.000000

36 0.452325

**Another Method**

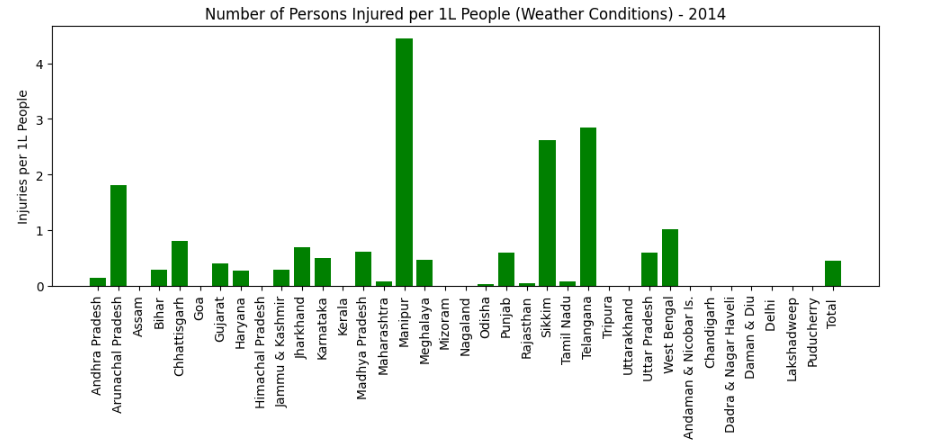
# Filter relevant columns  
columns\_weather = ['States/UTs',  
                   'Weather Condition-Total No. of Road Accidents - 2014',  
                   'Weather Condition-Total No. of Road Accidents - 2014 per 1L people',  
                   'Weather Condition-Number of Persons-Killed - 2014',  
                   'Weather Condition-Number of Persons-Killed - 2014 per 1L people',  
                   'Weather Condition-Number of Persons-Injured - 2014',  
                   'Weather Condition-Number of Persons-Injured - 2014 per 1L people']  
  
df\_weather = df5[columns\_weather]  
  
# Plotting the data for each metric: Accidents, Deaths, Injuries per 1L people  
states = df\_weather['States/UTs']  
  
# Create subplots to compare Accidents, Deaths, Injuries  
fig, ax = plt.subplots(3, 1, figsize=(10, 15))  
  
# Plotting Total Road Accidents per 1L people  
ax[0].bar(states, df\_weather['Weather Condition-Total No. of Road Accidents - 2014 per 1L people'], color='blue')  
ax[0].set\_title('Total Road Accidents per 1L People (Weather Conditions) - 2014')  
ax[0].set\_ylabel('Accidents per 1L People')  
ax[0].tick\_params(axis='x', rotation=90)  
  
# Plotting Number of Persons Killed per 1L people  
ax[1].bar(states, df\_weather['Weather Condition-Number of Persons-Killed - 2014 per 1L people'], color='red')  
ax[1].set\_title('Number of Persons Killed per 1L People (Weather Conditions) - 2014')  
ax[1].set\_ylabel('Deaths per 1L People')  
ax[1].tick\_params(axis='x', rotation=90)  
  
# Plotting Number of Persons Injured per 1L people  
ax[2].bar(states, df\_weather['Weather Condition-Number of Persons-Injured - 2014 per 1L people'], color='green')  
ax[2].set\_title('Number of Persons Injured per 1L People (Weather Conditions) - 2014')  
ax[2].set\_ylabel('Injuries per 1L People')  
ax[2].tick\_params(axis='x', rotation=90)  
  
# Layout adjustment  
plt.tight\_layout()  
  
# Display the plot  
plt.show()

**Output:**



A graph of a number of people killed by people

Description automatically generated



**24. Number of Total Accidents for each vehicle type per 1L people of that state.**

# Select relevant columns for total accidents per 1L people for 2014  
accidents\_per\_1L\_columns = [  
    'Two-Wheelers - Number of Road Accidents - Total - 2014 per 1L people',  
    'Auto-Rickshaws - Number of Road Accidents - Total - 2014 per 1L people',  
    'Cars, Jeeps,Taxis - Number of Road Accidents - Total - 2014 per 1L people',  
    'Buses - Number of Road Accidents - Total - 2014 per 1L people',  
    'Trucks, Tempos,MAVs,Tractors - Number of Road Accidents - Total - 2014 per 1L people',  
    'Other Motor Vehicles - Number of Road Accidents - Total - 2014 per 1L people',  
    'Other Vehicles/Objects - Number of Road Accidents - Total - 2014 per 1L people'  
]  
  
# Rename columns for readability  
df\_renamed = df6[['States/UTs'] + accidents\_per\_1L\_columns].rename(columns={  
    'Two-Wheelers - Number of Road Accidents - Total - 2014 per 1L people': 'Two-Wheelers per 1L people',  
    'Auto-Rickshaws - Number of Road Accidents - Total - 2014 per 1L people': 'Auto-Rickshaws per 1L people',  
    'Cars, Jeeps,Taxis - Number of Road Accidents - Total - 2014 per 1L people': 'Cars, Jeeps, Taxis per 1L people',  
    'Buses - Number of Road Accidents - Total - 2014 per 1L people': 'Buses per 1L people',  
    'Trucks, Tempos,MAVs,Tractors - Number of Road Accidents - Total - 2014 per 1L people': 'Trucks, Tempos, etc. per 1L people',  
    'Other Motor Vehicles - Number of Road Accidents - Total - 2014 per 1L people': 'Other Motor Vehicles per 1L people',  
    'Other Vehicles/Objects - Number of Road Accidents - Total - 2014 per 1L people': 'Other Vehicles/Objects per 1L people'  
})  
  
# Display the result  
print(df\_renamed)

**Output:**

States/UTs Two-Wheelers per 1L people \

0 Andhra Pradesh 13.812077

1 Arunachal Pradesh 2.457132

2 Assam 4.832470

3 Bihar 1.681085

4 Chhattisgarh 18.739334

5 Goa 130.746737

6 Gujarat 9.202562

7 Haryana 4.461281

8 Himachal Pradesh 11.857934

9 Jammu & Kashmir 8.778993

10 Jharkhand 3.607358

11 Karnataka 17.598736

12 Kerala 36.259887

13 Madhya Pradesh 27.108447

14 Maharashtra 12.537561

15 Manipur 7.038323

16 Meghalaya 1.516740

17 Mizoram 2.916499

18 Nagaland 2.830424

19 Odisha 6.282428

20 Punjab 5.759941

21 Rajasthan 8.687288

22 Sikkim 5.077165

23 Tamil Nadu 33.276491

24 Telangana 14.249007

25 Tripura 3.865085

26 Uttarakhand 2.290237

27 Uttar Pradesh 2.451801

28 West Bengal 3.849857

29 Andaman & Nicobar Islands 24.173566

30 Chandigarh 5.779525

31 Dadra & Nagar Haveli 5.236988

32 Daman & Diu 5.755467

33 Delhi 7.773437

34 Lakshadweep 1.551037

35 Puducherry 42.709942

36 Total 11.036334

Auto-Rickshaws per 1L people Cars, Jeeps, Taxis per 1L people \

0 8.475376 7.351559

1 1.228566 3.035281

2 0.935730 5.165743

3 0.340059 1.431324

4 1.217450 11.657768

5 2.331090 94.203470

6 3.616828 7.304802

7 1.597541 10.441212

8 11.610287 9.672811

9 0.510314 5.780899

10 0.760880 2.352361

11 4.276925 12.539427

12 15.078102 26.097061

13 2.815765 15.516309

14 2.959751 16.372066

15 1.470694 6.232943

16 0.910044 3.707587

17 0.182281 2.460796

18 2.931511 3.790747

19 1.369889 3.721332

20 0.666827 6.217709

21 0.716282 10.048369

22 0.000000 20.144879

23 4.033430 25.802864

24 7.958486 11.176603

25 2.449702 3.756209

26 0.138802 4.352442

27 0.738693 3.172477

28 0.279372 1.605020

29 6.831660 15.239857

30 0.852717 18.475532

31 1.163775 2.909438

32 0.411105 2.466629

33 1.358118 16.714378

34 0.000000 0.000000

35 3.044986 18.510312

36 2.574379 9.185080

Buses per 1L people Trucks, Tempos, etc. per 1L people \

0 3.304672 8.250231

1 1.084029 3.035281

2 2.012461 5.659245

3 0.923156 2.716633

4 2.642375 11.379830

5 24.270763 21.596865

6 1.927541 9.030489

7 2.540287 11.573297

8 5.958102 5.433673

9 12.678109 10.876064

10 1.794585 4.774444

11 6.299994 13.483853

12 14.497369 11.776306

13 4.627768 15.480509

14 3.923494 9.647221

15 1.330628 4.447099

16 1.752678 2.595311

17 0.455703 3.919045

18 1.061409 0.859236

19 1.824930 6.439667

20 2.014898 4.988585

21 2.655057 9.314581

22 0.000000 0.000000

23 9.664986 12.417698

24 4.517146 13.904873

25 0.734910 3.484020

26 1.447509 4.550731

27 1.165093 4.193935

28 1.382618 3.768784

29 5.517879 3.415830

30 3.316121 3.316121

31 1.454719 10.183033

32 0.411105 3.288838

33 3.162985 7.308818

34 0.000000 0.000000

35 7.291941 13.462045

36 3.416181 7.946038

Other Motor Vehicles per 1L people Other Vehicles/Objects per 1L people

0 2.997482 2.440344

1 0.000000 3.974772

2 3.489761 0.797934

3 0.572529 1.514898

4 6.451310 2.016034

5 0.205684 16.591878

6 2.916957 5.233316

7 5.013518 6.484833

8 0.000000 0.014567

9 7.582945 0.526261

10 1.133741 1.342907

11 9.067801 8.282143

12 4.900308 0.000000

13 7.001547 1.075360

14 6.367112 3.033611

15 5.497595 0.000000

16 5.628792 2.157142

17 2.096234 0.000000

18 3.942377 0.000000

19 2.401474 0.945819

20 2.083383 1.304818

21 4.484420 0.021882

22 7.861416 0.163780

23 2.885774 5.131188

24 5.747967 4.693863

25 3.892304 1.306507

26 0.892300 0.307348

27 1.812200 1.997374

28 2.128706 1.091195

29 2.102049 0.000000

30 3.221375 0.000000

31 4.073213 0.290944

32 3.699943 0.000000

33 1.888260 13.158254

34 0.000000 0.000000

35 4.006561 0.000000

36 3.632227 2.627482

**25. Number of Persons Killed for each vehicle type per 1L people of that state.**

# Select relevant columns for persons killed per 1L people for 2014  
persons\_killed\_per\_1L\_columns = [  
    'Two-Wheelers - Number of Persons - Killed - 2014 per 1L people',  
    'Auto-Rickshaws - Number of Persons - Killed - 2014 per 1L people',  
    'Cars, Jeeps,Taxis - Number of Persons - Killed - 2014 per 1L people',  
    'Buses - Number of Persons - Killed - 2014 per 1L people',  
    'Trucks, Tempos,MAVs,Tractors - Number of Persons - Killed - 2014 per 1L people',  
    'Other Motor Vehicles - Number of Persons - Killed - 2014 per 1L people',  
    'Other Vehicles/Objects - Number of Persons - Killed - 2014 per 1L people'  
]  
  
# Rename columns for readability  
df\_renamed = df6[['States/UTs'] + persons\_killed\_per\_1L\_columns].rename(columns={  
    'Two-Wheelers - Number of Persons - Killed - 2014 per 1L people': 'Two-Wheelers Killed per 1L people',  
    'Auto-Rickshaws - Number of Persons - Killed - 2014 per 1L people': 'Auto-Rickshaws Killed per 1L people',  
    'Cars, Jeeps,Taxis - Number of Persons - Killed - 2014 per 1L people': 'Cars, Jeeps, Taxis Killed per 1L people',  
    'Buses - Number of Persons - Killed - 2014 per 1L people': 'Buses Killed per 1L people',  
    'Trucks, Tempos,MAVs,Tractors - Number of Persons - Killed - 2014 per 1L people': 'Trucks, Tempos, etc. Killed per 1L people',  
    'Other Motor Vehicles - Number of Persons - Killed - 2014 per 1L people': 'Other Motor Vehicles Killed per 1L people',  
    'Other Vehicles/Objects - Number of Persons - Killed - 2014 per 1L people': 'Other Vehicles/Objects Killed per 1L people'  
})  
  
# Display the result  
print(df\_renamed)

**Output:**

States/UTs Two-Wheelers Killed per 1L people \

0 Andhra Pradesh 3.821742

1 Arunachal Pradesh 1.517640

2 Assam 1.720846

3 Bihar 0.787708

4 Chhattisgarh 4.407873

5 Goa 9.187238

6 Gujarat 2.794521

7 Haryana 1.254366

8 Himachal Pradesh 2.461905

9 Jammu & Kashmir 2.041255

10 Jharkhand 1.840056

11 Karnataka 4.975833

12 Kerala 2.882710

13 Madhya Pradesh 3.104914

14 Maharashtra 3.620044

15 Manipur 1.155546

16 Meghalaya 0.471875

17 Mizoram 1.458249

18 Nagaland 0.606519

19 Odisha 2.501536

20 Punjab 4.246064

21 Rajasthan 2.774680

22 Sikkim 0.818898

23 Tamil Nadu 6.520019

24 Telangana 4.337328

25 Tripura 1.061537

26 Uttarakhand 0.951787

27 Uttar Pradesh 1.247170

28 West Bengal 1.460404

29 Andaman & Nicobar Islands 1.313781

30 Chandigarh 1.894926

31 Dadra & Nagar Haveli 4.073213

32 Daman & Diu 3.288838

33 Delhi 0.851802

34 Lakshadweep 0.000000

35 Puducherry 4.407217

36 Total 2.686036

Auto-Rickshaws Killed per 1L people \

0 1.864125

1 0.505880

2 0.240342

3 0.138329

4 0.144841

5 0.137123

6 0.986107

7 0.710018

8 4.413949

9 0.119605

10 0.391050

11 0.721823

12 0.793269

13 0.257481

14 0.555287

15 0.175083

16 0.269643

17 0.000000

18 0.909779

19 0.357362

20 0.364051

21 0.147341

22 0.000000

23 0.433836

24 2.080305

25 0.517159

26 0.079316

27 0.388865

28 0.098602

29 0.788268

30 0.094746

31 0.581888

32 0.000000

33 0.160830

34 0.000000

35 0.240394

36 0.518559

Cars, Jeeps, Taxis Killed per 1L people Buses Killed per 1L people \

0 1.738196 1.081841

1 1.806715 0.505880

2 1.708028 0.788321

3 0.755047 0.521617

4 2.536680 0.739865

5 4.319373 1.576914

6 2.265068 0.724689

7 3.901156 1.301700

8 6.467964 1.835503

9 1.355521 0.837234

10 1.203463 0.821508

11 2.517379 1.700622

12 3.188044 2.457638

13 2.172751 0.956947

14 1.792224 0.424474

15 1.050496 0.350165

16 1.280803 0.370759

17 1.913952 0.455703

18 0.960323 0.353803

19 1.398478 0.790962

20 4.267691 1.402138

21 3.871715 1.204987

22 6.714960 0.000000

23 4.858135 2.748554

24 2.824379 1.624560

25 1.170413 0.244970

26 2.508355 1.338450

27 1.585488 0.626087

28 0.738419 0.734037

29 1.576537 0.788268

30 4.926808 2.179165

31 1.745663 1.163775

32 0.000000 0.411105

33 1.775084 0.673102

34 0.000000 0.000000

35 1.442362 0.961575

36 2.150381 1.006314

Trucks, Tempos, etc. Killed per 1L people \

0 3.354280

1 1.878983

2 2.041302

3 1.574456

4 4.235630

5 2.262529

6 3.443102

7 5.336970

8 2.287095

9 1.770151

10 2.503931

11 4.404594

12 1.882892

13 3.899387

14 2.260303

15 1.750827

16 1.145982

17 3.645624

18 0.404346

19 2.923223

20 3.665745

21 4.834538

22 0.000000

23 3.370894

24 6.222314

25 1.007100

26 3.053649

27 2.252614

28 1.966560

29 1.313781

30 1.421195

31 6.982651

32 1.233314

33 2.162266

34 0.000000

35 2.003281

36 2.949238

Other Motor Vehicles Killed per 1L people \

0 1.541671

1 0.000000

2 1.285027

3 0.279540

4 3.076899

5 0.000000

6 1.312052

7 2.386450

8 0.000000

9 1.714336

10 0.588090

11 1.484566

12 0.916001

13 1.320449

14 1.326815

15 1.400661

16 0.539285

17 1.913952

18 0.859236

19 1.022056

20 1.585966

21 2.160516

22 1.965354

23 1.835141

24 3.035200

25 0.707692

26 0.674182

27 0.948890

28 0.821683

29 0.262756

30 1.894926

31 2.327550

32 1.233314

33 0.327616

34 0.000000

35 3.044986

36 1.270177

Other Vehicles/Objects Killed per 1L people

0 1.686680

1 2.384863

2 0.298024

3 0.662828

4 0.602853

5 2.399652

6 1.636342

7 2.792738

8 0.000000

9 0.071763

10 0.618404

11 1.302883

12 0.000000

13 0.086745

14 1.414024

15 0.000000

16 0.674107

17 0.000000

18 0.000000

19 0.371657

20 1.124594

21 0.016047

22 0.163780

23 1.287648

24 1.286627

25 0.408284

26 0.099144

27 1.102034

28 0.616810

29 0.000000

30 0.000000

31 0.290944

32 0.000000

33 4.002873

34 0.000000

35 0.000000

36 0.954202

**26. Total accidents, fatal accidents, killed and injured for each state per 1L people of that state.**

# Select relevant columns for accidents, fatal, killed, and injured per 1L people for 2014  
accident\_stats\_columns = [  
    'States/UTs',  
    'Two-Wheelers - Number of Road Accidents - Total - 2014 per 1L people',  
    'Two-Wheelers - Number of Road Accidents - Fatal - 2014 per 1L people',  
    'Two-Wheelers - Number of Persons - Killed - 2014 per 1L people',  
    'Two-Wheelers - Number of Persons - Injured - 2014 per 1L people'  
]  
  
# Rename columns for readability  
df\_renamed = df6[accident\_stats\_columns].rename(columns={  
    'Two-Wheelers - Number of Road Accidents - Total - 2014 per 1L people': 'Total Accidents per 1L people',  
    'Two-Wheelers - Number of Road Accidents - Fatal - 2014 per 1L people': 'Fatal Accidents per 1L people',  
    'Two-Wheelers - Number of Persons - Killed - 2014 per 1L people': 'Persons Killed per 1L people',  
    'Two-Wheelers - Number of Persons - Injured - 2014 per 1L people': 'Persons Injured per 1L people'  
})  
  
# Display the result  
print(df\_renamed)

**Output:**

States/UTs Total Accidents per 1L people \

0 Andhra Pradesh 13.812077

1 Arunachal Pradesh 2.457132

2 Assam 4.832470

3 Bihar 1.681085

4 Chhattisgarh 18.739334

5 Goa 130.746737

6 Gujarat 9.202562

7 Haryana 4.461281

8 Himachal Pradesh 11.857934

9 Jammu & Kashmir 8.778993

10 Jharkhand 3.607358

11 Karnataka 17.598736

12 Kerala 36.259887

13 Madhya Pradesh 27.108447

14 Maharashtra 12.537561

15 Manipur 7.038323

16 Meghalaya 1.516740

17 Mizoram 2.916499

18 Nagaland 2.830424

19 Odisha 6.282428

20 Punjab 5.759941

21 Rajasthan 8.687288

22 Sikkim 5.077165

23 Tamil Nadu 33.276491

24 Telangana 14.249007

25 Tripura 3.865085

26 Uttarakhand 2.290237

27 Uttar Pradesh 2.451801

28 West Bengal 3.849857

29 Andaman & Nicobar Islands 24.173566

30 Chandigarh 5.779525

31 Dadra & Nagar Haveli 5.236988

32 Daman & Diu 5.755467

33 Delhi 7.773437

34 Lakshadweep 1.551037

35 Puducherry 42.709942

36 Total 11.036334

Fatal Accidents per 1L people Persons Killed per 1L people \

0 3.741606 3.821742

1 1.084029 1.517640

2 1.567028 1.720846

3 0.738717 0.787708

4 4.204313 4.407873

5 8.912992 9.187238

6 2.539722 2.794521

7 1.183364 1.254366

8 2.403635 2.461905

9 1.802046 2.041255

10 1.630889 1.840056

11 4.047775 4.975833

12 2.846789 2.882710

13 2.786850 3.104914

14 3.305915 3.620044

15 1.155546 1.155546

16 0.471875 0.471875

17 1.367109 1.458249

18 0.505433 0.606519

19 2.356208 2.501536

20 3.892826 4.246064

21 2.700280 2.774680

22 0.818898 0.818898

23 6.334287 6.520019

24 4.160611 4.337328

25 1.007100 1.061537

26 0.803070 0.951787

27 1.095528 1.247170

28 1.377140 1.460404

29 1.051025 1.313781

30 1.894926 1.894926

31 3.491326 4.073213

32 3.288838 3.288838

33 0.816062 0.851802

34 0.000000 0.000000

35 3.605905 4.407217

36 2.482874 2.686036

Persons Injured per 1L people

0 13.991430

1 2.168058

2 4.300513

3 1.086461

4 16.813336

5 57.454518

6 8.476218

7 5.404028

8 15.776588

9 5.653321

10 2.704003

11 22.800446

12 35.284016

13 25.822420

14 10.064576

15 11.310340

16 0.842633

17 2.460796

18 2.729338

19 6.227632

20 3.557611

21 8.987805

22 4.913385

23 34.777592

24 11.548640

25 3.919522

26 2.478612

27 1.758150

28 2.639245

29 25.750103

30 7.200720

31 4.364157

32 9.044305

33 8.107010

34 1.551037

35 47.998603

36 10.525786

**27. Number of Accidents happening in DAY and NIGHT TIME for 2014 and 2016.**

df7 = pd.read\_excel('timeOfOccurence.xls')  
  
# Define the columns for 2014  
daytime\_columns\_2014 = [  
    '06-900hrs - Day - 2014',  
    '09-1200hrs - Day - 2014',  
    '12-1500hrs - Day - 2014',  
    '15-1800hrs - Day - 2014'  
]  
  
nighttime\_columns\_2014 = [  
    '18-2100hrs - Night - 2014',  
    '21-2400hrs - Night - 2014',  
    '00-300hrs - Night - 2014',  
    '03-600hrs - Night - 2014'  
]  
  
# Calculate daytime and nighttime accidents for 2014  
df7['Total Daytime Accidents - 2014'] = df7[daytime\_columns\_2014].sum(axis=1)  
df7['Total Nighttime Accidents - 2014'] = df7[nighttime\_columns\_2014].sum(axis=1)  
  
# Define the columns for 2016  
daytime\_columns\_2016 = [  
    '06-900hrs - (Day) - 2016',  
    '09-1200hrs - (Day) - 2016',  
    '12-1500hrs - (Day) - 2016',  
    '15-1800hrs - (Day) - 2016'  
]  
  
nighttime\_columns\_2016 = [  
    '18-2100hrs - (Night) - 2016',  
    '21-2400hrs - (Night) - 2016',  
    '00-300hrs - (Night) - 2016',  
    '03-600hrs - (Night) - 2016'  
]  
  
# Calculate daytime and nighttime accidents for 2016  
df7['Total Daytime Accidents - 2016'] = df7[daytime\_columns\_2016].sum(axis=1)  
df7['Total Nighttime Accidents - 2016'] = df7[nighttime\_columns\_2016].sum(axis=1)  
  
# Display the results  
result = df7[['States/Uts', 'Total Daytime Accidents - 2014', 'Total Nighttime Accidents - 2014',  
             'Total Daytime Accidents - 2016', 'Total Nighttime Accidents - 2016']]  
print(result)

**Output:**

States/Uts Total Daytime Accidents - 2014 \

0 Andhra Pradesh 13093

1 Arunachal Pradesh 121

2 Assam 5215

3 Bihar 6063

4 Chhattisgarh 8607

5 Goa 2591

6 Gujarat 13897

7 Haryana 5758

8 Himachal Pradesh 1951

9 Jammu & Kashmir 4065

10 Jharkhand 3382

11 Karnataka 24860

12 Kerala 24638

13 Madhya Pradesh 35740

14 Maharashtra 34923

15 Manipur 501

16 Meghalaya 187

17 Mizoram 71

18 Nagaland 282

19 Orissa 5545

20 Punjab 3683

21 Rajasthan 15201

22 Sikkim 145

23 Tamil Nadu 39423

24 Telangana 10180

25 Tripura 483

26 Uttarakhand 914

27 Uttar Pradesh 17889

28 West Bengal 7633

29 A & N Islands 144

30 Chandigarh 192

31 D & N Haveli 50

32 Daman & Diu 22

33 Delhi 4306

34 Lakshadweep 1

35 Puducherry 651

36 Total 292407

Total Nighttime Accidents - 2014 Total Daytime Accidents - 2016 \

0 11347 14188

1 84 158

2 1929 4793

3 3493 5112

4 5214 8523

5 1638 2617

6 9815 12859

7 4918 6134

8 1107 1867

9 1796 4032

10 1819 2805

11 18853 26600

12 11644 27238

13 17732 34151

14 26704 21701

15 242 334

16 355 290

17 61 54

18 23 53

19 4103 6278

20 2708 3983

21 9427 14632

22 58 153

23 27827 41464

24 9898 12730

25 233 384

26 496 994

27 13145 19994

28 5242 10403

29 74 164

30 177 228

31 37 38

32 17 34

33 4317 3566

34 0 0

35 460 953

36 196993 289507

Total Nighttime Accidents - 2016

0 10700

1 91

2 2642

3 3110

4 5057

5 1687

6 9000

7 5100

8 1301

9 1469

10 2127

11 17803

12 12182

13 19821

14 18177

15 204

16 330

17 29

18 22

19 4254

20 2969

21 8434

22 57

23 29967

24 10081

25 173

26 597

27 15618

28 3177

29 74

30 200

31 32

32 37

33 3809

34 1

35 813

36 191145