EDA of India Census 2011 Dateset

```
Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns,to spot anomalies,to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.
```

Importing Libraries

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

MySQL connection with Python

In [2]:

```
import mysql.connector as mysql_py_connector

connection=mysql_py_connector.connect(
 host="localhost",
 user="root",
 password="Sql@246324"
)
print(connection)
```

```
<mysql.connector.connection_cext.CMySQLConnection object at 0x000001FFEA7DAD
F0>
```

Checking conection with MySQL

In [3]:

```
cursor=connection.cursor()
```

In [4]:

```
cursor.execute("select sum(population) from Indian_Census_2011.census_data_1 ")
```

In [5]:

```
for i in cursor:
    print(i)
```

```
(Decimal('1210854977'),)
```

Using Pandas to read tables from MySQL Database

In [6]:

```
full_data_query="select * from Indian_Census_2011.census_data_1 inner join Indian_Censu
df=pd.read_sql(full_data_query,connection)
df.head()
```

```
C:\Users\LENOVO\anaconda3\lib\site-packages\pandas\io\sql.py:761: UserWarnin
g: pandas only support SQLAlchemy connectable(engine/connection) ordatabase
string URI or sqlite3 DBAPI2 connectionother DBAPI2 objects are not tested,
please consider using SQLAlchemy
  warnings.warn(
```

Out[6]:

	District_code	State_name	District_name	Population	Male	Female	Literate	M
0	1	JAMMU_AND_KASHMIR	Kupwara	870354	474190	396164	439654	
1	2	JAMMU_AND_KASHMIR	Badgam	753745	398041	355704	335649	
2	3	JAMMU_AND_KASHMIR	Leh(Ladakh)	133487	78971	54516	93770	
3	4	JAMMU_AND_KASHMIR	Kargil	140802	77785	63017	86236	
4	5	JAMMU_AND_KASHMIR	Punch	476835	251899	224936	261724	

5 rows × 61 columns

In [7]:

```
df.shape
```

Out[7]:

```
(640, 61)
```

Conclusion:Datset has 640 rows and 61 Columns

In [8]:

```
1 df.columns
```

Out[8]:

```
Index(['District_code', 'State_name', 'District_name', 'Population', 'Male',
       'Female', 'Literate', 'Male_Literate', 'Female_Literate', 'SC',
       'Male_SC', 'Female_SC', 'ST', 'Male_ST', 'Female_ST', 'Workers',
       'Male_Workers', 'Female_Workers', 'Main_Workers', 'Marginal_Workers',
       'Non_Workers', 'Cultivator_Workers', 'Agricultural_Workers',
       'Household_Workers', 'Other_Workers', 'Hindus', 'Muslims', 'Christian
s',
       'Sikhs', 'Buddhists', 'Jains', 'Others_Religions',
       'Religion_Not_Stated', 'LPG_or_PNG_Households',
       'Housholds_with_Electric_Lighting', 'Households_with_Internet',
       'Households_with_Computer', 'Rural_Households', 'Urban_Households',
       'Households', 'Below_Primary_Education', 'Primary_Education',
       'Middle_Education', 'Secondary_Education', 'Higher_Education',
       'Graduate_Education', 'Other_Education', 'Literate_Education',
       'Illiterate_Education', 'Total_Education', 'Age_Group_0_29',
       'Age_Group_30_49', 'Age_Group_50', 'Age_not_stated', 'District_code',
       'State_name', 'District_name',
       'Type_of_bathing_facility_Enclosure_without_roof',
       'Not_having_bathing_facility_within_the_premises',
       'Not_having_latrine_facility_within_the_premises',
       'Main_source_of_drinking_water_Un_covered_well'],
      dtype='object')
```

In [9]:

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 640 entries, 0 to 639
Data columns (total 61 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   District_code                           640 non-null    int64
 1   State_name                              640 non-null    object
 2   District_name                           640 non-null    object
 3   Population                              640 non-null    int64
 4   Male                                    640 non-null    int64
 5   Female                                  640 non-null    int64
 6   Literate                                640 non-null    int64
 7   Male_Literate                           640 non-null    int64
 8   Female_Literate                         640 non-null    int64
 9   SC                                      640 non-null    int64
 10  Male_SC                                 640 non-null    int64
 11  Female_SC                               640 non-null    int64
 12  ST                                      640 non-null    int64
 13  Male_ST                                 640 non-null    int64
 14  Female_ST                               640 non-null    int64
 15  Workers                                 640 non-null    int64
 16  Male_Workers                            640 non-null    int64
 17  Female_Workers                          640 non-null    int64
 18  Main_Workers                            640 non-null    int64
 19  Marginal_Workers                        640 non-null    int64
 20  Non_Workers                             640 non-null    int64
 21  Cultivator_Workers                      640 non-null    int64
 22  Agricultural_Workers                    640 non-null    int64
 23  Household_Workers                       640 non-null    int64
 24  Other_Workers                           640 non-null    int64
 25  Hindus                                  640 non-null    int64
 26  Muslims                                 640 non-null    int64
 27  Christians                              640 non-null    int64
 28  Sikhs                                   640 non-null    int64
 29  Buddhists                               640 non-null    int64
 30  Jains                                   640 non-null    int64
 31  Others_Religions                        640 non-null    int64
 32  Religion_Not_Stated                     640 non-null    int64
 33  LPG_or_PNG_Households                   640 non-null    int64
 34  Housholds_with_Electric_Lighting        640 non-null    int64
 35  Households_with_Internet                640 non-null    int64
 36  Households_with_Computer                640 non-null    int64
 37  Rural_Households                        640 non-null    int64
 38  Urban_Households                        640 non-null    int64
 39  Households                              640 non-null    int64
 40  Below_Primary_Education                 640 non-null    int64
 41  Primary_Education                       640 non-null    int64
 42  Middle_Education                        640 non-null    int64
 43  Secondary_Education                     640 non-null    int64
 44  Higher_Education                        640 non-null    int64
 45  Graduate_Education                      640 non-null    int64
 46  Other_Education                         640 non-null    int64
 47  Literate_Education                      640 non-null    int64
 48  Illiterate_Education                    640 non-null    int64
 49  Total_Education                         640 non-null    int64
 50  Age_Group_0_29                          640 non-null    int64
 51  Age_Group_30_49                         640 non-null    int64
```

```
 52  Age_Group_50                                    640 non-null    int64
 53  Age_not_stated                                  640 non-null    int64
 54  District_code                                   640 non-null    int64
 55  State_name                                      640 non-null    object
 56  District_name                                   640 non-null    object
 57  Type_of_bathing_facility_Enclosure_without_roof 640 non-null    int64
 58  Not_having_bathing_facility_within_the_premises 640 non-null    int64
 59  Not_having_latrine_facility_within_the_premises 640 non-null    int64
 60  Main_source_of_drinking_water_Un_covered_well   640 non-null    int64
dtypes: int64(57), object(4)
memory usage: 305.1+ KB
```

In [10]:

```
1  df.describe()
```

Out[10]:

	District_code	Population	Male	Female	Literate	Male_Literate	F
count	640.000000	6.400000e+02	6.400000e+02	6.400000e+02	6.400000e+02	6.400000e+02	
mean	320.500000	1.891961e+06	9.738598e+05	9.181011e+05	1.193186e+06	6.793182e+05	
std	184.896367	1.544380e+06	8.007785e+05	7.449864e+05	1.068583e+06	5.924144e+05	
min	1.000000	8.004000e+03	4.414000e+03	3.590000e+03	4.436000e+03	2.614000e+03	
25%	160.750000	8.178610e+05	4.171682e+05	4.017458e+05	4.825982e+05	2.764365e+05	
50%	320.500000	1.557367e+06	7.986815e+05	7.589200e+05	9.573465e+05	5.483525e+05	
75%	480.250000	2.583551e+06	1.338604e+06	1.264277e+06	1.602260e+06	9.188582e+05	
max	640.000000	1.106015e+07	5.865078e+06	5.195070e+06	8.227161e+06	4.591396e+06	

8 rows × 57 columns

Checking if any Columns have null value in it or not

In [11]:

```
1  df.isnull().sum()==0
```

Out[11]:

```
District_code                                    True
State_name                                       True
District_name                                    True
Population                                       True
Male                                             True
                                                 ...
District_name                                    True
Type_of_bathing_facility_Enclosure_without_roof  True
Not_having_bathing_facility_within_the_premises  True
Not_having_latrine_facility_within_the_premises  True
Main_source_of_drinking_water_Un_covered_well    True
Length: 61, dtype: bool
```

Since all are 0 thus no column has null values

Total Population

In [12]:

```
df['Population'].sum()
```

Out[12]:

```
1210854977
```

Pie Chart for Male v/s Female Population

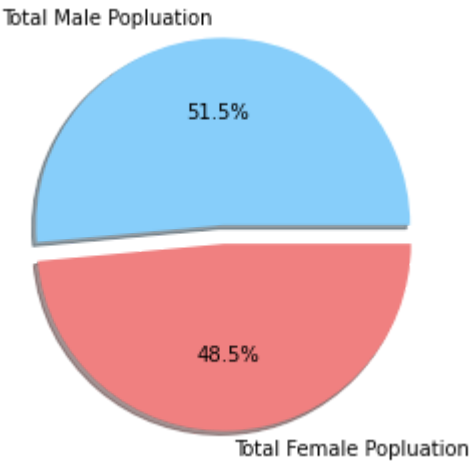
In [13]:

```
Total_Male=df['Male'].sum()
Total_Female=df['Female'].sum()
Population_array=[Total_Male,Total_Female]
labels = ['Total Male Popluation', 'Total Female Popluation']
colors = ['lightskyblue','lightcoral']
explode = (0.1, 0)
# Plot
plt.pie(Population_array, explode=explode, colors=colors, labels=labels,
autopct='%1.1f%%', shadow=True)
plt.axis('equal')

```

Out[13]:

```
(-1.1134881275574597,
 1.1006423422272433,
 -1.1166226006555118,
 1.2108741481208387)
```



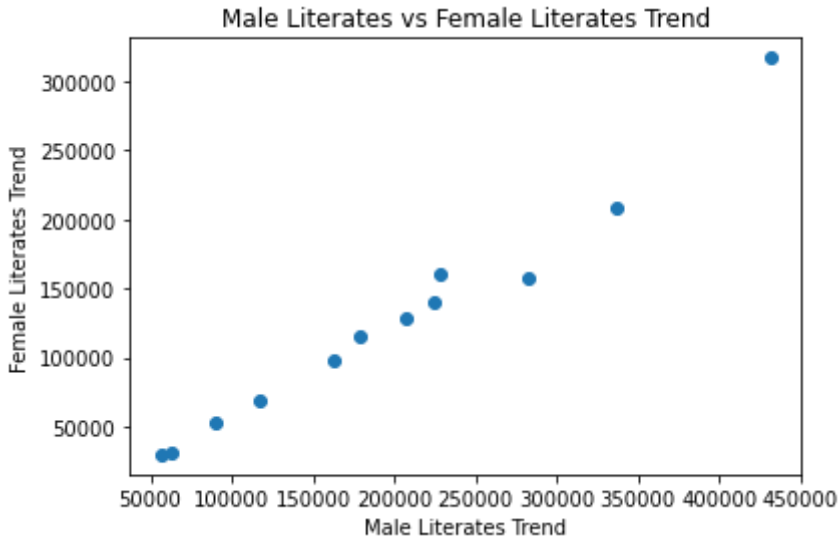
Male Literates vs Female Literates Trend

In [14]:

```
Male_Literate=df['Male_Literate'].head(12)
Female_Literate=df['Female_Literate'].head(12)
plt.scatter(np.array(Male_Literate),np.array(Female_Literate))
plt.xlabel("Male Literates Trend")
plt.ylabel("Female Literates Trend")
plt.title("Male Literates vs Female Literates Trend")
```

Out[14]:

```
Text(0.5, 1.0, 'Male Literates vs Female Literates Trend')
```



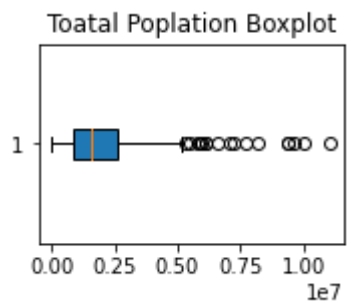
Boxplots

In [15]:

```
Population=df['Population']
Male_Population=df['Male']
Female_Population=df['Female']
plt.subplot(2,2,1)
plt.boxplot(Population,vert=False,patch_artist=True)
plt.title("Toatal Poplation Boxplot")
```

Out[15]:

```
Text(0.5, 1.0, 'Toatal Poplation Boxplot')
```



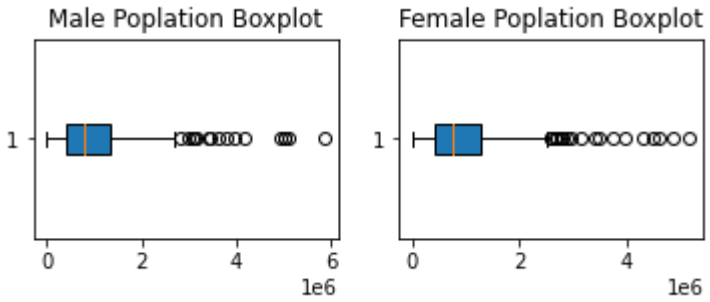
Conclusion:There are some Districts having Total Population above 50 Lakhs which are outliers

In [16]:

```
plt.subplot(2,2,1)
plt.boxplot(Male_Population,vert=False,patch_artist=True)
plt.title("Male Poplation Boxplot")
plt.subplot(2,2,2)
plt.boxplot(Female_Population,vert=False,patch_artist=True)
plt.title("Female Poplation Boxplot")
```

Out[16]:

```
Text(0.5, 1.0, 'Female Poplation Boxplot')
```



Conclusion:There are some Districts having Male Population even 60 Lakhs but, Female population is not maximum to 60 Lakhs as males

In [ ]:

```
```