

# INDEX

- DF - Analysing Accidents per Lakh Population-State-Year.
- DF1 - Analysing Offender and Victim Deaths per Gender, State.
- DF2 - Analysing Deaths occurred due to improper use of Safety Accessories.
- DF3 - Analysing rate of accidents from the year 2003 to 2016 for each state.
- DF4 - Analysing accidents/injuries/deaths occurring as per number of lanes.
- DF5 - Analysing accidents/injuries/deaths occurring due to various faults/reasons.
- DF6 - Analysing accidents/injuries/deaths as per the types of vehicles.
- DF7 - Analysing number of accidents which take place as per time of occurrence.

df = roadAccStats13-16.csv

df1 = Details\_of\_road\_accident\_deaths\_by\_situation\_state\_2014.csv

df2 = Persons\_killed\_due\_to\_Non-use\_of\_Safety\_Device\_2016.csv

df3 = datafile.xls - total number of accidents from 2003 to 16 per state.

df4 = laneAccidents.csv

df5 = reasonOfAccident.csv

df6 = typeOfVehicle.csv

df7 = timeOfOccurrence.csv

In [278...]

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl
import seaborn as sns
import sklearn
import csv
import os
import xlrd
from collections import defaultdict
import math as m
from jupyterthemes import jtplot
```

In [2]: df = pd.read\_csv('A:\Data Analysis Jupyter\Road-Accidents-Of-India\Database

In [3]: df1 = pd.read\_csv('A:\Data Analysis Jupyter\Road-Accidents-Of-India\Database

In [4]: df2 = pd.read\_csv('A:\Data Analysis Jupyter\Road-Accidents-Of-India\Database

```
In [6]: plt.rcParams['figure.figsize'] = [10, 5]
```

```
In [7]: jtplot.style()
jtplot.style(theme = 'onedork')
jtplot.style(context = 'talk', fscale = 1.4, spines = False, gridlines = '--'
jtplot.style(ticks = True, grid = False, figsize=(6, 4.5))
jtplot.reset()
```

df dataset tells about the **road accidents counts in each stage for years 2013 to 2016**.

```
In [8]: df.head()
```

Out[8]:

Sl. No.	States/UTs	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	Share of States/UTs in Total Number of Road Accidents - 2013	Share of States/UTs in Total Number of Road Accidents - 2014
		Number of Road Accidents during - 2013	Number of Road Accidents during - 2014	Number of Road Accidents during - 2015	Number of Road Accidents during - 2016	Number of Road Accidents - 2013	Number of Road Accidents - 2014
0	1	Andhra Pradesh	43482.0	24440	24258	24888	8.9
1	2	Arunachal Pradesh	308.0	205	284	249	0.1
2	3	Assam	7211.0	7144	6959	7435	1.5
3	4	Bihar	10200.0	9556	9555	8222	2.1
4	5	Chhattisgarh	13657.0	13821	14446	13580	2.8

```
In [9]: df.shape
```

Out[9]: (37, 20)

```
In [10]: df.dtypes
```

```
Out[10]: SI. No.                                     object
States/UTs                                         object
State/UT-Wise Total Number of Road Accidents during - 2013   float64
State/UT-Wise Total Number of Road Accidents during - 2014   int64
State/UT-Wise Total Number of Road Accidents during - 2015   int64
State/UT-Wise Total Number of Road Accidents during - 2016   int64
Share of States/UTs in Total Number of Road Accidents - 2013   float64
Share of States/UTs in Total Number of Road Accidents - 2014   float64
Share of States/UTs in Total Number of Road Accidents - 2015   float64
Share of States/UTs in Total Number of Road Accidents - 2016   float64
Total Number of Accidents Per Lakh Population - 2013      float64
Total Number of Accidents Per Lakh Population - 2014      float64
Total Number of Accidents Per Lakh Population - 2015      float64
Total Number of Accidents Per Lakh Population - 2016      float64
Total Number of Road Accidents per 10,000 Vehicles - 2013   float64
Total Number of Road Accidents per 10,000 Vehicles - 2014   float64
Total Number of Road Accidents per 10,000 Vehicles - 2015   float64
Total Number of Road Accidents per 10,000 Km of Roads - 2013   float64
Total Number of Road Accidents per 10,000 Km of Roads - 2014   float64
Total Number of Road Accidents per 10,000 Km of Roads - 2015   float64
dtype: object
```

```
In [11]: type(df['SI. No.'][0])
```

```
Out[11]: str
```

```
In [12]: type(df['States/UTs'][0])
```

```
Out[12]: str
```

```
In [13]: df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37 entries, 0 to 36
Data columns (total 20 columns):
SI. No.                                         37 non-null o
object                                         37 non-null o
States/UTs                                      37 non-null o
object                                         37 non-null o
State/UT-Wise Total Number of Road Accidents during - 2013   36 non-null f
float64
State/UT-Wise Total Number of Road Accidents during - 2014   37 non-null i
int64
State/UT-Wise Total Number of Road Accidents during - 2015   37 non-null i
int64
State/UT-Wise Total Number of Road Accidents during - 2016   37 non-null i
int64
Share of States/UTs in Total Number of Road Accidents - 2013   37 non-null f
float64
Share of States/UTs in Total Number of Road Accidents - 2014   37 non-null f
float64
Share of States/UTs in Total Number of Road Accidents - 2015   37 non-null f
float64
Share of States/UTs in Total Number of Road Accidents - 2016   37 non-null f
float64
Total Number of Accidents Per Lakh Population - 2013          36 non-null f
float64
Total Number of Accidents Per Lakh Population - 2014          36 non-null f
float64
Total Number of Accidents Per Lakh Population - 2015          36 non-null f
float64
Total Number of Accidents Per Lakh Population - 2016          36 non-null f
float64
Total Number of Road Accidents per 10,000 Vehicles - 2013       36 non-null f
float64
Total Number of Road Accidents per 10,000 Vehicles - 2014       37 non-null f
float64
Total Number of Road Accidents per 10,000 Vehicles - 2015       37 non-null f
float64
Total Number of Road Accidents per 10,000 Km of Roads - 2013     36 non-null f
float64
Total Number of Road Accidents per 10,000 Km of Roads - 2014     37 non-null f
float64
Total Number of Road Accidents per 10,000 Km of Roads - 2015     37 non-null f
float64
dtypes: float64(15), int64(3), object(2)
memory usage: 5.9+ KB

```

## Calculating the Usefulness of columns.

```
In [14]: row, column = df.shape
```

```
In [15]: # CALCULATE Usefullness of columns
round(((row - df.isnull().sum()) / row) * 100,2)
```

```
Out[15]: SI. No.          100.0
States/UTs                  100.0
State/UT-Wise Total Number of Road Accidents during - 2013    97.3
State/UT-Wise Total Number of Road Accidents during - 2014    100.0
State/UT-Wise Total Number of Road Accidents during - 2015    100.0
State/UT-Wise Total Number of Road Accidents during - 2016    100.0
Share of States/UTs in Total Number of Road Accidents - 2013   100.0
Share of States/UTs in Total Number of Road Accidents - 2014   100.0
Share of States/UTs in Total Number of Road Accidents - 2015   100.0
Share of States/UTs in Total Number of Road Accidents - 2016   100.0
Total Number of Accidents Per Lakh Population - 2013        97.3
Total Number of Accidents Per Lakh Population - 2014        97.3
Total Number of Accidents Per Lakh Population - 2015        97.3
Total Number of Accidents Per Lakh Population - 2016        97.3
Total Number of Road Accidents per 10,000 Vehicles - 2013    97.3
Total Number of Road Accidents per 10,000 Vehicles - 2014    100.0
Total Number of Road Accidents per 10,000 Vehicles - 2015    100.0
Total Number of Road Accidents per 10,000 Km of Roads - 2013  97.3
Total Number of Road Accidents per 10,000 Km of Roads - 2014  100.0
Total Number of Road Accidents per 10,000 Km of Roads - 2015  100.0
dtype: float64
```

```
In [16]: df.duplicated()
```

```
Out[16]: 0    False
         1    False
         2    False
         3    False
         4    False
         5    False
         6    False
         7    False
         8    False
         9    False
        10   False
        11   False
        12   False
        13   False
        14   False
        15   False
        16   False
        17   False
        18   False
        19   False
        20   False
        21   False
        22   False
        23   False
        24   False
        25   False
        26   False
        27   False
        28   False
        29   False
        30   False
        31   False
        32   False
        33   False
        34   False
        35   False
        36   False
dtype: bool
```

Check for repetition of values.

```
In [17]: df['States/UTs'].value_counts()
```

```
Out[17]: Jammu & Kashmir          1  
        Madhya Pradesh            1  
        Andhra Pradesh           1  
        West Bengal              1  
        Chhattisgarh             1  
        Arunachal Pradesh         1  
        Puducherry               1  
        Odisha                   1  
        Karnataka                1  
        Delhi                     1  
        Sikkim                   1  
        Total                     1  
        Uttar Pradesh            1  
        Kerala                   1  
        Andaman & Nicobar Islands 1  
        Telangana                 1  
        Punjab                   1  
        Mizoram                  1  
        Rajasthan                1  
        Maharashtra              1  
        Nagaland                 1  
        Himachal Pradesh          1  
        Bihar                     1  
        Goa                      1  
        Jharkhand                 1  
        Lakshadweep               1  
        Gujarat                  1  
        Assam                    1  
        Haryana                  1  
        Manipur                  1  
        Daman & Diu               1  
        Uttarakhand               1  
        Meghalaya                1  
        Dadra & Nagar Haveli      1  
        Chandigarh                1  
        Tripura                  1  
        Tamil Nadu                1  
Name: States/UTs, dtype: int64
```

No repetition present.

---

---

---

#

*df1* dataset shows **vehicle types involved in accidents around the country**.

```
In [18]: df1.head()
```

```
Out[18]:
```

	States/UTs	Year	CrimeHead	Offenders (Driver/Pedestrian) Died_Male	Offenders (Driver/Pedestrian) Died_Female	Offe (Driver/Pedes Died_Transg
0	Andhra Pradesh	2014	Truck/Lorry (Total)	445	30	
1	Andhra Pradesh	2014	Truck/Lorry - Normal Goods Carriers	357	25	
2	Andhra Pradesh	2014	Truck/Lorry - Trailer/Container Carriers	12	0	
3	Andhra Pradesh	2014	Truck/Lorry - Tankers	10	0	
4	Andhra Pradesh	2014	Truck/Lorry - Others	66	5	

```
In [19]:
```

```
df1['CrimeHead'].unique()  
df1['CrimeHead'].nunique()
```

```
Out[19]: 37
```

```
In [20]:
```

```
df1.shape
```

```
Out[20]: (1443, 11)
```

```
In [21]:
```

```
df1.dtypes
```

```
Out[21]: States/UTs          object  
Year              int64  
CrimeHead         object  
Offenders (Driver/Pedestrian) Died_Male    int64  
Offenders (Driver/Pedestrian) Died_Female   int64  
Offenders (Driver/Pedestrian) Died_Transgender int64  
Offenders (Driver/Pedestrian) Died_Total     int64  
Victims Died_Male      int64  
Victims Died_Female    int64  
Victims Died_Transgender int64  
Victims Died_Total     int64  
dtype: object
```

```
In [22]:
```

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1443 entries, 0 to 1442
Data columns (total 11 columns):
States/UTs                                1443 non-null object
Year                                         1443 non-null int64
CrimeHead                                    1443 non-null object
Offenders (Driver/Pedestrian) Died_Male      1443 non-null int64
Offenders (Driver/Pedestrian) Died_Female    1443 non-null int64
Offenders (Driver/Pedestrian) Died_Transgender 1443 non-null int64
Offenders (Driver/Pedestrian) Died_Total      1443 non-null int64
Victims Died_Male                           1443 non-null int64
Victims Died_Female                         1443 non-null int64
Victims Died_Transgender                    1443 non-null int64
Victims Died_Total                          1443 non-null int64
dtypes: int64(9), object(2)
memory usage: 124.1+ KB
```

```
In [23]: df1.duplicated()
```

```
Out[23]: 0    False
         1    False
         2    False
         3    False
         4    False
         5    False
         6    False
         7    False
         8    False
         9    False
        10   False
        11   False
        12   False
        13   False
        14   False
        15   False
        16   False
        17   False
        18   False
        19   False
        20   False
        21   False
        22   False
        23   False
        24   False
        25   False
        26   False
        27   False
        28   False
        29   False
         ...
       1413  False
       1414  False
       1415  False
       1416  False
       1417  False
       1418  False
       1419  False
       1420  False
       1421  False
       1422  False
       1423  False
       1424  False
       1425  False
       1426  False
       1427  False
       1428  False
       1429  False
       1430  False
       1431  False
       1432  False
       1433  False
       1434  False
       1435  False
       1436  False
```

```
1438    False
1439    False
1440    False
1441    False
1442    False
Length: 1443, dtype: bool
```

---

```
-----#
-----#
```

```
In [327]: df3.head()
```

```
Out[327]:
```

S. No.	States/Uts	06- 900hrs - Day - 2014	09- 1200hrs - Day - 2014	12- 1500hrs - Day - 2014	15- 1800hrs - Day - 2014	18- 2100hrs - Night - 2014	21- 2400hrs - Night - 2014	00- 300hrs - Night - 2014	600I - Nig - 20
0	1 Andhra Pradesh	2548	3448	3491	3606	4058	2989	2031	22
1	2 Arunachal Pradesh	43	30	28	20	16	23	19	
2	3 Assam	1034	1633	1092	1456	979	416	308	2
3	4 Bihar	1433	1661	1484	1485	1092	823	677	9
4	5 Chhattisgarh	1677	2316	2064	2550	2256	1394	755	8

```
In [25]: round(((row - df3.isnull().sum()) / row) * 100, 2)
```

```
Out[25]: States/Uts      100.00
2003          97.30
2004          97.30
2005          97.30
2006          97.30
2007          97.30
2008          97.30
2009          97.30
2010          97.30
2011          97.30
2012          94.59
2013          97.30
2014          100.00
2015          100.00
2016          100.00
dtype: float64
```

---

[Go to top](#)

# DF -

This dataset is about **Number of accidents** per state, accidents sharing multiple states and TOTAL number of accidents overall per year per state.

## Calculating the mean of State wise Total number of Road Accidents.

```
In [26]: mean13 = np.mean(df['State/UT-Wise Total Number of Road Accidents during - 2013'])
print("Mean of accidents happened in all states in year 2013: {}".format(mean13))
```

Mean of accidents happened in all states in year 2013: 27026.444444444445

```
In [27]: mean14 = np.mean(df['State/UT-Wise Total Number of Road Accidents during - 2014'])
print("Mean of accidents happened in all states in year 2014 : {}".format(mean14))
```

Mean of accidents happened in all states in year 2014 : 26454.054054054053

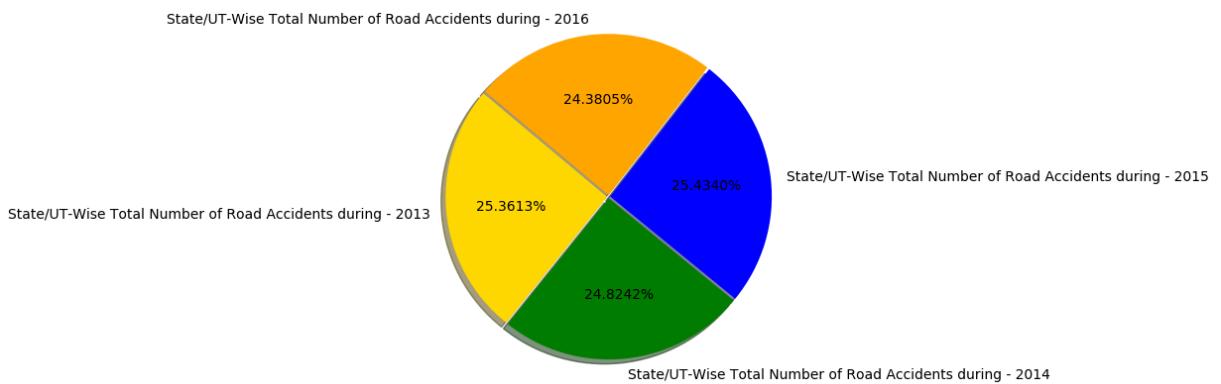
```
In [28]: mean15 = np.mean(df['State/UT-Wise Total Number of Road Accidents during - 2015'])
print("Mean of accidents happened in all states in year 2015 : {}".format(mean15))
```

Mean of accidents happened in all states in year 2015 : 27103.945945945947

```
In [29]: mean16 = np.mean(df['State/UT-Wise Total Number of Road Accidents during - 2016'])
print("Mean of accidents happened in all states in 2016 {}".format(mean16))
```

Mean of accidents happened in all states in 2016 25981.18918918919

```
In [30]: labels = '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'
sizes = [mean13, mean14, mean15, mean16]
colors = ['gold', 'green', 'blue', 'orange']
explode = (0.01, 0.01, 0.01, 0.01)
plt.pie(sizes, labels = labels, colors = colors, explode = explode,
        shadow = True, autopct = '%.4f%%', startangle = 140)
plt.axis('equal')
plt.show()
```



It is clearly visible that, the percentage of road accidents are **almost constant during all years**. It indicates that the government is making less efforts to prevent accidents by creating wider, good quality roads or creating new safety rules.

Calculating mean of total number of accidents per lakh population over the years.

```
In [31]: acc13 = np.mean(df['Total Number of Accidents Per Lakh Population - 2013'])
print("Mean of accidents per lakh population in 2013 : {}".format(acc13))
```

Mean of accidents per lakh population in 2013 : 41.91111111111111

```
In [32]: acc14 = np.mean(df['Total Number of Accidents Per Lakh Population - 2014'])
print("Mean of accidents per lakh population in 2014 : {}".format(acc14))
```

Mean of accidents per lakh population in 2014 : 39.87777777777778

```
In [33]: acc15 = np.mean(df['Total Number of Accidents Per Lakh Population - 2015'])
print("Mean of accidents per lakh population in 2015 : {}".format(acc15))
```

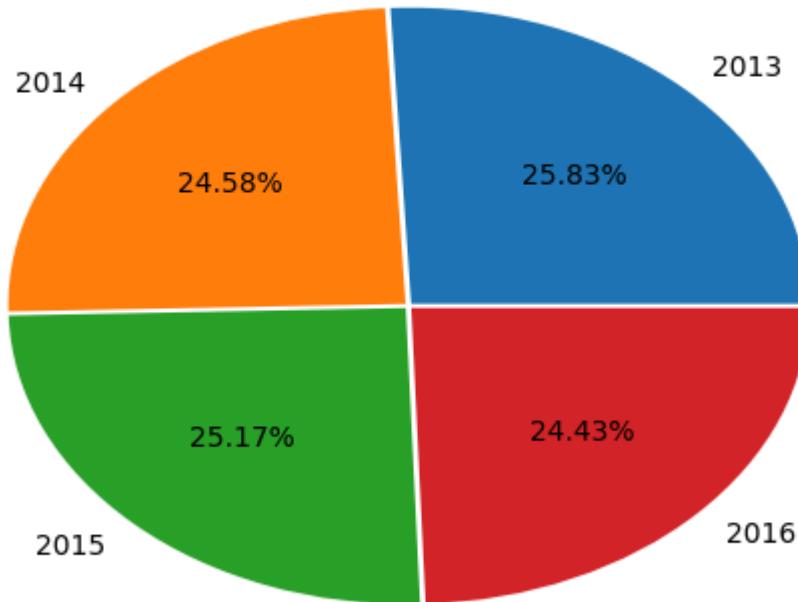
Mean of accidents per lakh population in 2015 : 40.83333333333333

```
In [34]: acc16 = np.mean(df['Total Number of Accidents Per Lakh Population - 2016'])
print("Mean of accidents per lakh population in 2016 : {}".format(acc16))
```

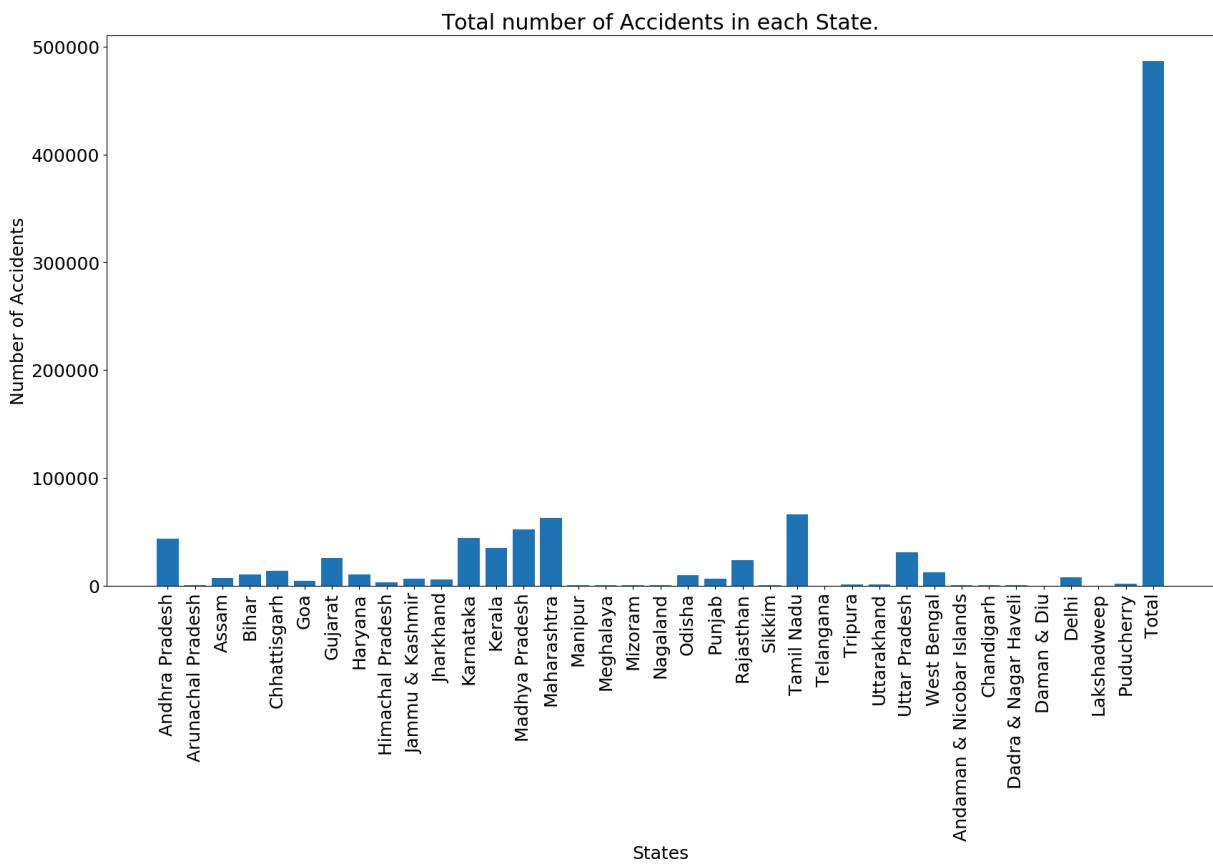
Mean of accidents per lakh population in 2016 : 39.63611111111106

```
In [35]: labels = ('2013','2014','2015','2016')
explode = (0.01,0.01,0.01,0.01)
performance = [acc13, acc14, acc15, acc16]
plt.pie(performance, labels = labels, autopct = '%.2f%%', center = (0,0), ex
plt.title("Mean Accidents per 1L population for each year.")
plt.show()
```

Mean Accidents per 1L population for each year.



```
In [36]: plt.figure(figsize = (20,10))
plt.rcParams.update({'font.size':18})
y = df['State/UT-Wise Total Number of Road Accidents during - 2013']
yd = df['States/UTs']
p = df['States/UTs'].nunique()
d = np.linspace(1,p,p) # refer notes
plt.bar(d, y, align = 'center')
plt.xticks(d, yd, rotation = 90)
plt.xlabel('States')
plt.ylabel('Number of Accidents')
plt.title('Total number of Accidents in each State.')
plt.show()
```



**Maharashtra** and **Tamil Nadu** have the highest number of accidents. Further investigation needs to be done to understand the case. **Arunachal, Manipur, Meghalaya, Mizoram, Nagaland, Tripura** have the least number of accidents. They surprisingly all belong to the **north-eastern area**.

### linspace()

$y = \text{linspace}(x_1, x_2, n)$  generates  $n$  points. The spacing between the points is  $(x_2 - x_1)/(n-1)$ .

`linspace` is similar to the colon operator, “`:`”, but gives direct control over the number of points and always includes the endpoints. “`lin`” in the name “`linspace`” refers to generating linearly spaced values as opposed to the sibling function `logspace`, which generates logarithmically

```
In [37]: min13 = np.min(df['Total Number of Accidents Per Lakh Population - 2013'])
max13 = np.max(df['Total Number of Accidents Per Lakh Population - 2013'])

In [38]: min14 = np.min(df['Total Number of Accidents Per Lakh Population - 2014'])
max14 = np.max(df['Total Number of Accidents Per Lakh Population - 2014'])

In [39]: min15 = np.min(df['Total Number of Accidents Per Lakh Population - 2015'])
max15 = np.max(df['Total Number of Accidents Per Lakh Population - 2015'])

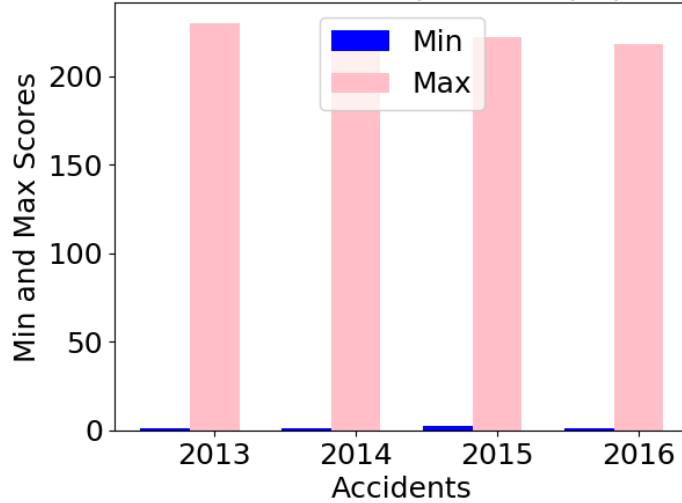
In [40]: min16 = np.min(df['Total Number of Accidents Per Lakh Population - 2016'])
max16 = np.max(df['Total Number of Accidents Per Lakh Population - 2016'])

In [41]: n = 4
minx = (min13, min14, min15, min16)
maxx = (max13, max14, max15, max16)
index = np.arange(n)
bar_width = 0.35

r1 = plt.bar(index, minx, bar_width, align = 'center', color = 'b', label = 'Min')
r2 = plt.bar(index + bar_width, maxx, bar_width, align = 'center', color = 'r', label = 'Max')

plt.xlabel("Accidents")
plt.ylabel("Min and Max Scores")
plt.title("Min and Max number of accidents per lakh population in resp year")
plt.xticks(index + bar_width, ('2013','2014','2015','2016'))
plt.legend(loc = 'upper center')
plt.show()
```

Min and Max number of accidents per lakh population in resp years




---

[Go to top](#)

## DF1 :

This dataset shows the number of **offenders** and **victims** who died according to gender as well as the total deaths.

```
In [42]: df1.head(2)
```

```
Out[42]:
```

	States/UTs	Year	CrimeHead	Offenders (Driver/Pedestrian) Died_Male	Offenders (Driver/Pedestrian) Died_Female	Offender: (Driver/Pedestrian Died_Transgende
0	Andhra Pradesh	2014	Truck/Lorry (Total)	445	30	1
1	Andhra Pradesh	2014	Truck/Lorry - Normal Goods Carriers	357	25	1

```
In [43]: df1.dtypes
```

```
Out[43]: States/UTs          object
Year              int64
CrimeHead        object
Offenders (Driver/Pedestrian) Died_Male    int64
Offenders (Driver/Pedestrian) Died_Female   int64
Offenders (Driver/Pedestrian) Died_Transgender int64
Offenders (Driver/Pedestrian) Died_Total     int64
Victims Died_Male      int64
Victims Died_Female    int64
Victims Died_Transgender int64
Victims Died_Total     int64
dtype: object
```

```
In [44]: meanOffMale = (np.mean(df1['Offenders (Driver/Pedestrian) Died_Male']))
```

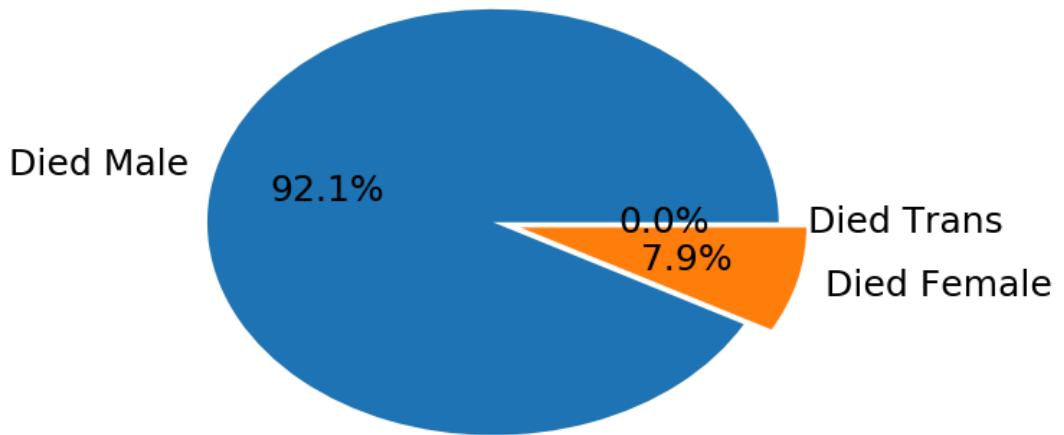
```
In [45]: meanOffFemale = (np.mean(df1['Offenders (Driver/Pedestrian) Died_Female']))
```

```
In [46]: meanOffTrans = (np.mean(df1['Offenders (Driver/Pedestrian) Died_Transgender']))
meanOffTrans, meanOffMale, meanOffFemale
```

```
Out[46]: (0.0, 251.59251559251558, 21.717255717255718)
```

```
In [47]: sizes = [meanOffMale, meanOffFemale, meanOffTrans]
labels = ['Died Male', 'Died Female', 'Died Trans']
explode = (0, 0.1, 0)
plt.pie(sizes, labels = labels, explode = explode, autopct = '%.1f%%')
plt.title("Percentage of offenders who died according to gender.")
plt.show()
```

## Percentage of offenders who died according to gender.



## Victims who died according to gender.

```
In [48]: meanVictMale = (np.mean(df1['Victims Died_Male']))
```

```
In [49]: meanVictFemale = (np.mean(df1['Victims Died_Female']))
```

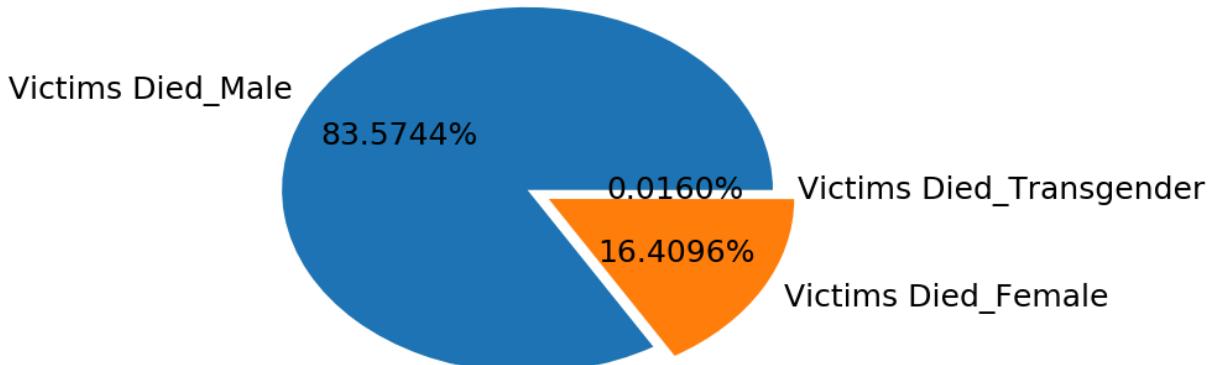
```
In [50]: meanVictTrans = (np.mean(df1['Victims Died_Transgender']))
meanVictMale, meanVictFemale, meanVictTrans
```

```
Out[50]: (705.8731808731809, 138.59667359667358, 0.13513513513513514)
```

```
In [51]: sizes = [meanVictMale, meanVictFemale, meanVictTrans]
labels = 'Victims Died_Male', 'Victims Died_Female', 'Victims Died_Transgender'
explode =(0, 0.1, 0)

plt.pie(sizes, labels = labels, explode = explode, autopct = '%.4f%')
plt.title("Percentage of Victims according to Gender who died.")
plt.show()
```

## Percentage of Victims according to Gender who died.

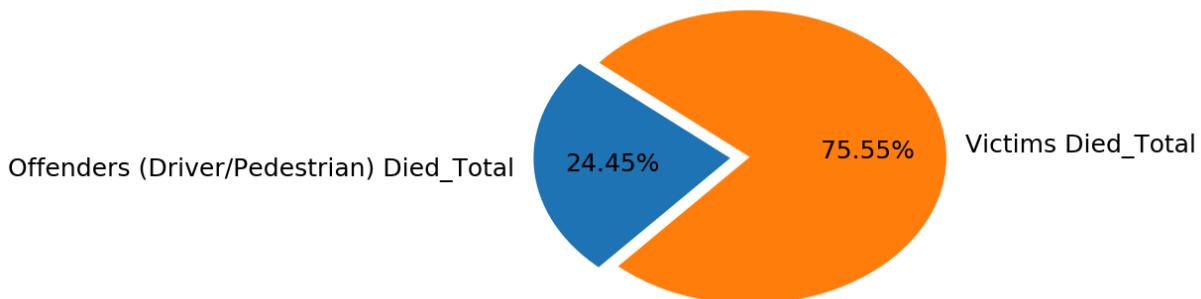


```
In [52]: meanOffTotal = (np.mean(df1['Offenders (Driver/Pedestrian) Died_Total']))
meanVictTotal = (np.mean(df1['Victims Died_Total']))
meanOffTotal, meanVictTotal
```

```
Out[52]: (273.3097713097713, 844.6049896049896)
```

```
In [53]: labels = 'Offenders (Driver/Pedestrian) Died_Total','Victims Died_Total'
sizes = [meanOffTotal, meanVictTotal]
explode = (0,0.1)
plt.title("Percentage of total offenders and victims who died in accidents.")
plt.pie(sizes, labels = labels, explode = explode, autopct = '%.2f%%', start
plt.show()
```

Percentage of total offenders and victims who died in accidents.



---

[Go to top](#)

## DF2 -

This dataset is about the deaths that occurred due to the **ignorance for safety accessories like Helmets, Belts** per gender.

```
In [54]: df2.head(2)
```

```
Out[54]:
```

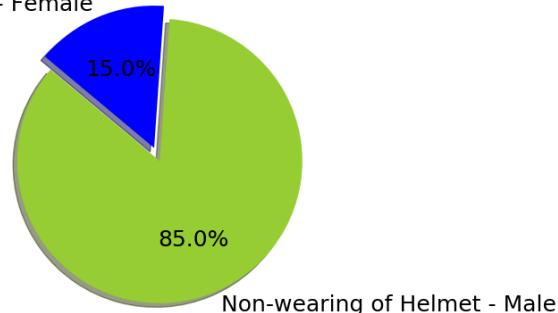
S. No.	State/ UT	Non- wearing of Helmet - Male	Non- wearing of Helmet - Female	Non- wearing of Helmet - Total	Non- wearing of seat belt - Male	Non- wearing of seat belt - Female	Non- wearing of seat belt - Total
0	1 Andhra Pradesh	459	18	477	222	15	237
1	2 Arunachal Pradesh	10	4	14	0	0	0

```
In [55]: helmetMale = (np.mean(df2['Non-wearing of Helmet - Male']))
helmetFemale = (np.mean(df2['Non-wearing of Helmet - Female']))
helmetFemale, helmetMale
```

```
Out[55]: (82.10810810810811, 465.72972972972974)
```

```
In [56]: labels = 'Non-wearing of Helmet - Male', 'Non-wearing of Helmet - Female'
sizes = [helmetMale, helmetFemale]
colors = ['yellowgreen', 'blue']
explode = (0.1,0)
plt.pie(sizes, explode=explode, labels=labels, colors=colors,
        autopct='%1.1f%%', shadow=True, startangle=140)
plt.title("Percentage of Deaths occuring due to non-wearing of helmets between male and female")
plt.axis('equal')
plt.show()
```

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



```
In [57]: max1 = df2['Non-wearing of Helmet - Total'].max()
max2 = df2['Non-wearing of seat belt - Total'].max()
max1, max2
```

```
Out[57]: (10135, 5638)
```

---

[Go to top.](#)

## DF3

This data set shows the number of accidents happening per state from the year 2003 to 2016.

```
In [331... df3 = pd.read_excel('A:\Data Analysis Jupyter\Road-Accidents-Of-India\Database.xlsx')
df3.head(2)
```

WARNING \*\*\* OLE2 inconsistency: SSCS size is 0 but SSAT size is non-zero

```
Out[331]:   States/Uts  2003  2004  2005  2006  2007  2008  2009  2010  2011  2012  2013  2014  2015  2016
0    Andhra Pradesh  8616.0  9343.0  9126.0  11378.0  11885.0  12233.0  12676.0  13932.0  13784.0  14020.0  14232.0  14420.0  14612.0
1  Arunachal Pradesh   65.0    79.0    63.0    87.0    81.0    90.0    114.0   108.0    91.0    84.0    80.0    75.0    70.0    65.0
```

Out[391]:	States/Uts	2003	2004	2005	2006	2007	2008	2009	2010
	31 Dadra & Nagar Haveli	34.0	46.0	51.0	43.0	61.0	62.0	43.0	6
	32 Daman & Diu	19.0	23.0	21.0	24.0	29.0	29.0	32.0	3
	33 Delhi	1730.0	1846.0	1771.0	2129.0	2081.0	2015.0	2272.0	210
	35 Puducherry	182.0	176.0	224.0	209.0	242.0	207.0	214.0	22
	36 All India	73589.0	79357.0	83491.0	93917.0	101161.0	106591.0	110993.0	11955

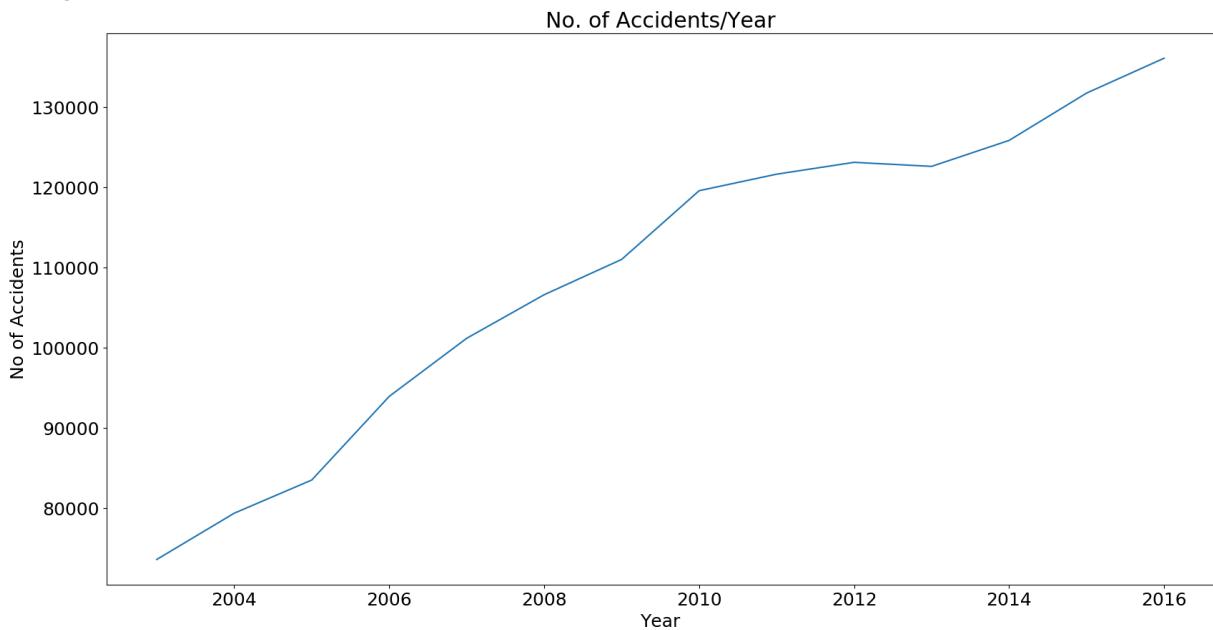
In [387... df3['States/Uts']

```
Out[387]: 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
25         Tripura
26         Uttarakhand
27         Uttar Pradesh
28         West Bengal
29    Andaman & Nicobar Islands
30         Chandigarh
31    Dadra & Nagar Haveli
32        Daman & Diu
33        Delhi
35        Puducherry
36        All India
Name: States/Uts, dtype: object
```

In [386... df8 = pd.DataFrame(columns = ['Year', 'Count'])

```
In [389]: df8['Year'] = df3[df3['States/Uts'] == 'All India'].columns[1:]  
  
In [409]: df8 = df3[df3['States/Uts'] == "All India"]  
df9 = df8.T  
  
In [446]: df9.to_csv('A:\\Data Analysis Jupyter\\Road-Accidents-Of-India\\Databases\\a  
  
In [441]: plt.figure(figsize = (20,10))  
plt.rcParams.update({'font.size' : 18})  
  
df9.iloc[1:].plot(figsize = (20,10), legend = False)  
plt.xlabel("Year")  
plt.ylabel("No of Accidents")  
plt.title("No. of Accidents/Year")  
plt.show()
```

<Figure size 2000x1000 with 0 Axes>



---

[Go to top](#)

## DF4

This data set contains number of **accidents/deaths/injuries** occurring as per number of lanes.

```
In [58]: df4 = pd.read_csv('A:\\Data Analysis Jupyter\\Road-Accidents-Of-India\\Database  
df4.head(2)
```

Out[58]:

S.No	State/UT	Single Lane - Accident - 2014	Single Lane - Accident - 2014 per 1L people	Single Lane - Killed - 2014	Single Lane - Killed - 2014 per 1L people	Single Lane - Injured - 2014	Single Lane - Injured - 2014 per 1L people	Single Lane - Injured - 2014 per 1L people	Two Lanes - Accident - 2014
0	1 Andhra Pradesh	8634.0	16.473750	2958.0	5.643891	13541.0	25.836350	10327.0	
1	2 Arunachal Pradesh	90.0	6.504173	60.0	4.336115	152.0	10.984826	115.0	

2 rows × 27 columns

In [59]: df4.dtypes

```
S.No          object
State/UT       object
Single Lane - Accident - 2014      float64
Single Lane - Accident - 2014 per 1L people  float64
Single Lane - Killed - 2014        float64
Single Lane - Killed - 2014 per 1L people  float64
Single Lane - Injured - 2014        float64
Single Lane - Injured - 2014 per 1L people  float64
Two Lanes - Accident - 2014        float64
Two Lanes - Accident - 2014 per 1L people  float64
Two Lanes - Killed - 2014         float64
Two Lanes - Killed - 2014 per 1L people  float64
Two Lanes - Injured - 2014         float64
Two Lanes - Injured - 2014 per 1L people  float64
Three Lanes or more w.o Median - Accident - 2014   float64
Three Lanes or more w.o Median - Accident - 2014 per 1L people  float64
Three Lanes or more w.o Median - Killed - 2014    float64
Three Lanes or more w.o Median - Killed - 2014 per 1L people  float64
Three Lanes or more w.o Median - Injured - 2014    float64
Three Lanes or more w.o Median - Injured - 2014 per 1L people  float64
Four Lanes with Median - Accident - 2014        float64
Four Lanes with Median - Accident - 2014 per 1L people  float64
Four Lanes with Median - Killed - 2014       float64
Four Lanes with Median - Killed - 2014 per 1L people  float64
Four Lanes with Median - Injured - 2014       float64
Four Lanes with Median - Injured - 2014 per 1L people  float64
Population          int64
dtype: object
```

In [60]: df4.dropna(axis = 0, how ='any', inplace = True)

Number of ACCIDENTS\* for 1,2,3,4 lanes per 1L population.\*

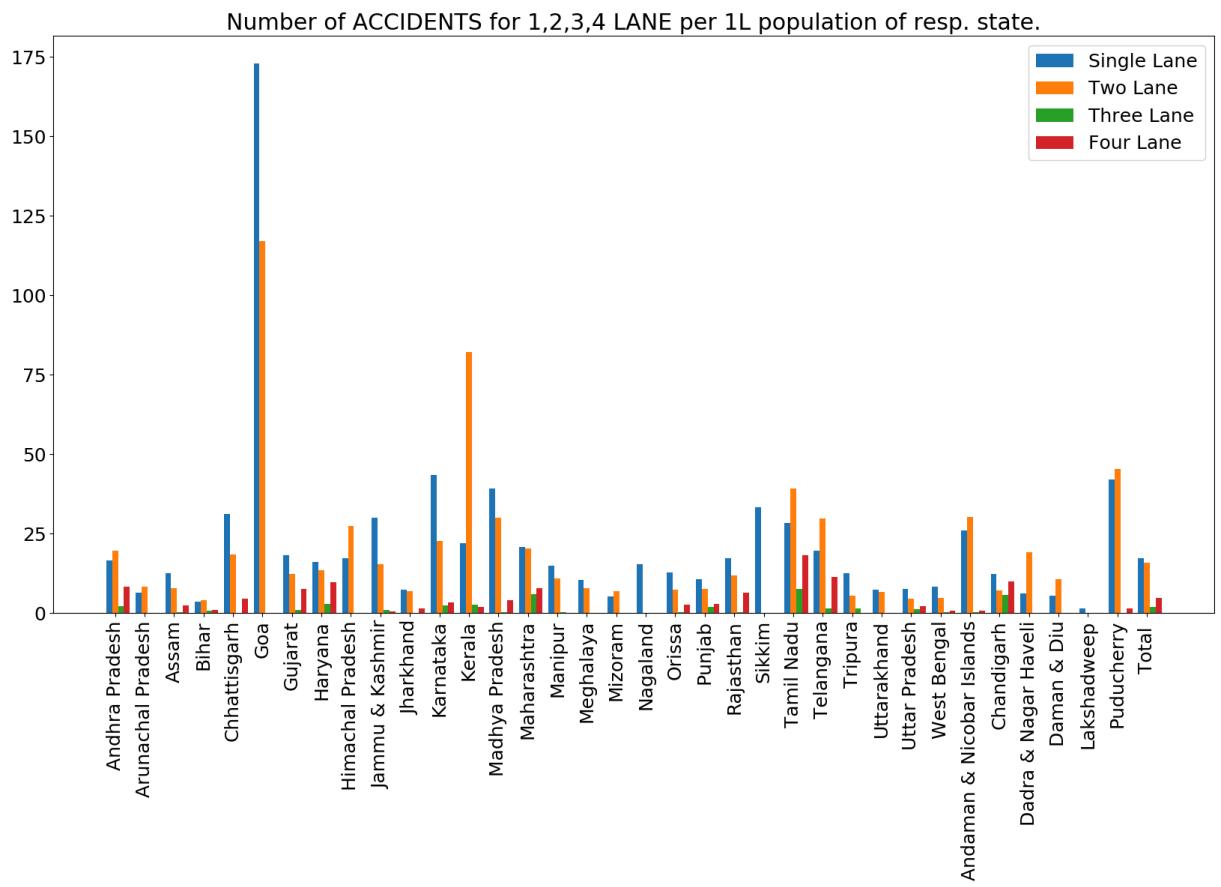
In [61]: singleLaneAcc = df4['Single Lane - Accident - 2014 per 1L people']
twoLaneAcc = df4['Two Lanes - Accident - 2014 per 1L people']

```
threeLaneAcc = df4['3 Lanes or more w.o Median - Accident - 2014 per 1L people']
fourLaneAcc = df4['4 Lanes with Median - Accident - 2014 per 1L people']
```

```
In [62]: plt.figure(figsize = (20,10))
plt.rcParams.update({'font.size':18})
UT = np.array([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24])

plt.bar(UT-0.2, singleLaneAcc, label='Single Lane', width = 0.2,
        align = 'center')
plt.bar(UT, twoLaneAcc, label = 'Two Lane', width = 0.2,
        align = 'center')
plt.bar(UT+0.2, threeLaneAcc, label = 'Three Lane', width = 0.2,
        align = 'center')
plt.bar(UT+0.4, fourLaneAcc, label = 'Four Lane', width = 0.2,
        align = 'center')

plt.xticks(UT, df4['State/UT'], rotation = 'vertical' )
plt.legend(loc = 'best')
plt.title("Number of ACCIDENTS for 1,2,3,4 LANE per 1L population of resp. state")
plt.show()
```



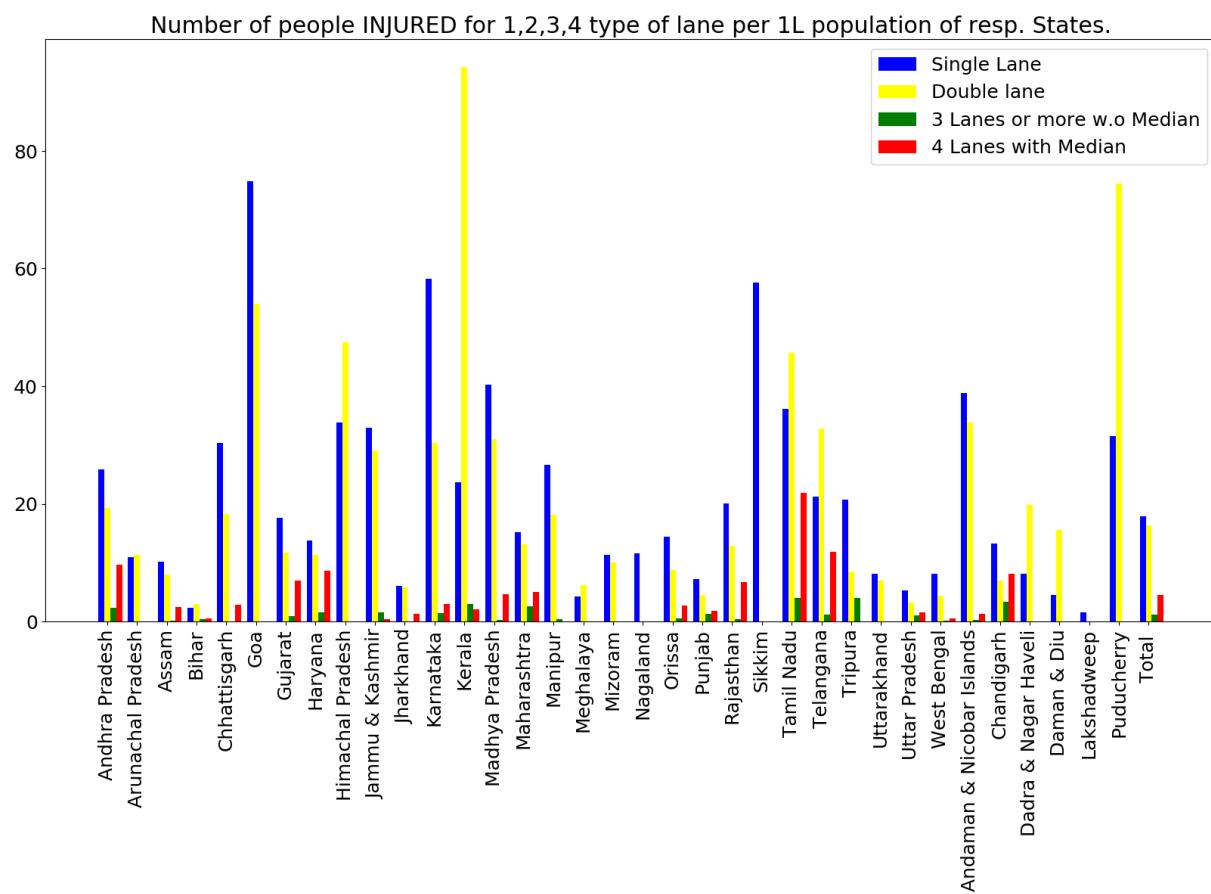
Number of people INJURED\* for 1,2,3,4 number of lanes per 1L population.\*

```
In [63]: singleInjured = df4['Single Lane - Injured - 2014 per 1L people']
twoInjured = df4['Two Lanes - Injured - 2014 per 1L people']
threeInjured = df4['3 Lanes or more w.o Median - Injured - 2014 per 1L people']
fourInjured = df4['4 Lanes with Median - Injured - 2014 per 1L people']
```

```
In [64]: plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

plt.bar(UT-0.2, singleInjured, width=0.2, color='b',
        align='center',label='Single Lane')
plt.bar(UT, twoInjured, width=0.2, color='yellow',
        align='center',label='Double lane')
plt.bar(UT+0.2, threeInjured, width=0.2, color='g',
        align='center',label='3 Lanes or more w.o Median')
plt.bar(UT+0.4, fourInjured, width=0.2, color='red',
        align='center',label='4 Lanes with Median')

plt.xticks(UT, df4['State/UT'], rotation='vertical')
plt.title("Number of people INJURED for 1,2,3,4 type of lane per 1L population")
plt.legend(loc = "best")
plt.show()
```



Number of people KILLED\* for 1,2,3,4 Lanes per 1L population.\*

```
In [65]: singleLaneKilled = df4['Single Lane - Killed - 2014 per 1L people']
twoLaneKilled = df4['Two Lanes - Killed - 2014 per 1L people']
threeLaneKilled = df4['3 Lanes or more w.o Median - Killed - 2014 per 1L people']
fourLaneKilled = df4['4 Lanes with Median - Killed - 2014 per 1L people']
```

```
In [66]: plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})
```

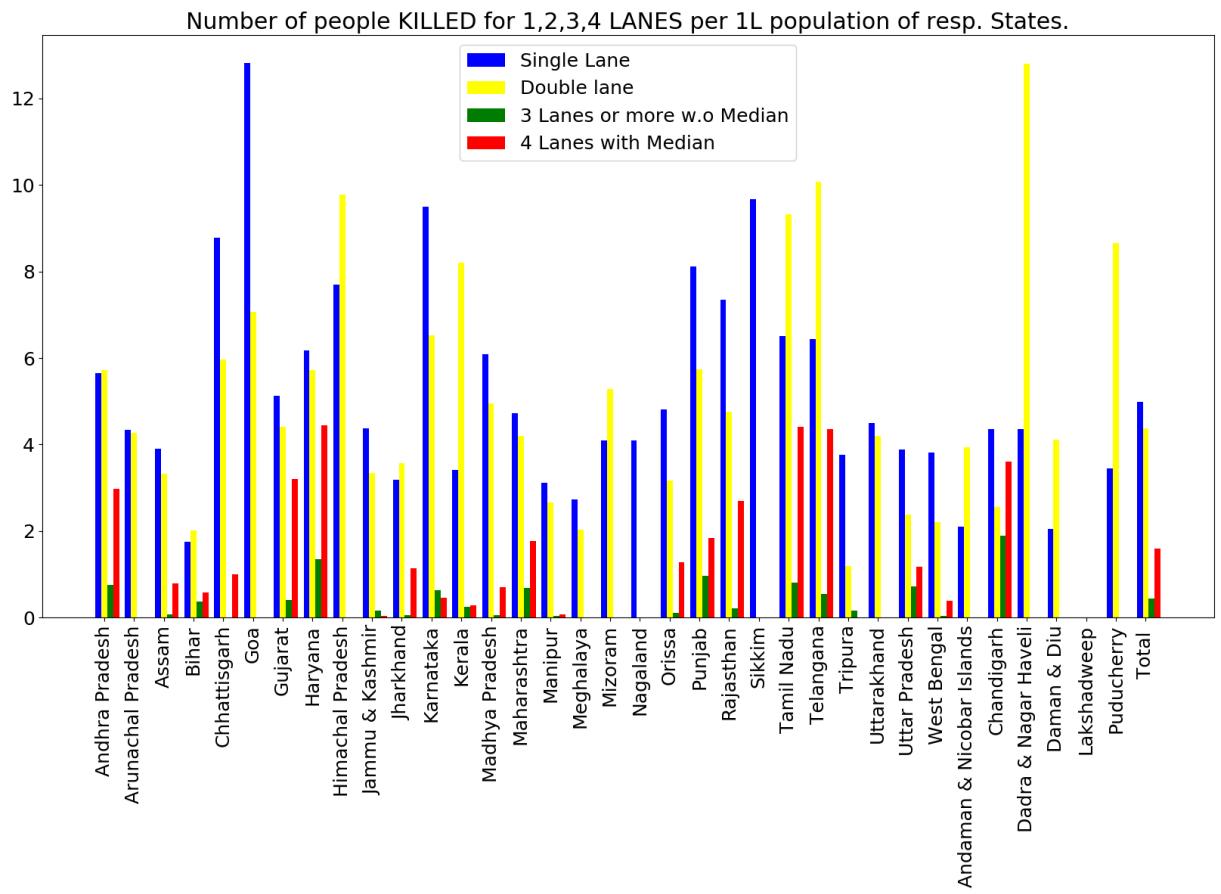
Loading [MathJax]/extensions/Safe.js 2, singleLaneKilled, width=0.2, color='b',

```

        align='center',label='Single Lane')
plt.bar(UT, twoLaneKilled, width=0.2, color='yellow',
        align='center',label='Double lane')
plt.bar(UT+0.2, threeLaneKilled, width=0.2, color='g',
        align='center',label='3 Lanes or more w.o Median')
plt.bar(UT+0.4, fourLaneKilled, width=0.2, color='red',
        align='center',label='4 Lanes with Median')

plt.xticks(UT, df4['State/UT'], rotation='vertical')
plt.title("Number of people KILLED for 1,2,3,4 LANES per 1L population of resp. States")
plt.legend(loc = "best")
plt.show()

```



Number of accidents, people injured, killed on SINGLE LANE\* per 1L population.\*

```

In [67]: singleLaneTotalInjured = df4['Single Lane - Injured - 2014 per 1L people']
singleLaneTotalKilled = df4['Single Lane - Killed - 2014 per 1L people']
singleLaneTotalAccidents = df4['Single Lane - Accident - 2014 per 1L people']

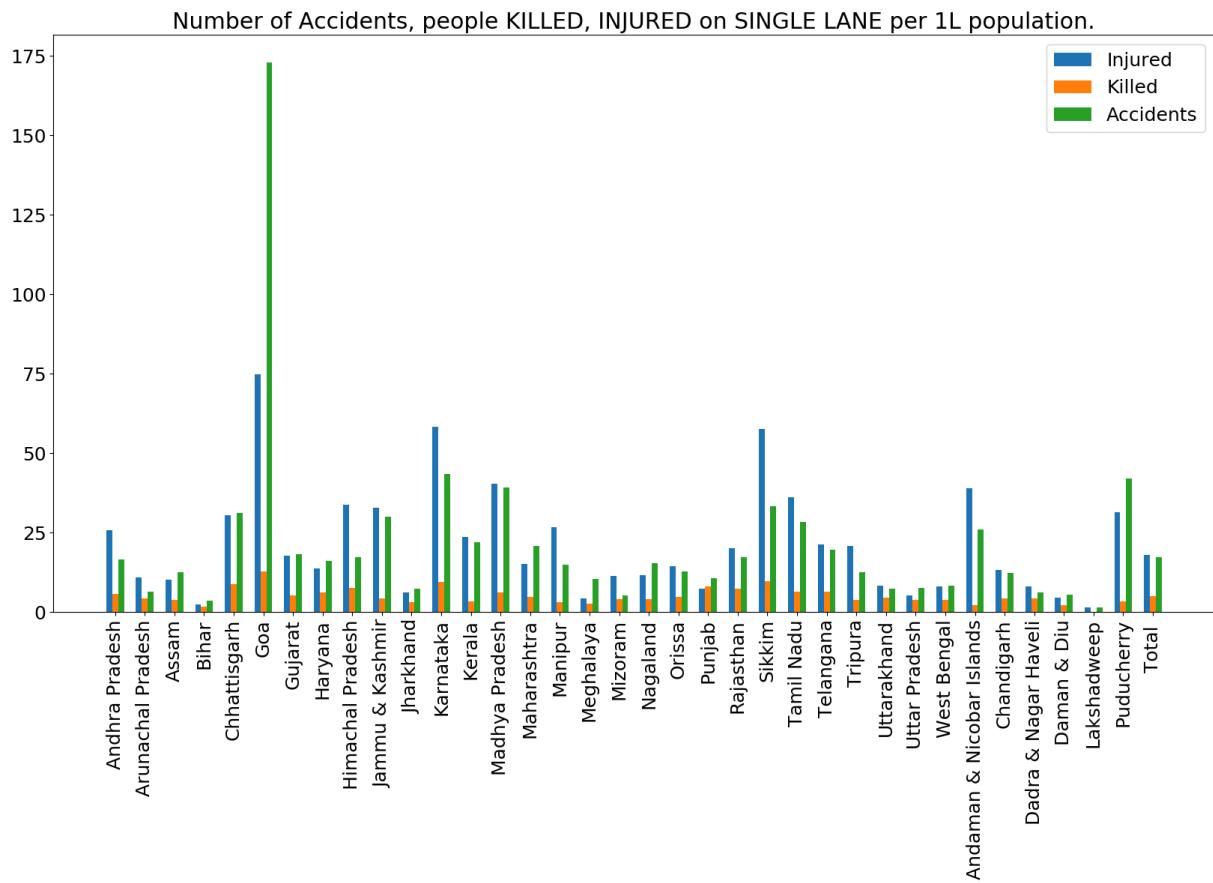
In [68]: plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

plt.bar(UT-0.2, singleLaneTotalInjured, width=0.2, align='center', label='Injured')
plt.bar(UT, singleLaneTotalKilled, width=0.2, align='center', label='Killed')
plt.bar(UT+0.2, singleLaneTotalAccidents, width=0.2, align='center', label='Accidents')

plt.xticks(UT, df4['State/UT'], rotation='vertical')
plt.title("Number of Accidents, people KILLED, INJURED on SINGLE LANE per 1L population")

```

```
plt.legend(loc = "best")
plt.show()
```



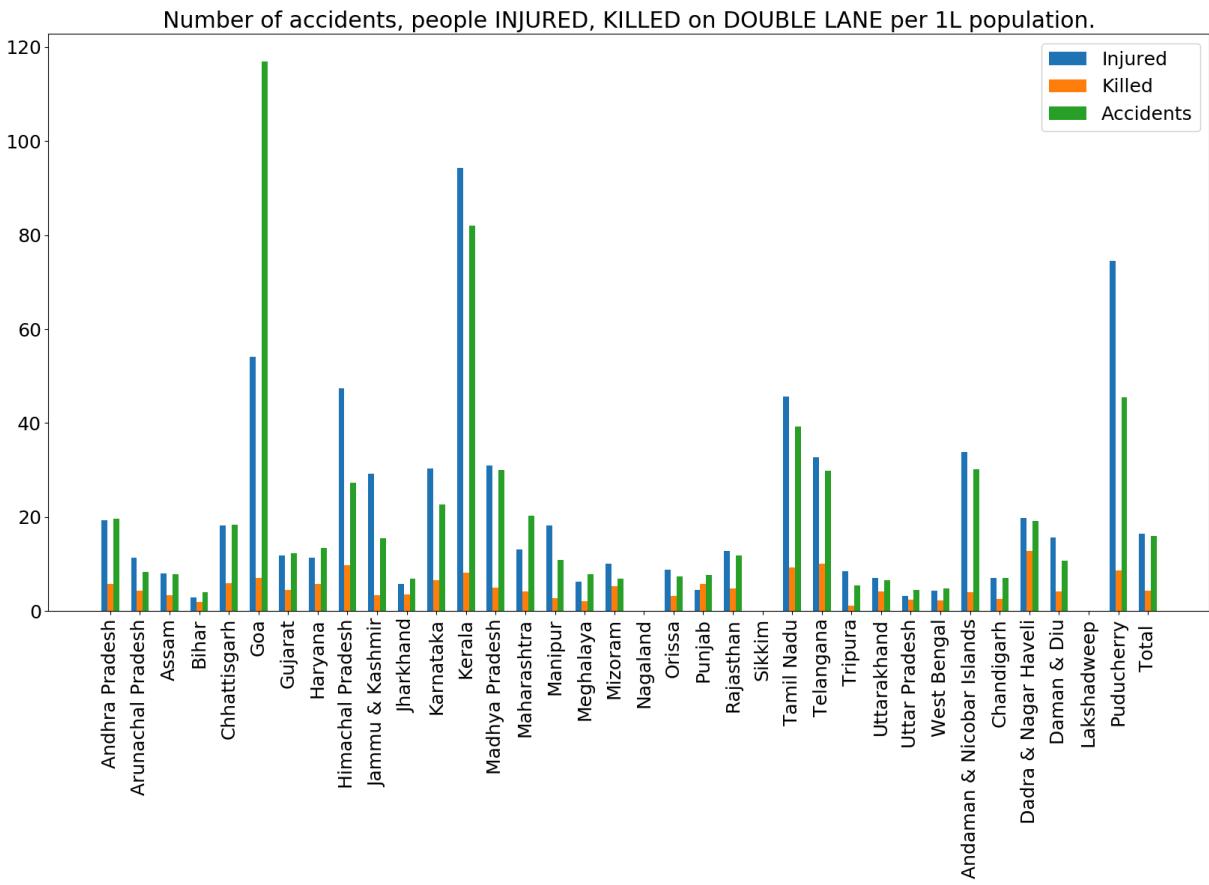
Number of accidents, people injured, killed on DOUBLE LANE\* per 1L population.\*

```
In [69]: twoLaneTotalInjured = df4['Two Lanes - Injured - 2014 per 1L people']
twoLaneTotalKilled = df4['Two Lanes - Killed - 2014 per 1L people']
twoLaneTotalAccidents = df4['Two Lanes - Accident - 2014 per 1L people']
```

```
In [70]: plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

plt.bar(UT-0.2, twoLaneTotalInjured, width=0.2, align='center', label='Injuri
plt.bar(UT, twoLaneTotalKilled, width=0.2, align='center', label='Killed')
plt.bar(UT+0.2, twoLaneTotalAccidents, width=0.2, align='center', label='Acc

plt.xticks(UT, df4['State/UT'], rotation='vertical')
plt.title("Number of accidents, people INJURED, KILLED on DOUBLE LANE per 1L
plt.legend(loc = "best")
plt.show()
```



Number of accidents, people injured, killed for THREE LANE\* per 1L population.\*

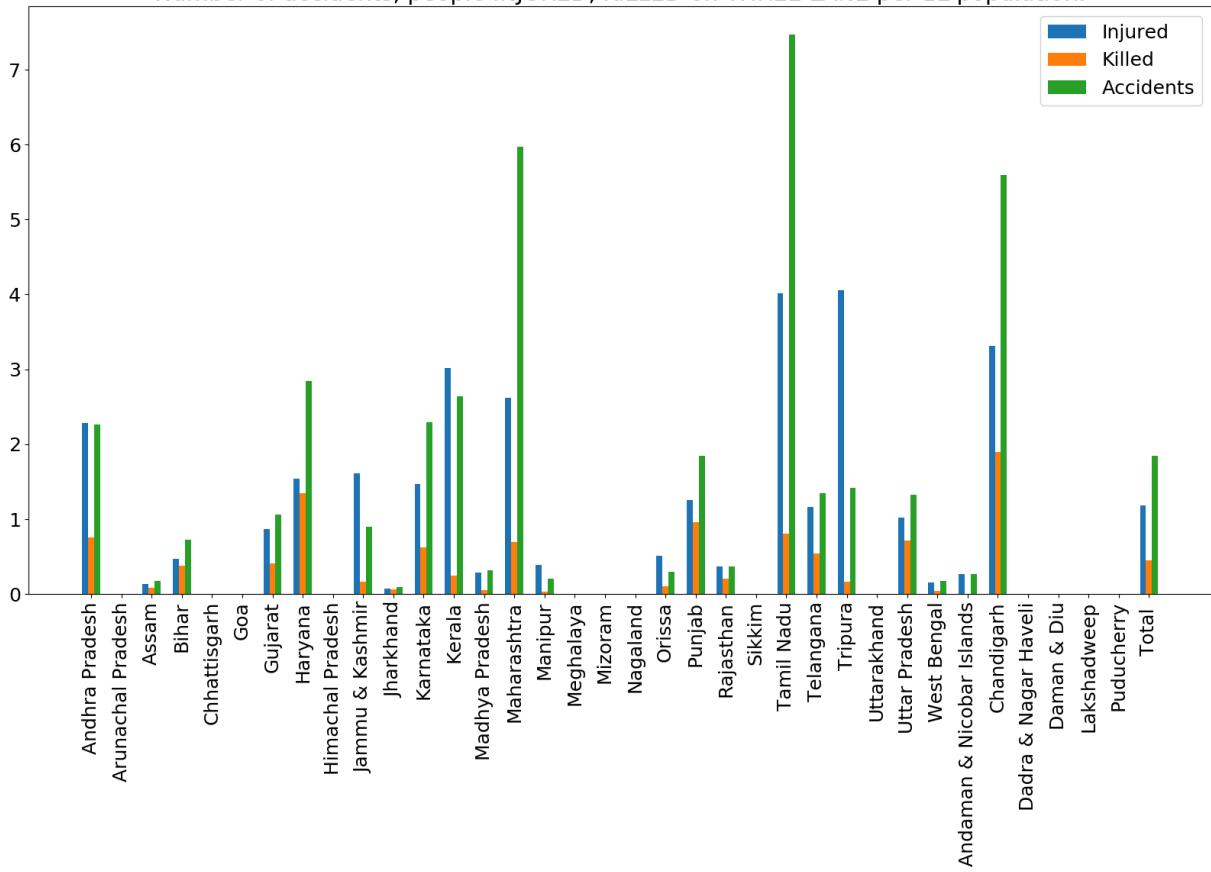
```
In [71]: threeLaneTotalInjured = df4['3 Lanes or more w/o Median - Injured - 2014 per 1L']
threeLaneTotalKilled = df4['3 Lanes or more w/o Median - Killed - 2014 per 1L']
threeLaneTotalAccidents = df4['3 Lanes or more w/o Median - Accident - 2014']
```

```
In [72]: plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

plt.bar(UT-0.2, threeLaneTotalInjured, width=0.2, align='center', label='Inj')
plt.bar(UT, threeLaneTotalKilled, width=0.2, align='center', label='Killed')
plt.bar(UT+0.2, threeLaneTotalAccidents, width=0.2, align='center', label="Acc")

plt.xticks(UT, df4['State/UT'], rotation='vertical')
plt.title("Number of accidents, people INJURED, KILLED on THREE LANE per 1L")
plt.legend(loc = "best")
plt.show()
```

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



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

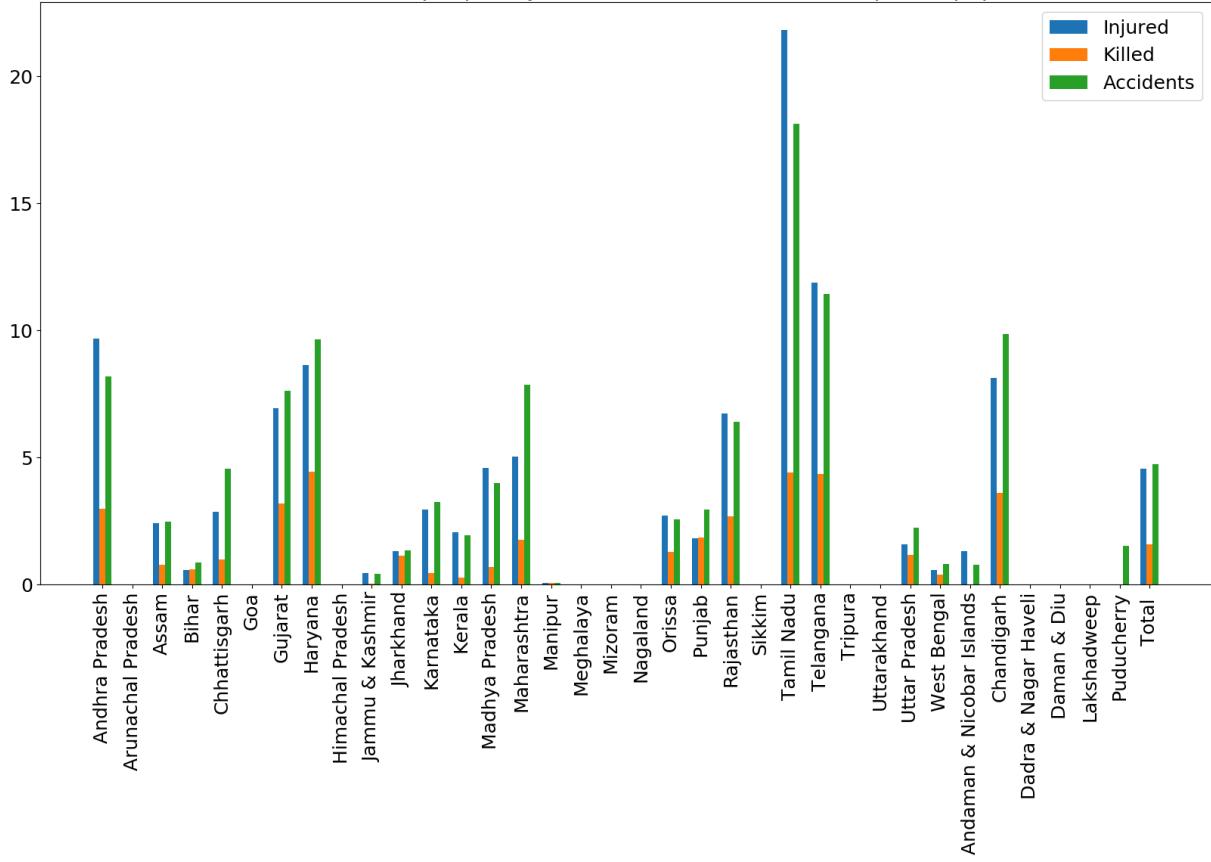
```
In [73]: fourLaneTotalInjured = df4['4 Lanes with Median - Injured - 2014 per 1L people']
fourLaneTotalKilled = df4['4 Lanes with Median - Killed - 2014 per 1L people']
fourLaneTotalAccidents = df4['4 Lanes with Median - Accident - 2014 per 1L people']
```

```
In [74]: plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

plt.bar(UT-0.2, fourLaneTotalInjured, width=0.2, align='center', label='Injured')
plt.bar(UT, fourLaneTotalKilled, width=0.2, align='center', label='Killed')
plt.bar(UT+0.2, fourLaneTotalAccidents, width=0.2, align='center', label="Accidents")

plt.xticks(UT, df4['State/UT'], rotation='vertical')
plt.title("Number of accidents, people INJURED, KILLED on FOUR LANE per 1L population")
plt.legend(loc = "best")
plt.show()
```

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



*Total number of accidents, people INJURED, KILLED irrespective of lanes\* per 1L population per state\**

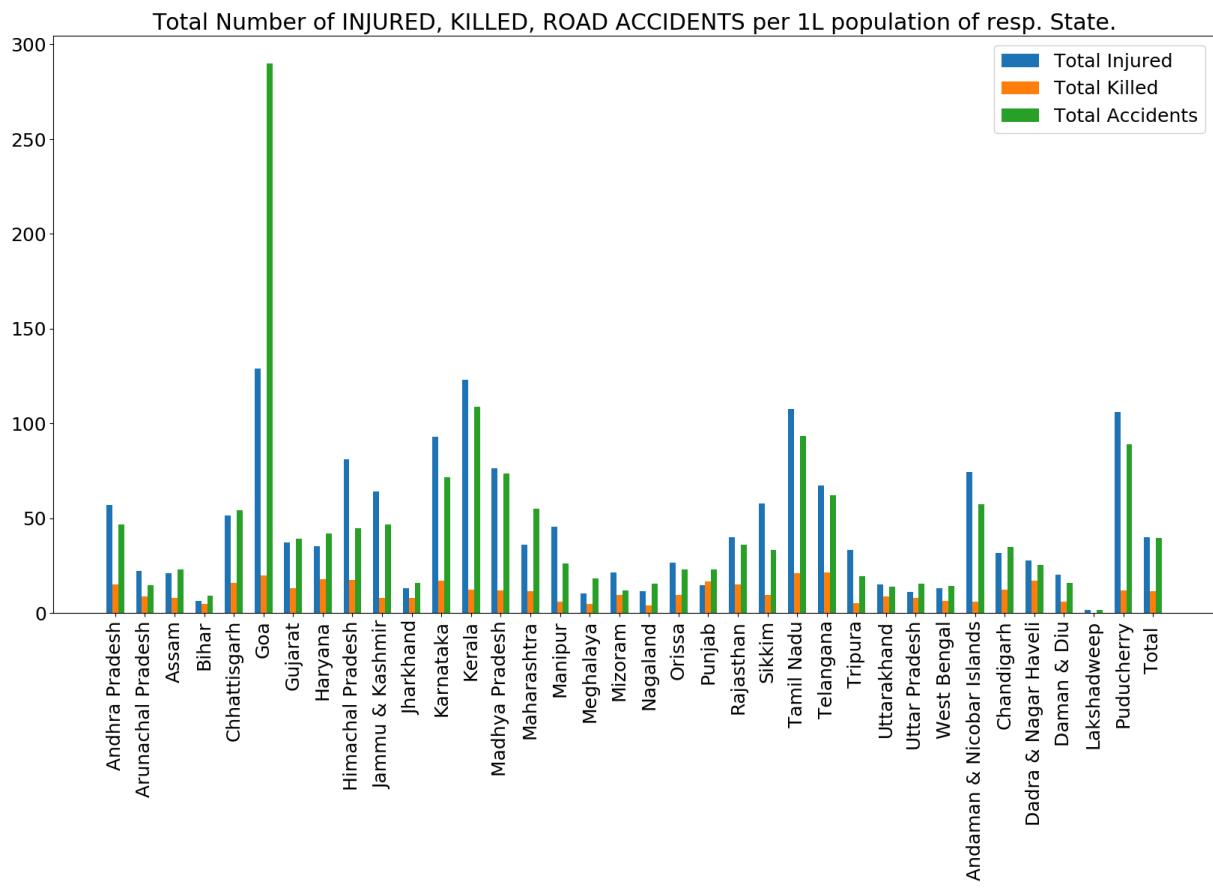
```
In [75]: df4['Sum Total Number of Persons Killed - 2014 per 1L people']=df4['Single L
df4['Sum Total Number of Persons Injured - 2014 per 1L people']=df4['Single
df4['Sum Total Road Accidents - 2014 per 1L people']=df4['Single Lane - Acci

totalKilled = df4['Sum Total Number of Persons Killed - 2014 per 1L people']
totalInjured = df4['Sum Total Number of Persons Injured - 2014 per 1L people']
totalAccidents = df4['Sum Total Road Accidents - 2014 per 1L people']
```

```
In [76]: plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

plt.bar(UT-0.2, totalInjured, width=0.2, align='center', label='Total Injured')
plt.bar(UT, totalKilled, width=0.2, align='center', label='Total Killed')
plt.bar(UT+0.2, totalAccidents, width=0.2, align='center', label="Total Accidents")

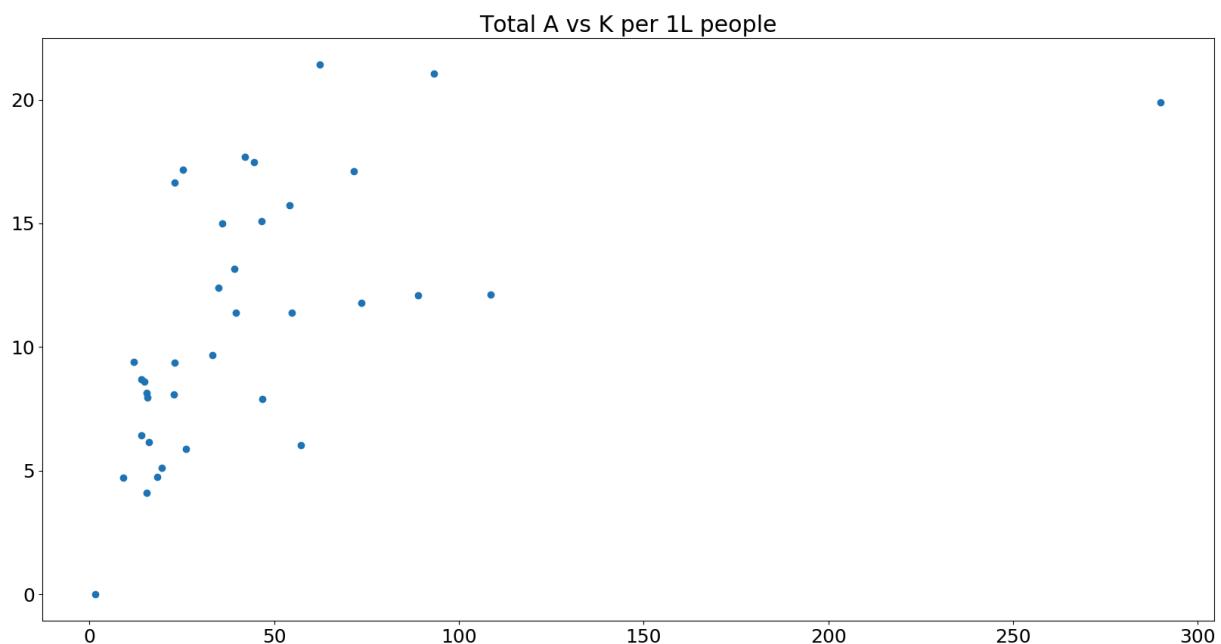
plt.xticks(UT, df4['State/UT'], rotation='vertical')
plt.title("Total Number of INJURED, KILLED, ROAD ACCIDENTS per 1L population")
plt.legend(loc = "best")
plt.show()
```



Scatter plot showing **Total Sum of Accidents vs Killed for all types of lanes for the accident (dots represent each state)**

```
In [77]: plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

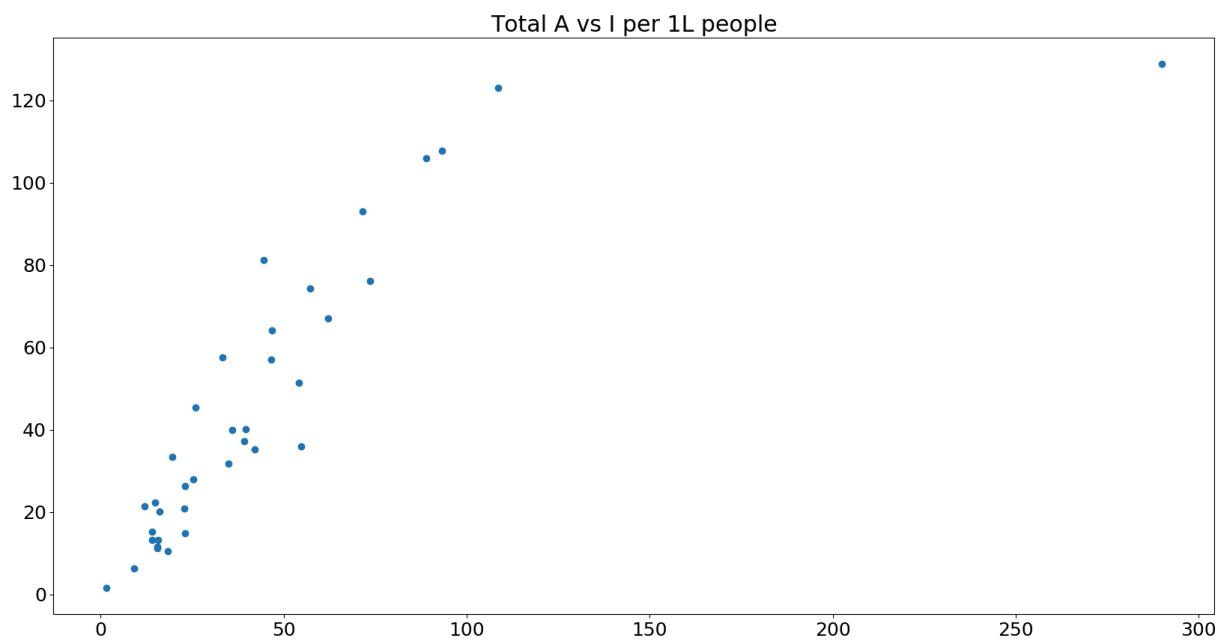
plt.scatter(df4['Sum Total Road Accidents - 2014 per 1L people'],df4['Sum Tc
plt.title("Total A vs K per 1L people")
plt.show()
```



Scatter plot showing *Total Sum of Accidents vs Injured for all types of lanes for the accident* (dots represent each state)

```
In [78]: plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

plt.scatter(df4['Sum Total Road Accidents - 2014 per 1L people'],df4['Sum To
plt.title("Total A vs I per 1L people")
plt.show()
```




---

[Go to top](#)

This dataset states the count of accidents, deaths, injuries occurred due to **various faults and reasons like Fault of Driver, weather condition, boulder, poor light etc.**

```
In [79]: df5 = pd.read_csv('A:\Data Analysis Jupyter\Road-Accidents-Of-India\Database')
df5.head()
```

Out[79]:

Sl. No	States/UTs	Fault of Driver-Total No. of Road Accidents - 2014	Fault of Driver-Total No. of Road Accidents - 2014 per 1L people	Fault of Driver-Number of Persons-Killed - 2014	Fault of Driver-Number of Persons-Killed - 2014 per 1L people	Fault of Driver-Number of Persons-Injured - 2014	Fault of Driver-Number of Persons-Injured - 2014 per 1L people	I D ve To c Ac
0	1 Andhra Pradesh	21359.0	40.753165	6743	12.865705	26287.0	50.155834	
1	2 Arunachal Pradesh	30.0	2.168058	19	1.373103	30.0	2.168058	
2	3 Assam	6895.0	22.095410	2429	7.783865	6281.0	20.127813	
3	4 Bihar	5008.0	4.810784	2646	2.541800	3374.0	3.241131	
4	5 Chhattisgarh	9108.0	35.654451	2458	9.622161	8710.0	34.096428	

5 rows × 63 columns

```
In [80]: df5.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37 entries, 0 to 36
Data columns (total 63 columns):
Sl. No
37 non-null object
States/UTs
37 non-null object
Fault of Driver-Total No. of Road Accidents - 2014
36 non-null float64
Fault of Driver-Total No. of Road Accidents - 2014 per 1L people
36 non-null float64
Fault of Driver-Number of Persons-Killed - 2014
37 non-null int64
Fault of Driver-Number of Persons-Killed - 2014 per 1L people
37 non-null float64
Fault of Driver-Number of Persons-Injured - 2014
36 non-null float64
Fault of Driver-Number of Persons-Injured - 2014 per 1L people
36 non-null float64
Fault of Driver of other vehicles-Total No. of Road Accidents - 2014
36 non-null float64
Fault of Driver of other vehicles-Total No. of Road Accidents - 2014 per 1L people
36 non-null float64
Fault of Driver of other vehicles-Number of Persons-Killed - 2014
37 non-null int64
Fault of Driver of other vehicles-Number of Persons-Killed - 2014 per 1L people
37 non-null float64
Fault of Driver of other vehicles-Number of Persons-Injured - 2014
36 non-null float64
Fault of Driver of other vehicles-Number of Persons-Injured - 2014 per 1L people
36 non-null float64
Fault of Pedestrian-Total No. of Road Accidents - 2014
36 non-null float64
Fault of Pedestrian-Total No. of Road Accidents - 2014 per 1L people
36 non-null float64
Fault of Pedestrian-Number of Persons-Killed - 2014
37 non-null int64
Fault of Pedestrian-Number of Persons-Killed - 2014 per 1L people
37 non-null float64
Fault of Pedestrian-Number of Persons-Injured - 2014
36 non-null float64
Fault of Pedestrian-Number of Persons-Injured - 2014 per 1L people
36 non-null float64
Defect in Condition of Motor Vehicle-Total No. of Road Accidents - 2014
36 non-null float64
Defect in Condition of Motor Vehicle-Total No. of Road Accidents - 2014 per 1L people
36 non-null float64
Defect in Condition of Motor Vehicle-Number of Persons-Killed - 2014
37 non-null int64
Defect in Condition of Motor Vehicle-Number of Persons-Killed - 2014 per 1L people
37 non-null float64
Defect in Condition of Motor Vehicle-Number of Persons-Injured - 2014
36 non-null float64
Defect in Condition of Motor Vehicle-Number of Persons-Injured - 2014 per 1L people
36 non-null float64
```

36 non-null float64  
Defect in Road Condition-Total No. of Road Accidents - 2014 per 1L people  
36 non-null float64  
Defect in Road Condition-Number of Persons-Killed - 2014  
37 non-null int64  
Defect in Road Condition-Number of Persons-Killed - 2014 per 1L people  
37 non-null float64  
Defect in Road Condition-Number of Persons-Injured - 2014  
36 non-null float64  
Defect in Road Condition-Number of Persons-Injured - 2014 per 1L people  
36 non-null float64  
Weather Condition-Total No. of Road Accidents - 2014  
36 non-null float64  
Weather Condition-Total No. of Road Accidents - 2014 per 1L people  
36 non-null float64  
Weather Condition-Number of Persons-Killed - 2014  
37 non-null int64  
Weather Condition-Number of Persons-Killed - 2014 per 1L people  
37 non-null float64  
Weather Condition-Number of Persons-Injured - 2014  
36 non-null float64  
Weather Condition-Number of Persons-Injured - 2014 per 1L people  
36 non-null float64  
Fault of Passenger-Total No. of Road Accidents - 2014  
36 non-null float64  
Fault of Passenger-Total No. of Road Accidents - 2014 per 1L people  
36 non-null float64  
Fault of Passenger-Number of Persons-Killed - 2014  
37 non-null int64  
Fault of Passenger-Number of Persons-Killed - 2014 per 1L people  
37 non-null float64  
Fault of Passenger-Number of Persons-Injured - 2014  
36 non-null float64  
Fault of Passenger-Number of Persons-Injured - 2014 per 1L people  
36 non-null float64  
Poor light-Total No. of Road Accidents - 2014  
36 non-null float64  
Poor light-Total No. of Road Accidents - 2014 per 1L people  
36 non-null float64  
Poor light-Number of Persons-Killed - 2014  
37 non-null int64  
Poor light-Number of Persons-Killed - 2014 per 1L people  
37 non-null float64  
Poor light-Number of Persons-Injured - 2014  
36 non-null float64  
Poor light-Number of Persons-Injured - 2014 per 1L people  
36 non-null float64  
Falling of boulders-Total No. of Road Accidents - 2014  
36 non-null float64  
Falling of boulders-Total No. of Road Accidents - 2014 per 1L people  
36 non-null float64  
Falling of boulders-Number of Persons-Killed - 2014  
37 non-null int64  
Falling of boulders-Number of Persons-Killed - 2014 per 1L people  
37 non-null float64  
Falling of boulders-Number of Persons-Injured - 2014

```
36 non-null float64
Falling of boulders-Number of Persons-Injured - 2014 per 1L people
36 non-null float64
Other causes/causes not known-Total No. of Road Accidents - 2014
36 non-null float64
Other causes/causes not known-Total No. of Road Accidents - 2014 per 1L people
36 non-null float64
Other causes/causes not known-Number of Persons-Killed - 2014
37 non-null int64
Other causes/causes not known-Number of Persons-Killed - 2014 per 1L people
37 non-null float64
Other causes/causes not known-Number of Persons-Injured - 2014
36 non-null float64
Other causes/causes not known-Number of Persons-Injured - 2014 per 1L people
36 non-null float64
Population
37 non-null object
dtypes: float64(50), int64(10), object(3)
memory usage: 18.3+ KB
```

*Reasons for death due to Accidents :*

- Driver
- Other Driver's
- Pedestrian
- Condition of Vehicle
- Road Condition
- Weather Condition
- Passenger
- Poor Light
- Boulders
- Other Causes

## IMPORTANT ERROR -

*plt.bar valueerror: shape mismatch: objects cannot be broadcast to a single shape.*

[The solution..](#)

```
In [81]: driverFault = df5['Fault of Driver-Number of Persons-Killed - 2014 per 1L pe
otherDriversFault = df5['Fault of Driver of other vehicles-Number of Persons
pedestrianFault = df5['Fault of Pedestrian-Number of Persons-Killed - 2014 p
conditionOfVehicleFault = df5['Defect in Condition of Motor Vehicle-Number o
roadConditionFault = df5['Defect in Road Condition-Number of Persons-Killed
weatherConditionFault = df5['Weather Condition-Number of Persons-Killed - 20
passengerFault = df5['Fault of Passenger-Number of Persons-Killed - 2014 per
poorLightFault = df5['Poor light-Number of Persons-Killed - 2014 per 1L peop
bouldersFault = df5['Falling of boulders-Number of Persons-Killed - 2014 per
otherCauses = df5['Other causes/causes not known-Number of Persons-Killed -
```

```
In [122... plt.figure(figsize=(20,10))
```

```
Loading [MathJax]/extensions/Safe.js update({'font.size':18})
```

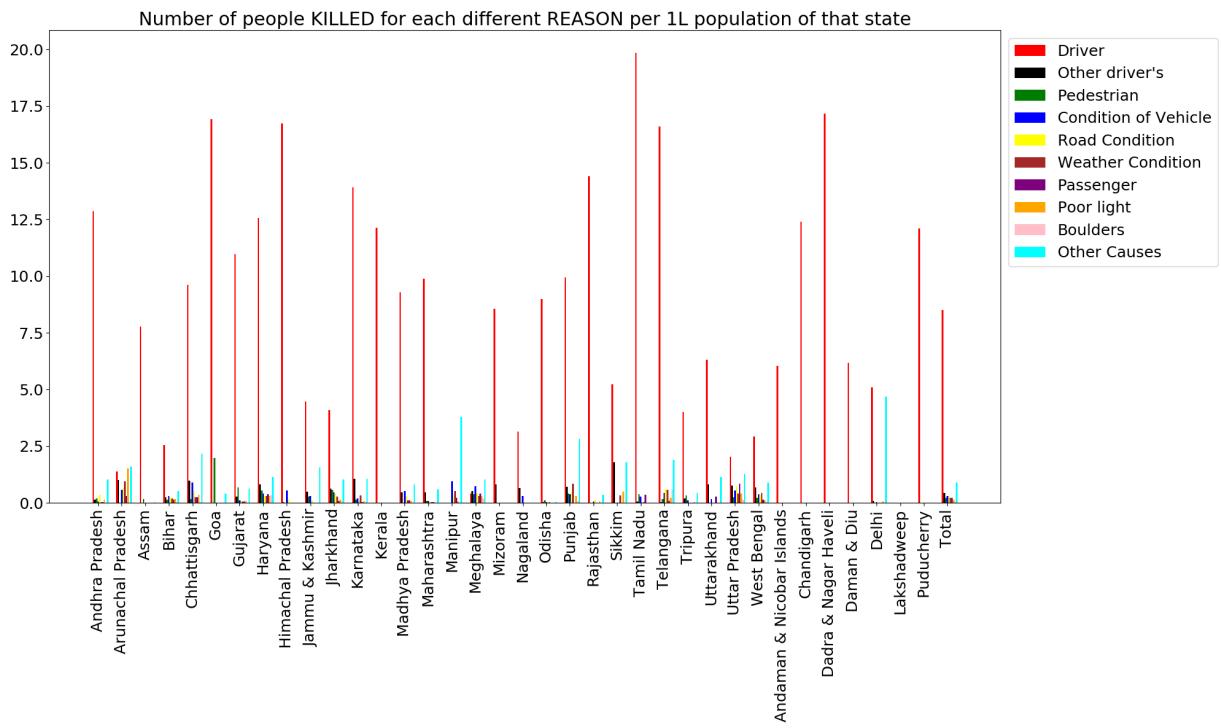
```

UT=np.array([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,
UT=UT*3

plt.bar(UT-0.6, driverFault, width = 0.2, color = 'r', align = 'center', lab
plt.bar(UT-0.4, otherDriversFault, width=0.2, color='black', align='center',
plt.bar(UT-0.2, pedestrianFault, width=0.2, color='g', align='center', label=
plt.bar(UT, conditionOfVehicleFault, width=0.2, color='b', align='center', la
plt.bar(UT+0.2, roadConditionFault, width=0.2, color='yellow', align='center'
plt.bar(UT+0.4, weatherConditionFault, width=0.2, color='brown', align='cente
plt.bar(UT+0.6, passengerFault, width=0.2, color='purple', align='center', la
plt.bar(UT+0.8, poorLightFault, width=0.2, color='orange', align='center', la
plt.bar(UT+1.0, bouldersFault, width=0.2, color='pink', align='center', label
plt.bar(UT+1.2, otherCauses, width=0.2, color='cyan', align='center', label='

plt.xticks(UT,df['States/UTs'], rotation='vertical')
plt.legend(loc='upper left', bbox_to_anchor=(1,1))
plt.title("Number of people KILLED for each different REASON per 1L populati
plt.show()

```



```

In [123... plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

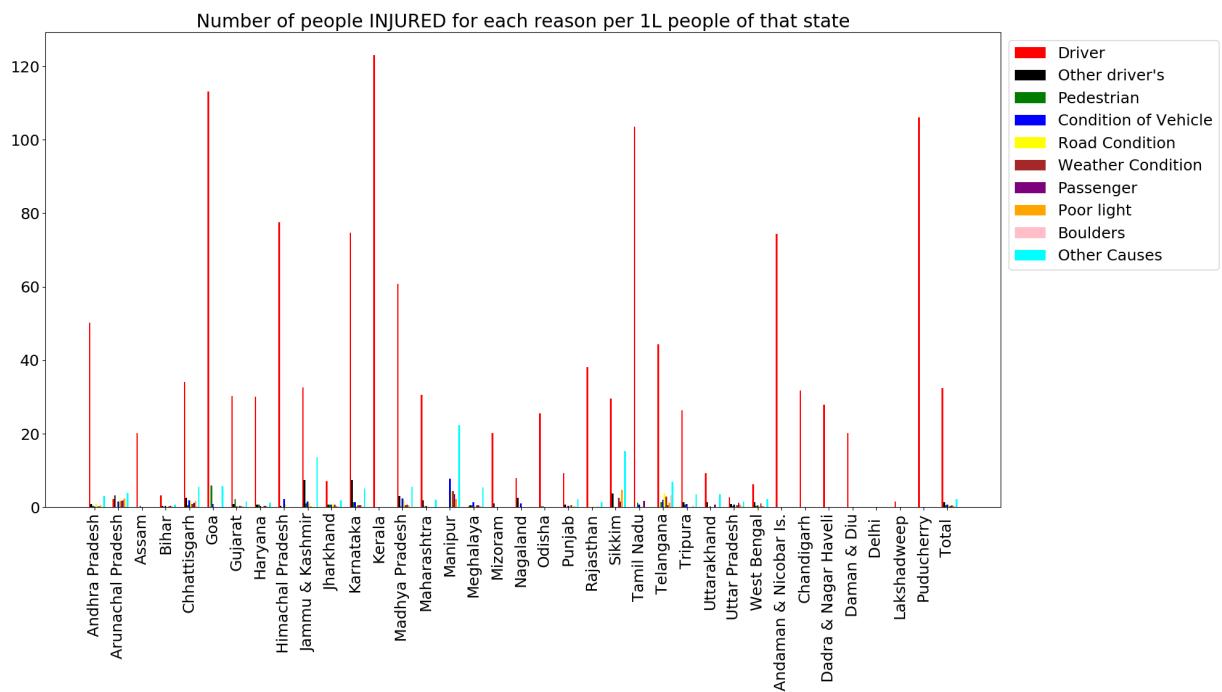
plt.bar(UT-0.6, df5['Fault of Driver-Number of Persons-Injured - 2014 per 1L
plt.bar(UT-0.4, df5['Fault of Driver of other vehicles-Number of Persons-Inj
plt.bar(UT-0.2, df5['Fault of Pedestrian-Number of Persons-Injured - 2014 pe
plt.bar(UT, df5['Defect in Condition of Motor Vehicle-Number of Persons-Inju
plt.bar(UT+0.2, df5['Defect in Road Condition-Number of Persons-Injured - 20
plt.bar(UT+0.4, df5['Weather Condition-Number of Persons-Injured - 2014 per
plt.bar(UT+0.6, df5['Fault of Passenger-Number of Persons-Injured - 2014 per
plt.bar(UT+0.8, df5['Poor light-Number of Persons-Injured - 2014 per 1L peop
plt.bar(UT+1.0, df5['Falling of boulders-Number of Persons-Injured - 2014 pe
plt.bar(UT+1.2, df5['Other causes/causes not known-Number of Persons-Injured

```

```

plt.xticks(UT,df5['States/UTs'],rotation='vertical')
plt.legend(loc='upper left', bbox_to_anchor=(1,1))
plt.title("Number of people INJURED for each reason per 1L people of that st
plt.show()

```



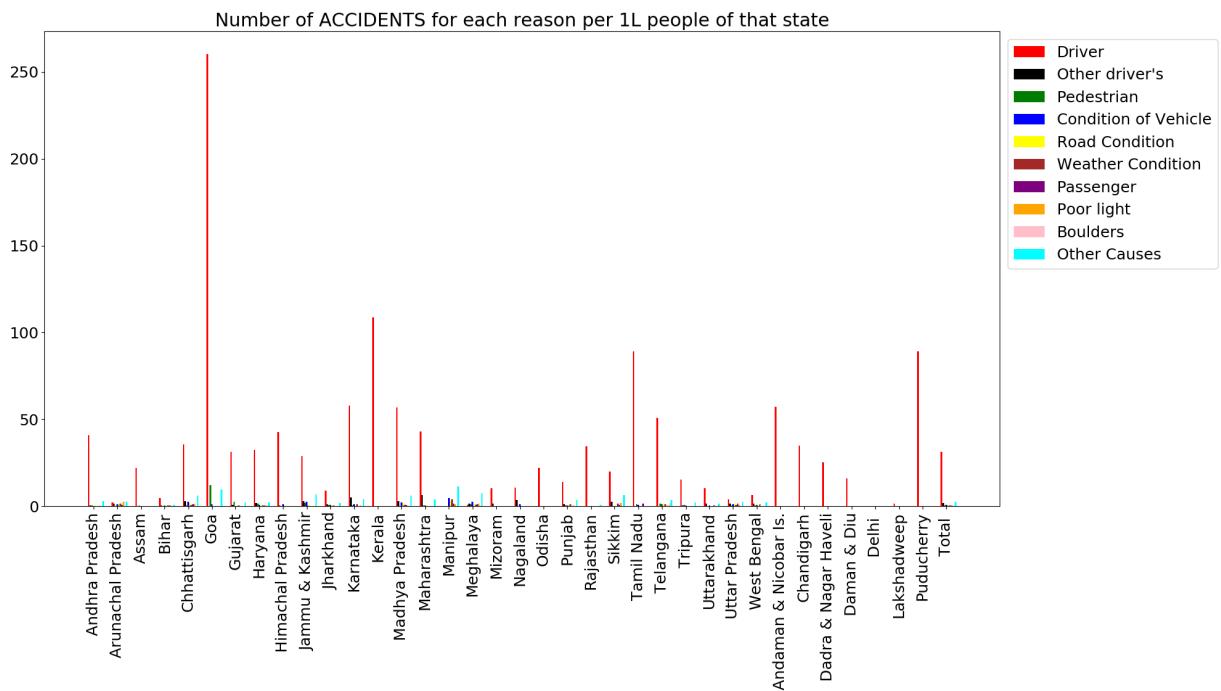
```

In [124... plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

plt.bar(UT-0.6,df5['Fault of Driver-Total No. of Road Accidents - 2014 per 1
plt.bar(UT-0.4,df5['Fault of Driver of other vehicles-Total No. of Road Acci
plt.bar(UT-0.2,df5['Fault of Pedestrian-Total No. of Road Accidents - 2014 p
plt.bar(UT,df5['Defect in Condition of Motor Vehicle-Total No. of Road Accid
plt.bar(UT+0.2,df5['Defect in Road Condition-Total No. of Road Accidents - 2
plt.bar(UT+0.4,df5['Weather Condition-Total No. of Road Accidents - 2014 per
plt.bar(UT+0.6,df5['Fault of Passenger-Total No. of Road Accidents - 2014 pe
plt.bar(UT+0.8,df5['Poor light-Total No. of Road Accidents - 2014 per 1L ped
plt.bar(UT+1.0,df5['Falling of boulders-Total No. of Road Accidents - 2014 p
plt.bar(UT+1.2,df5['Other causes/causes not known-Total No. of Road Accident

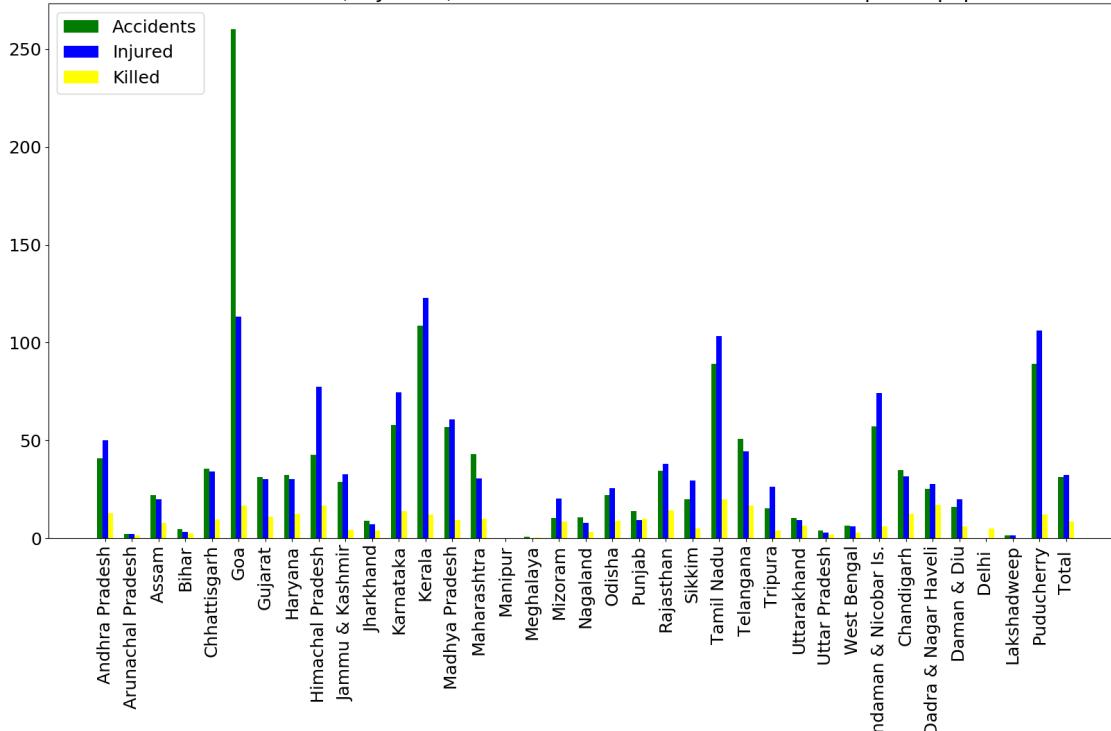
plt.xticks(UT,df5['States/UTs'],rotation='vertical')
plt.legend(loc='upper left', bbox_to_anchor=(1,1))
plt.title("Number of ACCIDENTS for each reason per 1L people of that state")
plt.show()

```



```
In [137...]: plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})
UT=np.array([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,
            25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50])
plt.bar(UT-0.2,df5['Fault of Driver-Total No. of Road Accidents - 2014 per 1L people'],color='red')
plt.bar(UT,df5['Fault of Driver-Number of Persons-Injured - 2014 per 1L people'],color='blue')
plt.bar(UT+0.2,df5['Fault of Driver-Number of Persons-Killed - 2014 per 1L people'],color='green')
plt.xticks(UT,df5['States/UTs'],rotation='vertical')
plt.legend(loc="best")
plt.title("Total number of ROAD ACCIDENTS, INJURIES, DEATHS due to FAULT OF DRIVERS")
plt.show()
```

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



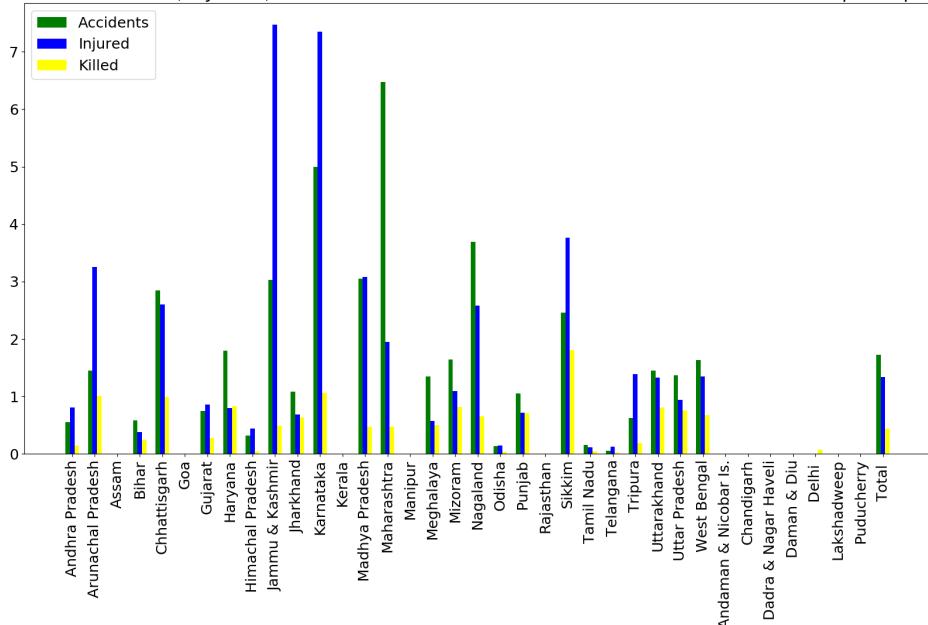
In [136]:

```
plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

plt.bar(UT-0.2,df5['Fault of Driver of other vehicles-Total No. of Road Acci
plt.bar(UT,df5['Fault of Driver of other vehicles-Number of Persons-Injured
plt.bar(UT+0.2,df5['Fault of Driver of other vehicles-Number of Persons-Kill

plt.xticks(UT,df5['States/UTs'],rotation='vertical')
plt.legend(loc="best")
plt.title("Total number of ROAD ACCIDENTS, INJURIES, DEATHS due to the FAULT
plt.show()
```

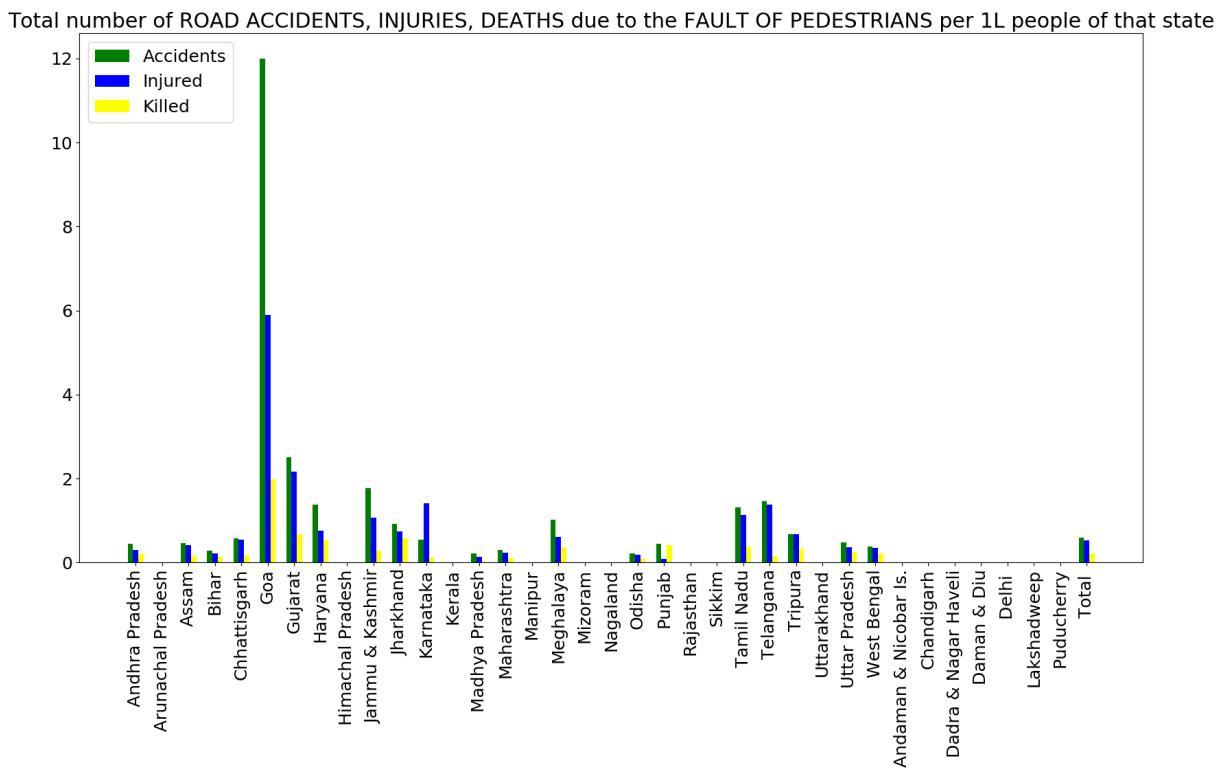
Total number of ROAD ACCIDENTS, INJURIES, DEATHS due to the FAULT OF DRIVER'S FROM OTHER VEHICLES per 1L people of that state.



```
In [138... plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

plt.bar(UT-0.2,df5['Fault of Pedestrian-Total No. of Road Accidents - 2014 per 1L people of that state'])
plt.bar(UT,df5['Fault of Pedestrian-Number of Persons-Injured - 2014 per 1L people of that state'])
plt.bar(UT+0.2,df5['Fault of Pedestrian-Number of Persons-Killed - 2014 per 1L people of that state'])

plt.xticks(UT,df5['States/UTs'],rotation='vertical')
plt.legend(loc="best")
plt.title("Total number of ROAD ACCIDENTS, INJURIES, DEATHS due to the FAULT OF PEDESTRIANS per 1L people of that state")
plt.show()
```

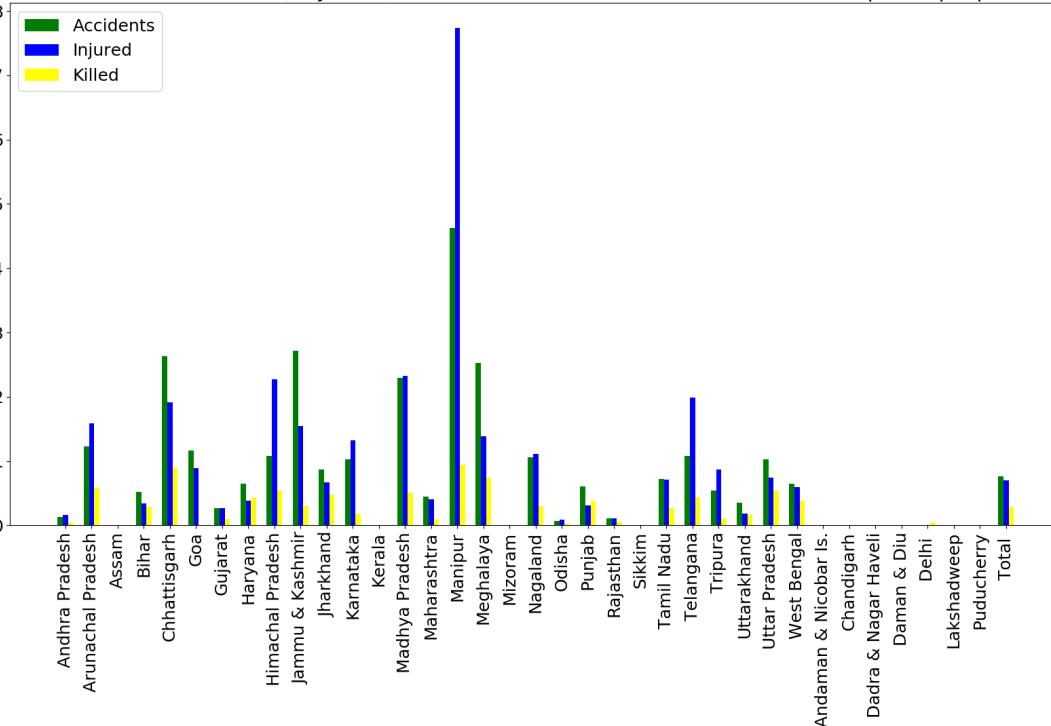


```
In [139... plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

plt.bar(UT-0.2,df5['Defect in Condition of Motor Vehicle-Total No. of Road Accidents - 2014 per 1L people of that state'])
plt.bar(UT,df5['Defect in Condition of Motor Vehicle-Number of Persons-Injured - 2014 per 1L people of that state'])
plt.bar(UT+0.2,df5['Defect in Condition of Motor Vehicle-Number of Persons-Killed - 2014 per 1L people of that state'])

plt.xticks(UT,df5['States/UTs'],rotation='vertical')
plt.legend(loc="best")
plt.title("Total number of ROAD ACCIDENTS, INJURIES, DEATHS due to the DEFECT IN CONDITION OF MOTOR VEHICLE per 1L people of that state")
plt.show()
```

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



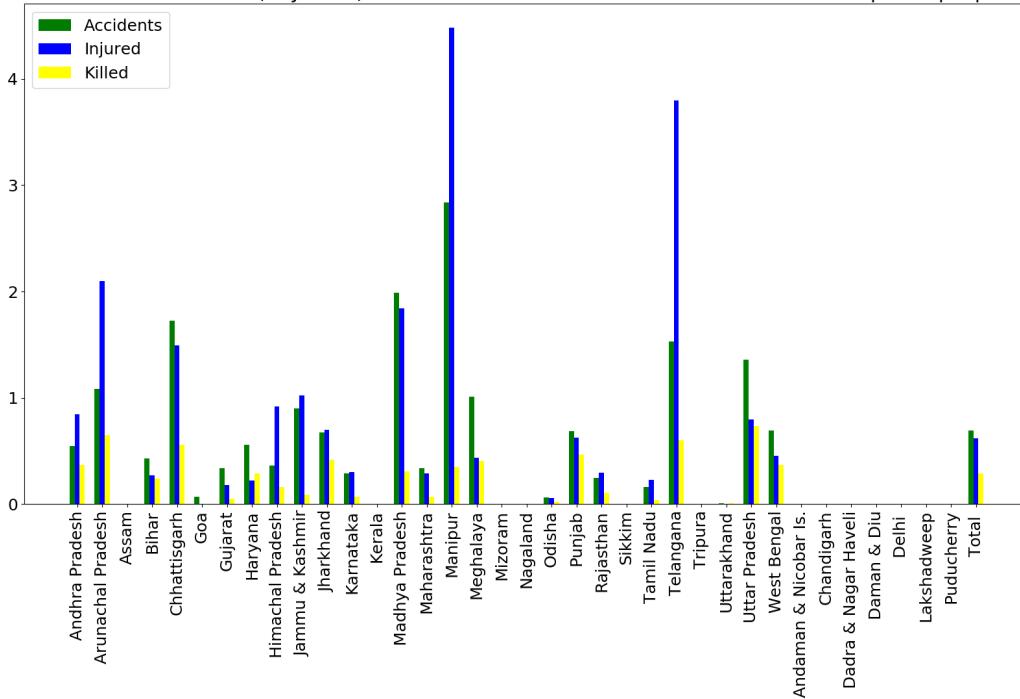
In [140]:

```
plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

plt.bar(UT-0.2,df5['Defect in Road Condition-Total No. of Road Accidents - 2014 per 1L people'])
plt.bar(UT,df5['Defect in Road Condition-Number of Persons-Injured - 2014 per 1L people'])
plt.bar(UT+0.2,df5['Defect in Road Condition-Number of Persons-Killed - 2014 per 1L people'])

plt.xticks(UT,df5['States/UTs'],rotation='vertical')
plt.legend(loc="best")
plt.title("Total number of ROAD ACCIDENTS, INJURIES, DEATHS due to DEFECTS IN THE VEHICLE per 1L people of that state")
plt.show()
```

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

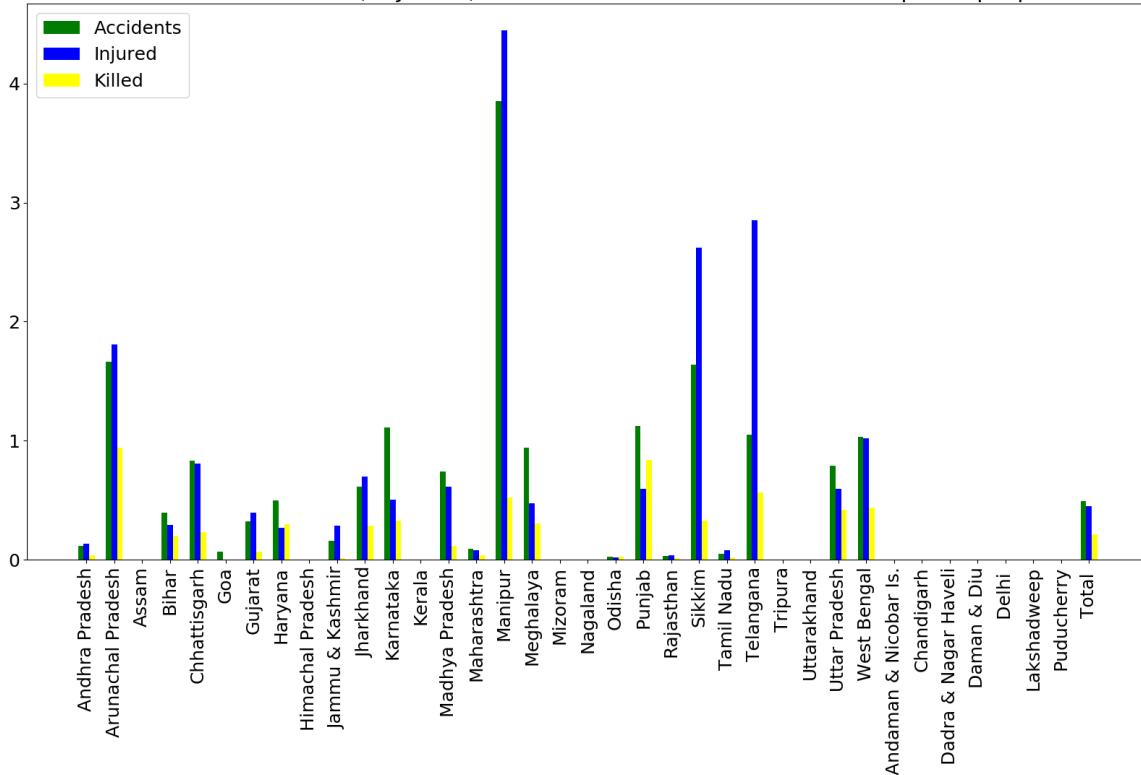


```
In [141]: plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

plt.bar(UT-0.2,df5['Weather Condition-Total No. of Road Accidents - 2014 per 1L people'])
plt.bar(UT,df5['Weather Condition-Number of Persons-Injured - 2014 per 1L people'])
plt.bar(UT+0.2,df5['Weather Condition-Number of Persons-Killed - 2014 per 1L people'])

plt.xticks(UT,df5['States/UTs'],rotation='vertical')
plt.legend(loc="best")
plt.title("Total number of ROAD ACCIDENTS, INJURIES, DEATHS due to WEATHER C")
plt.show()
```

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

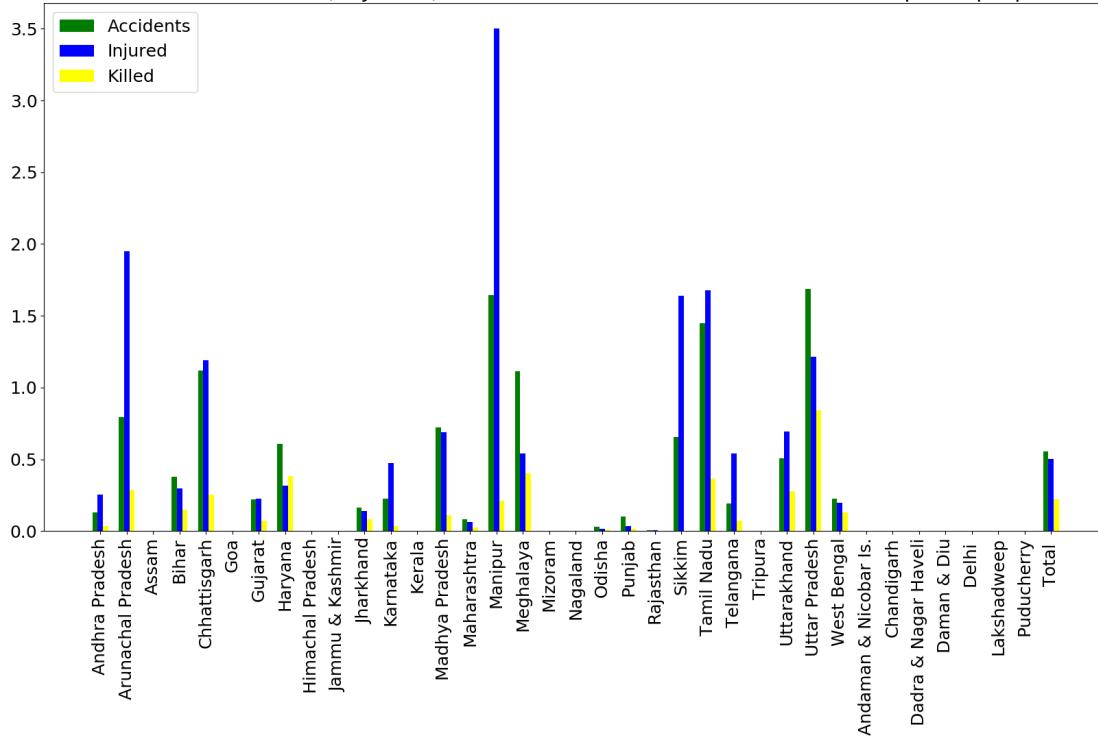


```
In [142]: plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

plt.bar(UT-0.2,df5['Fault of Passenger-Total No. of Road Accidents - 2014 per 1L people'])
plt.bar(UT,df5['Fault of Passenger-Number of Persons-Injured - 2014 per 1L people'])
plt.bar(UT+0.2,df5['Fault of Passenger-Number of Persons-Killed - 2014 per 1L people'])

plt.xticks(UT,df5['States/UTs'],rotation='vertical')
plt.legend(loc="best")
plt.title("Total number of ROAD ACCIDENTS, INJURIES, DEATHS due to the FAULT")
plt.show()
```

Total number of ROAD ACCIDENTS, INJURIES, DEATHS due to the FAULT OF PASSENGER/S per 1L people of that state



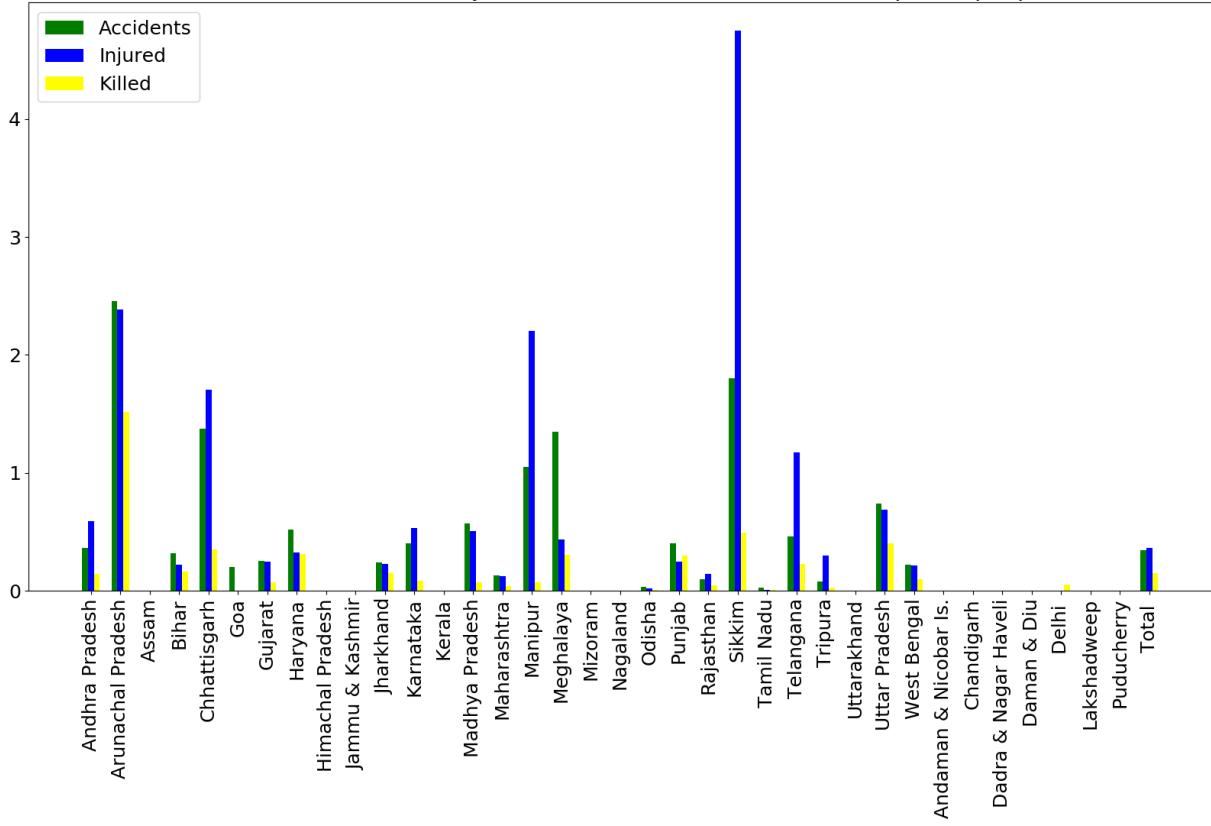
In [143]:

```
plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

plt.bar(UT-0.2,df5['Poor light-Total No. of Road Accidents - 2014 per 1L peo
plt.bar(UT,df5['Poor light-Number of Persons-Injured - 2014 per 1L people'],
plt.bar(UT+0.2,df5['Poor light-Number of Persons-Killed - 2014 per 1L people

plt.xticks(UT,df5['States/UTs'],rotation='vertical')
plt.legend(loc="best")
plt.title("Total number of ROAD ACCIDENTS, INJURIES, DEATHS due to POOR LIGH
plt.show()
```

Total number of ROAD ACCIDENTS, INJURIES, DEATHS due to POOR LIGHT per 1L people of that state

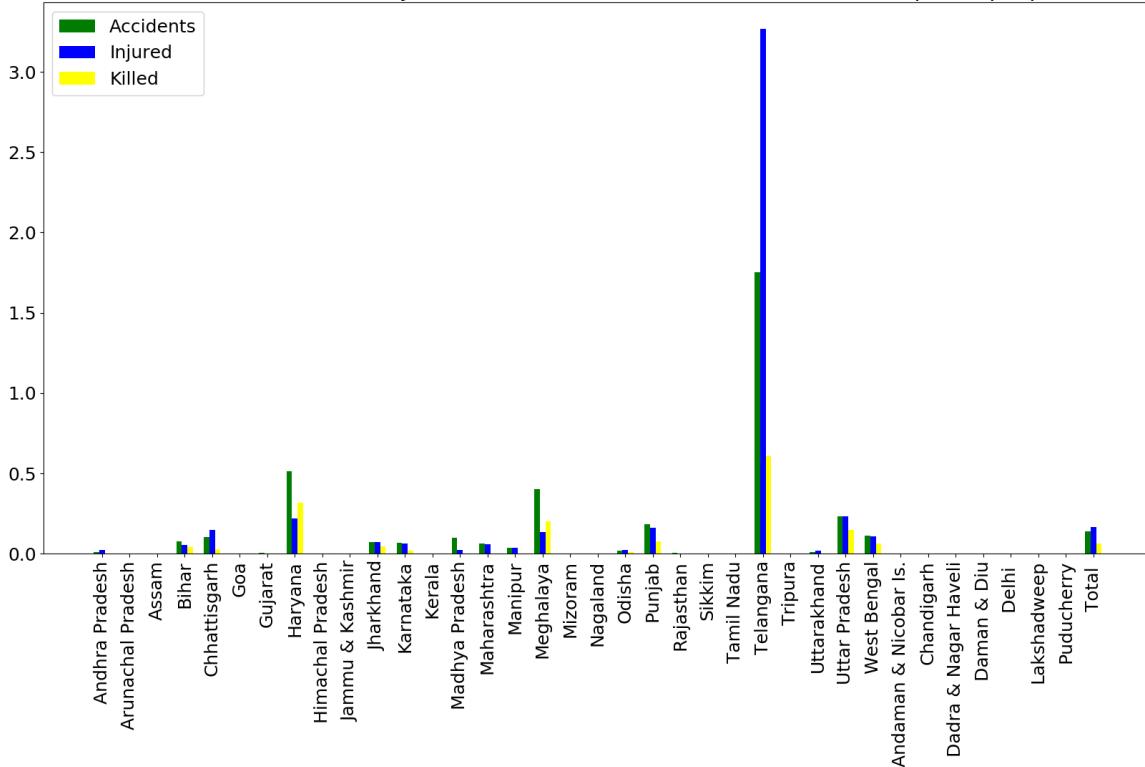


```
In [144]: plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

plt.bar(UT-0.2,df5['Falling of boulders-Total No. of Road Accidents - 2014 per 1L'])
plt.bar(UT,df5['Falling of boulders-Number of Persons-Injured - 2014 per 1L'])
plt.bar(UT+0.2,df5['Falling of boulders-Number of Persons-Killed - 2014 per 1L'])

plt.xticks(UT,df5['States/UTs'],rotation='vertical')
plt.legend(loc="best")
plt.title("Total number of ROAD ACCIDENTS, INJURIES, DEATHS due to FALLING OF BOULDERS")
plt.show()
```

Total number of ROAD ACCIDENTS, INJURIES, DEATHS due to FALLING OF BOULDERS per 1L people of that state

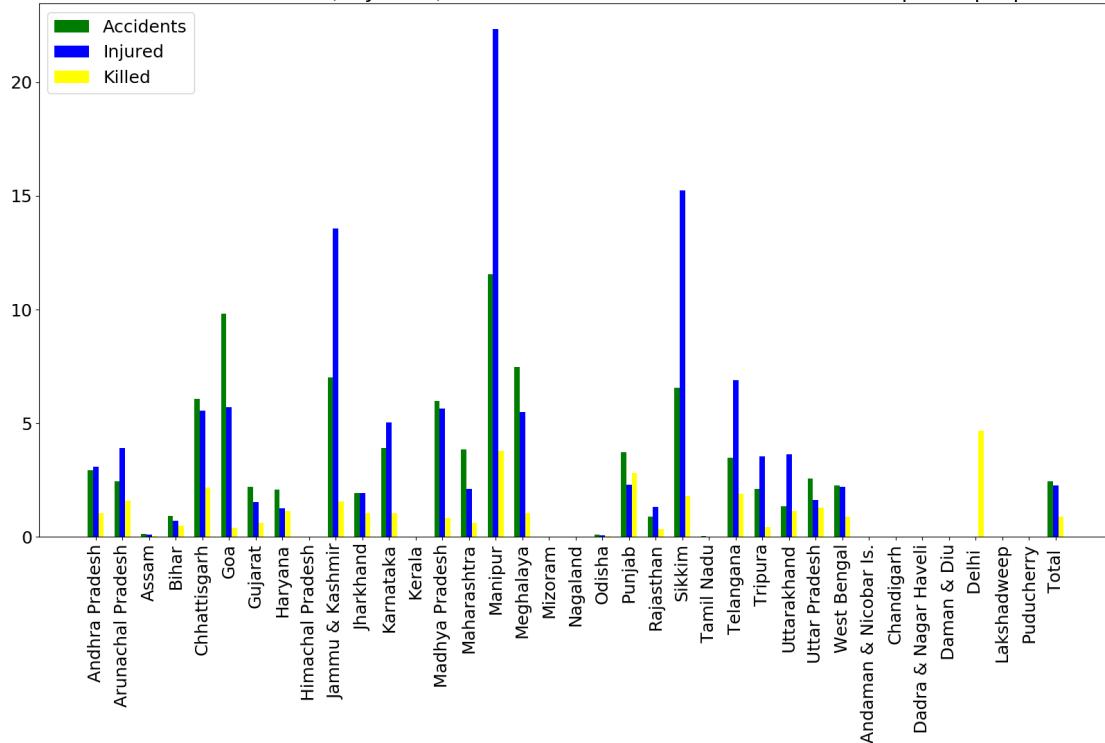


```
In [145...]: plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

plt.bar(UT-0.2,df5['Other causes/causes not known-Total No. of Road Accident'])
plt.bar(UT,df5['Other causes/causes not known-Number of Persons-Injured - 20'])
plt.bar(UT+0.2,df5['Other causes/causes not known-Number of Persons-Killed -'])

plt.xticks(UT,df5['States/UTs'],rotation='vertical')
plt.legend(loc="best")
plt.title("Total number of ROAD ACCIDENTS, INJURIES, DEATHS due to OTHER/UNK")
plt.show()
```

Total number of ROAD ACCIDENTS, INJURIES, DEATHS due to OTHER/UNKNOWN CAUSES per 1L people of that state



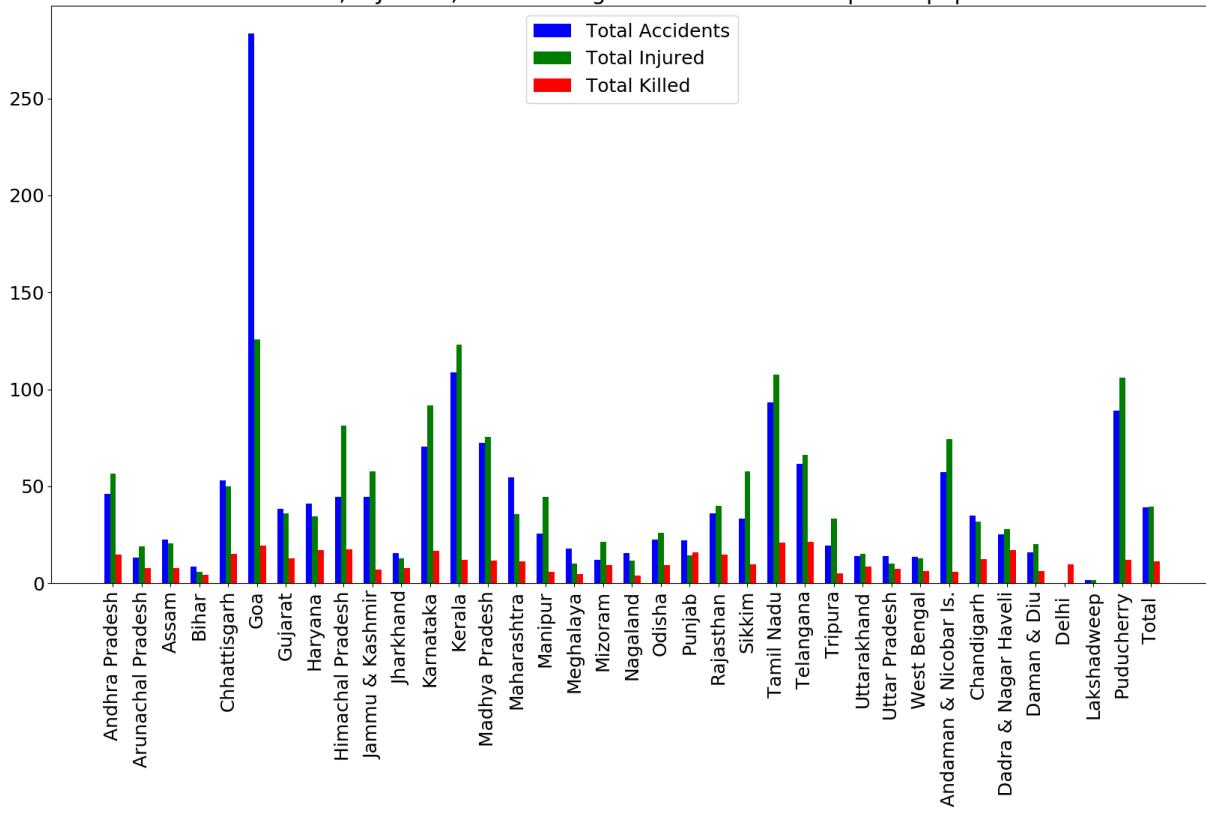
In [120]:

```
df5['Sum Total Road Accidents - 2014 per 1L people']=df5['Fault of Driver of  
df5['Defect in Road Condition-Total No. of Road Accidents - 2014 per 1L  
df5['Poor light-Total No. of Road Accidents - 2014 per 1L people']+df5['  
df5['Fault of Driver-Total No. of Road Accidents - 2014 per 1L people']  
  
df5['Sum Total Number of Persons Killed - 2014 per 1L people']=df5['Fault of  
df5['Defect in Road Condition-Number of Persons-Killed - 2014 per 1L pec  
df5['Poor light-Number of Persons-Killed - 2014 per 1L people']+df5['Fa  
df5['Fault of Driver-Number of Persons-Killed - 2014 per 1L people']  
  
df5['Sum Total Number of Persons Injured - 2014 per 1L people']=df5['Fault c  
df5['Defect in Road Condition-Number of Persons-Injured - 2014 per 1L pe  
df5['Poor light-Number of Persons-Injured - 2014 per 1L people']+df5['Fa  
df5['Fault of Driver-Number of Persons-Injured - 2014 per 1L people']
```

In [146]:

```
plt.figure(figsize=(20,10))  
plt.rcParams.update({'font.size':18})  
  
plt.bar(UT-0.2,df5['Sum Total Road Accidents - 2014 per 1L people'],width=0.  
plt.bar(UT,df5['Sum Total Number of Persons Injured - 2014 per 1L people'],w  
plt.bar(UT+0.2,df5['Sum Total Number of Persons Killed - 2014 per 1L people'  
  
plt.xticks(UT,df5['States/UTs'],rotation='vertical')  
plt.title("Total ROAD ACCIDENTS, INJURIES, DEATHS in general for each state  
plt.legend(loc="best")  
plt.show()
```

Total ROAD ACCIDENTS, INJURIES, DEATHS in general for each state per 1L population of that state



[Go to top.](#)

## DF6 -

This dataset tells about the number of accidents happening along **different types of vehicles**.

```
In [147]: df6 = pd.read_csv('A:\Data Analysis Jupyter\Road-Accidents-Of-India\Database')
df6.head()
```

Out[147]:

Sl. No.	States/UTs	Two- Wheeler s - Number of Road Accidents - Fatal - 2014	Two- Wheeler s - Number of Road Accidents - Fatal - 2014 per 1L people	Two- Wheeler s - Number of Road Accidents - Total - 2014	Two- Wheeler s - Number of Road Accidents - Total - 2014 per 1L people	Two- Wheeler s - Number of Persons - Killed - 2014	Two- Wheeler s - Number of Persons - Killed - 2014 per 1L people	
0	1	Andhra Pradesh	1961	3.741606	7239	13.812077	2003	3.821742
1	2	Arunachal Pradesh	15	1.084029	34	2.457132	21	1.517640
2	3	Assam	489	1.567028	1508	4.832470	537	1.720846
3	4	Bihar	769	0.738717	1750	1.681085	820	0.787708
4	5	Chhattisgarh	1074	4.204313	4787	18.739334	1126	4.407873

5 rows × 9 columns

In [157...]

```
row, column = df6.shape
round(((row - df6.isnull().sum()) / row) * 100)
```

Out[157]: Sl. No.  
100.0  
States/UTs  
100.0  
Two-Wheelers - Number of Road Accidents - Fatal - 2014  
100.0  
Two-Wheelers - Number of Road Accidents - Fatal - 2014 per 1L people  
100.0  
Two-Wheelers - Number of Road Accidents - Total - 2014  
100.0  
Two-Wheelers - Number of Road Accidents - Total - 2014 per 1L people  
100.0  
Two-Wheelers - Number of Persons - Killed - 2014  
100.0  
Two-Wheelers - Number of Persons - Killed - 2014 per 1L people  
100.0  
Two-Wheelers - Number of Persons - Injured - 2014  
100.0  
Two-Wheelers - Number of Persons - Injured - 2014 per 1L people  
100.0  
Auto-Rickshaws - Number of Road Accidents - Fatal - 2014  
100.0  
Auto-Rickshaws - Number of Road Accidents - Fatal - 2014 per 1L people  
100.0  
Auto-Rickshaws - Number of Road Accidents - Total - 2014  
100.0  
Auto-Rickshaws - Number of Road Accidents - Total - 2014 per 1L people  
100.0  
Auto-Rickshaws - Number of Persons - Killed - 2014  
100.0  
Auto-Rickshaws - Number of Persons - Killed - 2014 per 1L people  
100.0  
Auto-Rickshaws - Number of Persons - Injured - 2014  
100.0  
Auto-Rickshaws - Number of Persons - Injured - 2014 per 1L people  
100.0  
Cars, Jeeps,Taxis - Number of Road Accidents - Fatal - 2014  
100.0  
Cars, Jeeps,Taxis - Number of Road Accidents - Fatal - 2014 per 1L people  
100.0  
Cars, Jeeps,Taxis - Number of Road Accidents - Total - 2014  
100.0  
Cars, Jeeps,Taxis - Number of Road Accidents - Total - 2014 per 1L people  
100.0  
Cars, Jeeps,Taxis - Number of Persons - Killed - 2014  
100.0  
Cars, Jeeps,Taxis - Number of Persons - Killed - 2014 per 1L people  
100.0  
Cars, Jeeps,Taxis - Number of Persons - Injured - 2014  
100.0  
Cars, Jeeps,Taxis - Number of Persons - Injured - 2014 per 1L people  
100.0  
Buses - Number of Road Accidents - Fatal - 2014  
100.0  
Buses - Number of Road Accidents - Fatal - 2014 per 1L people

Buses - Number of Road Accidents - Total - 2014  
100.0  
Buses - Number of Road Accidents - Total - 2014 per 1L people  
100.0  
Buses - Number of Persons - Killed - 2014  
100.0  
Buses - Number of Persons - Killed - 2014 per 1L people  
100.0  
Buses - Number of Persons - Injured - 2014  
100.0  
Buses - Number of Persons - Injured - 2014 per 1L people  
100.0  
Trucks, Tempos, MAVs, Tractors - Number of Road Accidents - Fatal - 2014  
100.0  
Trucks, Tempos, MAVs, Tractors - Number of Road Accidents - Fatal - 2014 per  
1L people 100.0  
Trucks, Tempos, MAVs, Tractors - Number of Road Accidents - Total - 2014  
100.0  
Trucks, Tempos, MAVs, Tractors - Number of Road Accidents - Total - 2014 per  
1L people 100.0  
Trucks, Tempos, MAVs, Tractors - Number of Persons - Killed - 2014  
100.0  
Trucks, Tempos, MAVs, Tractors - Number of Persons - Killed - 2014 per 1L pe  
ople 100.0  
Trucks, Tempos, MAVs, Tractors - Number of Persons - Injured - 2014  
100.0  
Trucks, Tempos, MAVs, Tractors - Number of Persons - Injured - 2014 per 1L p  
eople 100.0  
Other Motor Vehicles - Number of Road Accidents - Fatal - 2014  
100.0  
Other Motor Vehicles - Number of Road Accidents - Fatal - 2014 per 1L peop  
le 100.0  
Other Motor Vehicles - Number of Road Accidents - Total - 2014  
100.0  
Other Motor Vehicles - Number of Road Accidents - Total - 2014 per 1L peop  
le 100.0  
Other Motor Vehicles - Number of Persons - Killed - 2014  
100.0  
Other Motor Vehicles - Number of Persons - Killed - 2014 per 1L people  
100.0  
Other Motor Vehicles - Number of Persons - Injured - 2014  
100.0  
Other Motor Vehicles - Number of Persons - Injured - 2014 per 1L people  
100.0  
Other Vehicles/Objects - Number of Road Accidents - Fatal - 2014  
100.0  
Other Vehicles/Objects - Number of Road Accidents - Fatal - 2014 per 1L pe  
ople 100.0  
Other Vehicles/Objects - Number of Road Accidents - Total - 2014  
100.0  
Other Vehicles/Objects - Number of Road Accidents - Total - 2014 per 1L pe  
ople 100.0  
Other Vehicles/Objects - Number of Persons - Killed - 2014  
100.0  
Other Vehicles/Objects - Number of Persons - Killed - 2014 per 1L people

```

Other Vehicles/Objects - Number of Persons - Injured - 2014
100.0
Other Vehicles/Objects - Number of Persons - Injured - 2014 per 1L people
100.0
Population
100.0
dtype: float64

```

```
In [158]: df6.dropna(axis = 0, how = 'any', inplace = True)
```

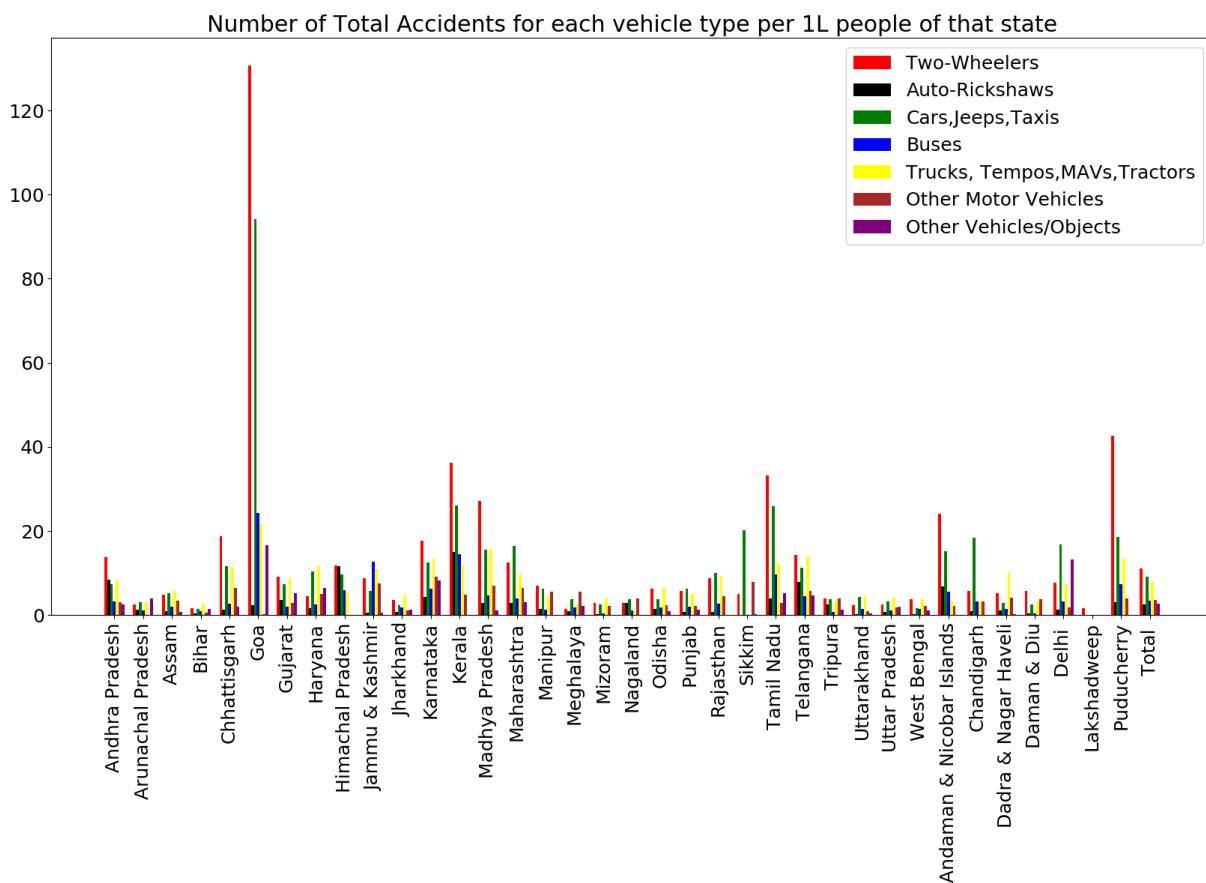
```

In [159]: plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})
UT=np.array([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,
UT=UT*2

plt.bar(UT-0.6,df6['Two-Wheelers - Number of Road Accidents - Total - 2014 per 1L people'])
plt.bar(UT-0.4,df6['Auto-Rickshaws - Number of Road Accidents - Total - 2014 per 1L people'])
plt.bar(UT-0.2,df6['Cars, Jeeps,Taxis - Number of Road Accidents - Total - 2014 per 1L people'])
plt.bar(UT,df6['Buses - Number of Road Accidents - Total - 2014 per 1L people'])
plt.bar(UT+0.2,df6['Trucks, Tempos,MAVs,Tractors - Number of Road Accidents - Total per 1L people'])
plt.bar(UT+0.4,df6['Other Motor Vehicles - Number of Road Accidents - Total per 1L people'])
plt.bar(UT+0.6,df6['Other Vehicles/Objects - Number of Road Accidents - Total per 1L people'])

plt.xticks(UT,df6['States/UTs'],rotation='vertical')
plt.legend(loc="best")
plt.title("Number of Total Accidents for each vehicle type per 1L people of that state")
plt.show()

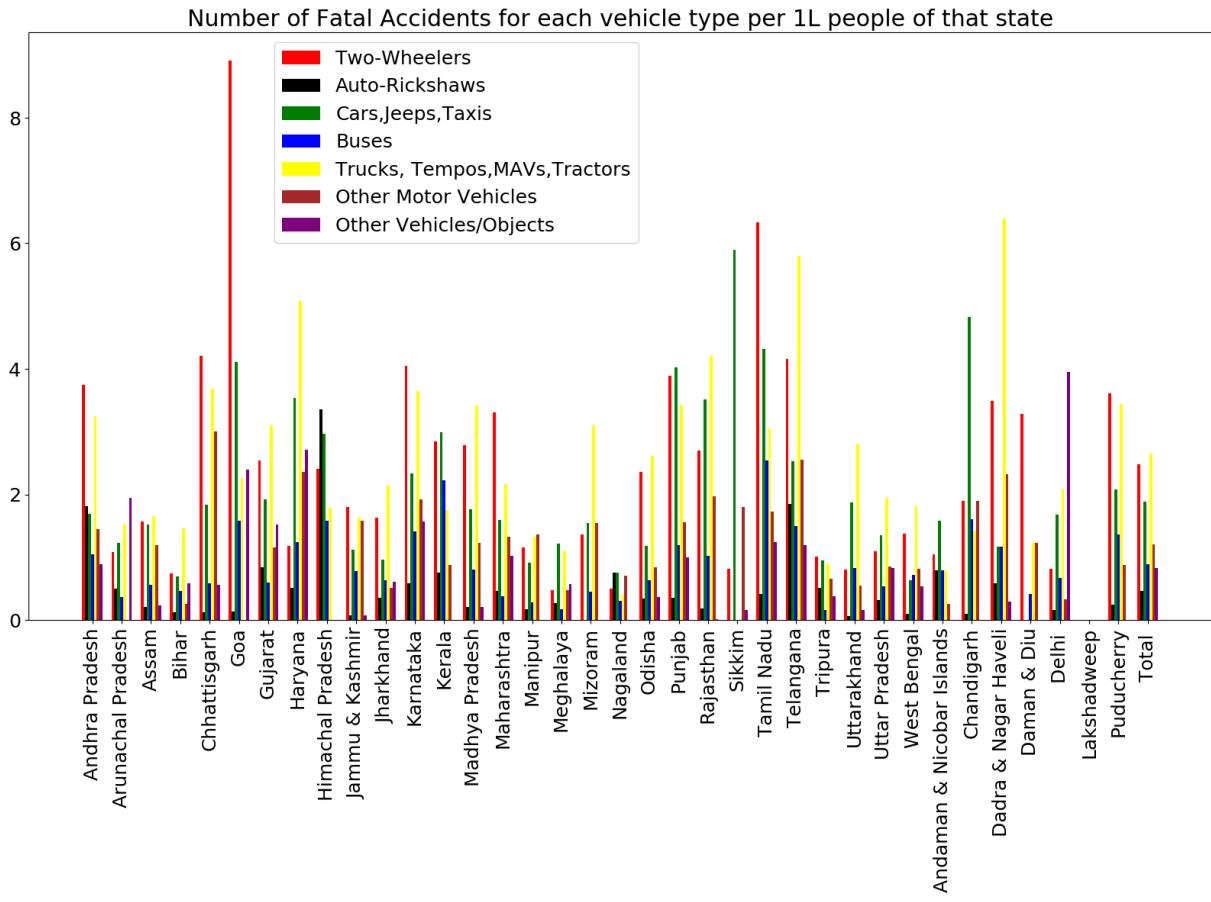
```



```
In [161... plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

plt.bar(UT-0.6,df6['Two-Wheelers - Number of Road Accidents - Fatal - 2014 per 1L people'],color='red')
plt.bar(UT-0.4,df6['Auto-Rickshaws - Number of Road Accidents - Fatal - 2014 per 1L people'],color='black')
plt.bar(UT-0.2,df6['Cars, Jeeps,Taxis - Number of Road Accidents - Fatal - 2014 per 1L people'],color='green')
plt.bar(UT,df6['Buses - Number of Road Accidents - Fatal - 2014 per 1L people'],color='blue')
plt.bar(UT+0.2,df6['Trucks, Tempos,MAVs,Tractors - Number of Road Accidents - Fatal - 2014 per 1L people'],color='yellow')
plt.bar(UT+0.4,df6['Other Motor Vehicles - Number of Road Accidents - Fatal - 2014 per 1L people'],color='brown')
plt.bar(UT+0.6,df6['Other Vehicles/Objects - Number of Road Accidents - Fatal - 2014 per 1L people'],color='purple')

plt.xticks(UT,df6['States/UTs'],rotation='vertical')
plt.legend(loc='upper left', bbox_to_anchor=(0.2,1))
plt.title("Number of Fatal Accidents for each vehicle type per 1L people of that state")
plt.show()
```



```
In [162... plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

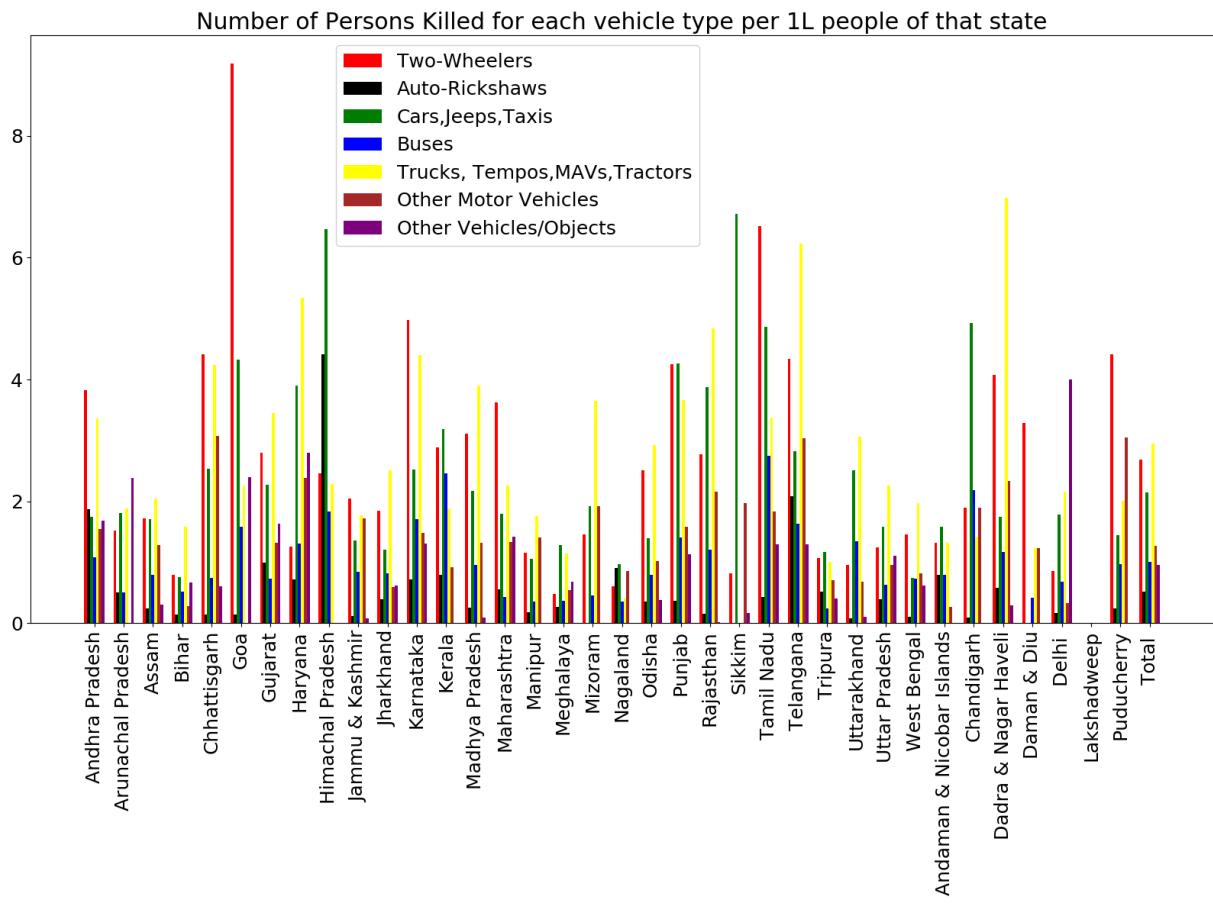
plt.bar(UT-0.6,df6['Two-Wheelers - Number of Persons - Killed - 2014 per 1L people'],color='red')
plt.bar(UT-0.4,df6['Auto-Rickshaws - Number of Persons - Killed - 2014 per 1L people'],color='black')
plt.bar(UT-0.2,df6['Cars, Jeeps,Taxis - Number of Persons - Killed - 2014 per 1L people'],color='green')
plt.bar(UT,df6['Buses - Number of Persons - Killed - 2014 per 1L people'],color='blue')
plt.bar(UT+0.2,df6['Trucks, Tempos, MAVs, Tractors - Number of Persons - Killed - 2014 per 1L people'],color='yellow')
plt.bar(UT+0.4,df6['Other Motor Vehicles - Number of Persons - Killed - 2014 per 1L people'],color='brown')
plt.bar(UT+0.6,df6['Other Vehicles/Objects - Number of Persons - Killed - 2014 per 1L people'],color='purple')

plt.xticks(UT,df6['States/UTs'],rotation='vertical')
plt.show()
```

```

plt.legend(loc='upper left', bbox_to_anchor=(0.25,1))
plt.title("Number of Persons Killed for each vehicle type per 1L people of t
plt.show()

```



```

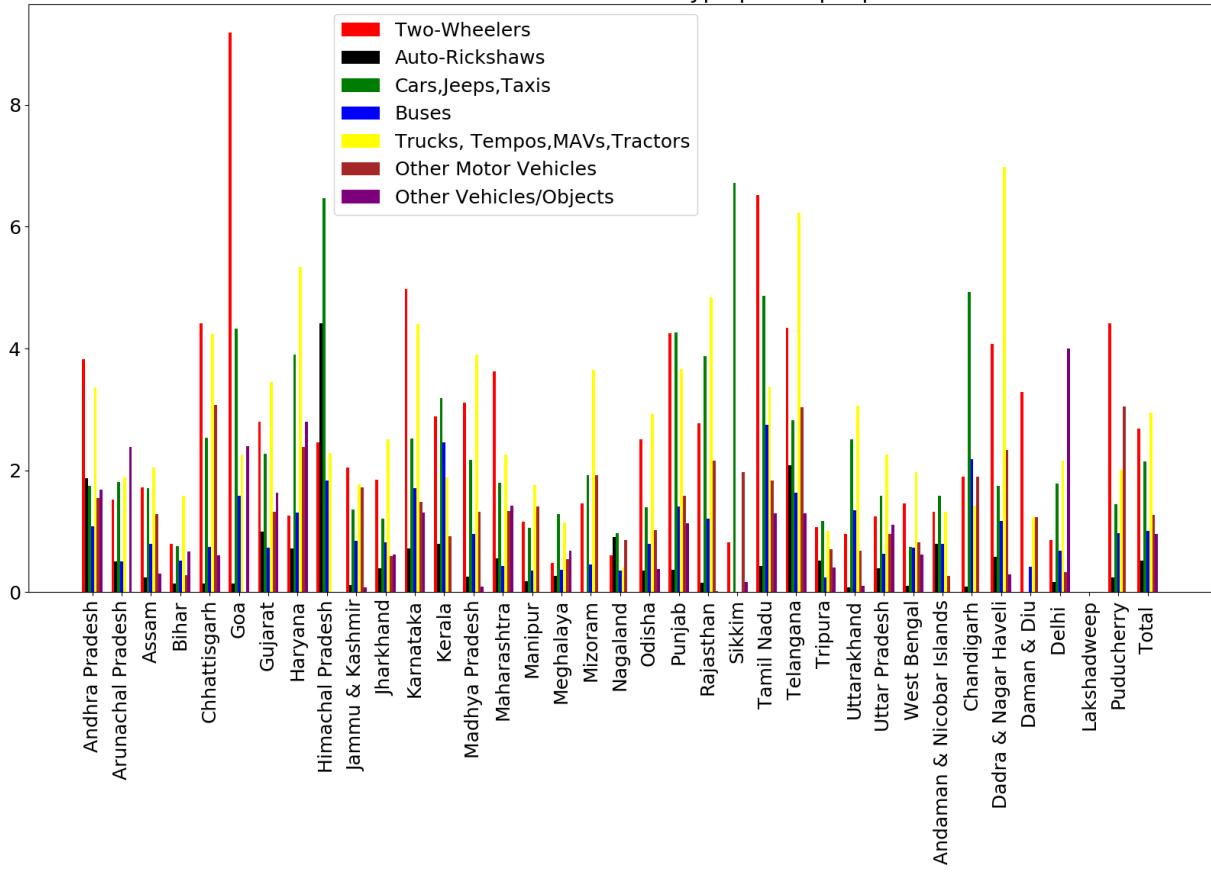
In [163]: plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

plt.bar(UT-0.6,df6['Two-Wheelers - Number of Persons - Killed - 2014 per 1L peop
plt.bar(UT-0.4,df6['Auto-Rickshaws - Number of Persons - Killed - 2014 per 1L peop
plt.bar(UT-0.2,df6['Cars, Jeeps,Taxis - Number of Persons - Killed - 2014 per 1L p
plt.bar(UT,df6['Buses - Number of Persons - Killed - 2014 per 1L people'],wi
plt.bar(UT+0.2,df6['Trucks, Tempos,MAVs,Tractors - Number of Persons - Kille
plt.bar(UT+0.4,df6['Other Motor Vehicles - Number of Persons - Killed - 2014 pe
plt.bar(UT+0.6,df6['Other Vehicles/Objects - Number of Persons - Killed - 2014 pe

plt.xticks(UT,df6['States/UTs'],rotation='vertical')
plt.legend(loc='upper left', bbox_to_anchor=(0.25,1))
plt.title("Number of Persons Killed for each vehicle type per 1L people of t
plt.show()

```

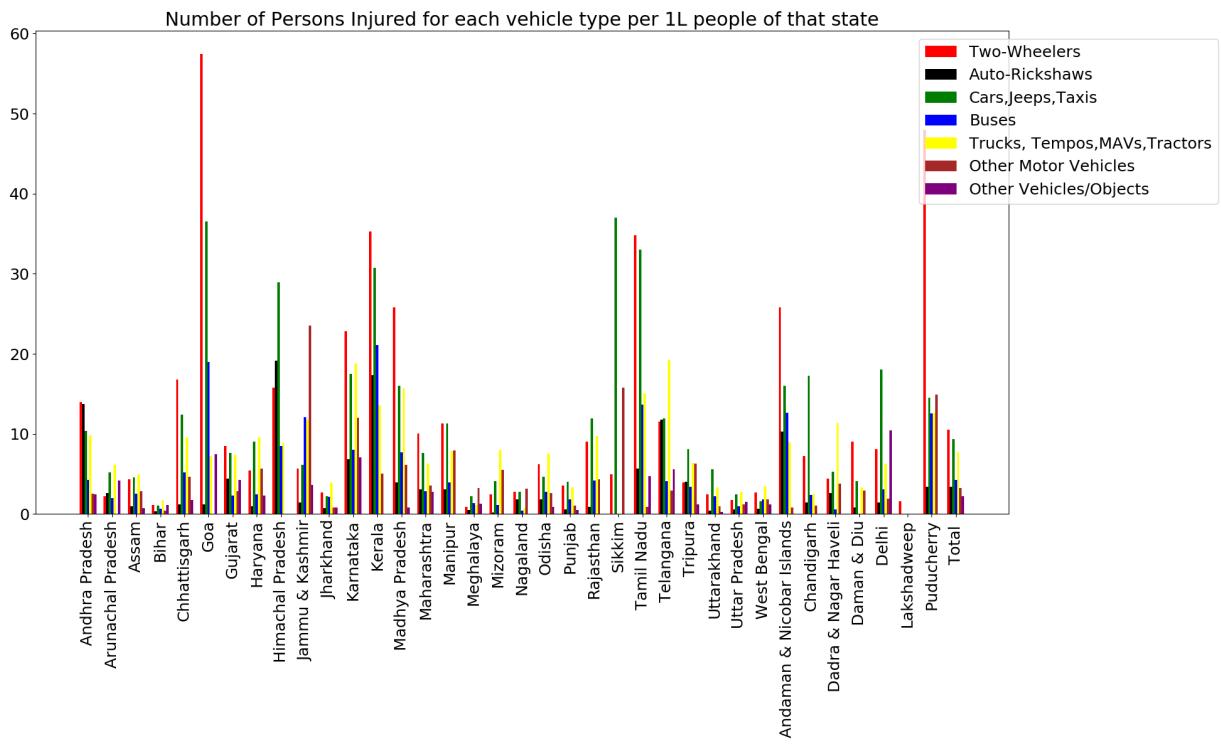
Number of Persons Killed for each vehicle type per 1L people of that state



```
In [164]: plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

plt.bar(UT-0.6,df6['Two-Wheelers - Number of Persons - Injured - 2014 per 1L people'])
plt.bar(UT-0.4,df6['Auto-Rickshaws - Number of Persons - Injured - 2014 per 1L people'])
plt.bar(UT-0.2,df6['Cars, Jeeps,Taxis - Number of Persons - Injured - 2014 per 1L people'])
plt.bar(UT,df6['Buses - Number of Persons - Injured - 2014 per 1L people'],width=0.2)
plt.bar(UT+0.2,df6['Trucks, Tempos,MAVs,Tractors - Number of Persons - Injured - 2014 per 1L people'])
plt.bar(UT+0.4,df6['Other Motor Vehicles - Number of Persons - Injured - 2014 per 1L people'])
plt.bar(UT+0.6,df6['Other Vehicles/Objects - Number of Persons - Injured - 2014 per 1L people'])

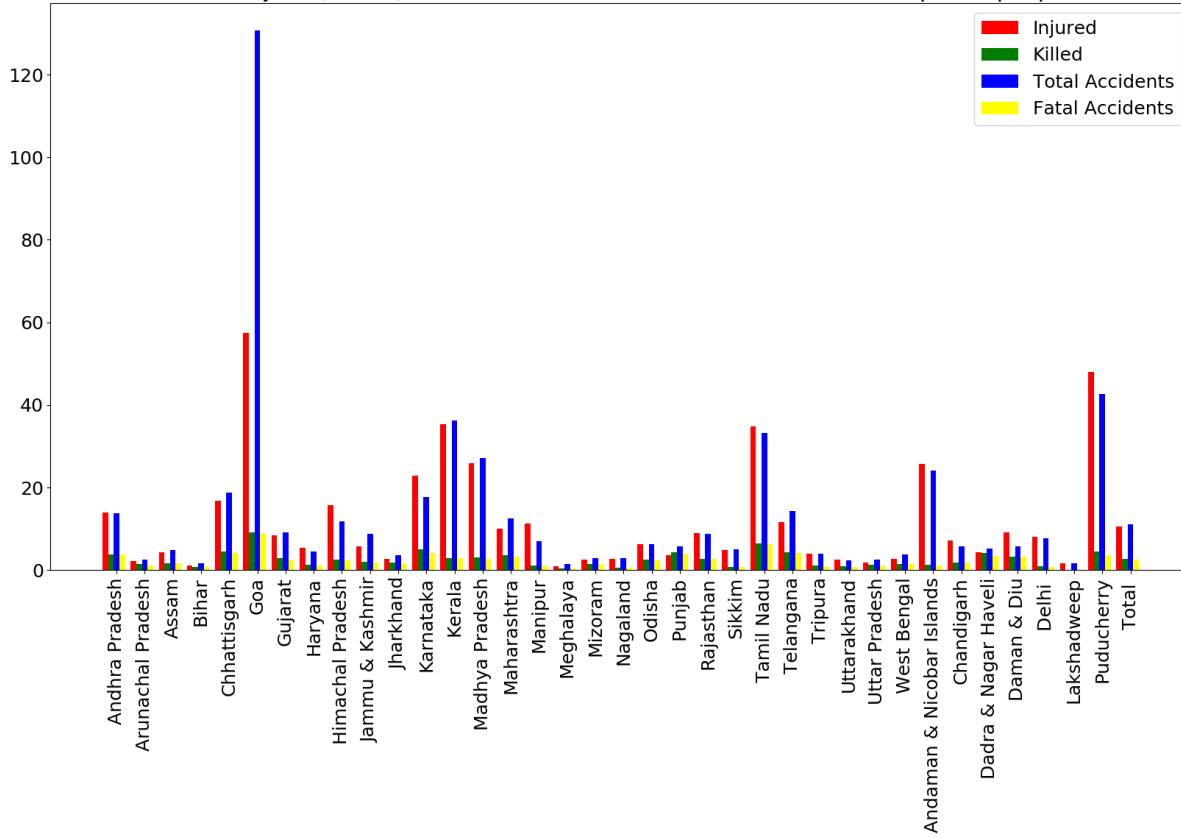
plt.xticks(UT,df6['States/UTs'],rotation='vertical')
plt.legend(loc='upper left', bbox_to_anchor=(0.9,1))
plt.title("Number of Persons Injured for each vehicle type per 1L people of that state")
plt.show()
```



```
In [165]: plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})
UT=np.array([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,
            25,26,27,28,29,30,31])
plt.bar(UT-0.4,df6['Two-Wheelers - Number of Persons - Injured - 2014 per 1L'])
plt.bar(UT-0.2,df6['Two-Wheelers - Number of Persons - Killed - 2014 per 1L'])
plt.bar(UT,df6['Two-Wheelers - Number of Road Accidents - Total - 2014 per 1L'])
plt.bar(UT+0.2,df6['Two-Wheelers - Number of Road Accidents - Fatal - 2014 per 1L'])

plt.xticks(UT,df6['States/UTs'],rotation='vertical')
plt.legend(loc="best")
plt.title("Number of Persons Injured, Killed; Road Accidents Total and fatal")
plt.show()
```

Number of Persons Injured, Killed; Road Accidents Total and fatal for 2 wheelers per 1L people of that state



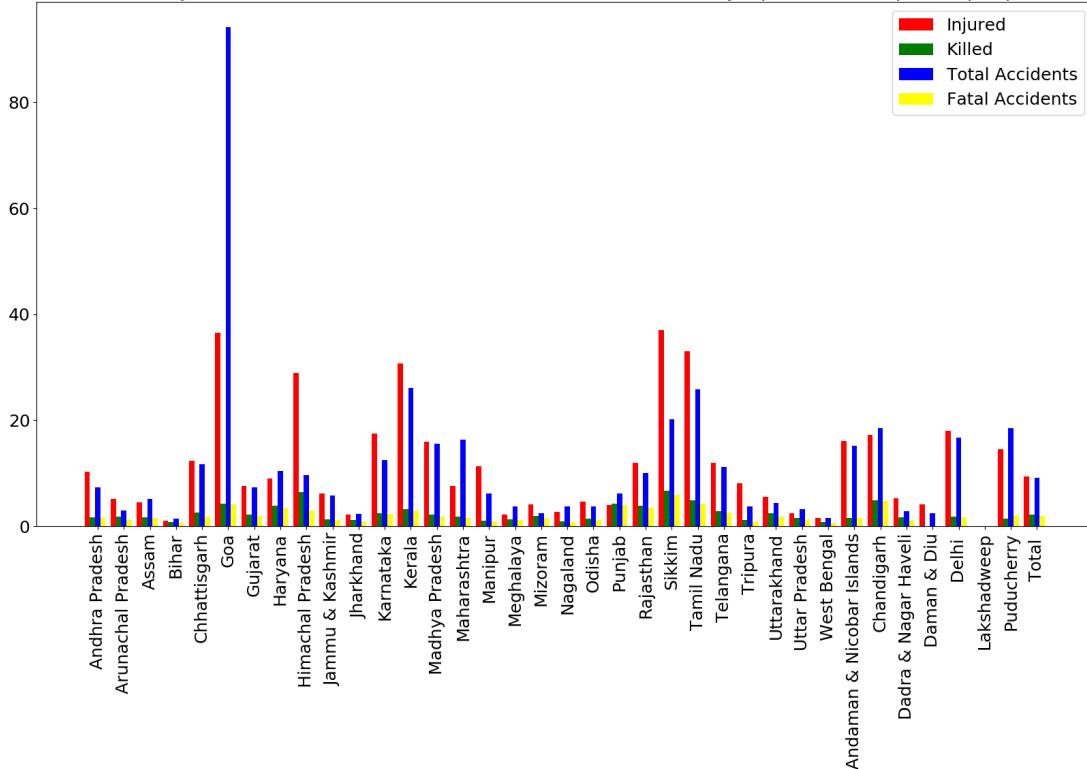
In [166]:

```
plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

plt.bar(UT-0.4,df6['Cars, Jeeps,Taxis - Number of Persons - Injured - 2014 per 1L people'])
plt.bar(UT-0.2,df6['Cars, Jeeps,Taxis - Number of Persons - Killed - 2014 per 1L people'])
plt.bar(UT,df6['Cars, Jeeps,Taxis - Number of Road Accidents - Total - 2014 per 1L people'])
plt.bar(UT+0.2,df6['Cars, Jeeps,Taxis - Number of Road Accidents - Fatal - 2014 per 1L people'])

plt.xticks(UT,df6['States/UTs'],rotation='vertical')
plt.legend(loc="best")
plt.title("Number of Persons Injured, Killed; Road Accidents Total and fatal per 1L people of that state")
plt.show()
```

Number of Persons Injured, Killed; Road Accidents Total and fatal for Cars, Jeeps and Taxis per 1L people of that state

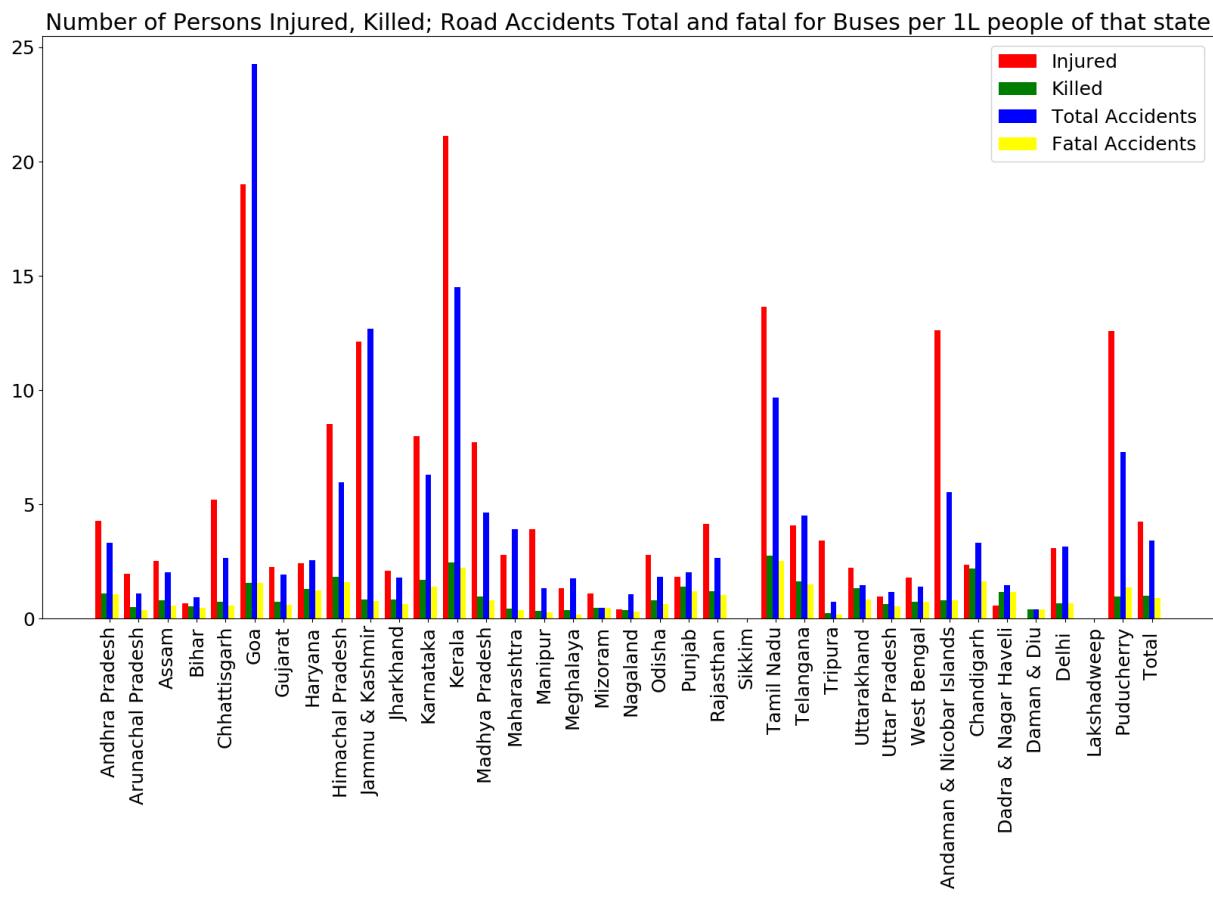


In [167]:

```
plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

plt.bar(UT-0.4,df6['Buses - Number of Persons - Injured - 2014 per 1L people']
plt.bar(UT-0.2,df6['Buses - Number of Persons - Killed - 2014 per 1L people'])
plt.bar(UT,df6['Buses - Number of Road Accidents - Total - 2014 per 1L people'])
plt.bar(UT+0.2,df6['Buses - Number of Road Accidents - Fatal - 2014 per 1L people'])

plt.xticks(UT,df6['States/UTs'],rotation='vertical')
plt.legend(loc="best")
plt.title("Number of Persons Injured, Killed; Road Accidents Total and fatal")
plt.show()
```

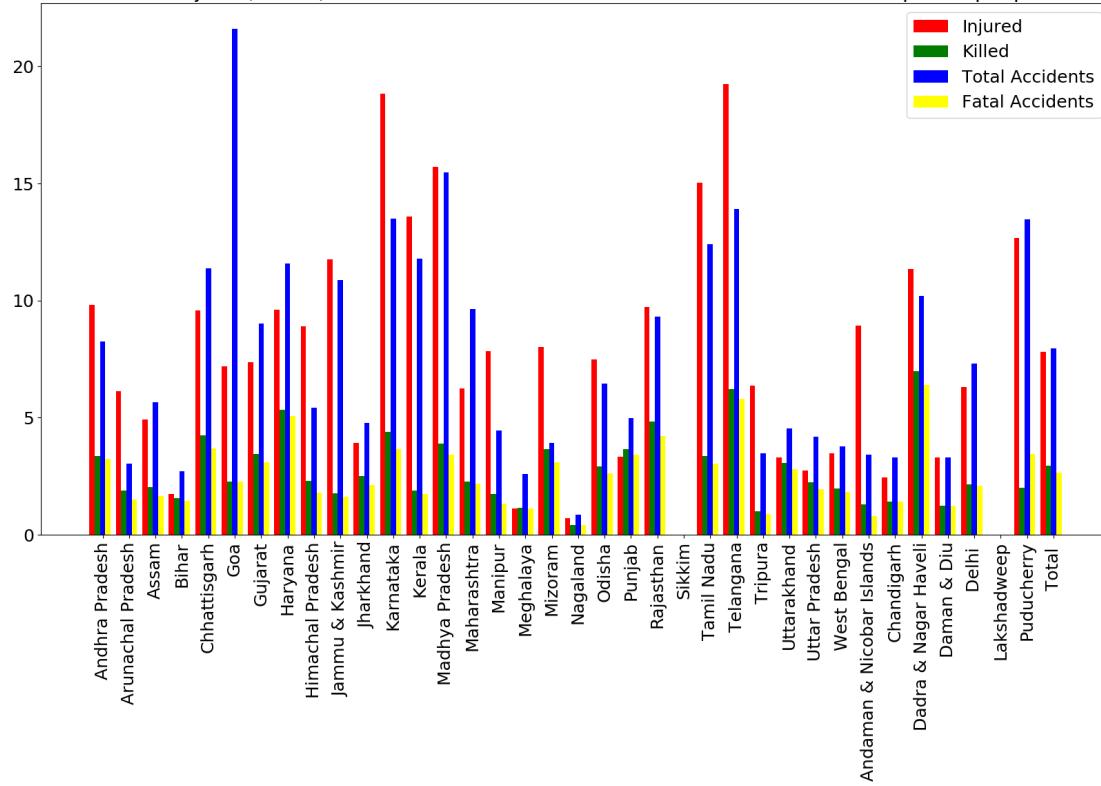


```
In [168]: plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

plt.bar(UT-0.4,df6['Trucks, Tempos, MAVs, Tractors - Number of Persons - Injured'], color='red')
plt.bar(UT-0.2,df6['Trucks, Tempos, MAVs, Tractors - Number of Persons - Killed'], color='green')
plt.bar(UT,df6['Trucks, Tempos, MAVs, Tractors - Number of Road Accidents - Total'], color='blue')
plt.bar(UT+0.2,df6['Trucks, Tempos, MAVs, Tractors - Number of Road Accidents - Fatal'], color='yellow')

plt.xticks(UT,df6['States/UTs'],rotation='vertical')
plt.legend(loc="best")
plt.title("Number of Persons Injured, Killed; Road Accidents Total and fatal per 1L people of that state")
plt.show()
```

Number of Persons Injured, Killed; Road Accidents Total and fatal for Trucks and Tractors per 1L people of that state

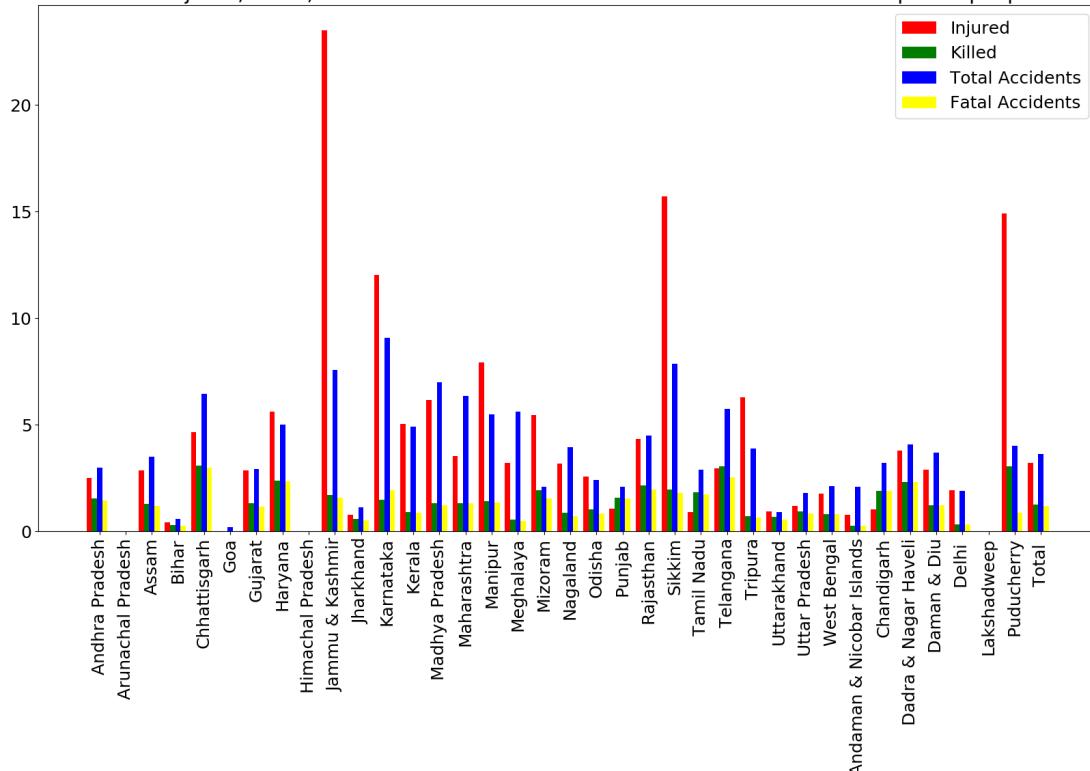


```
In [169]: plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

plt.bar(UT-0.4,df6['Other Motor Vehicles - Number of Persons - Injured - 2014'])
plt.bar(UT-0.2,df6['Other Motor Vehicles - Number of Persons - Killed - 2014'])
plt.bar(UT,df6['Other Motor Vehicles - Number of Road Accidents - Total - 2014'])
plt.bar(UT+0.2,df6['Other Motor Vehicles - Number of Road Accidents - Fatal'])

plt.xticks(UT,df6['States/UTs'],rotation='vertical')
plt.legend(loc="best")
plt.title("Number of Persons Injured, Killed; Road Accidents Total and fatal")
plt.show()
```

Number of Persons Injured, Killed; Road Accidents Total and fatal for Other Motor Vehicles per 1L people of that state



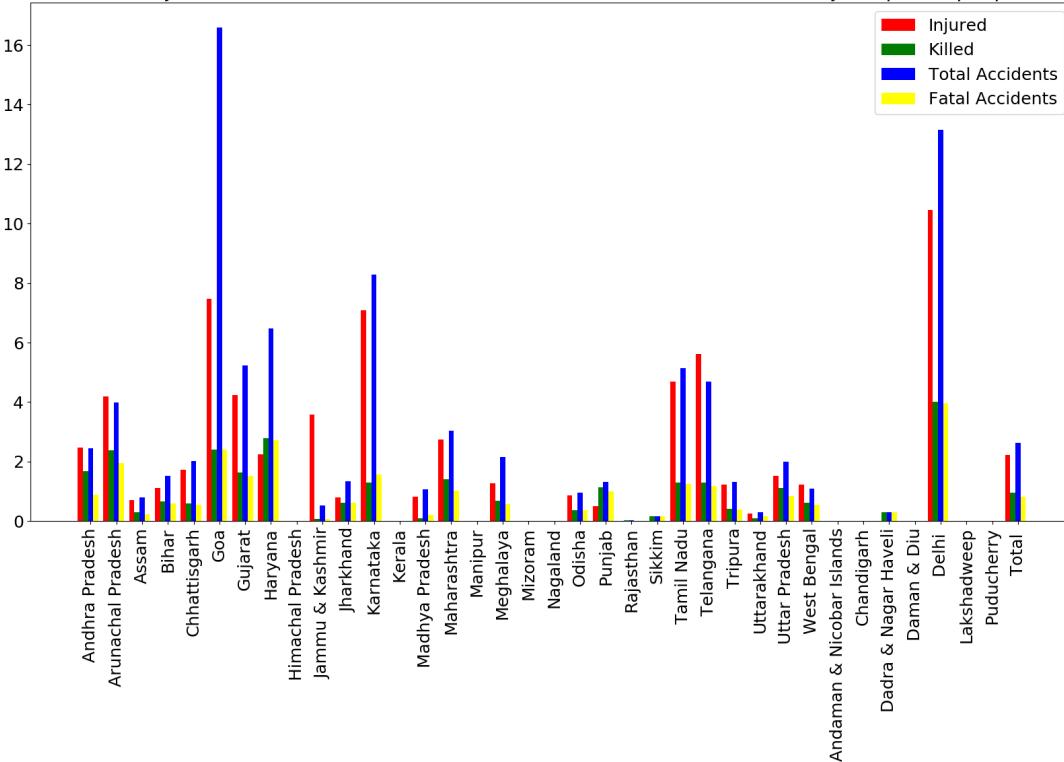
```
In [171]: plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

plt.bar(UT-0.4,df6['Other Vehicles/Objects - Number of Persons - Injured - 2018'])
plt.bar(UT-0.2,df6['Other Vehicles/Objects - Number of Persons - Killed - 2018'])
plt.bar(UT,df6['Other Vehicles/Objects - Number of Road Accidents - Total - 2018'])
plt.bar(UT+0.2,df6['Other Vehicles/Objects - Number of Road Accidents - Fatal - 2018'])

plt.xticks(UT,df6['States/UTs'],rotation='vertical')
plt.legend(loc="best")
plt.title("Number of Persons Injured, Killed; Road Accidents Total and fatal for Other Motor Vehicles per 1L people of that state")
plt.show()
```

<Figure size 2000x1000 with 0 Axes>

Number of Persons Injured, Killed; Road Accidents Total and fatal for Other Vehicles/Objects per 1L people of that state



```
In [173]: df6['Sum Total Road Accidents - 2014 per 1L people']=df6['Two-Wheelers - Number of Road Accidents - Total - 2014 per 1L people']+df6['Buses - Number of Road Accidents - Total - 2014 per 1L people']+df6['Other Vehicles/Objects - Number of Road Accidents - Total - 2014 per 1L people']

df6['Sum Total Number of Persons Killed - 2014 per 1L people']=df6['Two-Wheelers - Number of Persons - Killed - 2014 per 1L people']+df6['Buses - Number of Persons - Killed - 2014 per 1L people']+df6['Other Vehicles/Objects - Number of Persons - Killed - 2014 per 1L people']

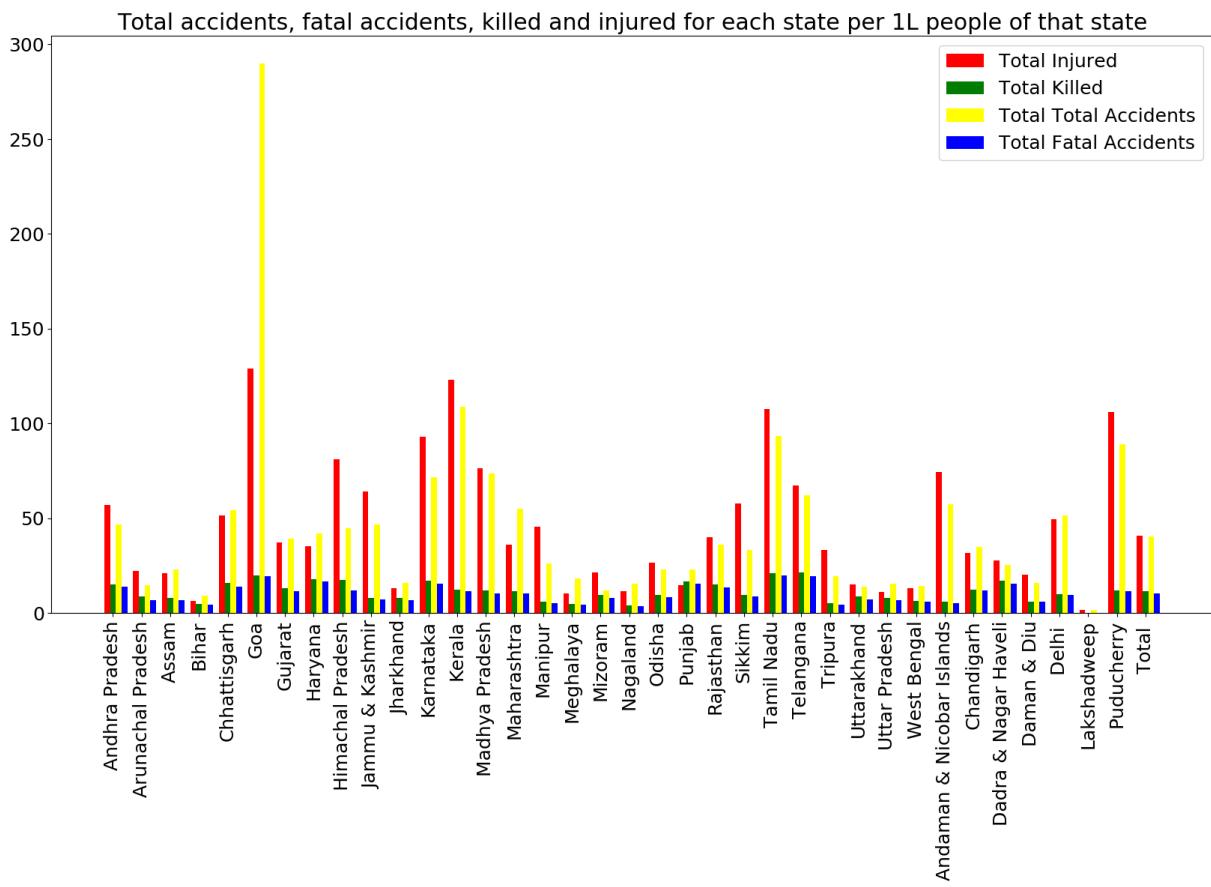
df6['Sum Total Number of Persons Injured - 2014 per 1L people']=df6['Two-Wheelers - Number of Persons - Injured - 2014 per 1L people']+df6['Buses - Number of Persons - Injured - 2014 per 1L people']+df6['Other Vehicles/Objects - Number of Persons - Injured - 2014 per 1L people']

df6['Sum Total Fatal Road Accidents - 2014 per 1L people']=df6['Two-Wheelers - Number of Road Accidents - Fatal - 2014 per 1L people']+df6['Buses - Number of Road Accidents - Fatal - 2014 per 1L people']+df6['Other Vehicles/Objects - Number of Road Accidents - Fatal - 2014 per 1L people']
```

```
In [174]: plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

plt.bar(UT-0.2,df6['Sum Total Number of Persons Injured - 2014 per 1L people'],width=0.4)
plt.bar(UT,df6['Sum Total Number of Persons Killed - 2014 per 1L people'],width=0.4)
plt.bar(UT+0.2,df6['Sum Total Road Accidents - 2014 per 1L people'],width=0.4)
plt.bar(UT+0.4,df6['Sum Total Fatal Road Accidents - 2014 per 1L people'],width=0.4)

plt.xticks(UT,df6['States/UTs'],rotation='vertical')
plt.legend(loc="best")
plt.title("Total accidents, fatal accidents, killed and injured for each state")
plt.show()
```

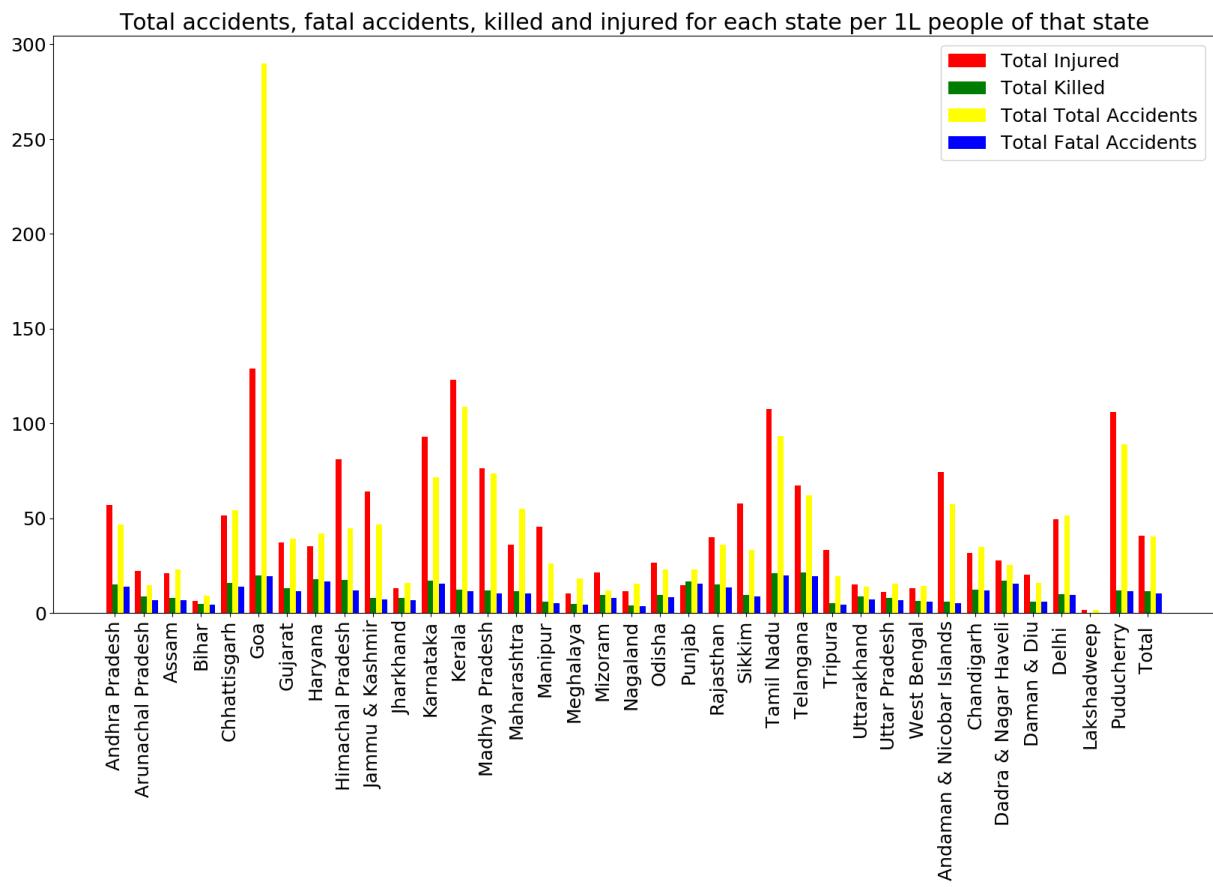


```
In [176]: plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size':18})

plt.bar(UT-0.2,df6['Sum Total Number of Persons Injured - 2014 per 1L people'],
        plt.bar(UT,df6['Sum Total Number of Persons Killed - 2014 per 1L people'],width=0.2),
        plt.bar(UT+0.2,df6['Sum Total Road Accidents - 2014 per 1L people'],width=0.2),
        plt.bar(UT+0.4,df6['Sum Total Fatal Road Accidents - 2014 per 1L people'],width=0.2)

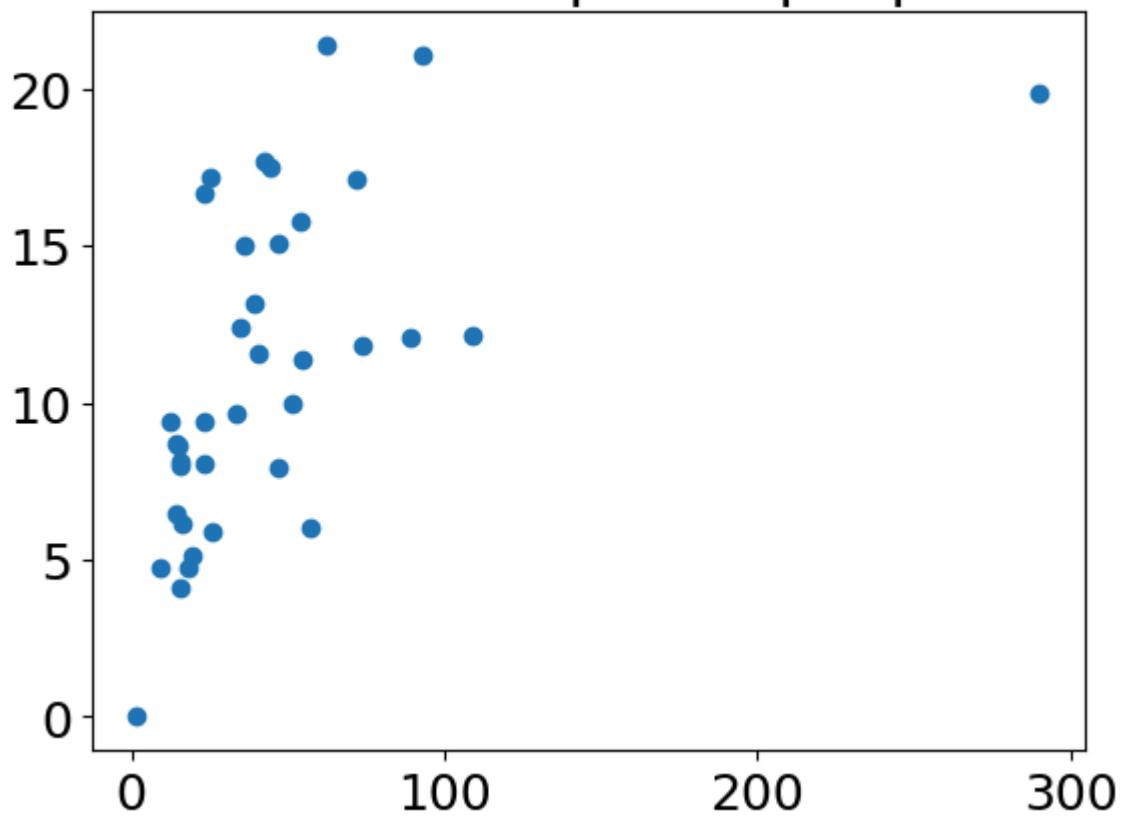
plt.xticks(UT,df6['States/UTs'],rotation='vertical')
plt.legend(loc="best")
plt.title("Total accidents, fatal accidents, killed and injured for each state/UT")
plt.show()
```

<Figure size 2000x1000 with 0 Axes>



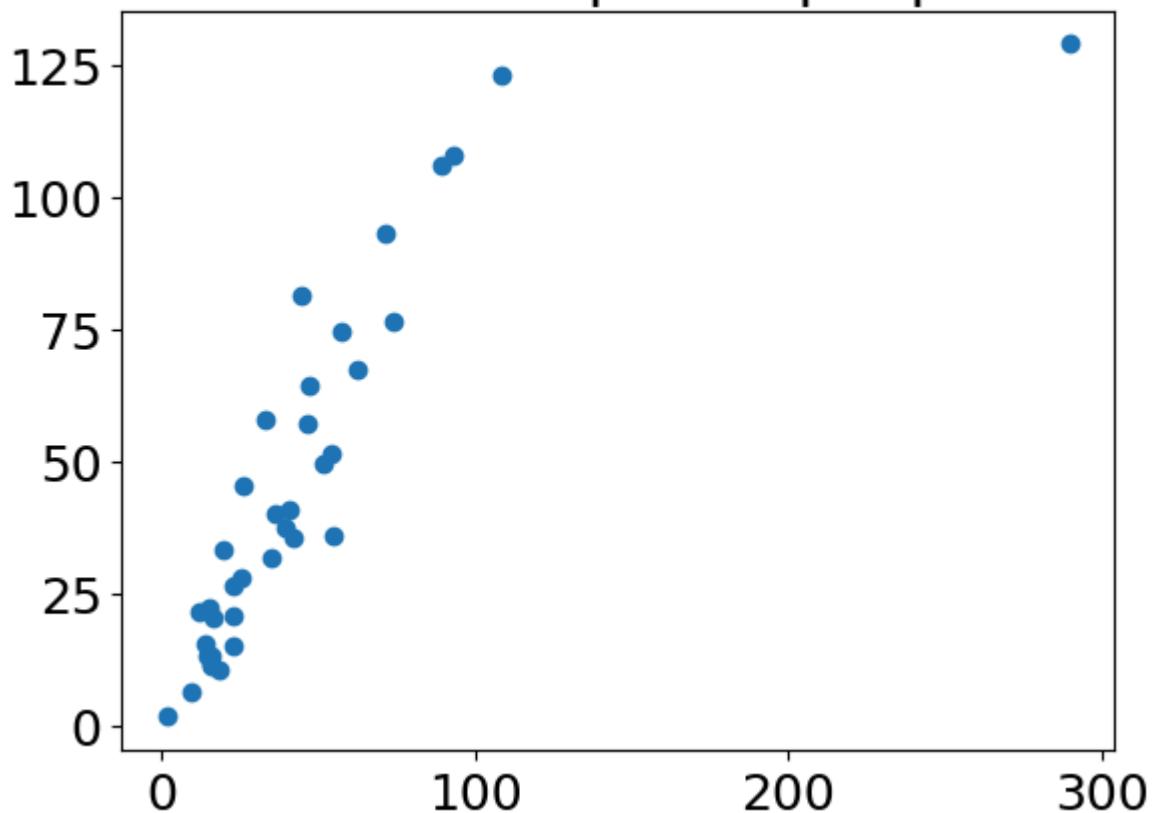
```
In [177]: plt.scatter(df6['Sum Total Road Accidents - 2014 per 1L people'], df6['Sum Tc'])
plt.title("Total A vs K per 1L people")
plt.show()
```

## Total A vs K per 1L people



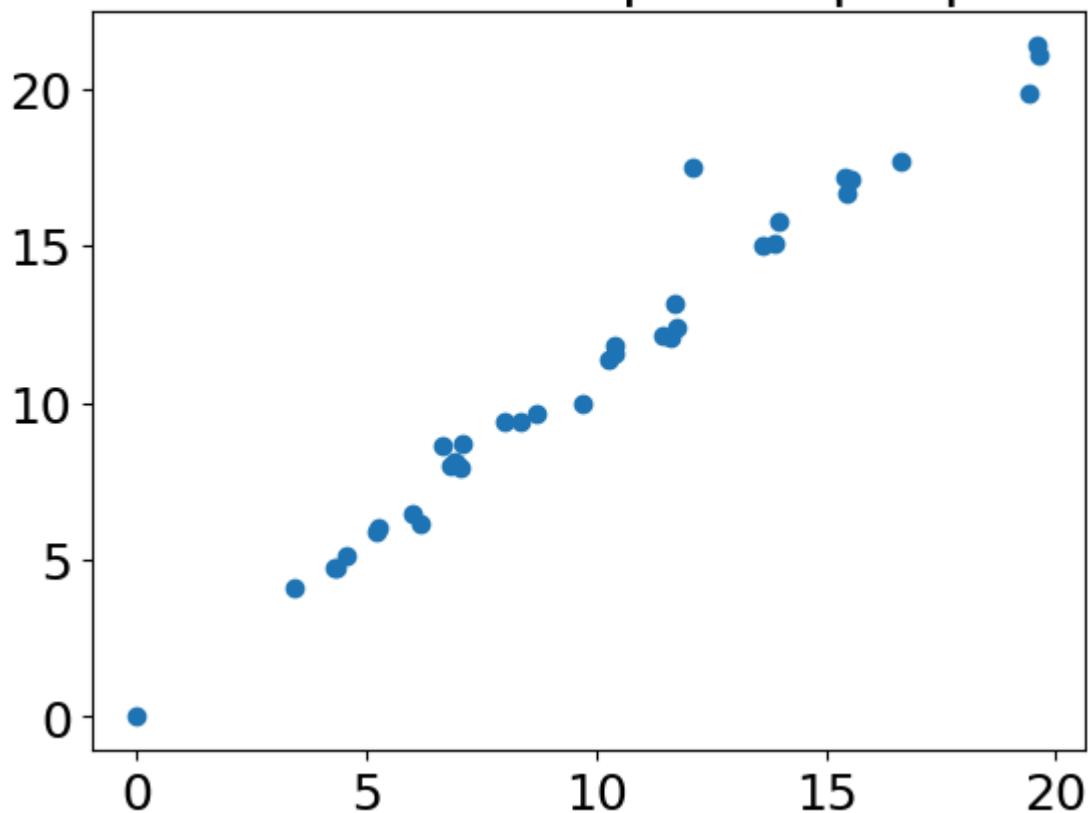
```
In [178]: plt.scatter(df6['Sum Total Road Accidents - 2014 per 1L people'], df6['Sum Total Fatalities - 2014 per 1L people'])
plt.title("Total A vs I per 1L people")
plt.show()
```

## Total A vs I per 1L people



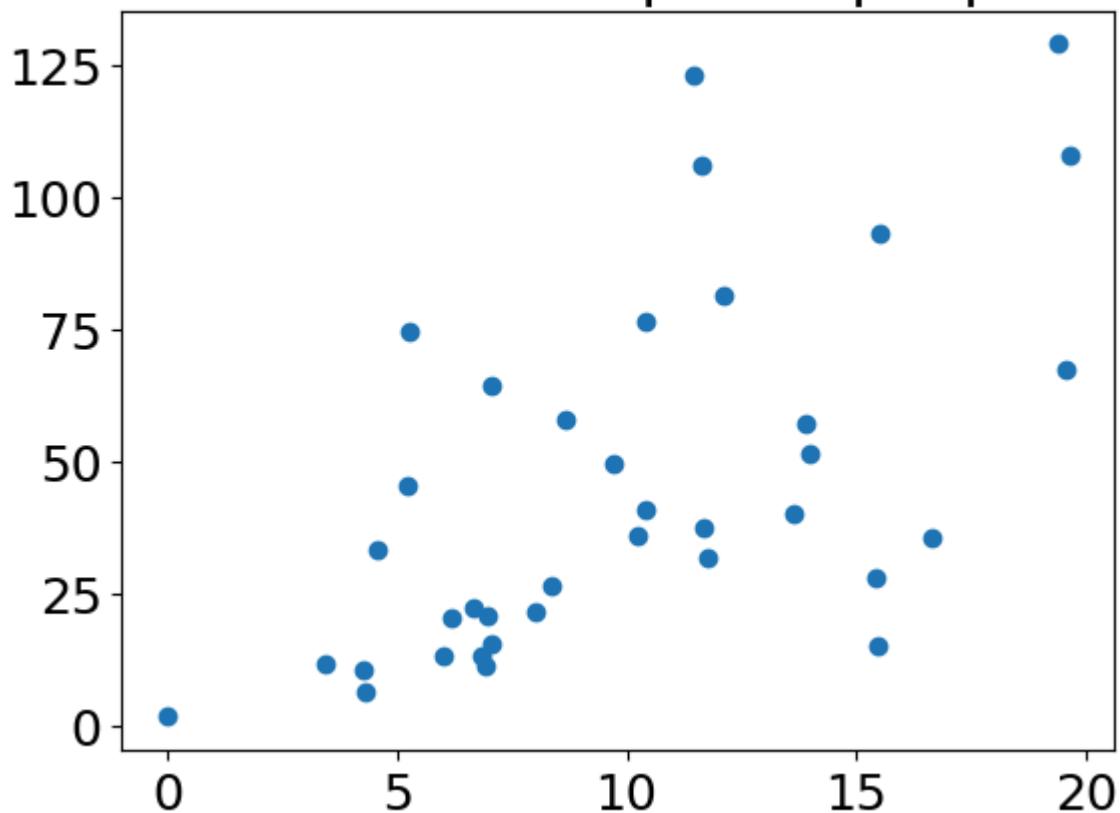
```
In [179]: plt.scatter(df6['Sum Total Fatal Road Accidents - 2014 per 1L people'], df6['Injury Fatal Road Accidents - 2014 per 1L people'])
plt.title("Total Fatal vs K per 1L people")
plt.show()
```

## Total Fatal vs K per 1L people



```
In [180]: plt.scatter(df6['Sum Total Fatal Road Accidents - 2014 per 1L people'], df6['K per 1L people'])
plt.title("Total Fatal vs K per 1L people")
plt.show()
```

## Total Fatal vs I per 1L people



---

[Go to top.](#)

### DF7 -

This dataset tells how many accidents happen according to the **time of occurrence** in different states.

In [186]:

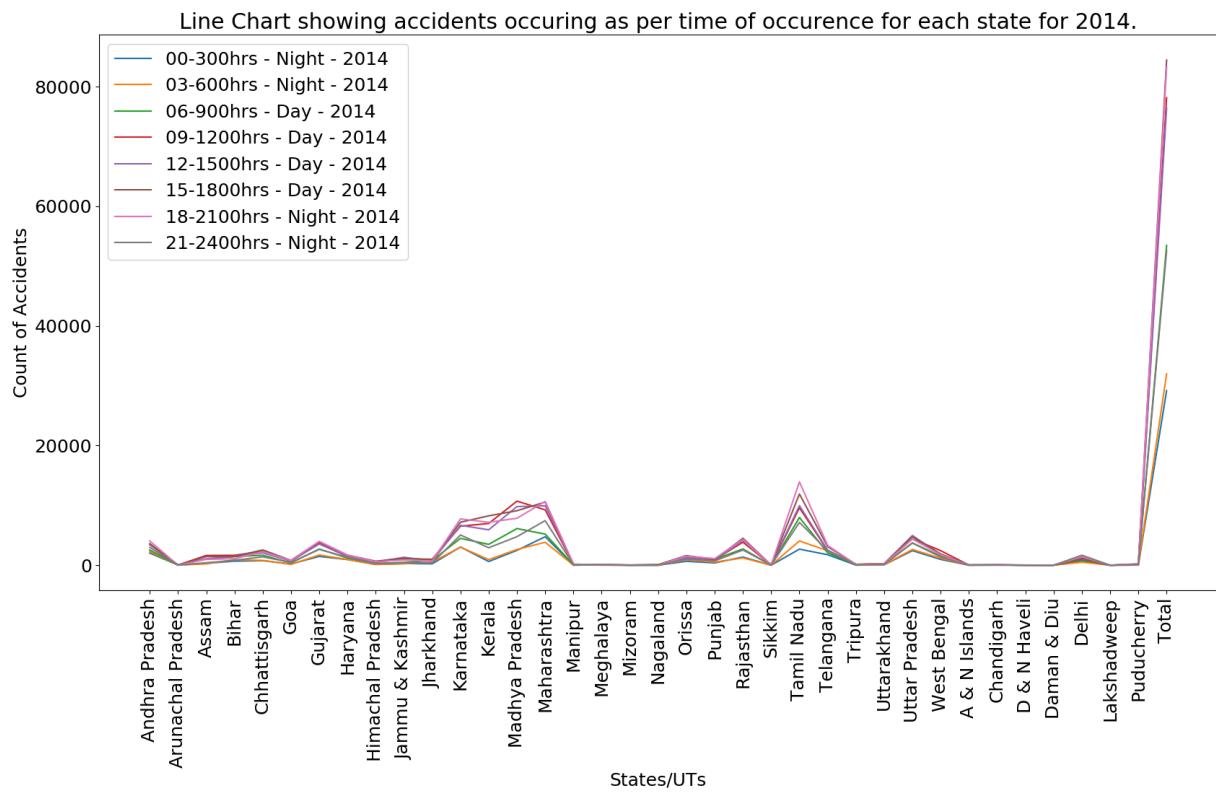
```
df7 = pd.read_excel('A:\Data Analysis Jupyter\Road-Accidents-Of-India\Database.xlsx')
df7.info()
```

```
WARNING *** OLE2 inconsistency: SSCS size is 0 but SSAT size is non-zero
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37 entries, 0 to 36
Data columns (total 20 columns):
S. No.                      37 non-null object
States/Uts                   37 non-null object
06-900hrs - Day - 2014      37 non-null int64
09-1200hrs - Day - 2014     37 non-null int64
12-1500hrs - Day - 2014     37 non-null int64
15-1800hrs - Day - 2014     37 non-null int64
18-2100hrs - Night - 2014   37 non-null int64
21-2400hrs - Night - 2014   37 non-null int64
00-300hrs - Night - 2014    37 non-null int64
03-600hrs - Night - 2014    37 non-null int64
Total Accidents - 2014       37 non-null int64
06-900hrs - (Day) - 2016    37 non-null int64
09-1200hrs - (Day) - 2016    37 non-null int64
12-1500hrs - (Day) - 2016    37 non-null int64
15-1800hrs - (Day) - 2016    37 non-null int64
18-2100hrs - (Night) - 2016  37 non-null int64
21-2400hrs - (Night) - 2016  37 non-null int64
00-300hrs - (Night) - 2016   37 non-null int64
03-600hrs - (Night) - 2016   37 non-null int64
Total Accidents - 2016       37 non-null int64
dtypes: int64(18), object(2)
memory usage: 5.9+ KB
```

```
In [324]: plt.rcParams.update({'font.size' : 18})
plt.figure(figsize = (20,10))

plt.plot(df7['States/Uts'], df7['00-300hrs - Night - 2014'])
plt.plot(df7['03-600hrs - Night - 2014'])
plt.plot(df7['06-900hrs - Day - 2014'])
plt.plot(df7['09-1200hrs - Day - 2014'])
plt.plot(df7['12-1500hrs - Day - 2014'])
plt.plot(df7['15-1800hrs - Day - 2014'])
plt.plot(df7['18-2100hrs - Night - 2014'])
plt.plot(df7['21-2400hrs - Night - 2014'])

plt.title("Line Chart showing accidents occurring as per time of occurrence for 2014")
plt.xlabel("States/UTs")
plt.ylabel("Count of Accidents")
plt.xticks(rotation = 90)
plt.legend(loc = 'best')
plt.show()
```

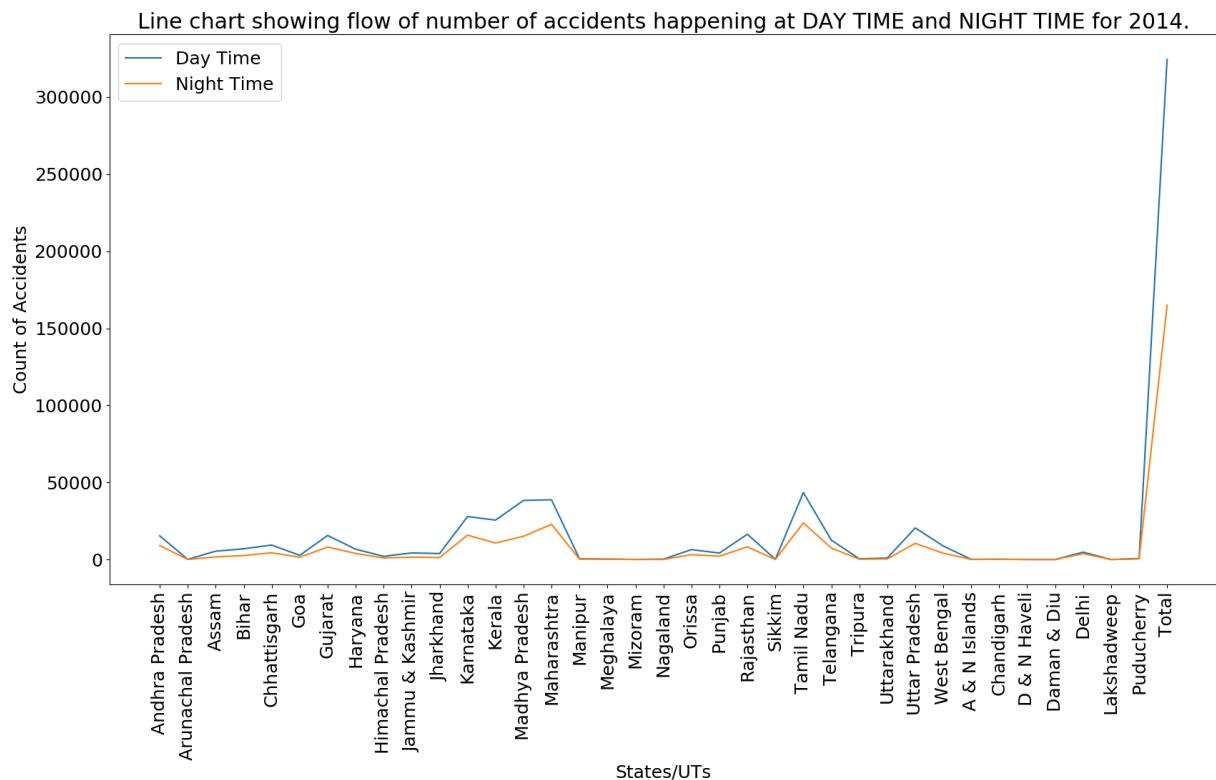


```
In [267]: dayTime2014 = df7['03-600hrs - Night - 2014']+df7['06-900hrs - Day - 2014']+df7['09-1200hrs - Day - 2014']+df7['12-1500hrs - Day - 2014']+nightTime2014 = df7['00-300hrs - Night - 2014'] + df7['18-2100hrs - Night - 2014']
```

```
In [323]: plt.rcParams.update({'font.size' : 18})
plt.figure(figsize = (20,10))

plt.plot(df7['States/Uts'], dayTime2014, label = 'Day Time')
plt.plot(nightTime2014, label = 'Night Time')

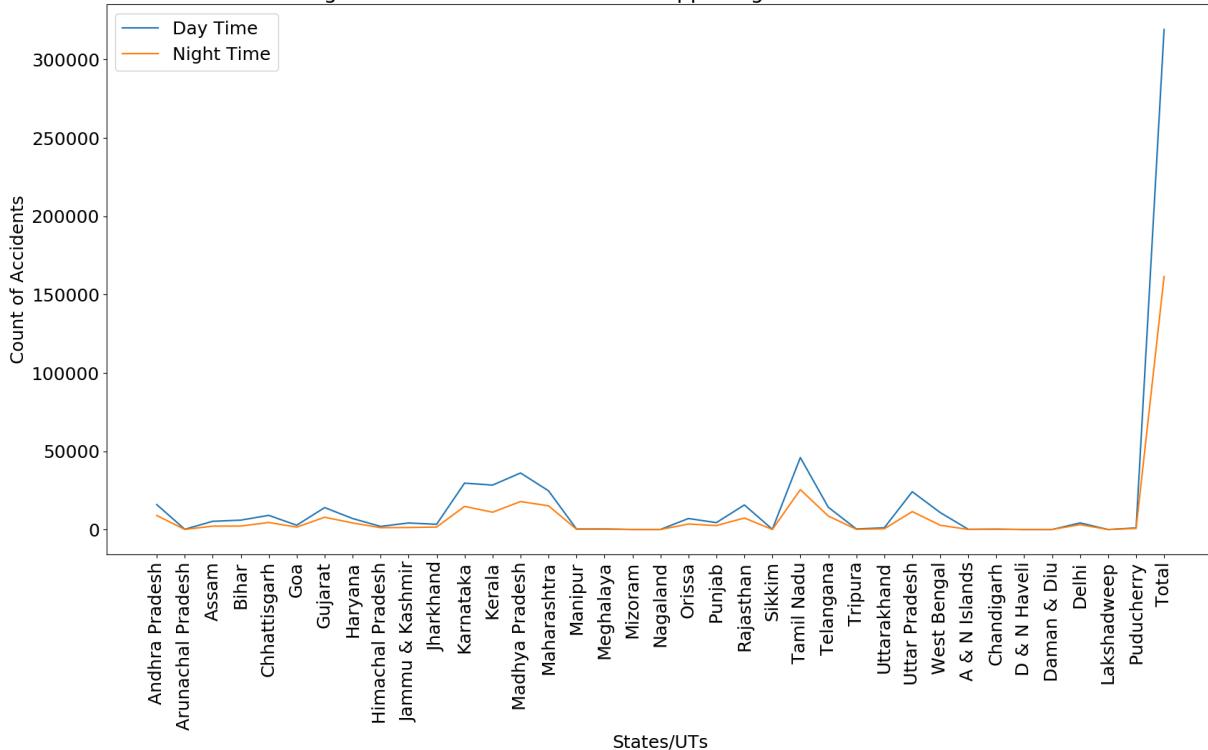
plt.title("Line chart showing flow of number of accidents happening at DAY TIME")
plt.xlabel("States/UTs")
plt.ylabel("Count of Accidents")
plt.xticks(rotation = 90)
plt.legend( loc = 'best')
plt.show()
```



```
In [283]:  
dayTime2016 = df7['03-600hrs - (Night) - 2016']+df7['06-900hrs - (Day) - 2016']+df7['09-1200hrs - (Day) - 2016']+df7['12-1500hrs - (Day) - 2016']  
nightTime2016 = df7['00-300hrs - (Night) - 2016'] + df7['18-2100hrs - (Night) - 2016']
```

```
In [321]:  
plt.rcParams.update({'font.size' : 18})  
plt.figure(figsize = (20,10))  
  
plt.plot(df7['States/Uts'], dayTime2016, label = 'Day Time')  
plt.plot(nightTime2016, label = 'Night Time')  
  
plt.title("Line chart showing flow of number of accidents happening at DAY TIME and NIGHT TIME for 2014")  
plt.xlabel("States/UTs")  
plt.ylabel("Count of Accidents")  
plt.xticks(rotation = 90)  
plt.legend(loc = 'best')  
plt.show()
```

Line chart showing flow of number of accidents happening at DAY TIME and NIGHT TIME for 2016.

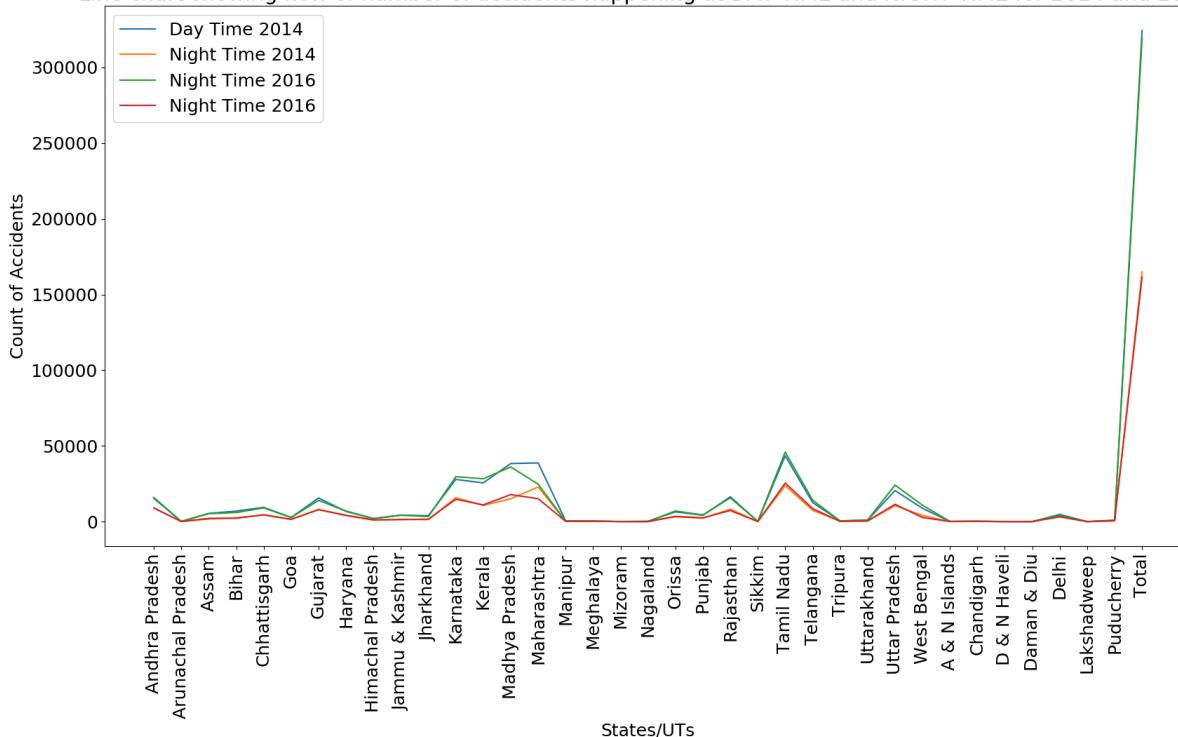


```
In [322]: plt.rcParams.update({'font.size' : 18})
plt.figure(figsize = (20,10))

plt.plot(df7['States/UTs'], dayTime2014, label = 'Day Time 2014')
plt.plot(nightTime2014, label = 'Night Time 2014')
plt.plot(dayTime2016, label = 'Night Time 2016')
plt.plot(nightTime2016, label = 'Night Time 2016')

plt.title("Line chart showing flow of number of accidents happening at DAY T")
plt.xlabel("States/UTs")
plt.ylabel("Count of Accidents")
plt.xticks(rotation = 90)
plt.legend(loc = 'best')
plt.show()
```

Line chart showing flow of number of accidents happening at DAY TIME and NIGHT TIME for 2014 and 2016.

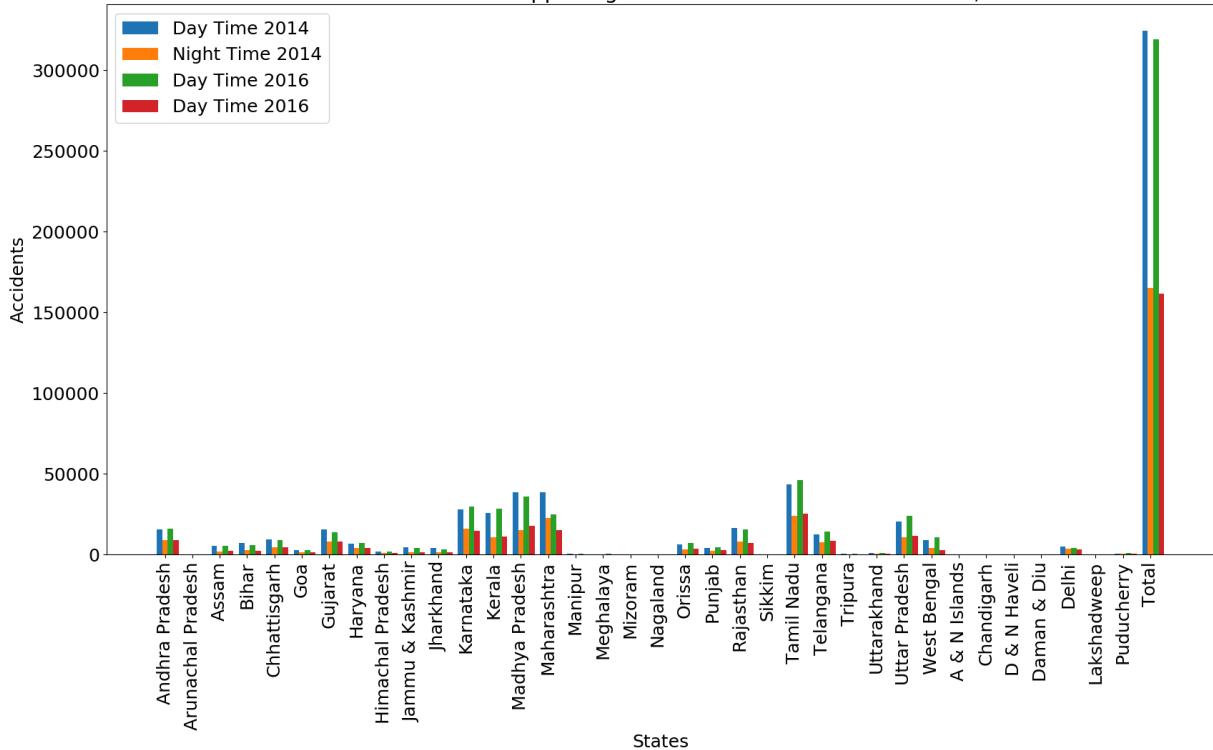


```
In [319]: plt.rcParams.update({'font.size' : 18})
plt.figure(figsize = (20,10))
UT = np.array([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24])

plt.bar(UT-0.2, dayTime2014, width = 0.2,label = 'Day Time 2014')
plt.bar(UT, nightTime2014, width = 0.2, label = 'Night Time 2014')
plt.bar(UT+0.2, dayTime2016, width = 0.2, label = 'Day Time 2016')
plt.bar(UT+0.4, nightTime2016, width = 0.2, label = 'Night Time 2016')

plt.xlabel("States")
plt.ylabel("Accidents")
plt.legend(loc = 'best')
plt.title("Number of Accidents happening in DAY and NIGHT TIME for 2014, 2016")
plt.xticks(UT, df7['States/Uts'], rotation=90)
plt.show()
```

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



In [ ]:

In [ ]:

In [ ]:

In [ ]: