### Importing the Required Libraries

```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
```

### Reading the Datasets

```
df1 = pd.read_csv('Primary.csv',encoding='ISO-8859-1',header=1)
df2 = pd.read_csv('Secondary.csv',encoding='ISO-8859-1',header=1)
df3 = pd.read_csv('Total School Age.csv',encoding='ISO-8859-
1',header=1)
```

**observation:**

1. Reading all the Datasets and We are giving a name to these Datasets.
2. There are 3 Dataset Given we can work on this separately rather than merging it together. So we can have a clear idea about the dataset; So I have planned to do EDA first and then do the required steps later.
3. Therefore we can proceed further with the DataFrame Named df1,df2 and df3.

```
df1.head()
```

```
  ISO3 Countries and areas Region Sub-region              Income Group
Total  \
0  AGO              Angola    SSA        ESA  Lower middle income (LM)
15%
1  ARG           Argentina    LAC        LAC  Upper middle income (UM)
39%
2  ARM             Armenia    ECA       EECA  Upper middle income (UM)
81%
3  BGD          Bangladesh     SA         SA  Lower middle income (LM)
34%
4  BRB            Barbados    LAC        LAC           High income (H)
63%

  Rural (Residence) Urban (Residence) Poorest (Wealth quintile)  \
0                2%                22%                         0%
1               NaN               NaN                        NaN
2               69%               89%                        46%
3               30%               49%                         7%
4               54%               68%                         9%

  Richest (Wealth quintile)                        Data source Time
period
0                       61%      Demographic and Health Survey
2015-16
1                       NaN  Multiple Indicator Cluster Survey
2011-12
```

```
2                       99%      Demographic and Health Survey
2015-16
3                       75%  Multiple Indicator Cluster Survey
2019
4                       97%  Multiple Indicator Cluster Survey
2012
```

```
df2.head(5)
```

```
   ISO3 Countries and areas Region Sub-region              Income
Group Total  \
0  AGOA              Angola    SSA        ESA  Lower middle income
(LM)   24%
1   ARG           Argentina    LAC        LAC  Upper middle income
(UM)   45%
2   ARM             Armenia    ECA       EECA  Upper middle income
(UM)   85%
3   BGD          Bangladesh     SA         SA  Lower middle income
(LM)   42%
4   BRB            Barbados    LAC        LAC           High income
(H)   76%

  Rural (Residence) Urban (Residence) Poorest (Wealth quintile)  \
0                2%                33%                          0%
1               NaN               NaN                         NaN
2               78%               91%                         54%
3               38%               57%                         13%
4               76%               76%                          4%

  Richest (Wealth quintile)                       Data source Time
period
0                       69%      Demographic and Health Survey
2015-16
1                       NaN  Multiple Indicator Cluster Survey
2011-12
2                      100%      Demographic and Health Survey
2015-16
3                       79%  Multiple Indicator Cluster Survey
2019
4                      100%  Multiple Indicator Cluster Survey
2012
```

```
df3.head(5)
```

```
  ISO3 Countries and areas Region Sub-region              Income Group
Total  \
0  DZA             Algeria   MENA       MENA  Upper middle income (UM)
24%
1  AGO              Angola    SSA        ESA  Lower middle income (LM)
17%
2  ARG           Argentina    LAC        LAC  Upper middle income (UM)
```

```
40%
3  ARM             Armenia    ECA       EECA  Upper middle income (UM)
81%
4  BGD          Bangladesh     SA         SA  Lower middle income (LM)
37%

  Rural (Residence) Urban (Residence) Poorest (Wealth quintile)  \
0                9%               32%                         1%
1                2%               24%                         0%
2               NaN              NaN                        NaN
3               71%              88%                        47%
4               33%              52%                         9%

  Richest (Wealth quintile)                       Data source Time
period
0                       77%  Multiple Indicator Cluster Survey
2018-19
1                       62%      Demographic and Health Survey
2015-16
2                       NaN  Multiple Indicator Cluster Survey
2011-12
3                       99%      Demographic and Health Survey
2015-16
4                       76%  Multiple Indicator Cluster Survey
2019
```

```
print('Number of Rows:',df1.shape[0])
print('Number of Columns:',df1.shape[1])
```

```
Number of Rows: 87
Number of Columns: 12
```

**observation:**

1. Therefore we can see that there are about 87 records and 12 columns present.
2. We can see that there 12 columns all are kind of Categorical while looking the top 5 records will make sure in further analysis.
3. Encoding is done for all these datasets and header is given for the better understanding of the datas we have.
4. So the Next step is to clean the DataFrame 1 and Treat all the Null values.
5. Therefore we can do the visualization for the required stuffs and we can draw some insights.

```
print('Categorical
Columns:',df1.select_dtypes(exclude=np.number).columns)
print('-----------------------------------------------------------------
-------')
print('Numerical
Columns:',df1.select_dtypes(include=np.number).columns)
```

```
Categorical Columns: Index(['ISO3', 'Countries and areas', 'Region',
'Sub-region', 'Income Group',
       'Total', 'Rural (Residence)', 'Urban (Residence)',
       'Poorest (Wealth quintile)', 'Richest (Wealth quintile)', 'Data
source',
       'Time period'],
      dtype='object')
------------------------------------------------------------------------
Numerical Columns: Index([], dtype='object')
```

**Observation:**

1. From the above dataset 1 we can see that there is no presence of Numerical Columns.
2. All the 12 columns are in the Categorical types but we can see that Total', 'Rural (Residence)', 'Urban (Residence)','Poorest (Wealth quintile)', 'Richest (Wealth quintile) are needed to be converted into the Numerical type and therefore we do the required stuffs to convert these into Numerical Stuffs.

```
df1['Total'] = df1['Total'].str.split('%',expand=True)
[0].astype(float)

df1['Rural (Residence)'] = df1['Rural
(Residence)'].str.split('%',expand=True)[0].astype(float)

df1['Urban (Residence)'] = df1['Urban
(Residence)'].str.split('%',expand=True)[0].astype(float)

df1['Poorest (Wealth quintile)'] = df1['Poorest (Wealth
quintile)'].str.split('%',expand=True)[0].astype(float)

df1['Richest (Wealth quintile)'] = df1['Richest (Wealth
quintile)'].str.split('%',expand=True)[0].astype(float)

print('Categorical
Columns:',df1.select_dtypes(exclude=np.number).columns)
print('------------------------------------------------------------------
-------')
print('Numerical
Columns:',df1.select_dtypes(include=np.number).columns)

Categorical Columns: Index(['ISO3', 'Countries and areas', 'Region',
'Sub-region', 'Income Group',
       'Data source', 'Time period'],
      dtype='object')
------------------------------------------------------------------------
Numerical Columns: Index(['Total', 'Rural (Residence)', 'Urban
(Residence)',
       'Poorest (Wealth quintile)', 'Richest (Wealth quintile)'],
      dtype='object')
```

**Observation:**

1. Therefore we have converted the Total', 'Rural (Residence)', 'Urban (Residence)','Poorest (Wealth quintile)', 'Richest (Wealth quintile) to the Numerical type with the help of split method and I have removed the '%' and then converted these to float type.
2. We can see that the type of this converted variables in the Numerical Column and therefore the variables are converted to required type and therefore we can proceed further.

```
df1['Time period'].value_counts()
```

```
2018         17
2019          9
2018-19       7
2017          7
2013          6
2012          5
2017-18       5
2015          4
2015-16       4
2014          3
2011-12       3
2010          3
2016          3
2014-15       3
2016-17       2
2076          1
2018-2019     1
2562          1
2027          1
2011          1
2012-99       1
Name: Time period, dtype: int64
```

**observation:**

1. In the Time Period column we can see there are some anomalies present.
2. These anomalies are needed to be cleared and therefore I have palnned to convert these anamolies to the null type.

```
df1 ['Time period'] = df1['Time
period'].replace({'2027':np.nan,'2076':np.nan,'2012-
99':np.nan,'2562':np.nan})
```

**observation:**

1. Therefore we have converted the Time period column to the required ones and can proceed further.
2. The values of the anamolies have changed to the Null.

```
df1['Data source'].value_counts()
```

Multiple Indicator Cluster Survey
44
Demographic and Health Survey
16
STEP Skills Measurement Household Survey 2012 (Wave 1)
2
ENSANUT
2
Nicaragua National Demographic and Health Survey 2011-2012
1
UK Data Archive Information for the Study 8298. Statistical Bulletin: Internet Access Ð Households and Individuals, 2016. 1
STEP Skills Measurement Household Survey 2013 (Wave 1)
1
LSMS
1
South Africa Living Conditions Survey 2014-15
1
Somalia High Frequency Survey
1
Russia Longitudinal Monitoring Survey, 2018
1
ENDES
1
General Household Survey, Panel 2018-2019, Wave 4
1
National Survey on Household Living Conditions and Agriculture 2014, Wave 2 Panel Data 1
STEP Skills Measurement Household Survey 2013 (Wave 2)
1
Morocco Household and Youth Survey 2010
1
The Japan Household Panel Survey
1
Multiscopo sulle famiglie: aspetti della vita quotidiana
1
SUSENAS
1
2015 Household Income, Expenditure and Consumption Survey
1
Enqu□te Djiboutienne Aupr□s des M□nages pour les Indicateurs Sociaux 2012 - Donn□es pour utilisation publique 1
CHARLS 2018
1
CASEN
1
Bulgarian Longitudinal Inclusive Society Survey (BLISS) 2013
1
Brazil Continuous National Household Sample Survey (Continuous PNAD) 2018, IBGE. 1

```
EDSA
1
UNICEF Nutrition Survey 2017
1
Name: Data source, dtype: int64
```

```
threshold = 3
counts = df1['Data source'].value_counts()
other_cities = counts[counts < threshold].index.tolist()
df1['Data source'] = df1['Data source'].apply(lambda x: 'Other' if x
in other_cities else x)
```

```
df1['Data source'].value_counts()
```

```
Multiple Indicator Cluster Survey    44
Other                                27
Demographic and Health Survey        16
Name: Data source, dtype: int64
```

**Observation:**

1. As there are Many Data source type. I have planned to convert the Lesser Data Source type in 'other' category; So that we can visualize it better and produce some insight.
2. Therefore we keep a threshold of 3 and then convert the lesser count into 'Other' Category.
3. Then we have done the conversion and we can see that the Data source is converted into 3 category and can be used for analysis of the Data source more easier.

```
df1 ['Income Group'] = df1['Income Group'].str.split('(',expand=True)
[0]
```

**Observation:**

1. The Income Group is changed to the required condition and removed the Unwanted stuffs.
2. The Income group is sorted and we can proceed further.

```
df1['Countries and areas'].value_counts().sum()
```

```
87
```

**Here I have tried alot of variation to bring the Countries and Area into 5 sub-divisions. But founded that some are not named Properly and even I can't sort these Countries into sub-divisions. I tried many variation like developed,undeveloped,develpoing and etc for easy access but these are not proper and Planned to drop these out.**

```
df1['ISO3'].value_counts().sum()
```

```
87
```

**Observation:**

1. The Countries and areas and then ISO3 are so unique and we can't use this for any analysis.
2. These are said to be reductant and can be removed.
3. So I am removing these columns.

```
df1.drop(columns='Countries and areas',inplace=True)
```

```
df1.drop(columns='ISO3',inplace=True)
```

```
df1.head()
```

```
  Region Sub-region          Income Group  Total  Rural (Residence)  \
0    SSA        ESA  Lower middle income    15.0                2.0
1    LAC        LAC  Upper middle income    39.0                NaN
2    ECA       EECA  Upper middle income    81.0               69.0
3     SA         SA  Lower middle income    34.0               30.0
4    LAC        LAC          High income    63.0               54.0

   Urban (Residence)  Poorest (Wealth quintile)  Richest (Wealth
quintile)  \
0               22.0                        0.0
61.0
1                NaN                        NaN
NaN
2               89.0                       46.0
99.0
3               49.0                        7.0
75.0
4               68.0                        9.0
97.0

                        Data source Time period
0      Demographic and Health Survey     2015-16
1  Multiple Indicator Cluster Survey     2011-12
2      Demographic and Health Survey     2015-16
3  Multiple Indicator Cluster Survey        2019
4  Multiple Indicator Cluster Survey        2012
```

**observation**

1. After dropping the reductant columns; We can have look into the top 5 records.
2. Then we look into the Null values and treat them.

```
df1.isnull().sum()
```

```
Region                         0
Sub-region                     0
Income Group                   0
Total                          0
Rural (Residence)             11
Urban (Residence)              8
Poorest (Wealth quintile)     18
```

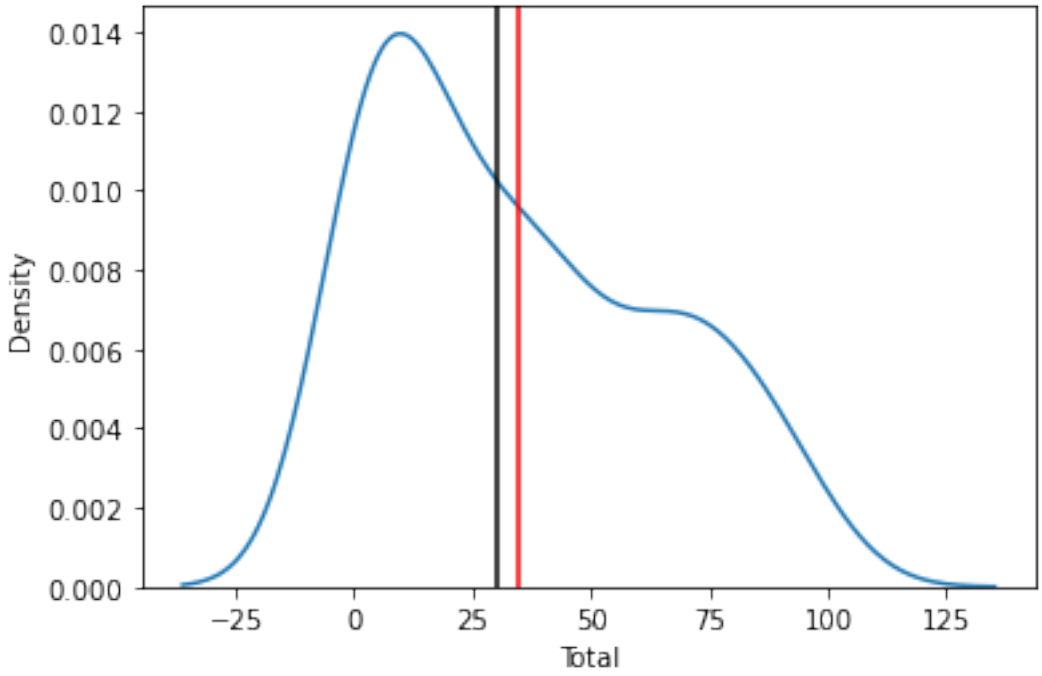
```
Richest (Wealth quintile)    21
Data source                   0
Time period                   4
dtype: int64
```

**observation:**

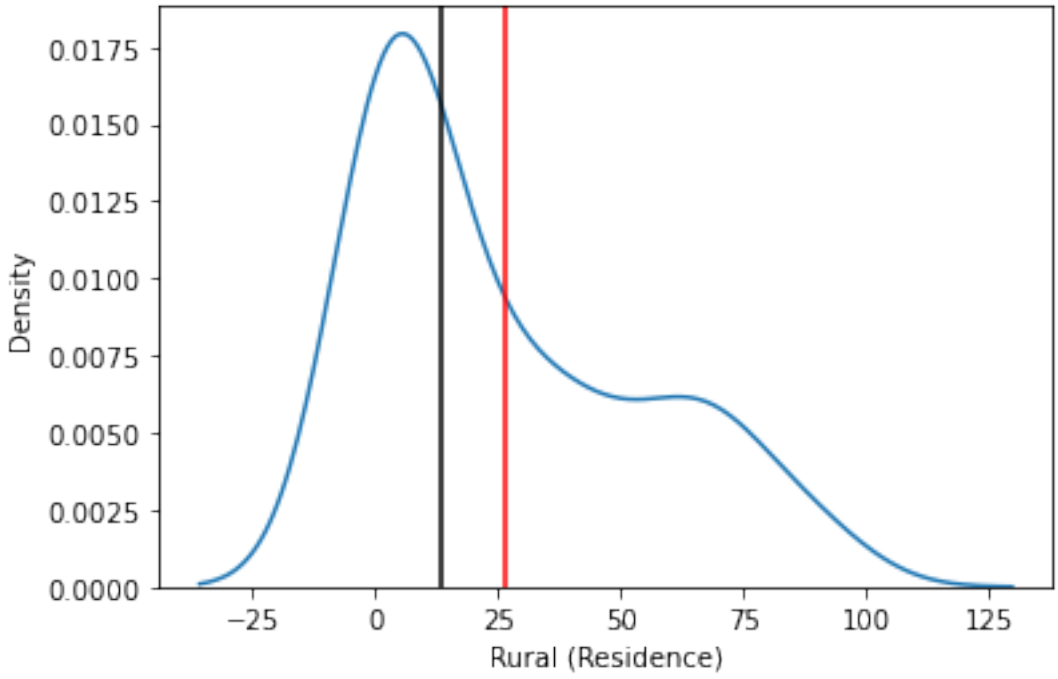
1. The Rural(Residence) as 11,Urban(Residence)as 8,Poorest(Wealth quintile)as 18,Richest(Wealth quintile)as 21,Time period as 4 Null values and these are needed to be imputed.
2. Before we impute we need to look into the skew for imputing the Mean or Median values for the Numerical Columns.

```
for i in df1.select_dtypes(include=np.number):
    sns.kdeplot(x= df1[i])
    plt.axvline(df1[i].mean(),color='red')
    plt.axvline(df1[i].median(),color='black')
    print('Column Name:',i,'Skewness:',df1[i].skew())
    plt.show()
```

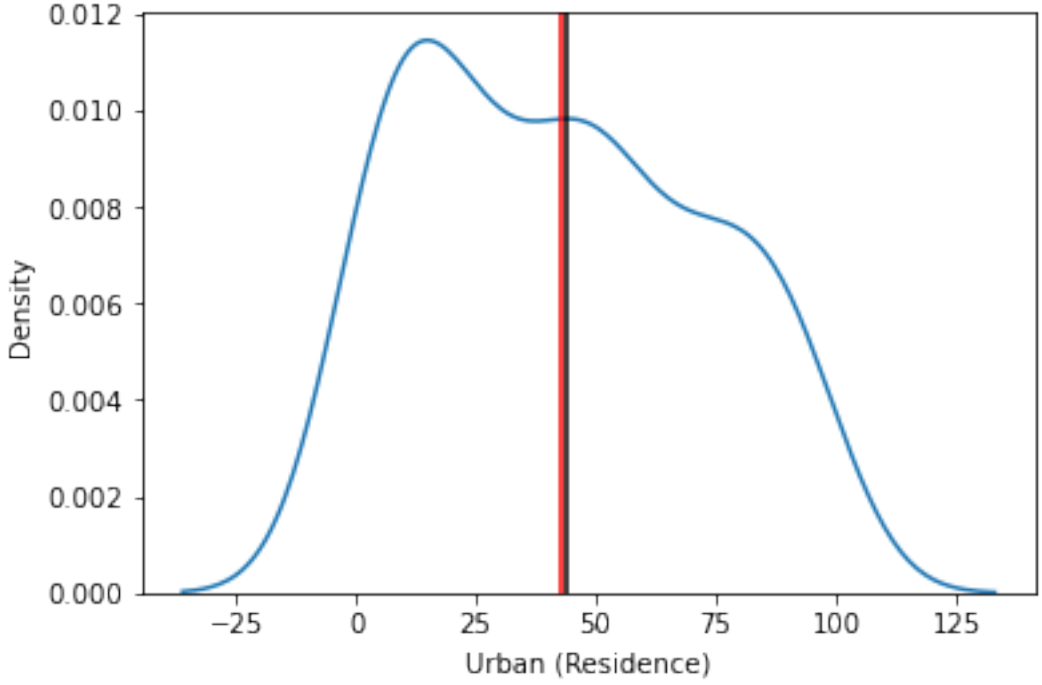
```
Column Name: Total Skewness: 0.5401312883061952
```



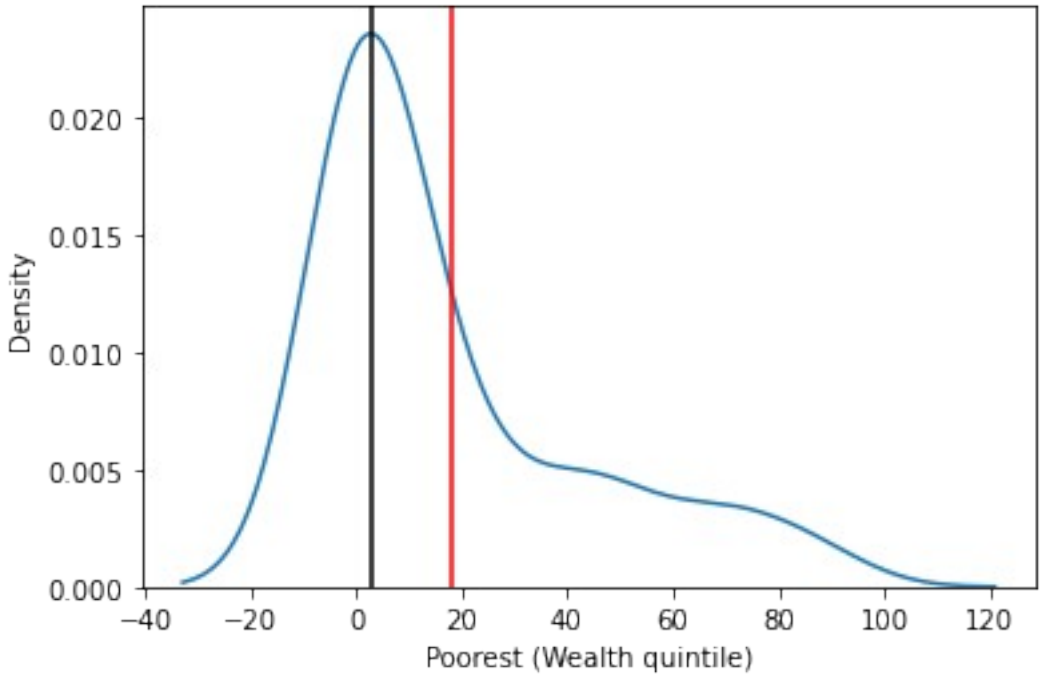

```
Column Name: Rural (Residence) Skewness: 0.8562030328393314
```



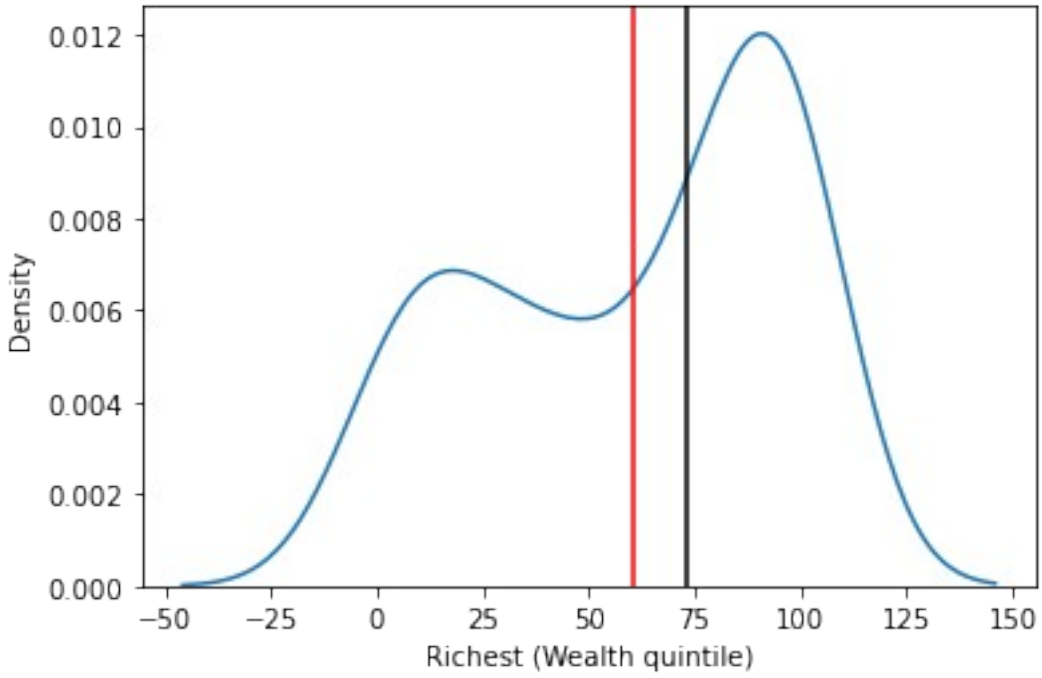

Column Name: Urban (Residence) Skewness: 0.25656895871095015




Column Name: Poorest (Wealth quintile) Skewness: 1.4091340335146407




```
Column Name: Richest (Wealth quintile) Skewness: -0.42341421599345697
```




**Observation:**

1. As we can see that they all are skewed for Numerical values and we can't impute the Missing values with the Mean value.
2. The black Line represents the Median value and the Red line represents the Mean values.

3. The Line showcases us that how the variables are present in the dataset we have and we can impute the nulls via but if we impute the Nulls like this the values that are filled either be a mean or median value but this doesn't make sense.
4. So we will use groupby and then impute the Necessary values to the columns.
5. By this method we can have a better value than imputing the null values from the columns's Median or Mean.

```
 df1['Urban (Residence)'] = df1.groupby('Region')['Urban
(Residence)'].fillna(df1['Urban (Residence)'].median()).values

df1['Poorest (Wealth quintile)'] = df1.groupby('Region')['Poorest
(Wealth quintile)'].fillna(df1['Poorest (Wealth
quintile)'].median()).values

df1['Richest (Wealth quintile)'] = df1.groupby('Region')['Richest
(Wealth quintile)'].fillna(df1['Richest (Wealth
quintile)'].median()).values

 df1['Rural (Residence)'] = df1.groupby('Region')['Rural
(Residence)'].fillna(df1['Rural (Residence)'].median()).values

df1.isnull().sum()
```

```
Region                          0
Sub-region                      0
Income Group                    0
Total                           0
Rural (Residence)               0
Urban (Residence)               0
Poorest (Wealth quintile)       0
Richest (Wealth quintile)       0
Data source                     0
Time period                     4
dtype: int64
```

**All the Null values have been Treated except the Time Period and it 4 Null values and we can't impute the mode of the column as it gives out Wrong information. I have planned to drop it out**

```
df1.dropna(axis=0,inplace=True)

df1.isnull().sum()
```

```
Region                          0
Sub-region                      0
Income Group                    0
Total                           0
Rural (Residence)               0
Urban (Residence)               0
Poorest (Wealth quintile)       0
Richest (Wealth quintile)       0
Data source                     0
```

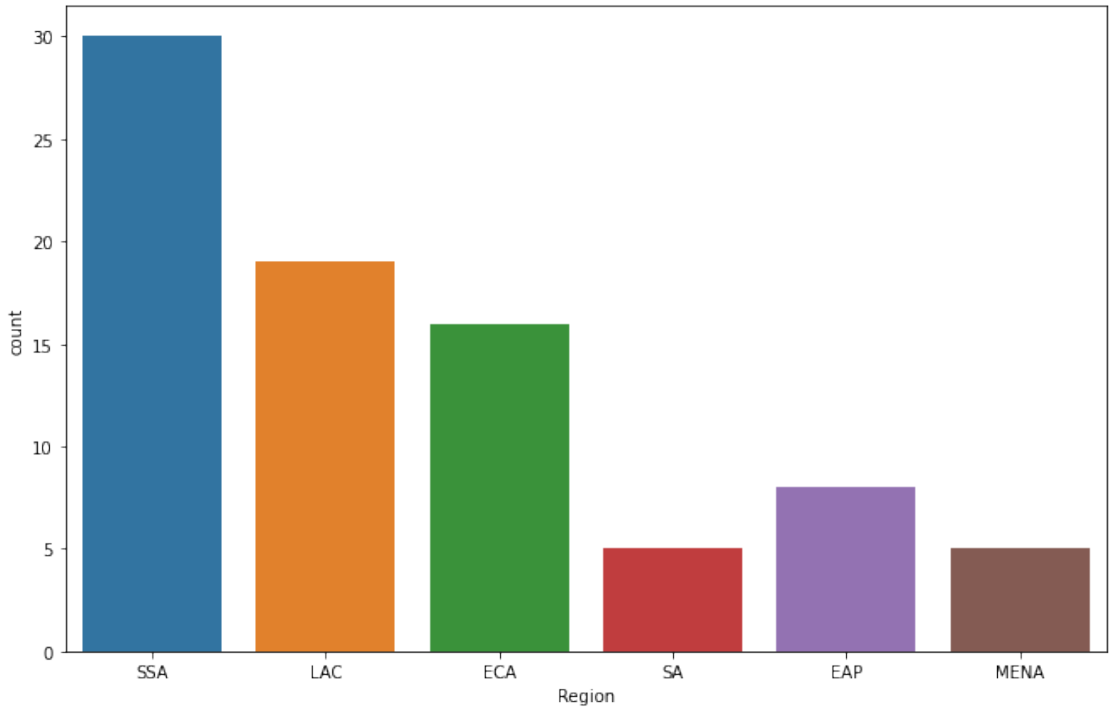
```
Time period                  0
dtype: int64
```

## Data Exploration

```
df1.head()
```

```
  Region Sub-region          Income Group  Total  Rural (Residence)  \
0    SSA        ESA  Lower middle income    15.0                2.0
1    LAC        LAC  Upper middle income    39.0               13.5
2    ECA       EECA  Upper middle income    81.0               69.0
3     SA         SA  Lower middle income    34.0               30.0
4    LAC        LAC          High income    63.0               54.0

   Urban (Residence)  Poorest (Wealth quintile)  Richest (Wealth
quintile)  \
0               22.0                        0.0
61.0
1               44.0                        3.0
73.0
2               89.0                       46.0
99.0
3               49.0                        7.0
75.0
4               68.0                        9.0
97.0

                        Data source Time period
0      Demographic and Health Survey     2015-16
1  Multiple Indicator Cluster Survey     2011-12
2      Demographic and Health Survey     2015-16
3  Multiple Indicator Cluster Survey        2019
4  Multiple Indicator Cluster Survey        2012
```

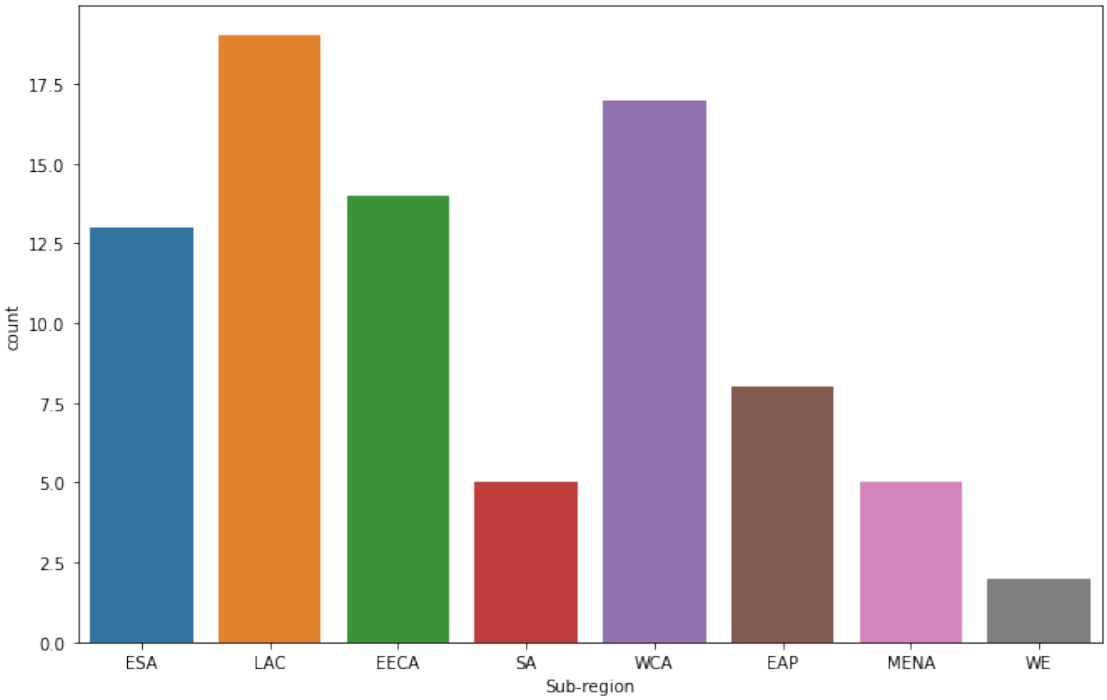
```
plt.figure(figsize=(11,7))
sns.countplot(df1['Region'])
plt.show()
```




**Inference:**

1. The SSA is the largest count and therefore we have the LAC as the second Region in this dataFrame.
2. This shows that the first Region in this df1 is the SSA which means most the members around 30 are from this and then we have LAC in the second where the count is around 20.
3. The Last Region in this dataFrame is SA with the count of 5 which indicate that least members are from SA.

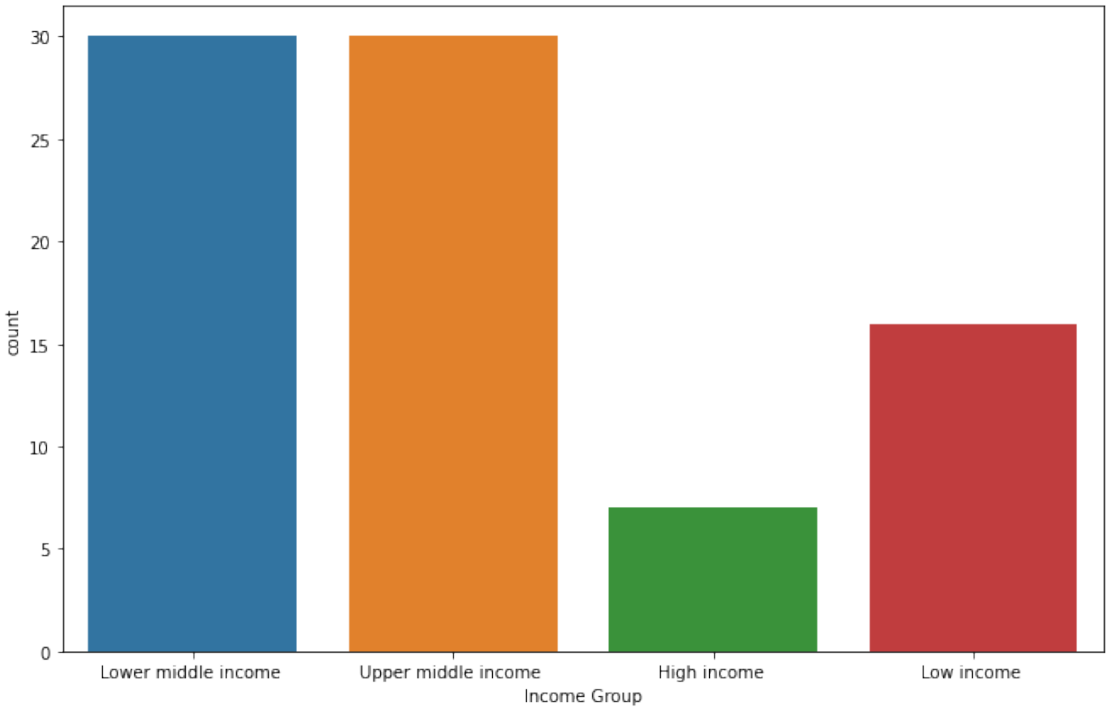
```
plt.figure(figsize=(11,7))
sns.countplot(df1['Sub-region'])
plt.show()
```




**Inference:**

1. The LAC count is more in the sub-Region category and then comes the WCA where the count is quite Lesser in terms of the LAC.
2. Then we have in the last is the WE where the count is less than 2.5 and thus we can see it via the countplot.
3. The most important Sub-Region are LAC,WCA and EECA.

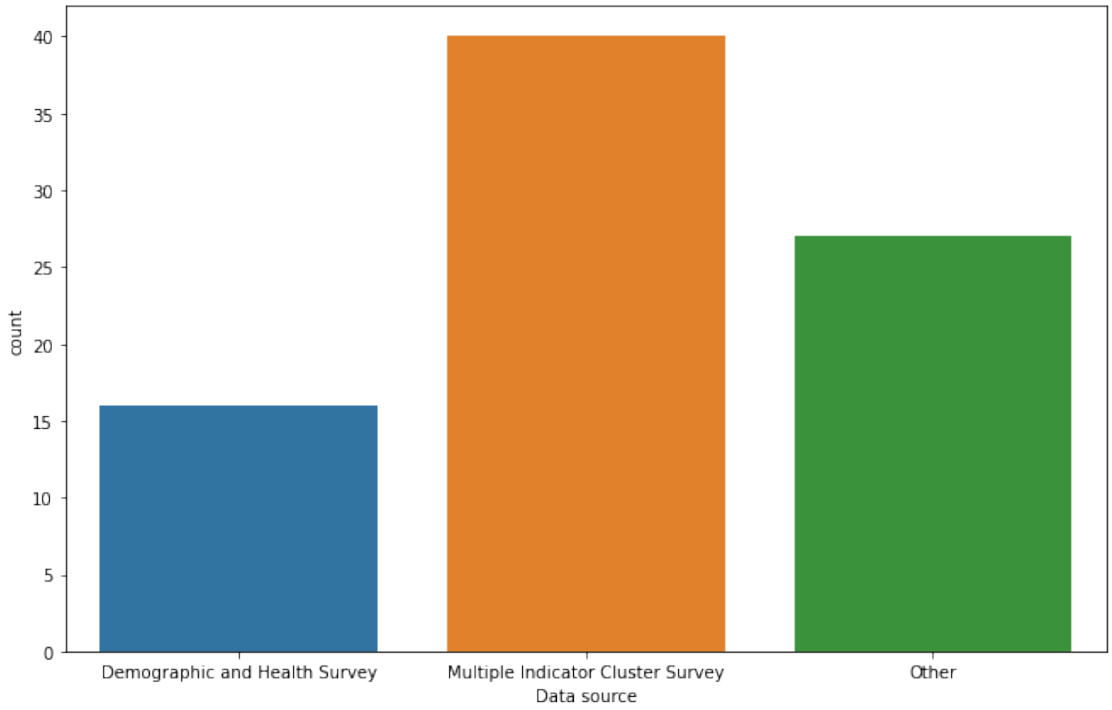
```
plt.figure(figsize=(11,7))
sns.countplot(df1['Income Group'])
plt.show()
```




**Inference:**

1. It is very suprising to see that Lower Middle Income and the Upper Middle Income are falling into the same count around 30Each.
2. Then we have Low Income in the 3rd place where the count is lesser than 20.
3. Then we have High Income category where the count is too low than other 3 category and the count is also lower than 10.
4. Therefore we can conclude that we have a DataFrame where most people are not in High Income Category.

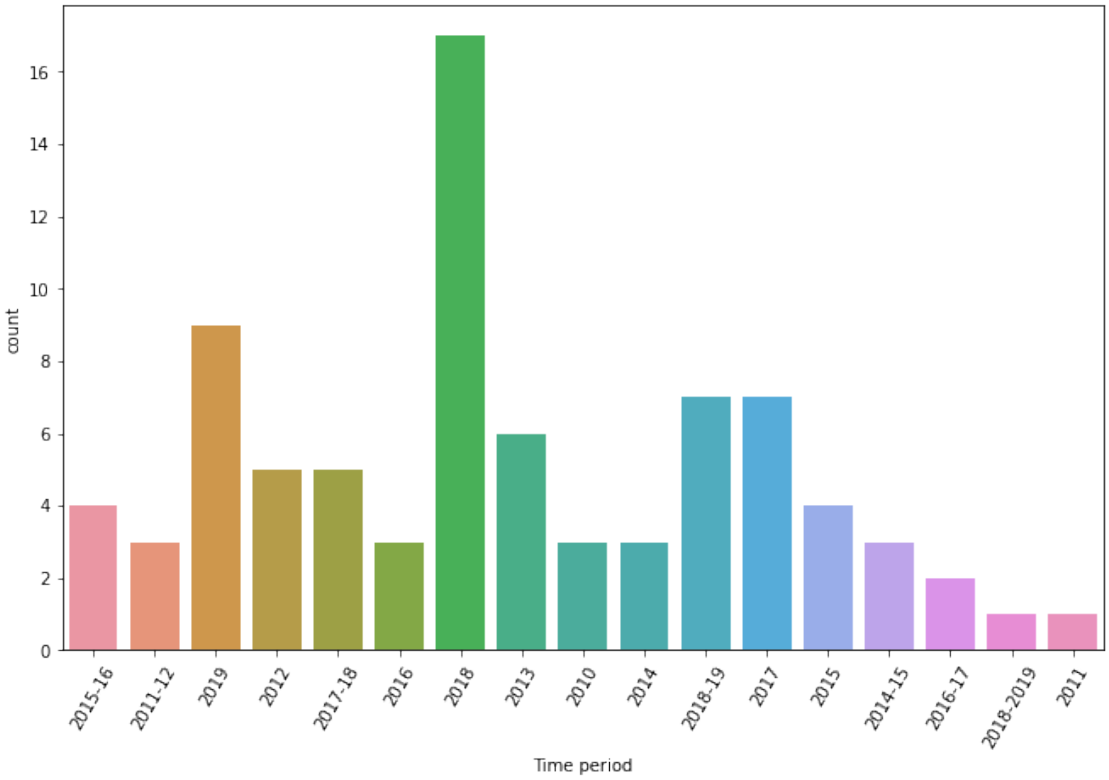
```
plt.figure(figsize=(11,7))
sns.countplot(df1['Data source'])
plt.show()
```




**Inference:**

1. The DataSource is where the Data is collected and most of the datas are collected via Multiple Indicator Cluster Survey.
2. Then we have the Demographic and Health Survey; In this 2 Data Source almost all the Datas are collected.

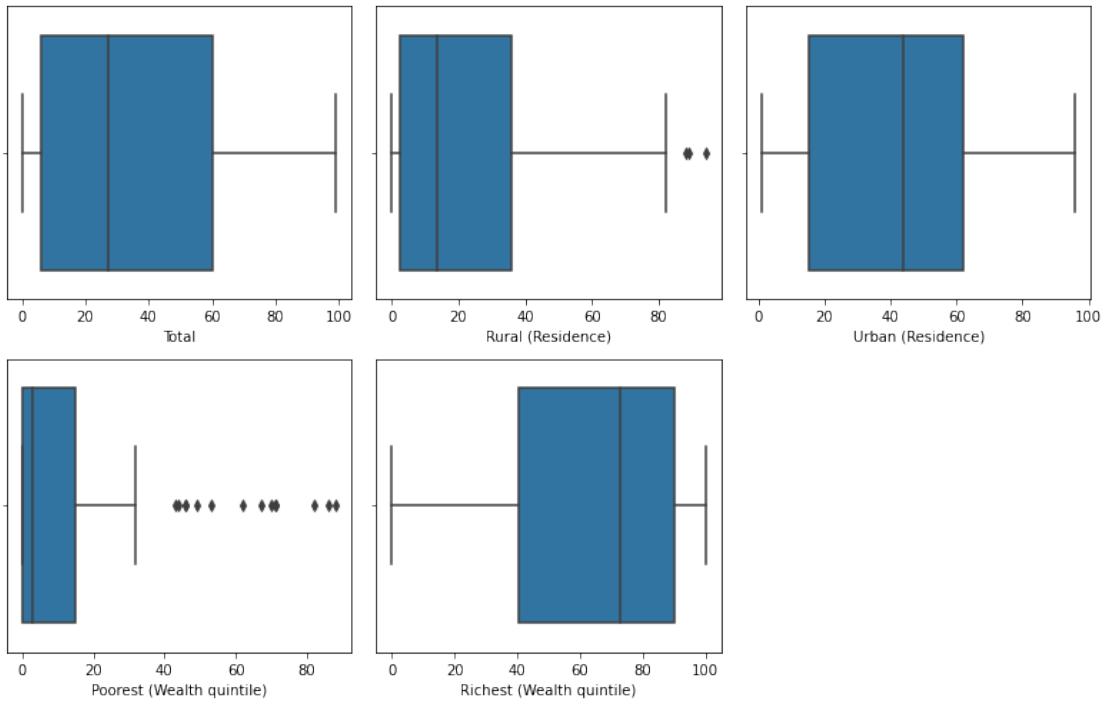
```
plt.figure(figsize=(11,7))
sns.countplot(df1['Time period'])
plt.xticks(rotation=60)
plt.show()
```




**Inference:**

1. There are many Time Period but the 2018 Time Period has the most number of the count and it is around 16+ count and then next time period is that we have 2019 and it contributes around 8+ count.
2. The least is the 2011 count and it has less than 2 count and same for the 2018-2019 Time Period also.

```
it=1
plt.figure(figsize=(11,7))
for i in df1.select_dtypes(include=np.number):
    plt.subplot(2,3,it)
    sns.boxplot(df1[i])
    it=it+1
plt.tight_layout()
plt.show()
```




**Inference:**

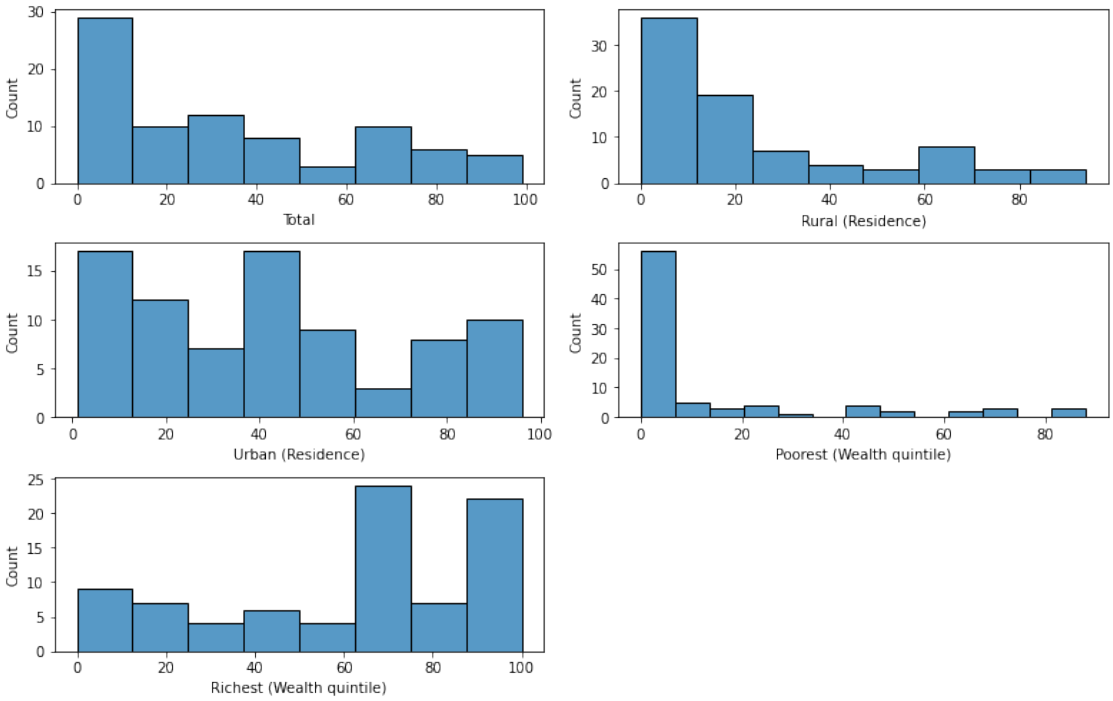
1. From the above Boxplot we can see that there is presence of outliers in the DataFrame1.
2. There are some outliers present in the Rural and Poorest we can reduce those by using IQR method but it doesn't make sense.
3. When we do the IQR some records are lost and thus we can proceed further with these Outliers.
4. They all fall in the same scale and there is no need of scaling.

```
df1.head()
```

```
  Region Sub-region          Income Group  Total  Rural (Residence)  \
0    SSA        ESA  Lower middle income    15.0                2.0
1    LAC        LAC  Upper middle income    39.0               13.5
2    ECA       EECA  Upper middle income    81.0               69.0
3     SA         SA  Lower middle income    34.0               30.0
4    LAC        LAC          High income    63.0               54.0

   Urban (Residence)  Poorest (Wealth quintile)  Richest (Wealth
quintile)  \
0               22.0                        0.0
61.0
1               44.0                        3.0
73.0
2               89.0                       46.0
99.0
3               49.0                        7.0
75.0
```

```
4                68.0                          9.0
97.0

                         Data source Time period
0      Demographic and Health Survey     2015-16
1  Multiple Indicator Cluster Survey     2011-12
2      Demographic and Health Survey     2015-16
3  Multiple Indicator Cluster Survey        2019
4  Multiple Indicator Cluster Survey        2012
```

```
it=1
plt.figure(figsize=(11,7))
for i in df1.select_dtypes(include=np.number):
    plt.subplot(3,2,it)
    sns.histplot(df1[i])
    it=it+1
plt.tight_layout()
plt.show()
```

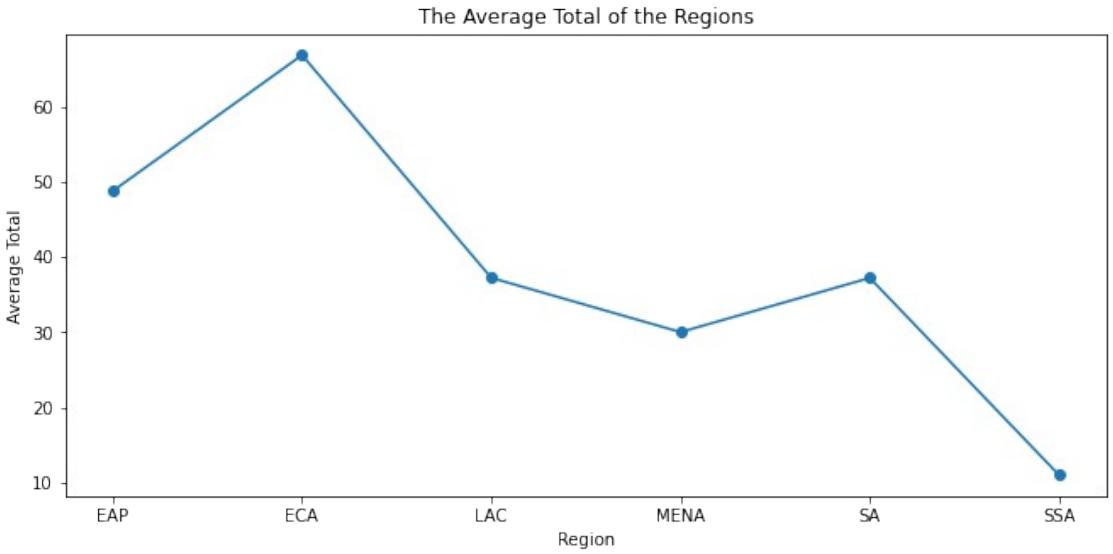



**Inference:**

1. From the Histplot we can get some inference about the spread of the values.
2. Here we can see that the all the values are spread across the range of 0-100.
3. The Poorest have a lesser count down the spread.
4. In the Richest we have value starts from lower value to the higher range.

```
plt.figure(figsize=(11,5))
df1.groupby('Region')['Total'].mean().plot(kind='line',marker='o')
plt.ylabel('Average Total')
```

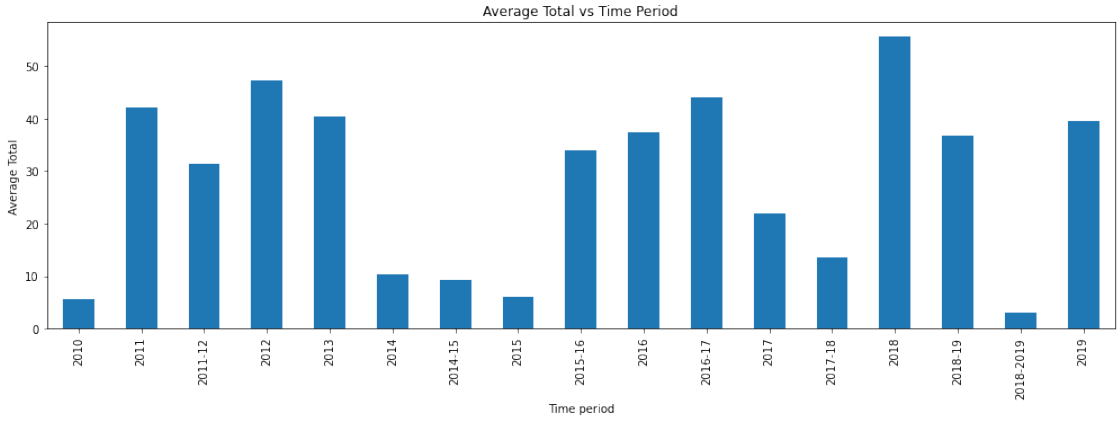
```
plt.title('The Average Total of the Regions')
plt.show()
```




**Inference:**

1. The value of Average total and the Region is plotted. We can see that ECA region has the Higher Total value and the least is the SSA with the total of lesser than 10.

```
plt.figure(figsize=(17,5))
df1.groupby('Time period')['Total'].mean().plot(kind='bar')
plt.ylabel('Average Total')
plt.title('Average Total vs Time Period',color='black')
plt.show()
```

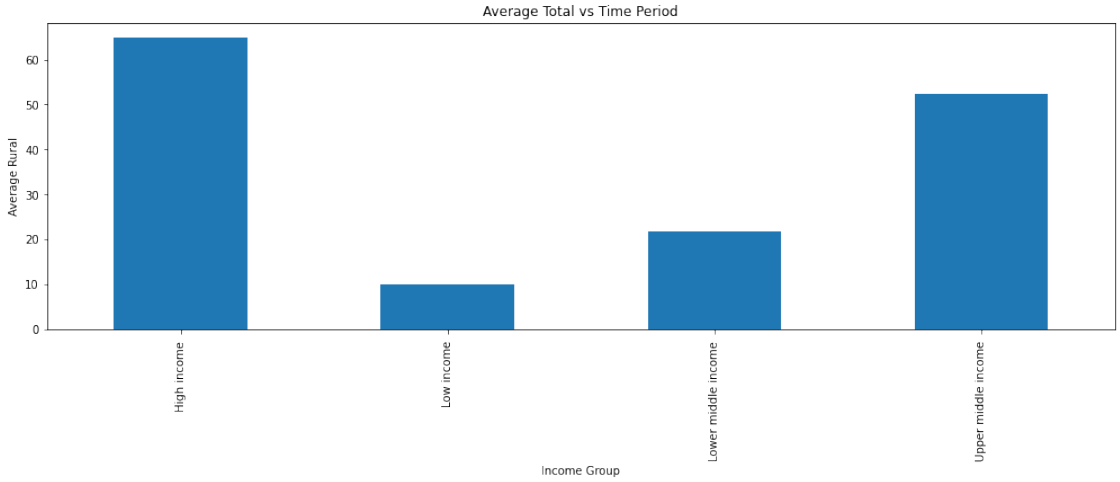



**Inference:**

1. The 2018 Time period is more in the Average total and therefore we can say that the contribution in 2018 is more than the other Time period.
2. The Least Time period is 2018-2019 with the total average is lesser than 10.

```
plt.figure(figsize=(17,5))
df1.groupby('Income Group')['Total'].mean().plot(kind='bar')
```

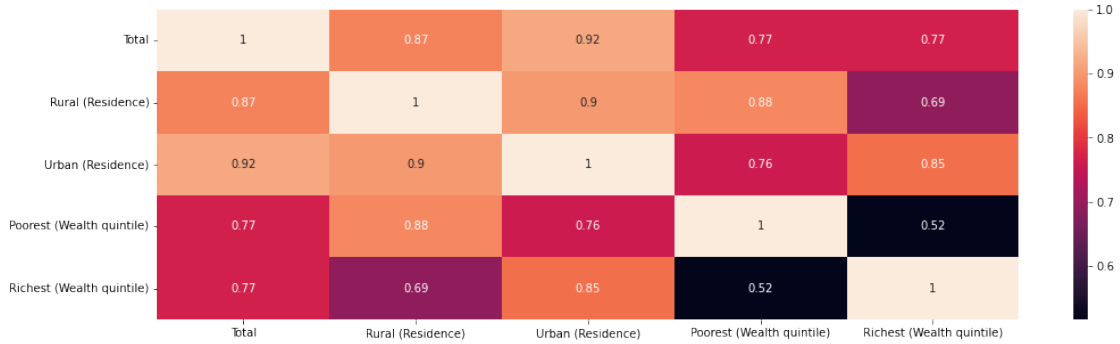
```
plt.ylabel('Average Rural')
plt.title('Average Total vs Time Period',color='black')
plt.show()
```




**Inference:**

1. The contribution is more for the category High Income.
2. As we known that the total will be more the High Income people and that is what seen over here.
3. The least is the Low Income and thus total is less than what we expect to have.

```
plt.figure(figsize=(17,5))
sns.heatmap(df1.corr(),annot=True)
plt.show()
```




**inference:**

1. All the variables are Highly correlated to each other.
2. This shows that there are chances of one influence the other.
3. Possible of Multicollinearity.

## EDA on DataFrame 2

```
df2.head()
```

```
   ISO3 Countries and areas Region Sub-region              Income
Group Total  \
0  AGOA              Angola    SSA        ESA  Lower middle income
(LM)   24%
1   ARG           Argentina    LAC        LAC  Upper middle income
(UM)   45%
2   ARM             Armenia    ECA       EECA  Upper middle income
(UM)   85%
3   BGD          Bangladesh     SA         SA  Lower middle income
(LM)   42%
4   BRB            Barbados    LAC        LAC           High income
(H)   76%

  Rural (Residence) Urban (Residence) Poorest (Wealth quintile)  \
0                2%               33%                         0%
1               NaN               NaN                        NaN
2               78%               91%                        54%
3               38%               57%                        13%
4               76%               76%                         4%

  Richest (Wealth quintile)                       Data source Time
period
0                       69%      Demographic and Health Survey
2015-16
1                       NaN  Multiple Indicator Cluster Survey
2011-12
2                      100%      Demographic and Health Survey
2015-16
3                       79%  Multiple Indicator Cluster Survey
2019
4                      100%  Multiple Indicator Cluster Survey
2012
```

**Inference:**

1. Therefore we Have top 5 records of the DataFrame 2 and we can see that is quite similar to the previous DataFrame 1 and therefore we will perform the same structured EDA method that we followed upon during the DataFrame 1.

```
print('Number of Records:',df2.shape[0])
print('Number of columns:',df2.shape[1])
```

```
Number of Records: 82
Number of columns: 12
```

**observation:**

1. Therefore we can see that there are about 87 records and 12 columns present.
2. We can see that there 12 columns all are kind of Categorical while looking the top 5 records will make sure in further analysis.

3. Encoding is done for all these datasets and header is given for the better understanding of the datas we have.
4. So the Next step is to clean the DataFrame 1 and Treat all the Null values.
5. Therefore we can do the visualization for the required stuffs and we can draw some insights.

```
print('Categorical
Columns:',df2.select_dtypes(exclude=np.number).columns)
print('----------------------------------------------------------------
-------')
print('Numerical
Columns:',df2.select_dtypes(include=np.number).columns)
```

```
Categorical Columns: Index(['ISO3', 'Countries and areas', 'Region',
'Sub-region', 'Income Group',
       'Total', 'Rural (Residence)', 'Urban (Residence)',
       'Poorest (Wealth quintile)', 'Richest (Wealth quintile)', 'Data
source',
       'Time period'],
      dtype='object')
----------------------------------------------------------------------
Numerical Columns: Index([], dtype='object')
```

**Observation:**

1. From the above dataset 2 we can see that there is no presence of Numerical Columns.
2. All the 12 columns are in the Categorical types but we can see that Total', 'Rural (Residence)', 'Urban (Residence)','Poorest (Wealth quintile)', 'Richest (Wealth quintile) are needed to be converted into the Numerical type and therefore we do the required stuffs to convert these into Numerical Stuffs.

```
df2['Total'] = df2['Total'].str.split('%',expand=True)
[0].astype(float)

df2['Rural (Residence)'] = df2['Rural
(Residence)'].str.split('%',expand=True)[0].astype(float)

df2['Poorest (Wealth quintile)'] = df2['Poorest (Wealth
quintile)'].str.split('%',expand=True)[0].astype(float)

df2['Richest (Wealth quintile)'] = df2['Richest (Wealth
quintile)'].str.split('%',expand=True)[0].astype(float)

df2['Urban (Residence)'] = df2['Urban
(Residence)'].str.split('%',expand=True)[0].astype(float)

df2['Time period'].value_counts()
```

```
2018         15
2017          8
2018-19       7
```

```
2019          7
2013          6
2012          5
2017-18       5
2015-16       4
2014          3
2010          3
2011-12       3
2014-15       3
2015          3
2016          2
2016-17       2
2076          1
2018-2019     1
2562          1
2011          1
3019          1
2012-99       1
Name: Time period, dtype: int64
```

```
df2['Time period'] = df2['Time
period'].replace({'2076':np.nan,'2562':np.nan,'2012-
99':np.nan,'3019':np.nan})
```

**Inference:**

1. Therefore we can see that there are some anomaly present and I have planned to convert these stuffs into the Null values and later we can Impute these anomaly.

```
df2['Data source'].value_counts()
```

```
Multiple Indicator Cluster Survey
44
Demographic and Health Survey
16
STEP Skills Measurement Household Survey 2012 (Wave 1)
2
Nicaragua National Demographic and Health Survey 2011-2012
1
UK Data Archive Information for the Study 8298. Statistical Bulletin:
Internet Access Ð Households and Individuals, 2016.     1
STEP Skills Measurement Household Survey 2013 (Wave 1)
1
LSMS
1
South Africa Living Conditions Survey 2014-15
1
Somalia High Frequency Survey
1
Russia Longitudinal Monitoring Survey, 2018
1
General Household Survey, Panel 2018-2019, Wave 4
```

```
1
National Survey on Household Living Conditions and Agriculture 2014,
Wave 2 Panel Data                                                    1
Morocco Household and Youth Survey 2010
1
STEP Skills Measurement Household Survey 2013 (Wave 2)
1
The Japan Household Panel Survey
1
Multiscopo sulle famiglie: aspetti della vita quotidiana
1
SUSENAS
1
Enqu□te Djiboutienne Aupr□s des M□nages pour les Indicateurs Sociaux
2012 - Donn□es pour utilisation publique                             1
CHARLS 2018
1
CASEN
1
Bulgarian Longitudinal Inclusive Society Survey (BLISS) 2013
1
Brazil Continuous National Household Sample Survey (Continuous PNAD)
2018, IBGE.                                                          1
UNICEF Nutrition Survey 2017
1
Name: Data source, dtype: int64
```

```
threshold = 3
counts = df2['Data source'].value_counts()
other_cities = counts[counts < threshold].index.tolist()
df2['Data source'] = df2['Data source'].apply(lambda x: 'Other' if x
in other_cities else x)
```

**Observation:**

1. As there are Many Data source type. I have planned to convert the Lesser Data Source type in 'other' category; So that we can visualize it better and produce some insight.
2. Therefore we keep a threshold of 3 and then convert the lesser count into 'Other' Category.
3. Then we have done the conversion and we can see that the Data source is converted into 3 category and can be used for analysis of the Data source more easier.

```
df2 ['Income Group'] = df2['Income Group'].str.split('(',expand=True)
[0]
```

```
df2.head()
```

```
   ISO3 Countries and areas Region Sub-region          Income Group
Total  \
0  AGOA              Angola    SSA        ESA  Lower middle income
```

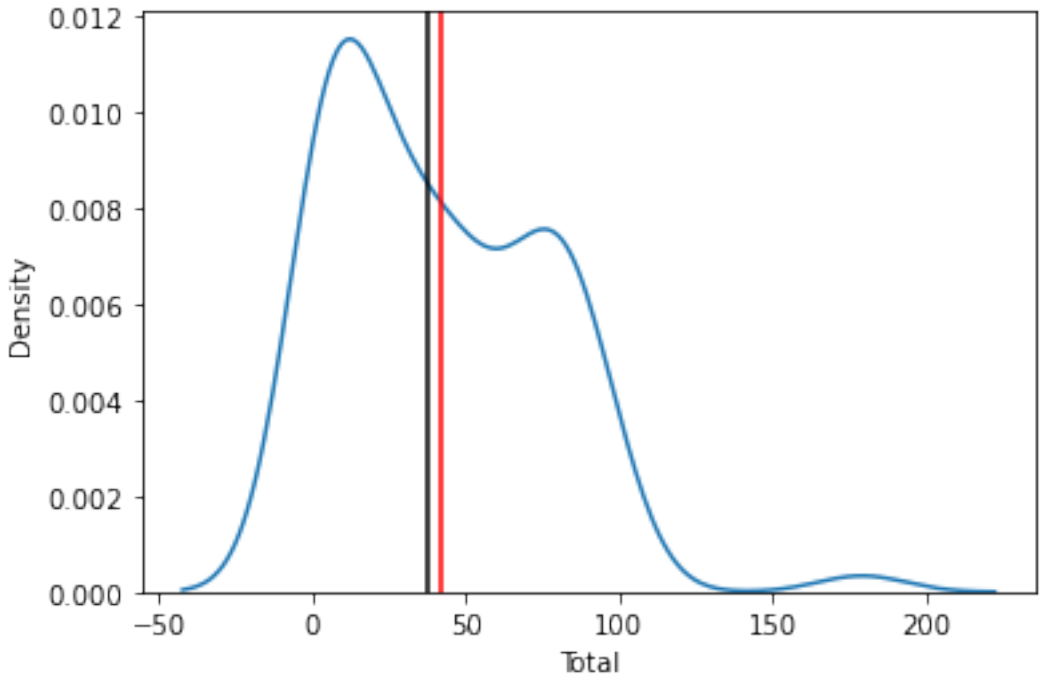
```
24.0
1   ARG           Argentina    LAC        LAC  Upper middle income
45.0
2   ARM             Armenia    ECA       EECA  Upper middle income
85.0
3   BGD          Bangladesh     SA         SA  Lower middle income
42.0
4   BRB            Barbados    LAC        LAC          High income
76.0

   Rural (Residence)  Urban (Residence)  Poorest (Wealth quintile)  \
0                2.0               33.0                        0.0
1                NaN                NaN                        NaN
2               78.0               91.0                       54.0
3               38.0               57.0                       13.0
4               76.0               76.0                        4.0

   Richest (Wealth quintile)                       Data source Time
period
0                       69.0      Demographic and Health Survey
2015-16
1                        NaN  Multiple Indicator Cluster Survey
2011-12
2                      100.0      Demographic and Health Survey
2015-16
3                       79.0  Multiple Indicator Cluster Survey
2019
4                      100.0  Multiple Indicator Cluster Survey
2012
```

```
df2.drop(columns='Countries and areas',inplace=True)
df2.drop(columns='ISO3',inplace=True)
```

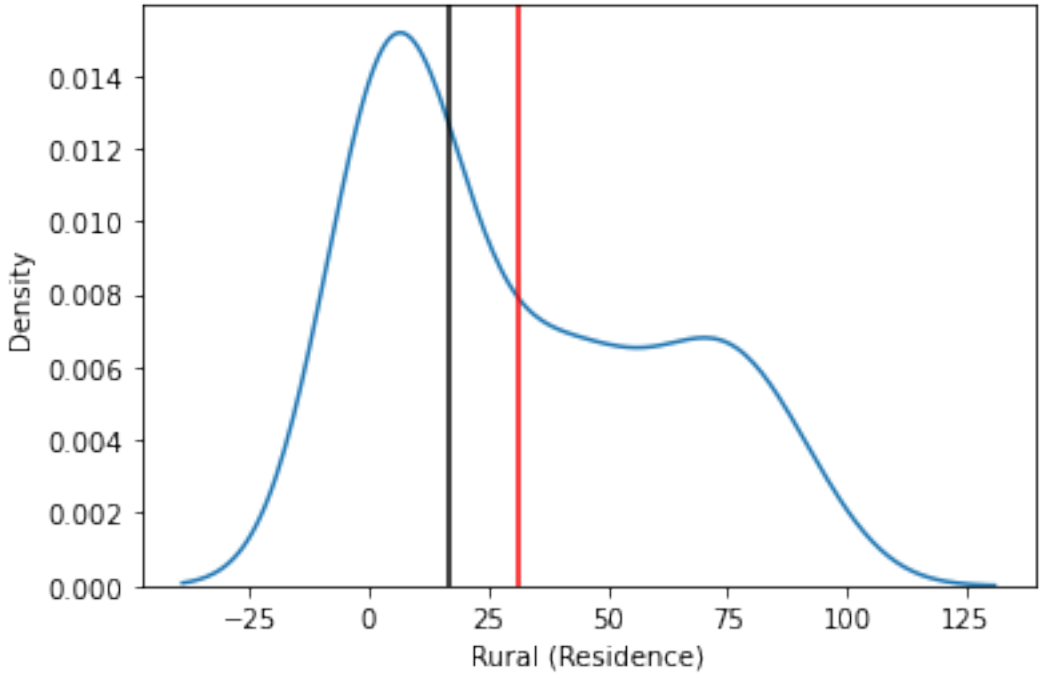
**As these are not significant columns we can drop these out and proceed futher**

```
for i in df2.select_dtypes(include=np.number):
    sns.kdeplot(x= df2[i])
    plt.axvline(df2[i].mean(),color='red')
    plt.axvline(df2[i].median(),color='black')
    print('Column Name:',i,'Skewness:',df2[i].skew())
    plt.show()
```

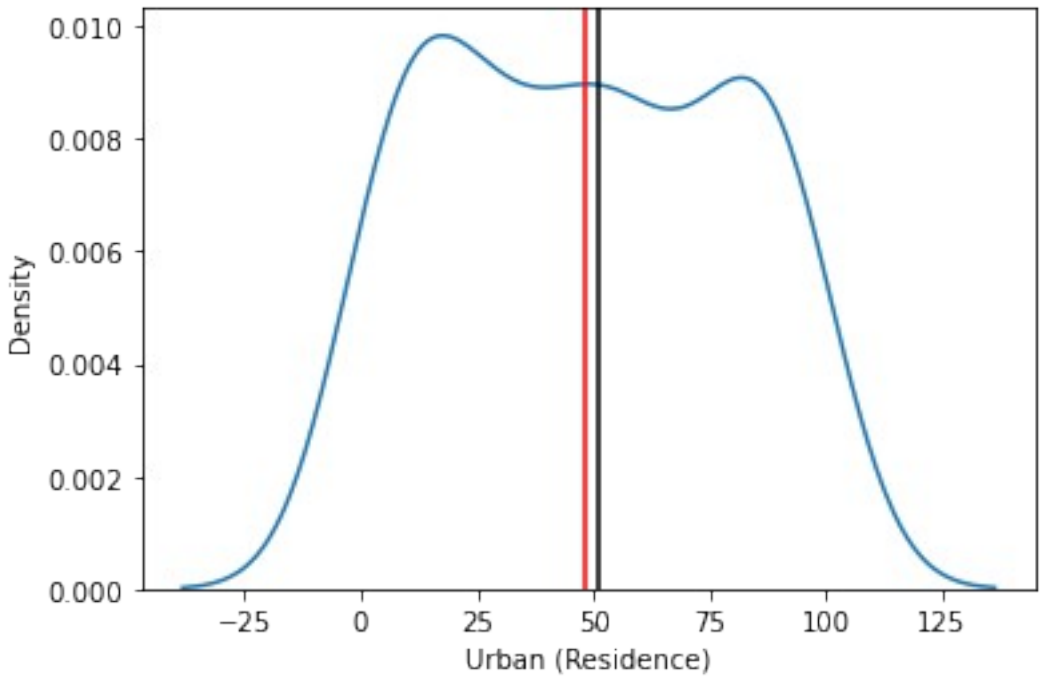
```
Column Name: Total Skewness: 0.9048632152428207
```



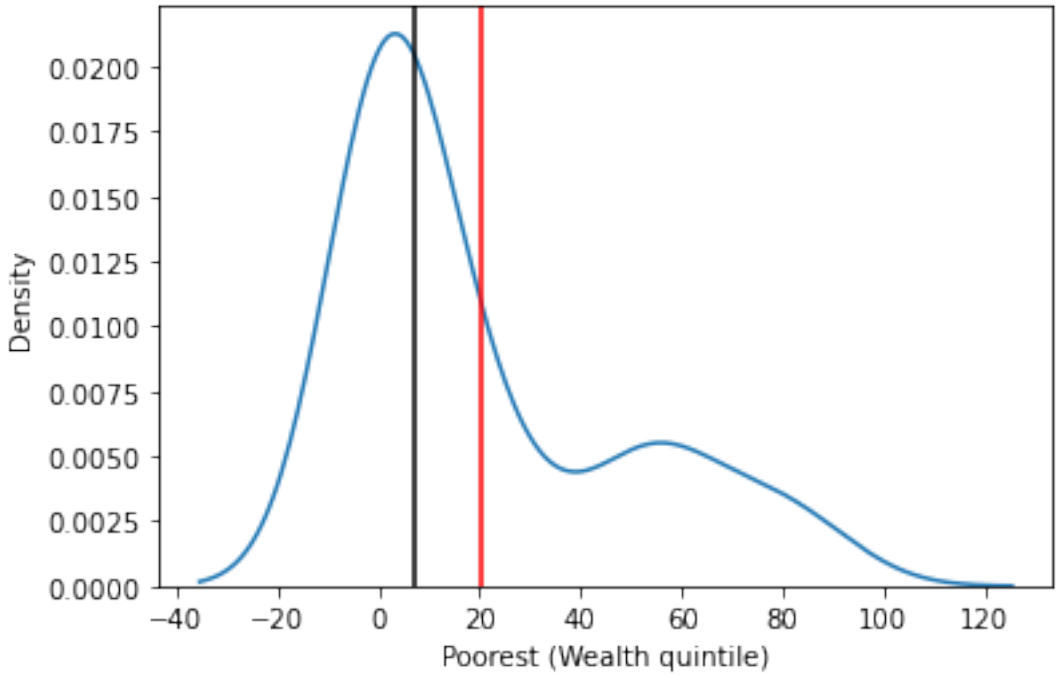

Column Name: Rural (Residence) Skewness: 0.5997982890985141



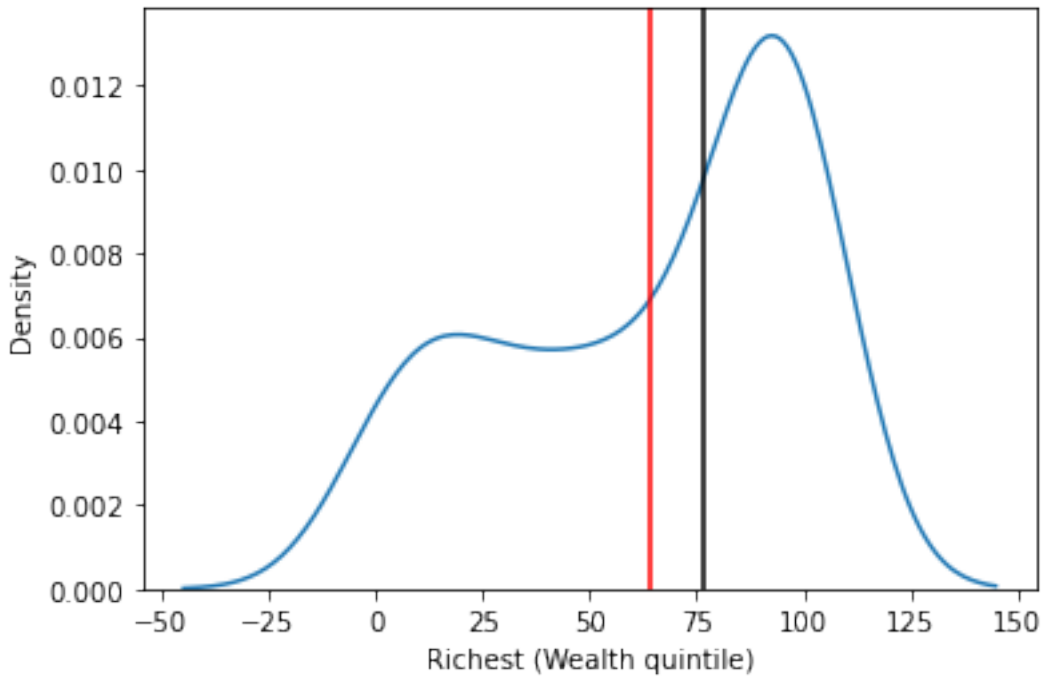

Column Name: Urban (Residence) Skewness: 0.053141267464263196




Column Name: Poorest (Wealth quintile) Skewness: 1.171076596558539




Column Name: Richest (Wealth quintile) Skewness: -0.5764189374945966




```
df2['Urban (Residence)'] = df2.groupby('Region')['Urban
(Residence)'].fillna(df1['Urban (Residence)'].median()).values
df2['Poorest (Wealth quintile)'] = df2.groupby('Region')['Poorest
(Wealth quintile)'].fillna(df1['Poorest (Wealth
quintile)'].median()).values
df2['Richest (Wealth quintile)'] = df2.groupby('Region')['Richest
(Wealth quintile)'].fillna(df1['Richest (Wealth
quintile)'].median()).values
df2['Rural (Residence)'] = df2.groupby('Region')['Rural
(Residence)'].fillna(df1['Rural (Residence)'].median()).values
```

**Inference:**

1. The Null values are filled with the groupby clause as we can't impute the mean and median values for this.
2. If mean or median is imputed we may impute wrong values so I went further wiht the Groupby and then taking the median value and then we are imputing it further.

```
df2.isnull().sum()
```

```
Region                          0
Sub-region                      0
Income Group                    0
Total                           0
Rural (Residence)               0
Urban (Residence)               0
Poorest (Wealth quintile)       0
Richest (Wealth quintile)       0
Data source                     0
```

```
Time period                  4
dtype: int64
```

```
df2.dropna(axis=0,inplace=True)
```

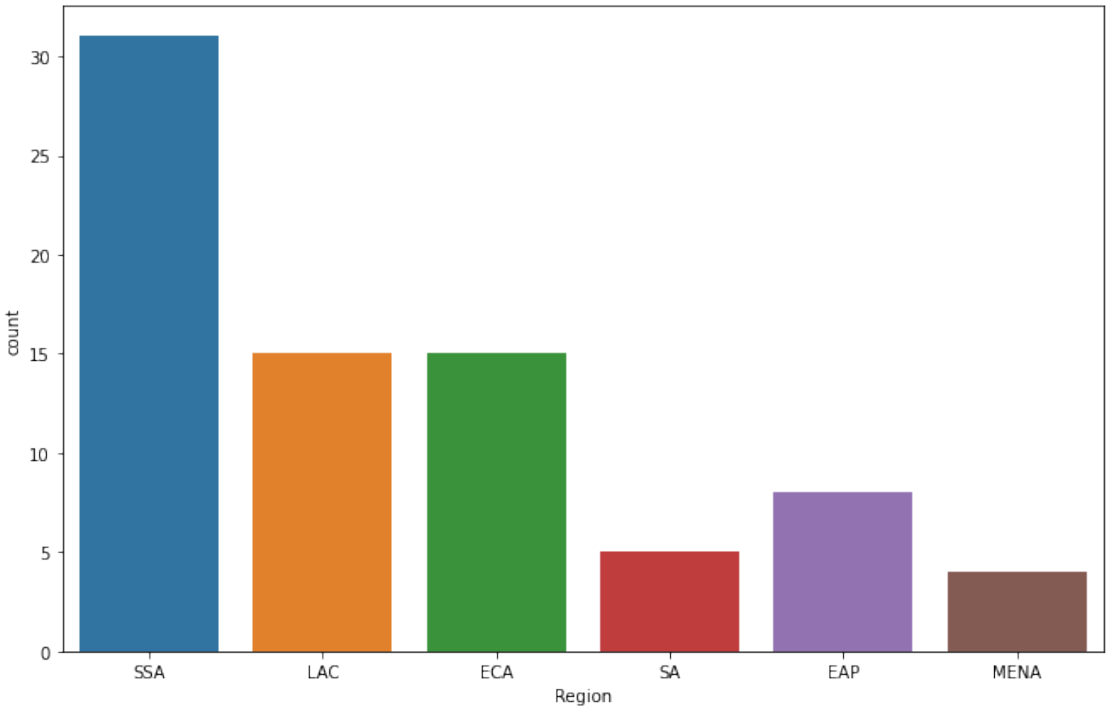
**Inference:**

1. The Null values present in the Time period column as we can't impute any values it makes an huge error so we planned to drop it and proceed further. As there are only 4 reocrds with these Null values it doesn't make any differences.

```
df2.head()
```

```
  Region Sub-region          Income Group  Total  Rural (Residence)  \
0    SSA        ESA  Lower middle income    24.0                2.0
1    LAC        LAC  Upper middle income    45.0               13.5
2    ECA       EECA  Upper middle income    85.0               78.0
3     SA         SA  Lower middle income    42.0               38.0
4    LAC        LAC          High income    76.0               76.0

   Urban (Residence)  Poorest (Wealth quintile)  Richest (Wealth
quintile)  \
0               33.0                        0.0
69.0
1               44.0                        3.0
73.0
2               91.0                       54.0
100.0
3               57.0                       13.0
79.0
4               76.0                        4.0
100.0

                        Data source Time period
0      Demographic and Health Survey     2015-16
1  Multiple Indicator Cluster Survey     2011-12
2      Demographic and Health Survey     2015-16
3  Multiple Indicator Cluster Survey        2019
4  Multiple Indicator Cluster Survey        2012
```

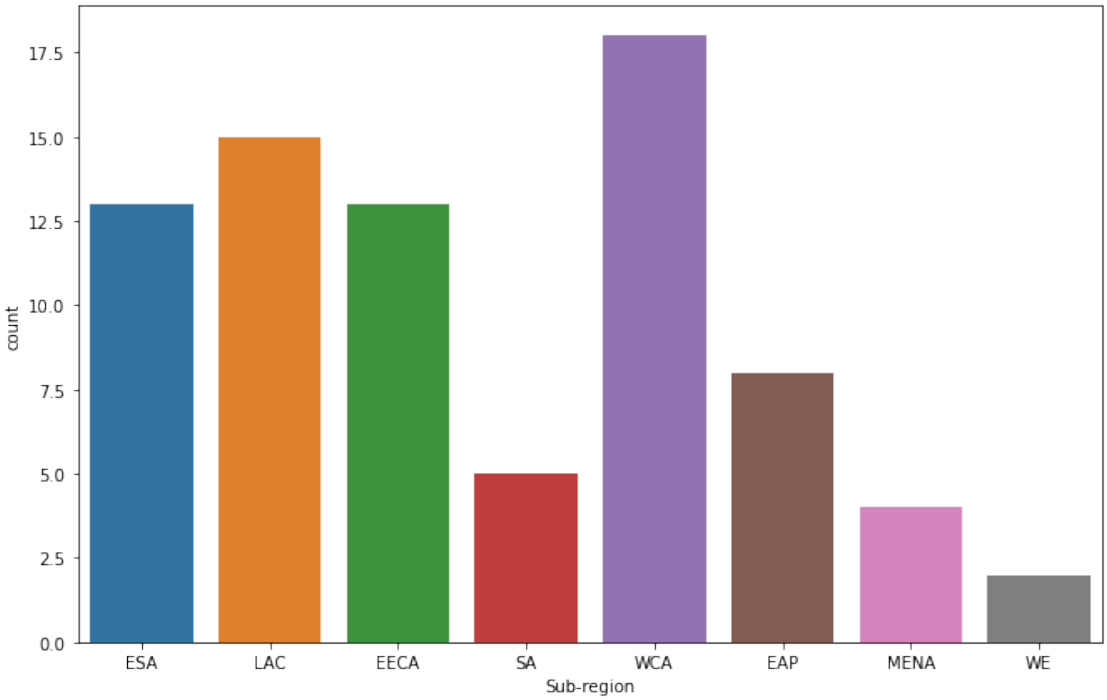
```
plt.figure(figsize=(11,7))
sns.countplot(df2['Region'])
plt.show()
```




**Inference:**

1. The SSA is the largest count and therefore we have the ECA as the second Region in this dataFrame.
2. This shows that the first Region in this df2 is the SSA which means most the members around 30 are from this and then we have LAC in the second where the count is around 20.
3. The Last Region in this dataFrame is MENA with the count of 5 which indicate that least members are from SA.

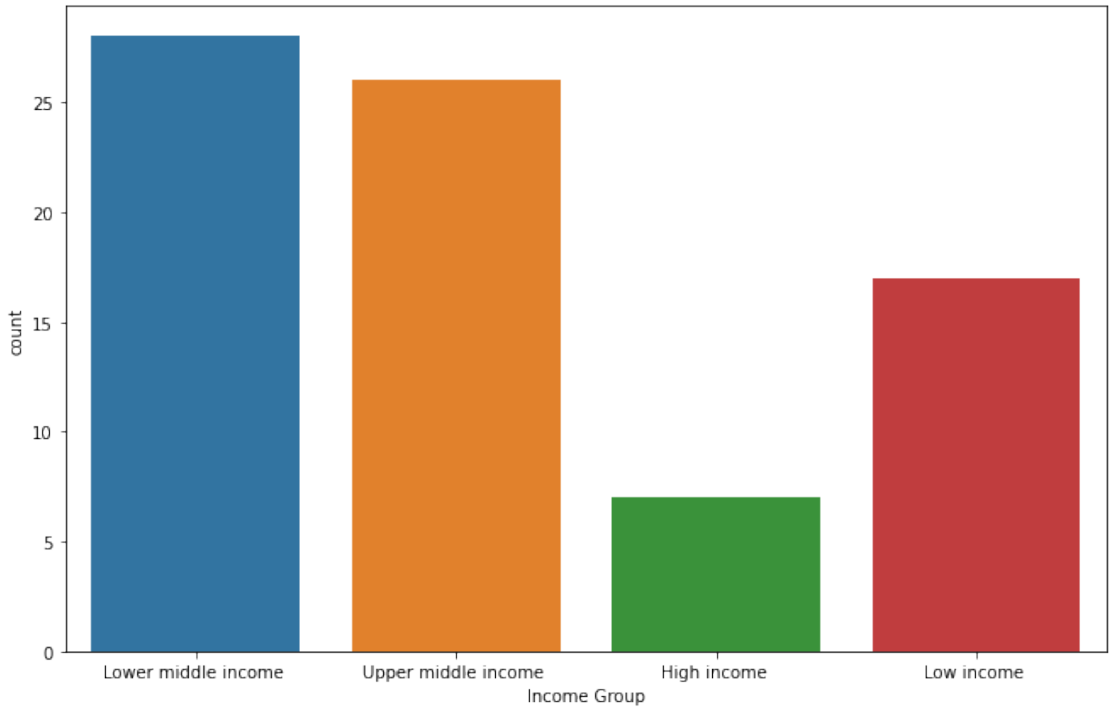
```
plt.figure(figsize=(11,7))
sns.countplot(df2['Sub-region'])
plt.show()
```




**Inference:**

1. The WCA count is more in the sub-Region category and then comes the LAC where the count is quite Lesser in terms of the LAC.
2. Then we have in the last is the WE where the count is less than 2.5 and thus we can see it via the countplot.
3. The most important Sub-Region are LAC,WCA and EECA.

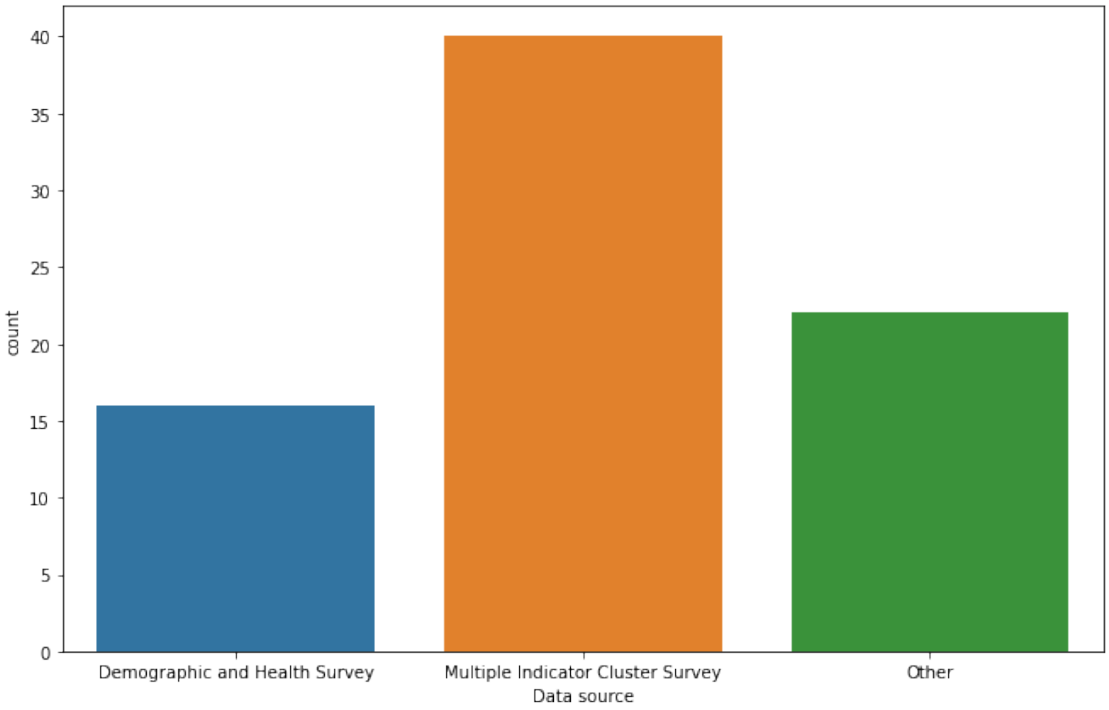
```
plt.figure(figsize=(11,7))
sns.countplot(df2['Income Group'])
plt.show()
```




**Inference:**

1. It is very suprising to see that Lower Middle Income and the Upper Middle Income are Higher than other two category.
2. Then we have Low Income in the 3rd place where the count is lesser than 20.
3. Then we have High Income category where the count is too low than other 3 category and the count is also lower than 10.
4. Therefore we can conclude that we have a DataFrame where most people are not in High Income Category

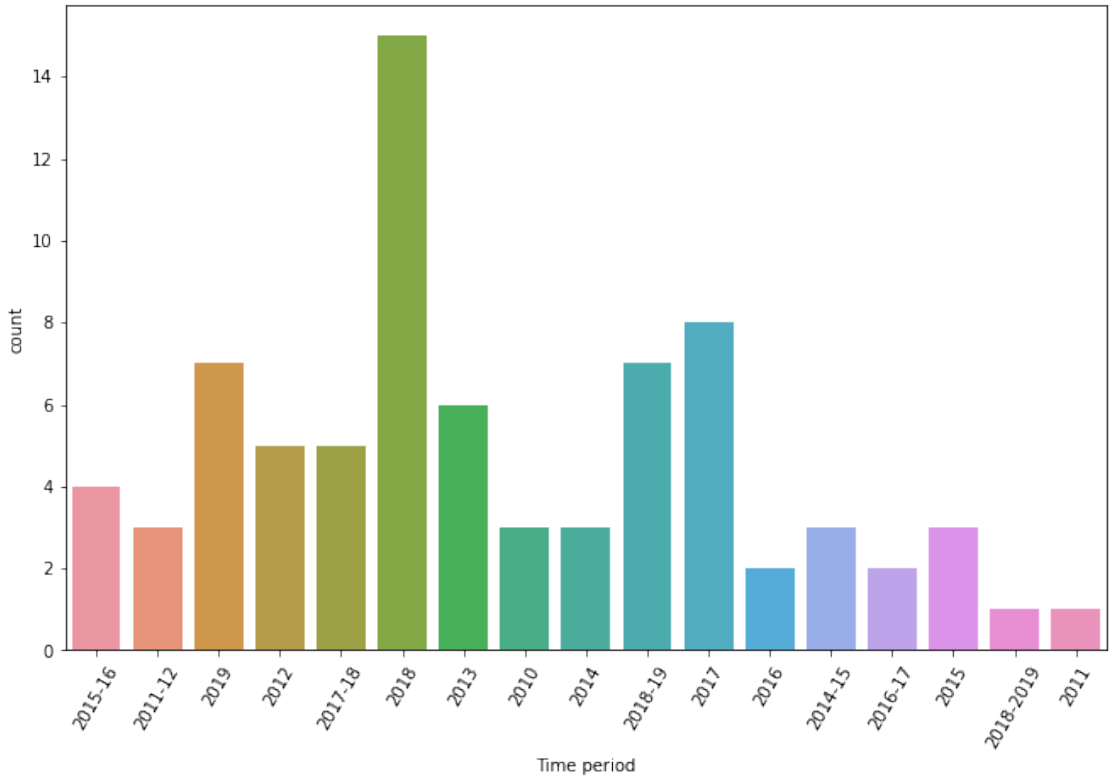
```
plt.figure(figsize=(11,7))
sns.countplot(df2['Data source'])
plt.show()
```




**Inference:**

1. The DataSource is where the Data is collected and most of the datas are collected via Multiple Indicator Cluster Survey.
2. Then we have the Demographic and Health Survey; In this 2 Data Source almost all the Datas are collected.

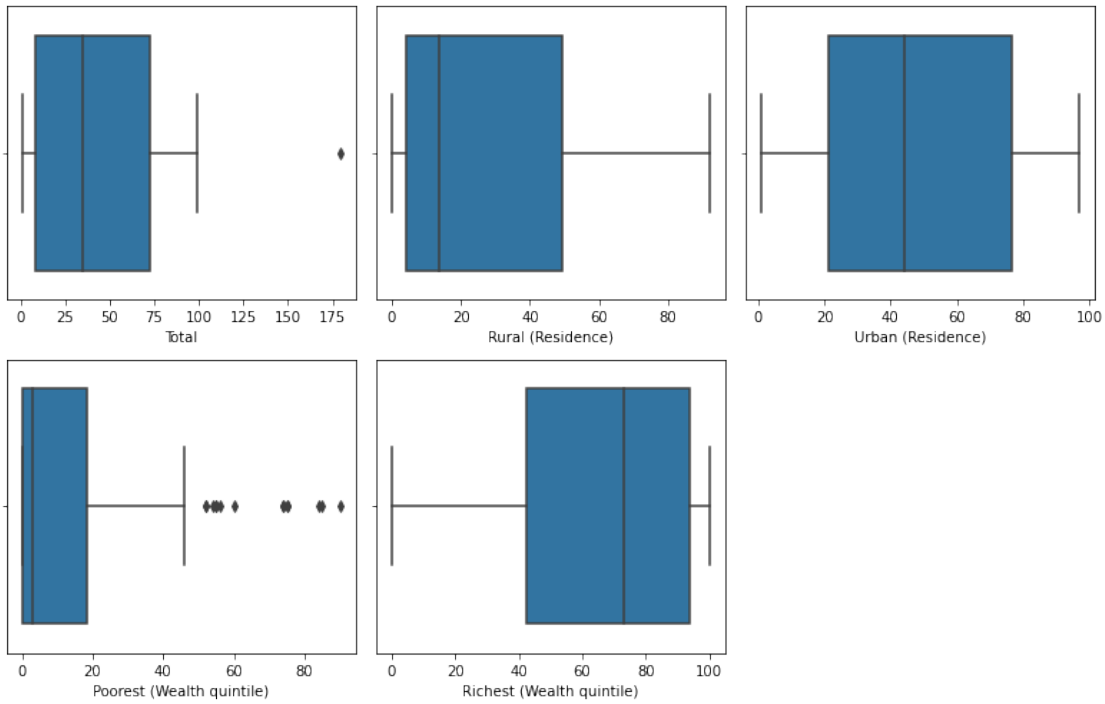
```
plt.figure(figsize=(11,7))
sns.countplot(df2['Time period'])
plt.xticks(rotation=60)
plt.show()
```




**Inference:**

1. There are many Time Period but the 2018 Time Period has the most number of the count and it is around 16+ count and then next time period is that we have 2019 and it contributes around 8+ count.
2. The least is the 2011 count and it has less than 2 count and same for the 2018-2019 Time Period also.

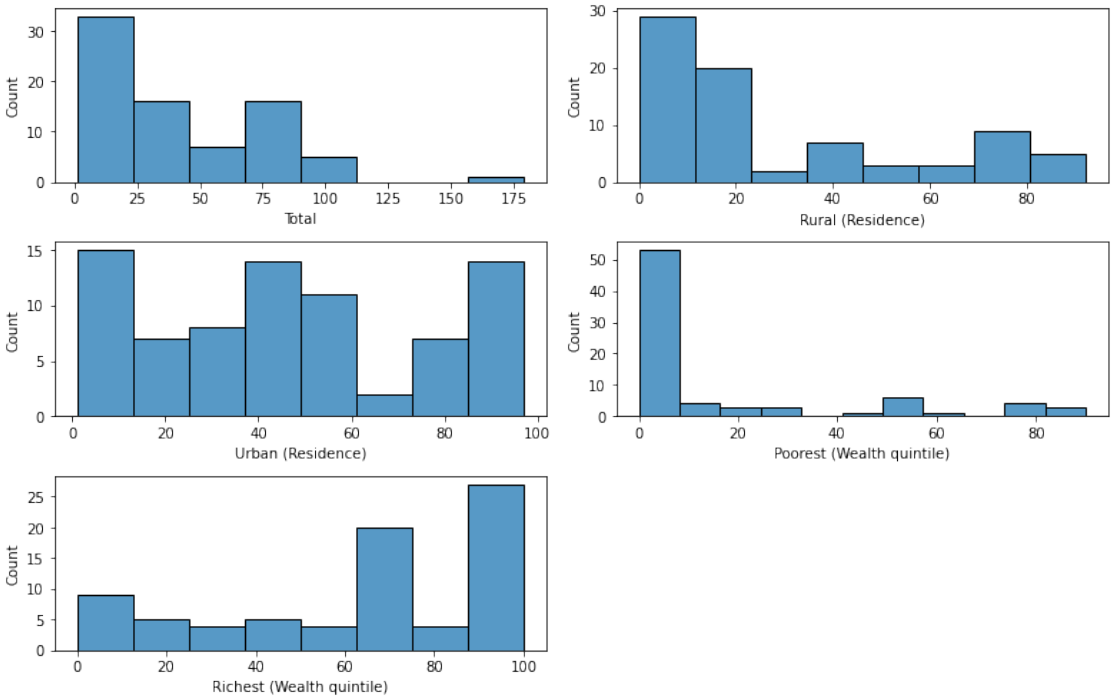
```
it=1
plt.figure(figsize=(11,7))
for i in df2.select_dtypes(include=np.number):
    plt.subplot(2,3,it)
    sns.boxplot(df2[i])
    it=it+1
plt.tight_layout()
plt.show()
```




**Inference:**

1. From the above Boxplot we can see that there is presence of outliers in the DataFrame2.
2. There are some outliers present in the Rural and Poorest we can reduce those by using IQR method but it doesn't make sense.
3. When we do the IQR some records are lost and thus we can proceed further with these Outliers.
4. They all fall in the same scale and there is no need of scaling.

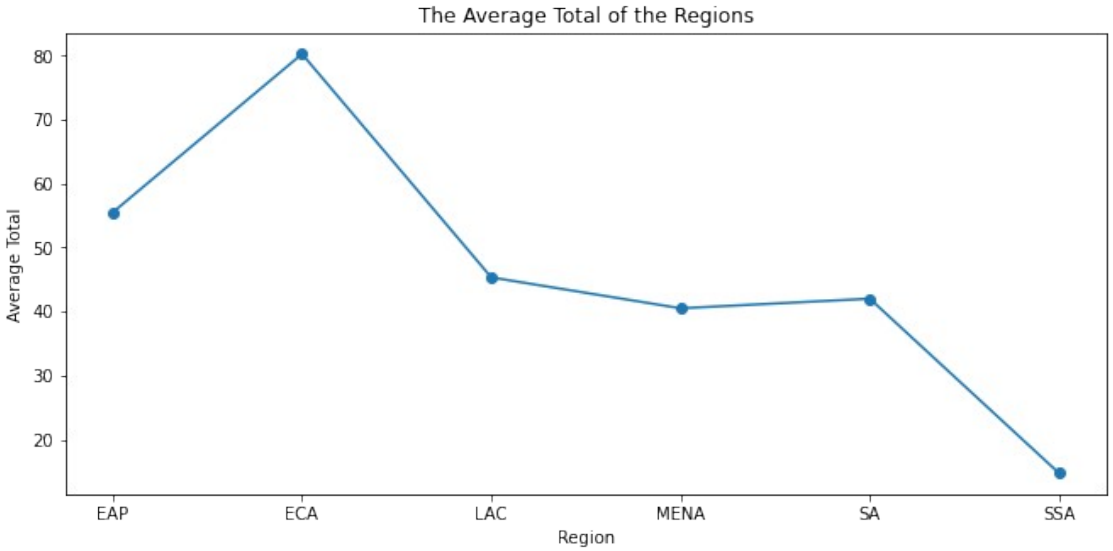
```
it=1
plt.figure(figsize=(11,7))
for i in df2.select_dtypes(include=np.number):
    plt.subplot(3,2,it)
    sns.histplot(df2[i])
    it=it+1
plt.tight_layout()
plt.show()
```




**Inference:**

1. From the Histplot we can get some inference about the spread of the values.
2. Here we can see that the all the values are spread across the range of 0-100.
3. The Poorest have a lesser count down the spread.
4. In the Richest we have value starts from lower value to the higher range.

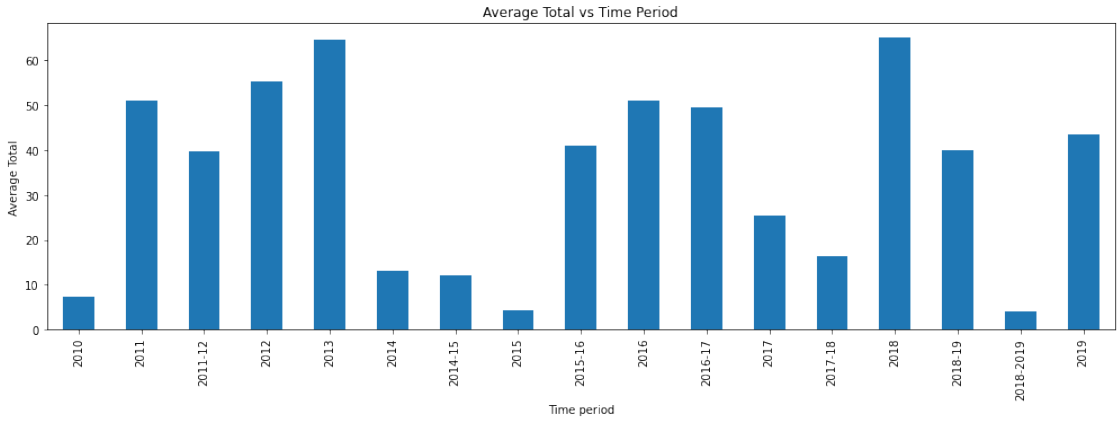
```
plt.figure(figsize=(11,5))
df2.groupby('Region')['Total'].mean().plot(kind='line',marker='o')
plt.ylabel('Average Total')
plt.title('The Average Total of the Regions')
plt.show()
```



**Inference:**

1. The value of Average total and the Region is plotted. We can see that ECA region has the Higher Total value and the least is the SSA with the total of lesser than 10.

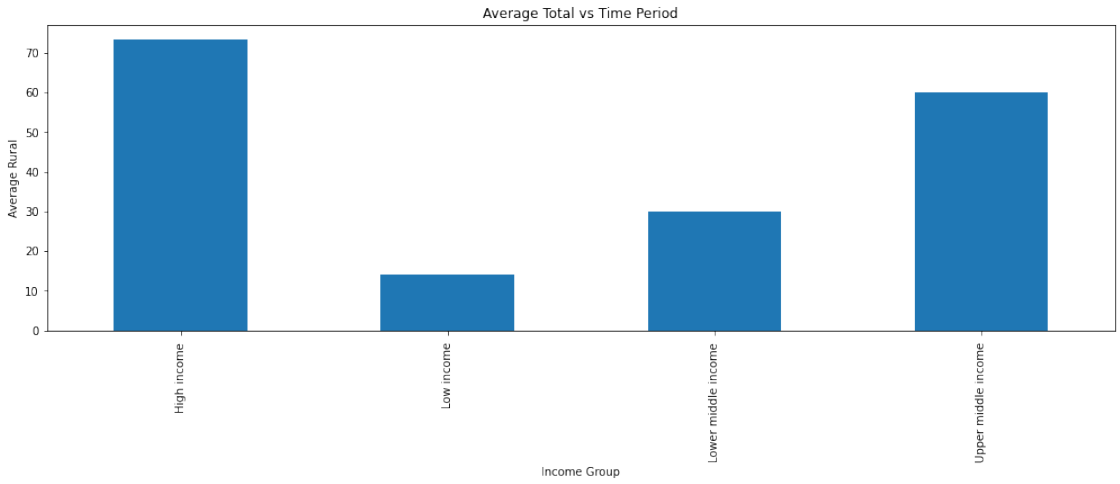
```
plt.figure(figsize=(17,5))
df2.groupby('Time period')['Total'].mean().plot(kind='bar')
plt.ylabel('Average Total')
plt.title('Average Total vs Time Period',color='black')
plt.show()
```




**Inference:**

1. The 2018 Time period is more in the Average total and therefore we can say that the contribution in 2018 is more than the other Time period.
2. The Least Time period is 2018-2019 with the total average is lesser than 10.

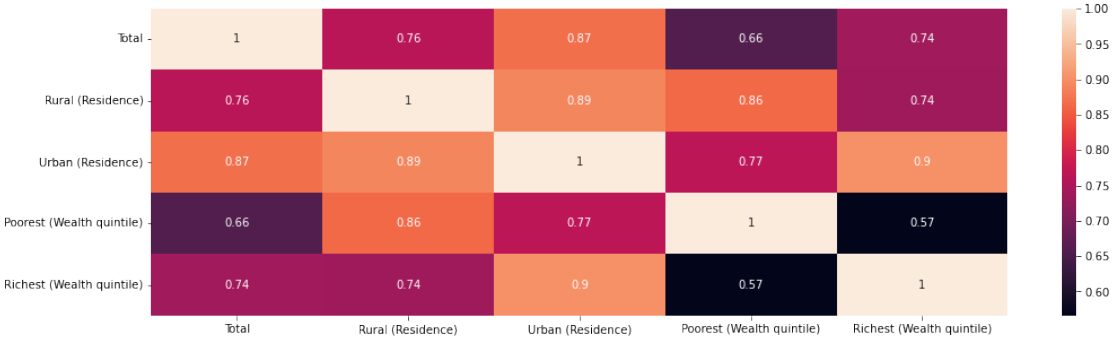
```
plt.figure(figsize=(17,5))
df2.groupby('Income Group')['Total'].mean().plot(kind='bar')
plt.ylabel('Average Rural')
plt.title('Average Total vs Time Period',color='black')
plt.show()
```




**Inference:**

1. The contribution is more for the category High Income.
2. As we known that the total will be more the High Income people and that is what seen over here.
3. The least is the Low Income and thus total is less than what we expect to have.

```
plt.figure(figsize=(17,5))
sns.heatmap(df2.corr(),annot=True)
plt.show()
```




**inference:**

1. All the variables are Highly correlated to each other.
2. This shows that there are chances of one influence the other.
3. Possible of Multicollinearity.

## EDA on DataFrame 3

```
df3.head()
```

```
  ISO3 Countries and areas Region Sub-region                Income Group
Total  \
0  DZA             Algeria   MENA       MENA  Upper middle income (UM)
24%
1  AGO              Angola    SSA        ESA  Lower middle income (LM)
17%
2  ARG           Argentina    LAC        LAC  Upper middle income (UM)
40%
3  ARM             Armenia    ECA       EECA  Upper middle income (UM)
81%
4  BGD          Bangladesh     SA         SA  Lower middle income (LM)
37%

  Rural (Residence) Urban (Residence) Poorest (Wealth quintile)  \
0                9%                32%                        1%
1                2%                24%                        0%
2               NaN               NaN                       NaN
3               71%               88%                       47%
4               33%               52%                        9%

  Richest (Wealth quintile)                      Data source Time
```

```
period
0                        77%  Multiple Indicator Cluster Survey
2018-19
1                        62%       Demographic and Health Survey
2015-16
2                        NaN  Multiple Indicator Cluster Survey
2011-12
3                        99%       Demographic and Health Survey
2015-16
4                        76%  Multiple Indicator Cluster Survey
2019
```

**Inference**

1. The above same steps are followed by for this DataFrame 3 also.
2. we will do the same stuffs as we and then we will analyse further.

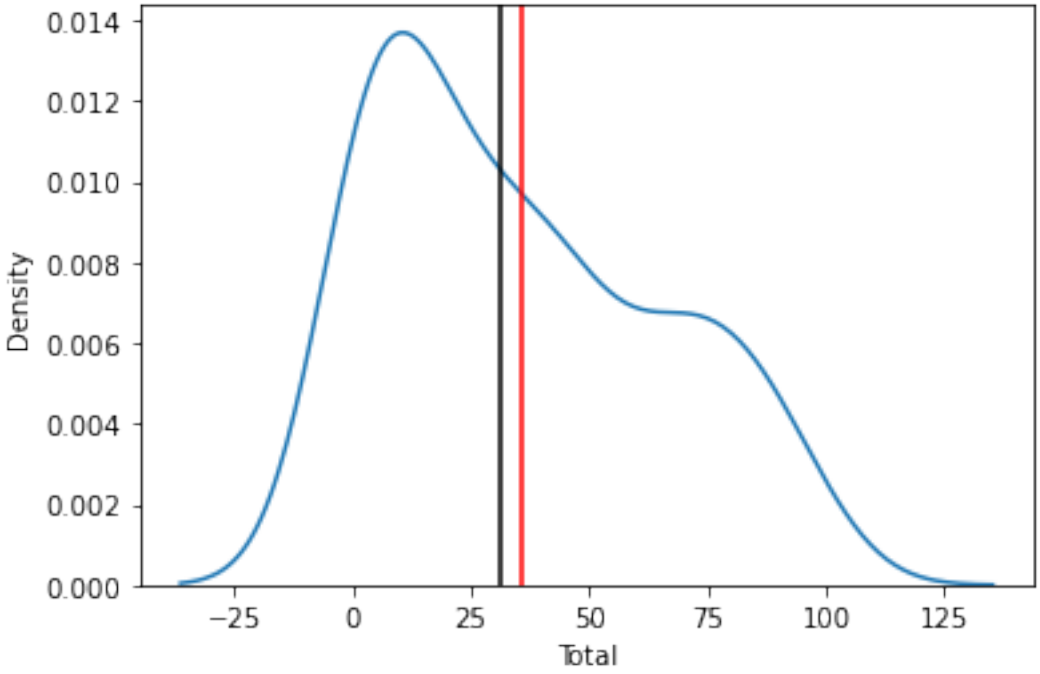
```
df3['Total'] = df3['Total'].str.split('%',expand=True)
[0].astype(float)
df3['Urban (Residence)'] = df3['Urban
(Residence)'].str.split('%',expand=True)[0].astype(float)
df3['Rural (Residence)'] = df3['Rural
(Residence)'].str.split('%',expand=True)[0].astype(float)
df3['Poorest (Wealth quintile)'] = df3['Poorest (Wealth
quintile)'].str.split('%',expand=True)[0].astype(float)
df3['Richest (Wealth quintile)'] = df3['Richest (Wealth
quintile)'].str.split('%',expand=True)[0].astype(float)
print('Categorical
Columns:',df3.select_dtypes(exclude=np.number).columns)
print('----------------------------------------------------------------
-------')
print('Numerical
Columns:',df3.select_dtypes(include=np.number).columns)
```

```
Categorical Columns: Index(['ISO3', 'Countries and areas', 'Region',
'Sub-region', 'Income Group',
       'Data source', 'Time period'],
      dtype='object')
------------------------------------------------------------------------
Numerical Columns: Index(['Total', 'Rural (Residence)', 'Urban
(Residence)',
       'Poorest (Wealth quintile)', 'Richest (Wealth quintile)'],
      dtype='object')
```

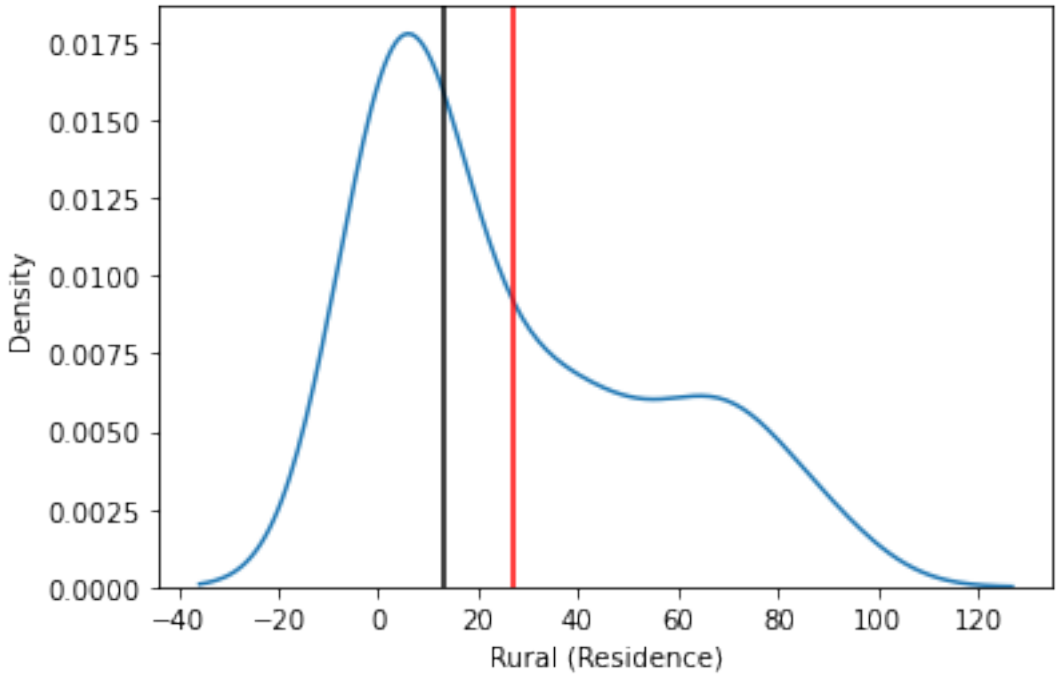
```
threshold = 3
counts = df3['Data source'].value_counts()
other_cities = counts[counts < threshold].index.tolist()
df3['Data source'] = df3['Data source'].apply(lambda x: 'Other' if x
in other_cities else x)

df3['Income Group'] = df3['Income Group'].str.split('(',expand=True)
[0]
```

```
df3.drop(columns='Countries and areas',inplace=True)
df3.drop(columns='ISO3',inplace=True)

for i in df3.select_dtypes(include=np.number):
    sns.kdeplot(x= df3[i])
    plt.axvline(df3[i].mean(),color='red')
    plt.axvline(df3[i].median(),color='black')
    print('Column Name:',i,'Skewness:',df3[i].skew())
    plt.show()
```

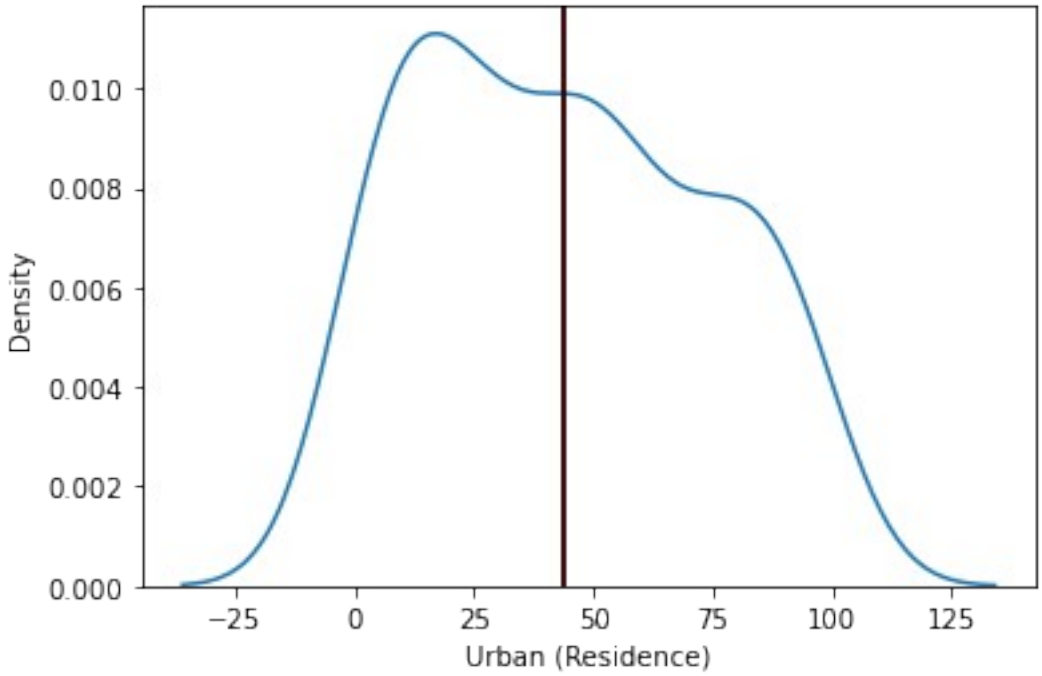
```
Column Name: Total Skewness: 0.5263535919201434
```



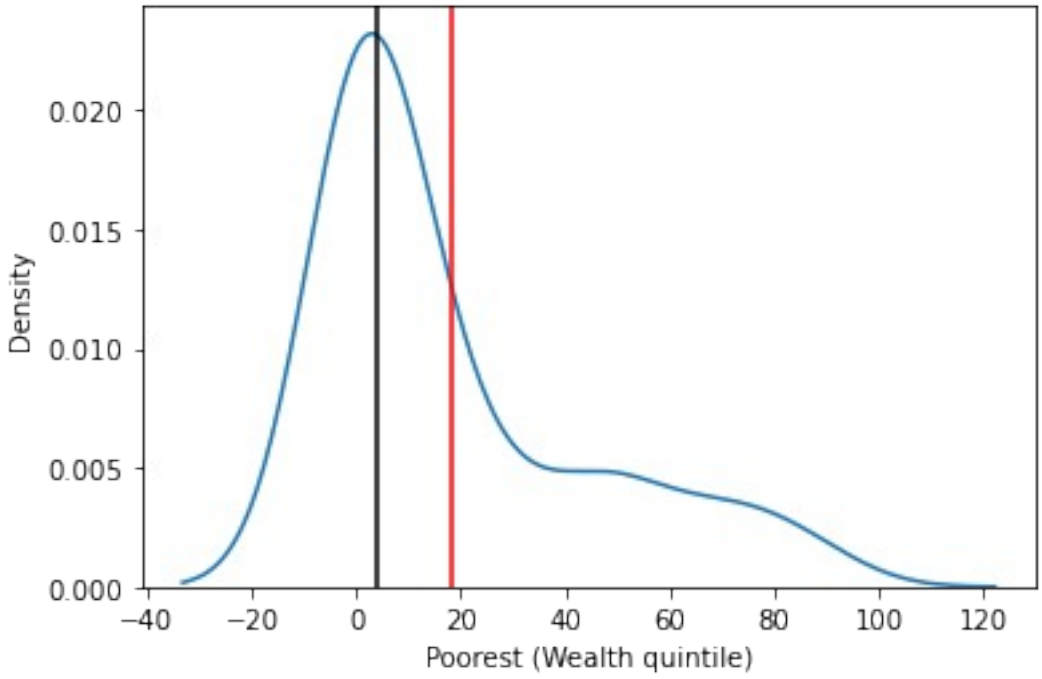

```
Column Name: Rural (Residence) Skewness: 0.8272835712084216
```



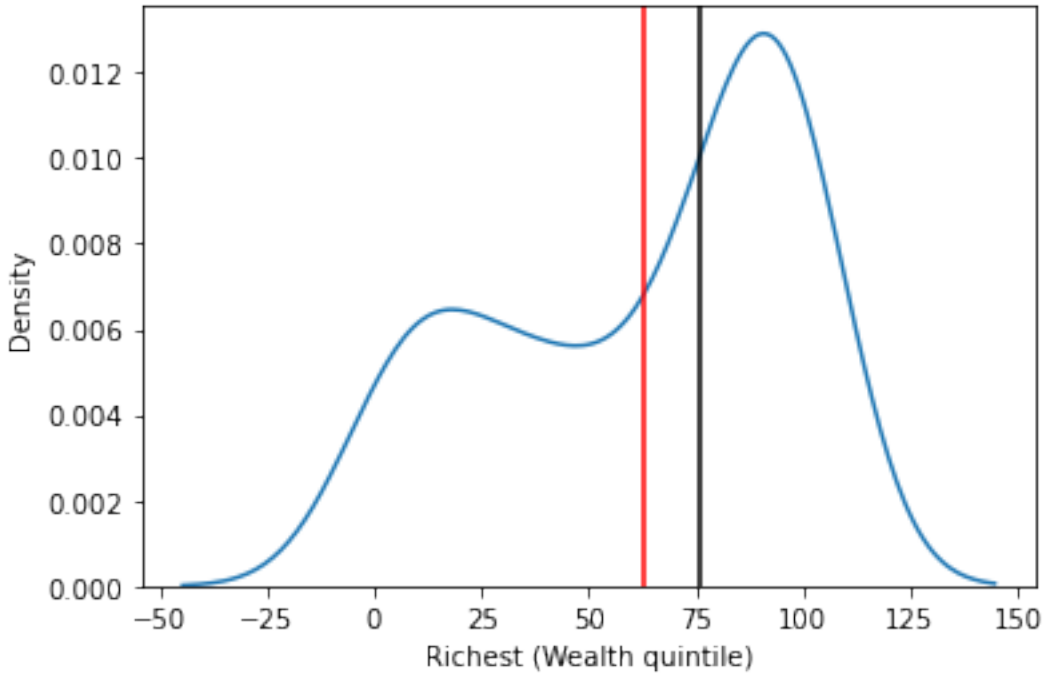

Column Name: Urban (Residence) Skewness: 0.22467473245582037




Column Name: Poorest (Wealth quintile) Skewness: 1.3567399143150374




```
Column Name: Richest (Wealth quintile) Skewness: -0.524026730732908
```




**Observation:**

1. As we can see that they all are skewed for Numerical values and we can't impute the Missing values with the Mean value.
2. The black Line represents the Median value and the Red line represents the Mean values.

3. The Line showcases us that how the variables are present in the dataset we have and we can impute the nulls via but if we impute the Nulls like this the values that are filled either be a mean or median value but this doesn't make sense.
4. So we will use groupby and then impute the Necessary values to the columns.
5. By this method we can have a better value than imputing the null values from the columns's Median or Mean.

```
df3['Urban (Residence)'] = df3.groupby('Region')['Urban
(Residence)'].fillna(df3['Urban (Residence)'].median()).values
df3['Poorest (Wealth quintile)'] = df3.groupby('Region')['Poorest
(Wealth quintile)'].fillna(df3['Poorest (Wealth
quintile)'].median()).values
df3['Richest (Wealth quintile)'] = df3.groupby('Region')['Richest
(Wealth quintile)'].fillna(df3['Richest (Wealth
quintile)'].median()).values
df3['Rural (Residence)'] = df3.groupby('Region')['Rural
(Residence)'].fillna(df3['Rural (Residence)'].median()).values
```

```
df3.head()
```

```
  Region Sub-region          Income Group  Total  Rural (Residence)  \
0   MENA       MENA  Upper middle income    24.0                9.0
1    SSA        ESA  Lower middle income    17.0                2.0
2    LAC        LAC  Upper middle income    40.0               13.0
3    ECA       EECA  Upper middle income    81.0               71.0
4     SA         SA  Lower middle income    37.0               33.0

   Urban (Residence)  Poorest (Wealth quintile)  Richest (Wealth
quintile)  \
0               32.0                        1.0
77.0
1               24.0                        0.0
62.0
2               44.0                        4.0
76.0
3               88.0                       47.0
99.0
4               52.0                        9.0
76.0

                        Data source Time period
0  Multiple Indicator Cluster Survey     2018-19
1      Demographic and Health Survey     2015-16
2  Multiple Indicator Cluster Survey     2011-12
3      Demographic and Health Survey     2015-16
4  Multiple Indicator Cluster Survey        2019
```

```
df3.isnull().sum()
```

```
Region                      0
Sub-region                  0
```

```
Income Group                 0
Total                        0
Rural (Residence)            0
Urban (Residence)            0
Poorest (Wealth quintile)    0
Richest (Wealth quintile)    0
Data source                  0
Time period                  0
dtype: int64
```

```
df3['Time period'].value_counts()
```

```
2018         16
2019          9
2018-19       8
2017          8
2013          6
2012          5
2017-18       5
2015-16       4
2015          4
2014-15       3
2014          3
2010          3
2016          3
2011-12       3
2016-17       2
2076          1
2018-2019     1
2562          1
2011          1
2012-99       1
Name: Time period, dtype: int64
```

```
df3['Time period'] = df3['Time
period'].replace({'2076':np.nan,'2562':np.nan,'2012-99':np.nan})
```

**Inference:**

1. The Time Period column has some anomly and these anomly have been treated and there are some invalied Dates and thererfore we can convert these records into null and then repalce them or drop this out.
2. There are not many records in this we can drop this off and proceed further.

```
df3.dropna(axis=0,inplace=True)
```

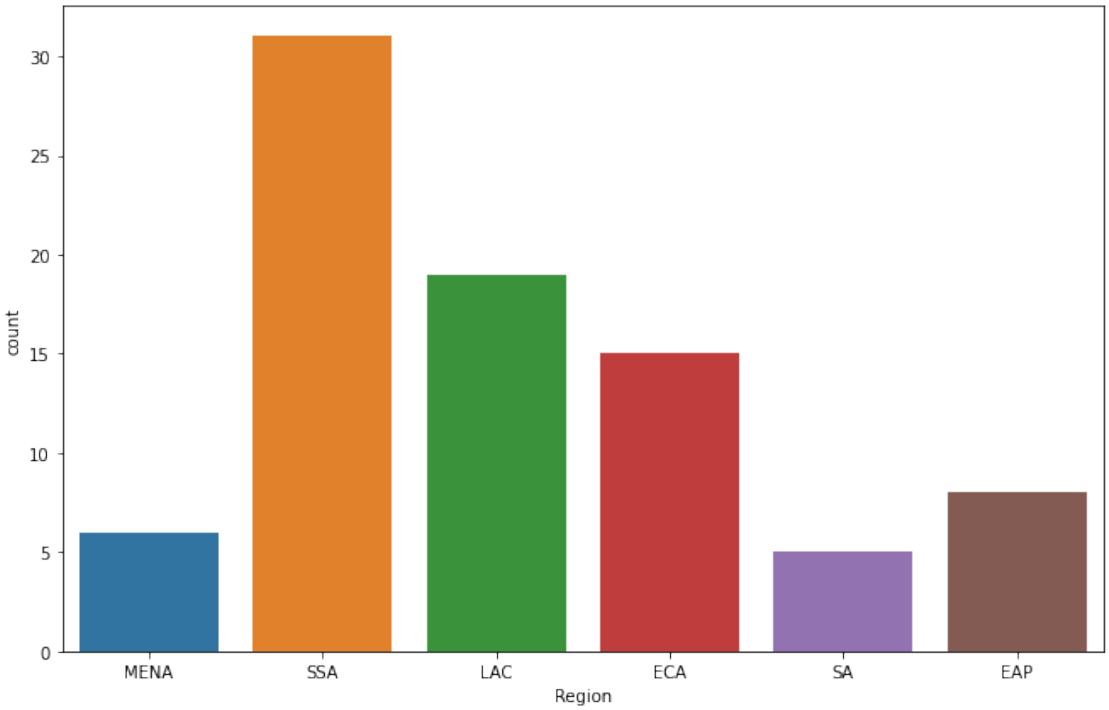
```
df3.shape
```

```
(84, 10)
```

```
df3.head()
```

```
  Region Sub-region          Income Group  Total  Rural (Residence)  \
0   MENA       MENA  Upper middle income    24.0                9.0
1    SSA        ESA  Lower middle income    17.0                2.0
2    LAC        LAC  Upper middle income    40.0               13.0
3    ECA       EECA  Upper middle income    81.0               71.0
4     SA         SA  Lower middle income    37.0               33.0

   Urban (Residence)  Poorest (Wealth quintile)  Richest (Wealth
quintile)  \
0               32.0                        1.0
77.0
1               24.0                        0.0
62.0
2               44.0                        4.0
76.0
3               88.0                       47.0
99.0
4               52.0                        9.0
76.0

                        Data source Time period
0  Multiple Indicator Cluster Survey     2018-19
1      Demographic and Health Survey     2015-16
2  Multiple Indicator Cluster Survey     2011-12
3      Demographic and Health Survey     2015-16
4  Multiple Indicator Cluster Survey        2019
```

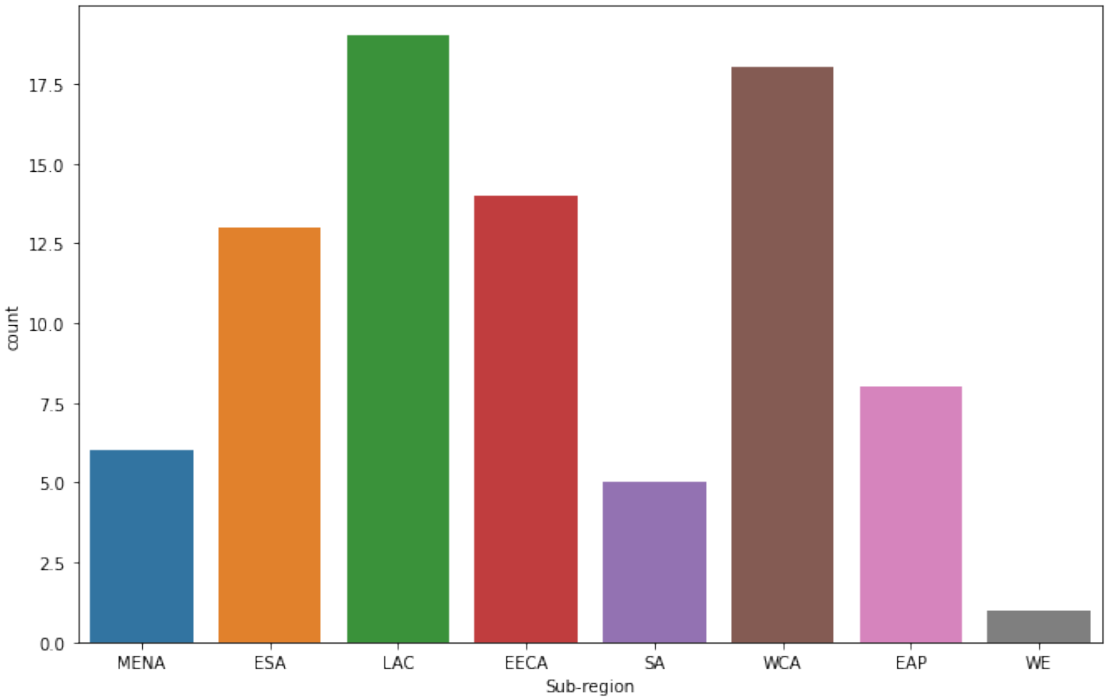
```
plt.figure(figsize=(11,7))
sns.countplot(df3['Region'])
plt.show()
```




**Inference:**

1. The SSA is the largest count and therefore we have the LAC as the second Region in this dataFrame.
2. This shows that the first Region in this df1 is the SSA which means most the members around 30 are from this and then we have LAC in the second where the count is around 20.
3. The Last Region in this dataFrame is SA with the count of 5 which indicate that least members are from SA.

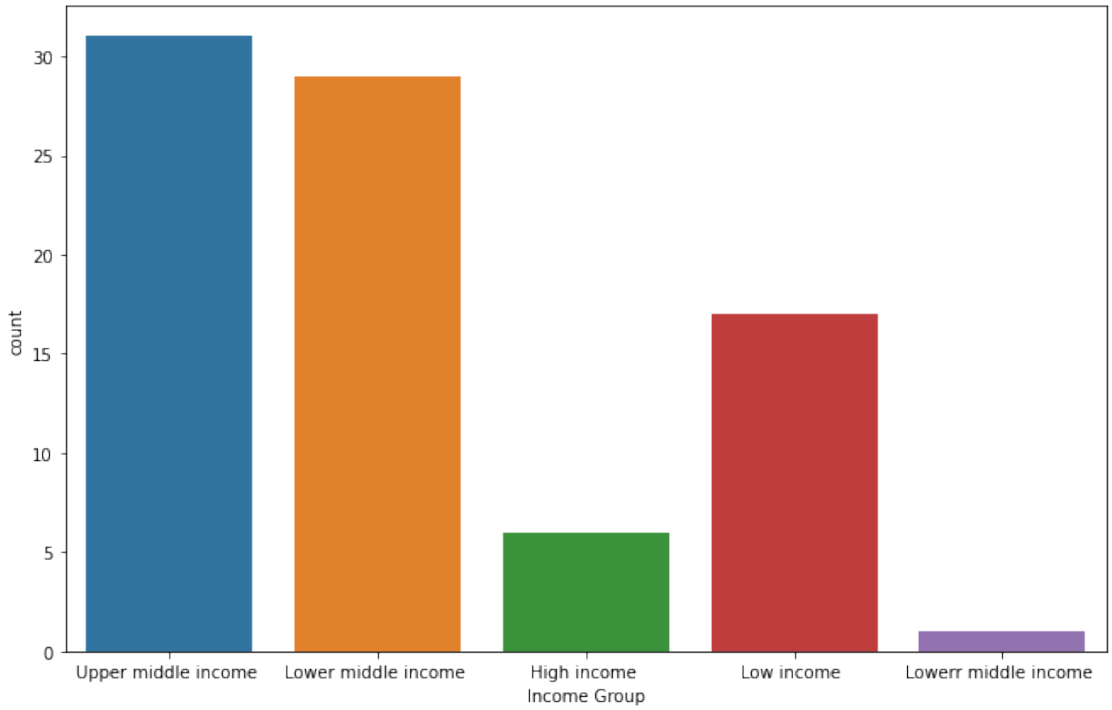
```
plt.figure(figsize=(11,7))
sns.countplot(df3['Sub-region'])
plt.show()
```




**Inference:**

1. The LAC count is more in the sub-Region category and then comes the WCA where the count is quite Lesser in terms of the LAC.
2. Then we have in the last is the WE where the count is less than 2.5 and thus we can see it via the countplot.
3. The most important Sub-Region are LAC,WCA and EECA.

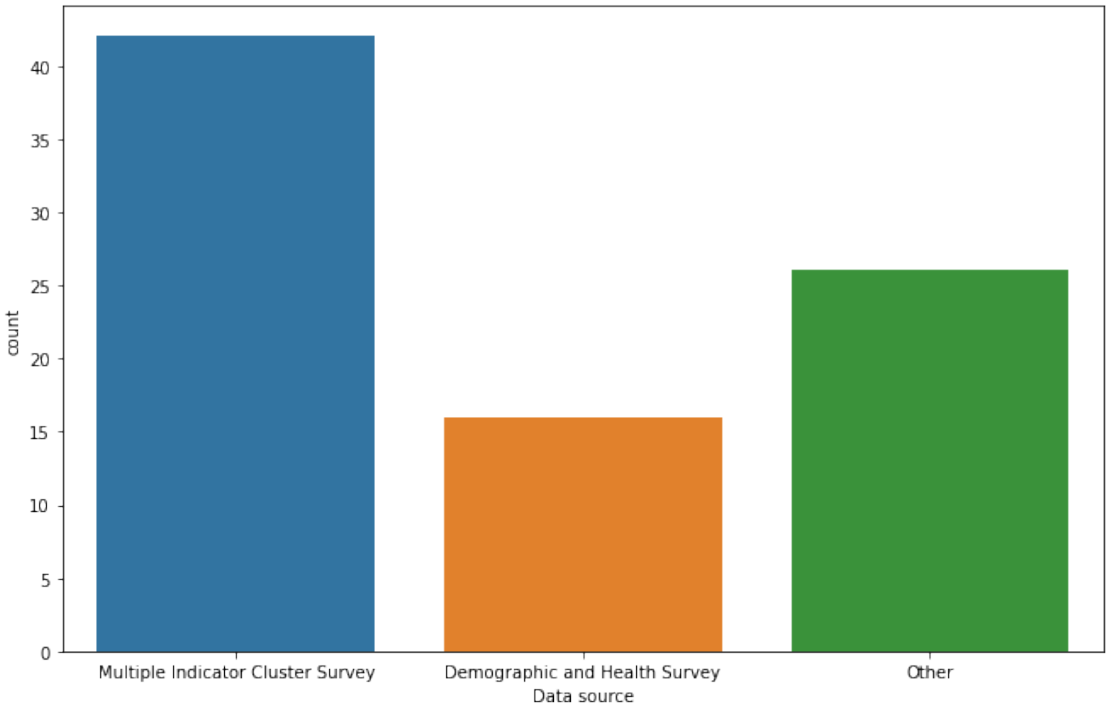
```
plt.figure(figsize=(11,7))
sns.countplot(df3['Income Group'])
plt.show()
```




**Inference:**

1. It is very suprising to see that Lower Middle Income and the Upper Middle Income are falling into the same count around 30Each.
2. Then we have Low Income in the 3rd place where the count is lesser than 20.
3. Then we have High Income category where the count is too low than other 3 category and the count is also lower than 10.
4. Therefore we can conclude that we have a DataFrame where most people are not in High Income Category.

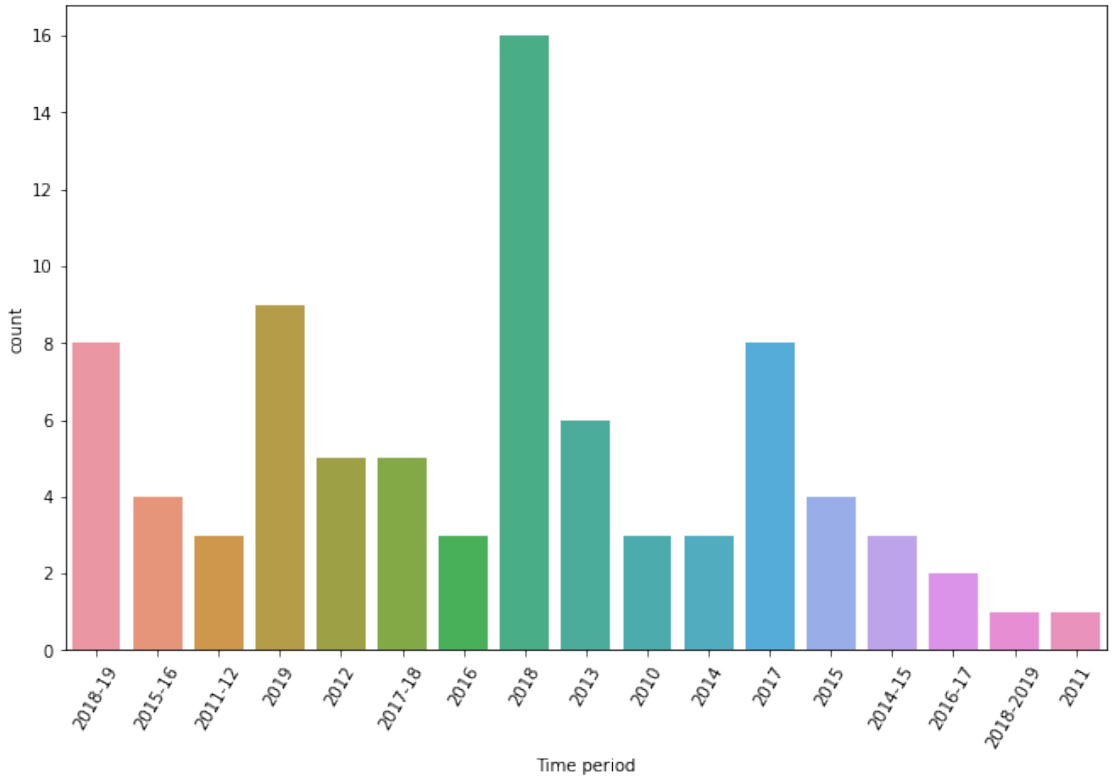
```
plt.figure(figsize=(11,7))
sns.countplot(df3['Data source'])
plt.show()
```




**Inference:**

1. The DataSource is where the Data is collected and most of the datas are collected via Multiple Indicator Cluster Survey.
2. Then we have the Demographic and Health Survey; In this 2 Data Source almost all the Datas are collected.

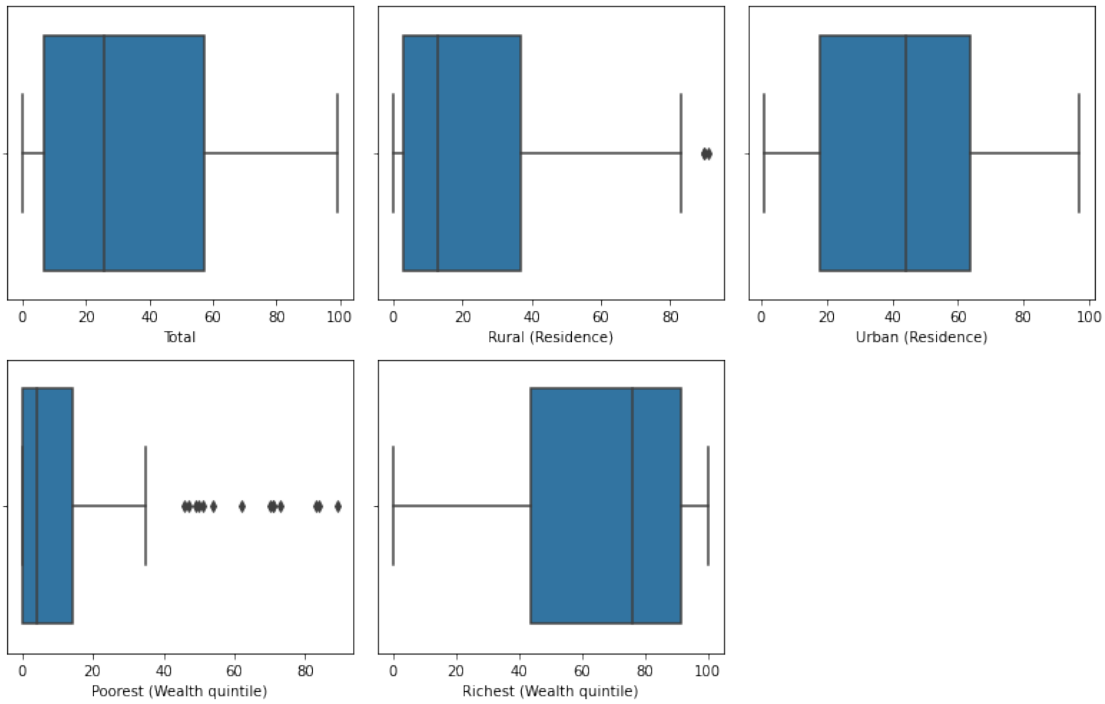
```
plt.figure(figsize=(11,7))
sns.countplot(df3['Time period'])
plt.xticks(rotation=60)
plt.show()
```




**Inference:**

1. There are many Time Period but the 2018 Time Period has the most number of the count and it is around 16+ count and then next time period is that we have 2019 and it contributes around 8+ count.
2. The least is the 2011 count and it has less than 2 count and same for the 2018-2019 Time Period also.

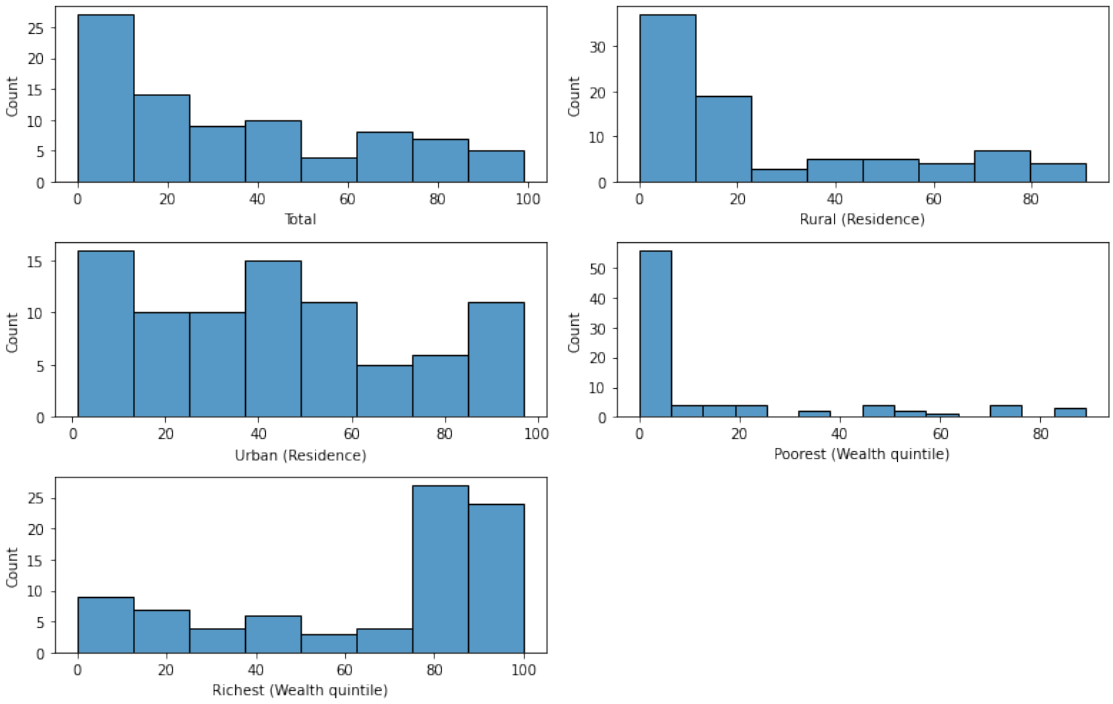
```
it=1
plt.figure(figsize=(11,7))
for i in df3.select_dtypes(include=np.number):
    plt.subplot(2,3,it)
    sns.boxplot(df3[i])
    it=it+1
plt.tight_layout()
plt.show()
```




**Inference:**

1. From the above Boxplot we can see that there is presence of outliers in the DataFrame1.
2. There are some outliers present in the Rural and Poorest we can reduce those by using IQR method but it doesn't make sense.
3. When we do the IQR some records are lost and thus we can proceed further with these Outliers.
4. They all fall in the same scale and there is no need of scaling.

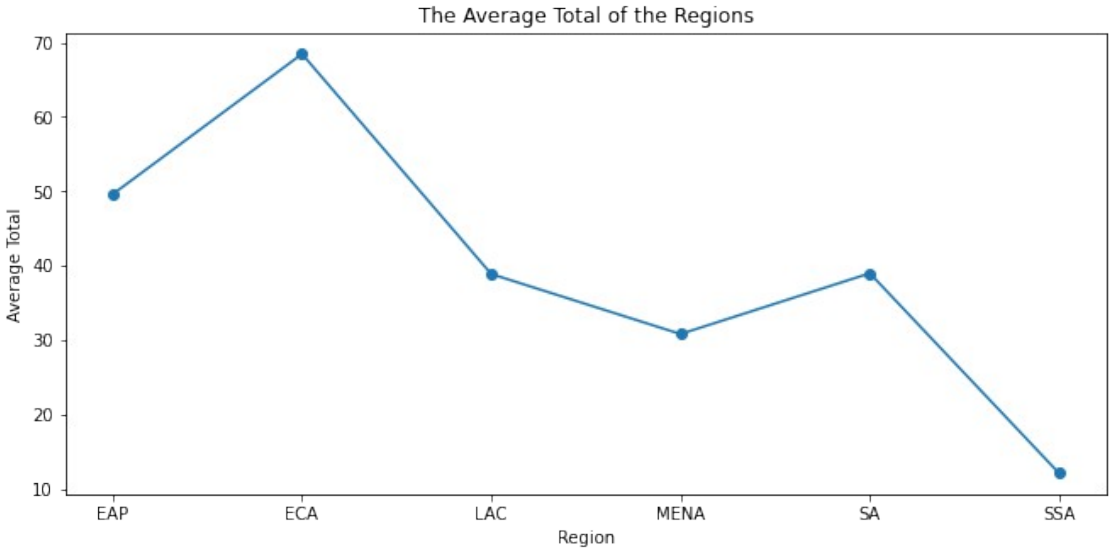
```
it=1
plt.figure(figsize=(11,7))
for i in df3.select_dtypes(include=np.number):
    plt.subplot(3,2,it)
    sns.histplot(df3[i])
    it=it+1
plt.tight_layout()
plt.show()
```




**Inference:**

1. From the Histplot we can get some inference about the spread of the values.
2. Here we can see that the all the values are spread across the range of 0-100.
3. The Poorest have a lesser count down the spread.
4. In the Richest we have value starts from lower value to the higher range.

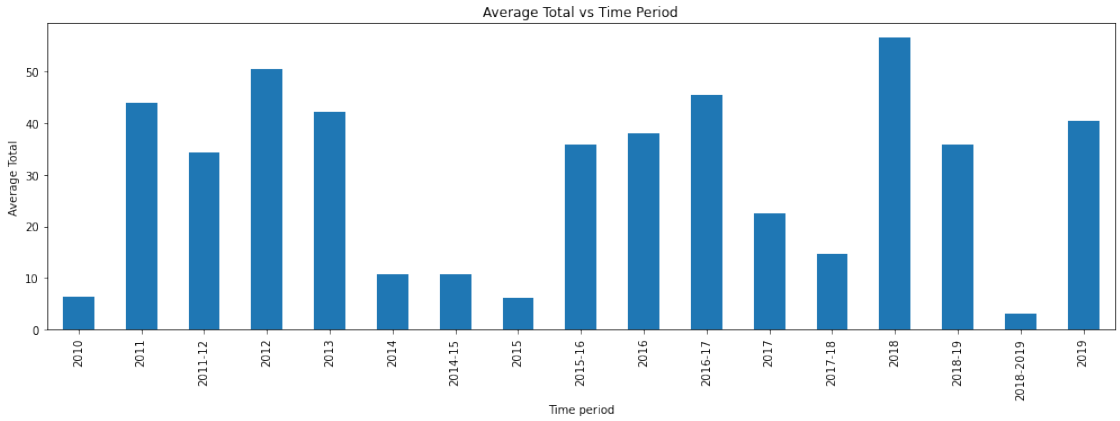
```
plt.figure(figsize=(11,5))
df3.groupby('Region')['Total'].mean().plot(kind='line',marker='o')
plt.ylabel('Average Total')
plt.title('The Average Total of the Regions')
plt.show()
```



**Inference:**

1. The value of Average total and the Region is plotted. We can see that ECA region has the Higher Total value and the least is the SSA with the total of lesser than 10.

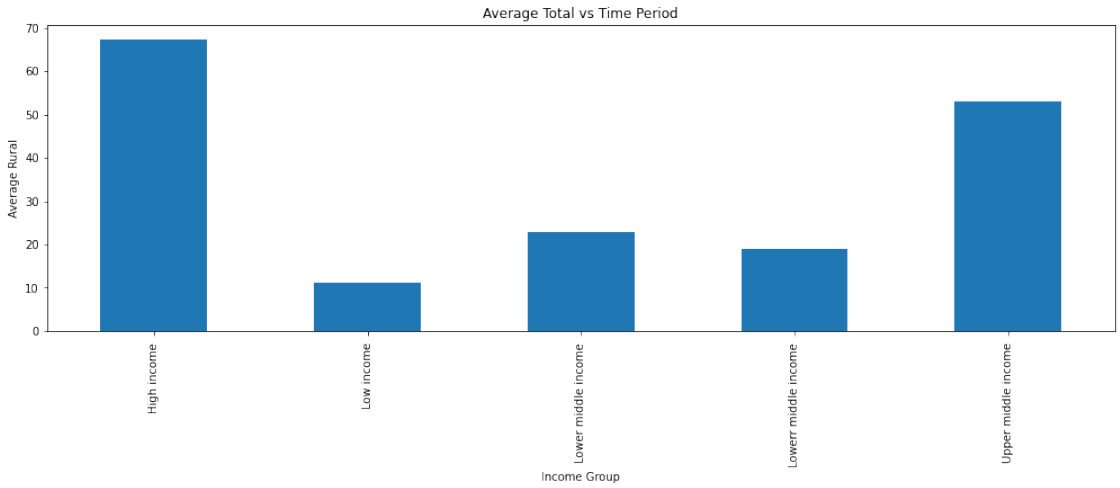
```
plt.figure(figsize=(17,5))
df3.groupby('Time period')['Total'].mean().plot(kind='bar')
plt.ylabel('Average Total')
plt.title('Average Total vs Time Period',color='black')
plt.show()
```




**Inference:**

1. The 2018 Time period is more in the Average total and therefore we can say that the contribution in 2018 is more than the other Time period.
2. The Least Time period is 2018-2019 with the total average is lesser than 10.

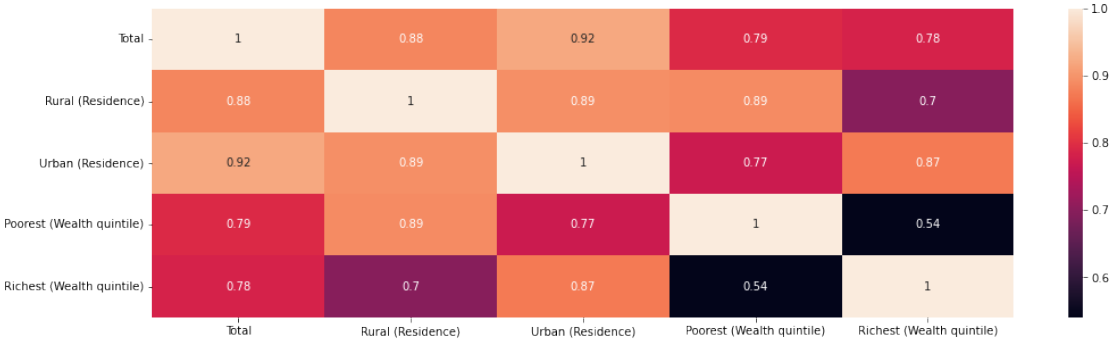
```
plt.figure(figsize=(17,5))
df3.groupby('Income Group')['Total'].mean().plot(kind='bar')
plt.ylabel('Average Rural')
plt.title('Average Total vs Time Period',color='black')
plt.show()
```




**Inference:**

1. The contribution is more for the category High Income.
2. As we known that the total will be more the High Income people and that is what seen over here.
3. The least is the Low Income and thus total is less than what we expect to have.

```
plt.figure(figsize=(17,5))
sns.heatmap(df3.corr(),annot=True)
plt.show()
```




**inference:**

1. All the variables are Highly correlated to each other.
2. This shows that there are chances of one influence the other.
3. Possible of Multicollinearity.

## Merging all the DataSets

```
merged_df = pd.concat([df1, df2, df3], axis=0)

merged_df = merged_df.reset_index(drop=True)

merged_df.shape

(245, 10)
```

**Inference:**

1. Therefore we have merged dataset and combined all the 3 DataFrame into single DataFrame.
2. Therefore in the merged DataSet we have 245 records and 10 columns

```
merged_df.tail()
```

```
    Region Sub-region           Income Group  Total  Rural (Residence)
\
240    ECA         WE            High income    99.0               13.0

241    ECA       EECA  Lowerr middle income    19.0               16.0

242    EAP        EAP   Lower middle income    62.0               13.0

243    SSA        ESA   Lower middle income     6.0                2.0
```

```
244    SSA        ESA   Lower middle income    26.0                18.0


     Urban (Residence)  Poorest (Wealth quintile)  Richest (Wealth
quintile)  \
240                44.0                         4.0
76.0
241                29.0                         1.0
69.0
242                62.0                         4.0
76.0
243                13.0                         0.0
28.0
244                49.0                         4.0
62.0

                           Data source Time period
240                              Other        2016
241                              Other        2017
242                              Other        2012
243      Demographic and Health Survey     2018-19
244  Multiple Indicator Cluster Survey     2018-19

merged_df.head()

  Region Sub-region          Income Group  Total  Rural (Residence)  \
0    SSA        ESA  Lower middle income    15.0                2.0
1    LAC        LAC  Upper middle income    39.0               13.5
2    ECA       EECA  Upper middle income    81.0               69.0
3     SA         SA  Lower middle income    34.0               30.0
4    LAC        LAC          High income    63.0               54.0

   Urban (Residence)  Poorest (Wealth quintile)  Richest (Wealth
quintile)  \
0                22.0                        0.0
61.0
1                44.0                        3.0
73.0
2                89.0                       46.0
99.0
3                49.0                        7.0
75.0
4                68.0                        9.0
97.0

                         Data source Time period
0      Demographic and Health Survey     2015-16
1  Multiple Indicator Cluster Survey     2011-12
2      Demographic and Health Survey     2015-16
```

```
3  Multiple Indicator Cluster Survey        2019
4  Multiple Indicator Cluster Survey        2012
```

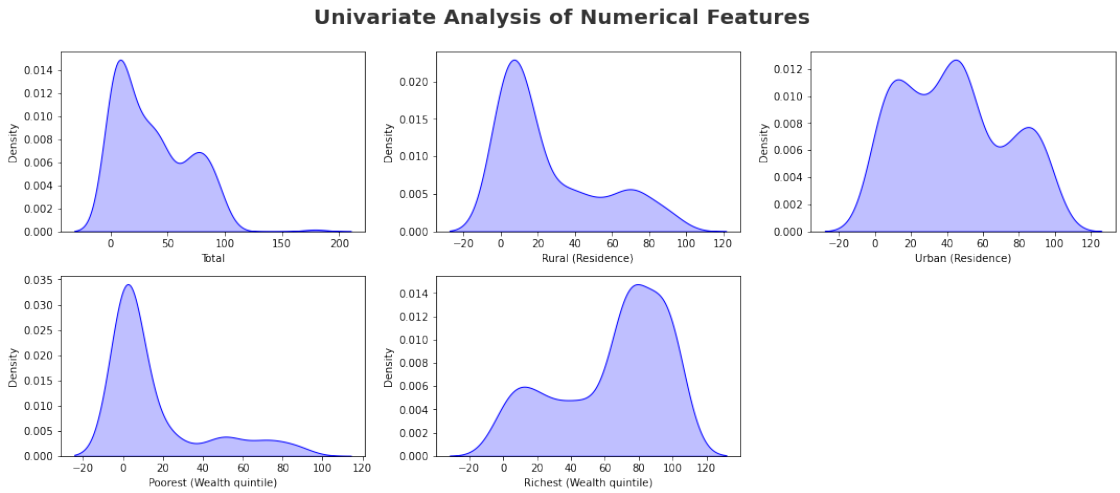
```
print(merged_df.select_dtypes(include=np.number).columns)
print('-----------------------------------------------------------')
print(merged_df.select_dtypes(exclude=np.number).columns)
```

```
Index(['Total', 'Rural (Residence)', 'Urban (Residence)',
       'Poorest (Wealth quintile)', 'Richest (Wealth quintile)'],
      dtype='object')
-----------------------------------------------------------
Index(['Region', 'Sub-region', 'Income Group', 'Data source', 'Time
period'], dtype='object')
```

**Inference:**

1. Therefore we can see that there is presence of Numerical and Categorical columns in the datasets.
2. The Numerical columns are : 'Total', 'Rural (Residence)', 'Urban (Residence)','Poorest (Wealth quintile)', 'Richest (Wealth quintile)'
3. The categorical Columns are : 'Region', 'Sub-region', 'Income Group', 'Data source', 'Time period'.

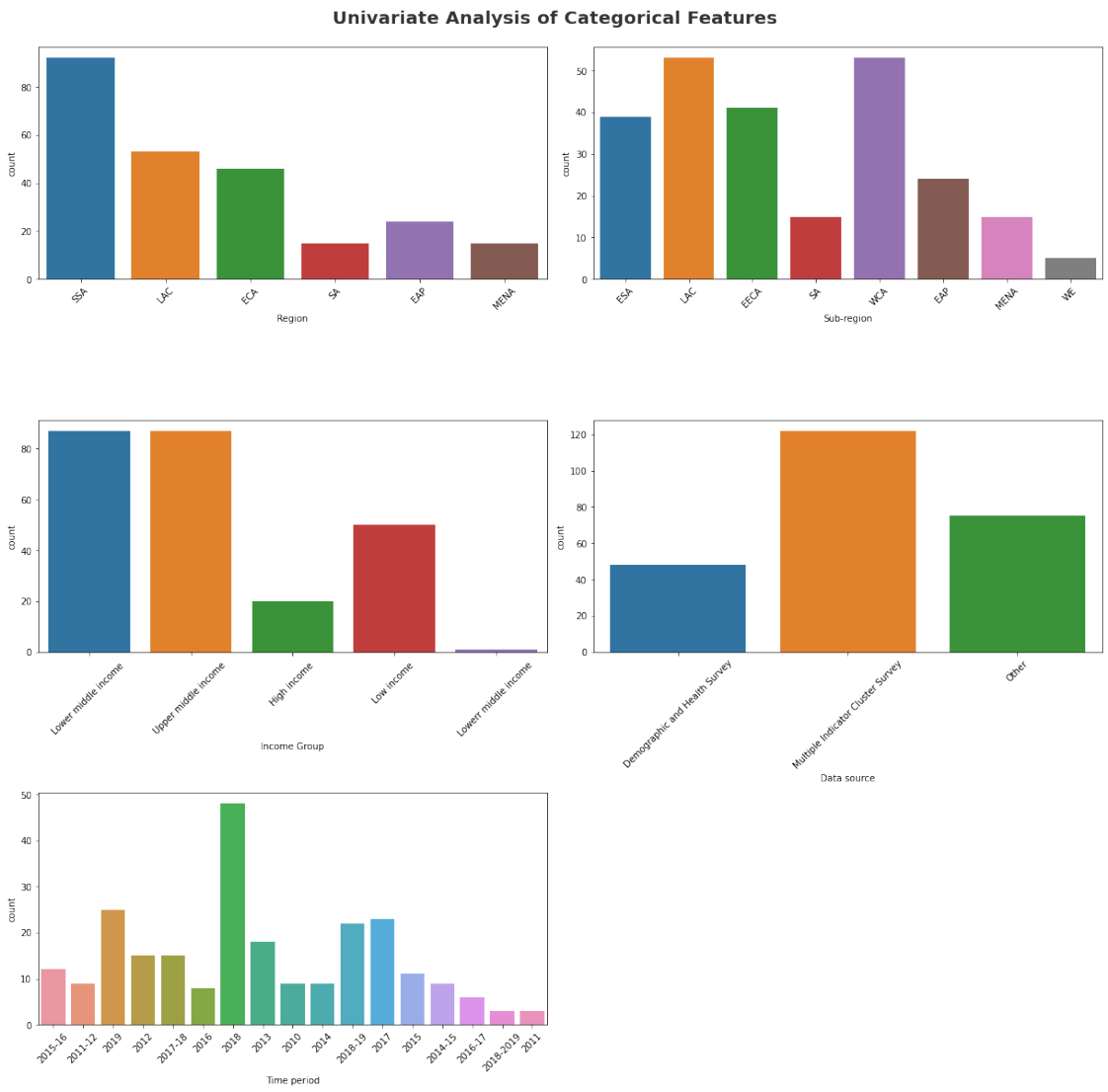
```
# define numerical & categorical columns
numeric_features = [feature for feature in merged_df.columns if
merged_df[feature].dtype != 'O']
categorical_features = [feature for feature in merged_df.columns if
merged_df[feature].dtype == 'O']
plt.figure(figsize=(15, 15))
plt.suptitle('Univariate Analysis of Numerical Features', fontsize=20,
fontweight='bold', alpha=0.8, y=1.)

for i in range(0, len(numeric_features)):
    plt.subplot(5, 3, i+1)
    sns.kdeplot(x=merged_df[numeric_features[i]],shade=True,
color='b')
    plt.xlabel(numeric_features[i])
    plt.tight_layout()
```




**Inference:**

1. Therefore We have done univariate Analysis of Numerical Features.
2. Therefore we can see that Some are right skewed and Richest alone tends to be in Left skew.
3. The merged dataset gives a overall distribution of Numerical variables.

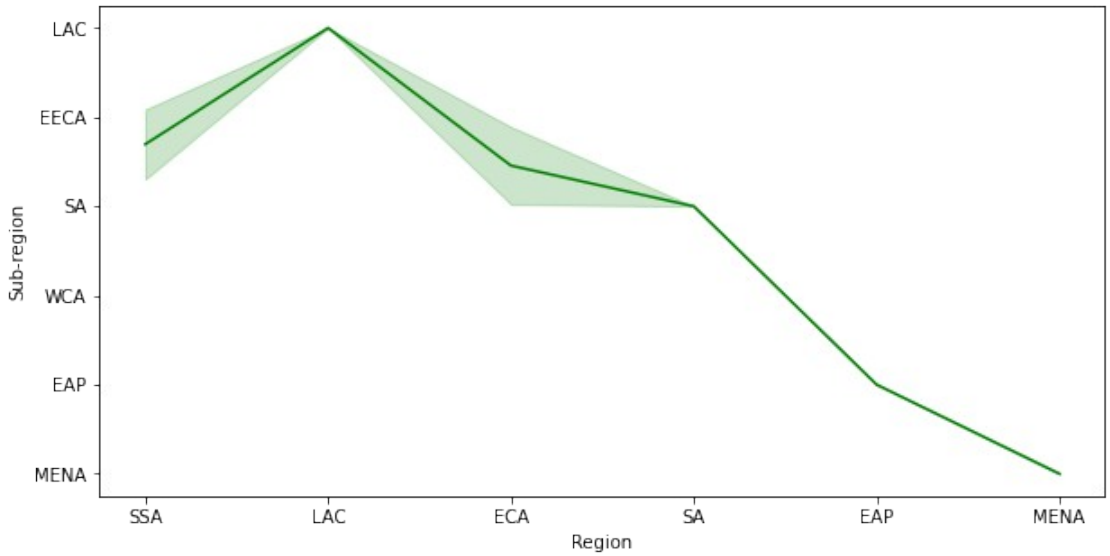
```
# categorical columns
plt.figure(figsize=(17, 27))
plt.suptitle('Univariate Analysis of Categorical Features',
fontsize=20, fontweight='bold', alpha=0.8, y=1.)
for i in range(0, len(categorical_features)):
    plt.subplot(5, 2, i+1)
    sns.countplot(x=merged_df[categorical_features[i]])
    plt.xlabel(categorical_features[i])
    plt.xticks(rotation=45)
    plt.tight_layout()
```




## Inference:

1. Therefore We have done univariate Analysis of Categorical Features.
2. Therefore we can see that count of region,sub-Region and other category based columns.
3. The merged dataset gives a overall Count/Frequency of Categorical variables.

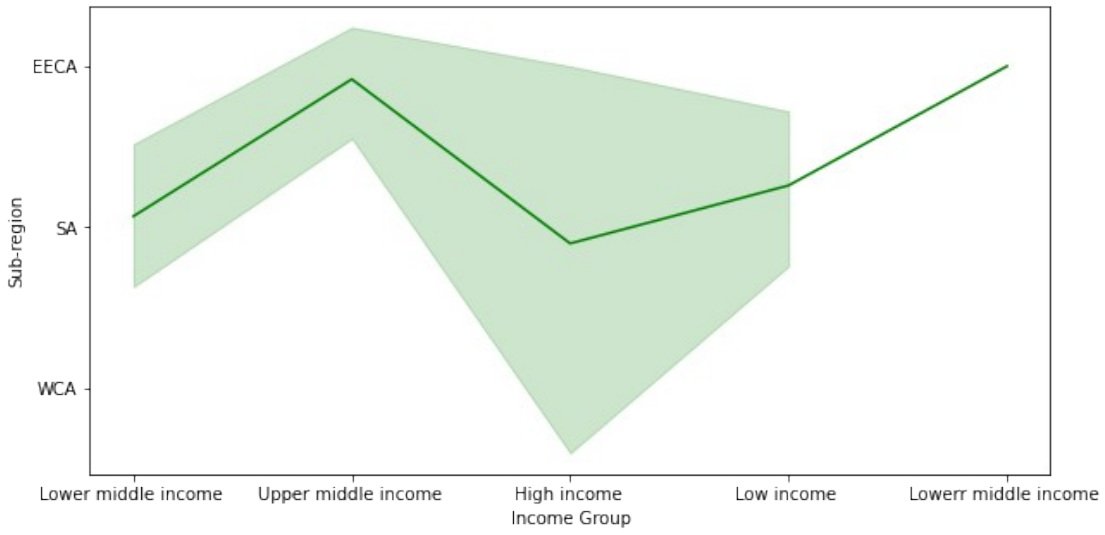
```
plt.subplots(figsize=(10,5))
sns.lineplot(x='Region',y='Sub-region',data=merged_df,color='g')
plt.show()
```




**Inference:**

1. The plot displays the information about the 2 categorical features that is Region and Sub-Region.
2. This shows that how the region and sub-region are related to each other and therefore we can there is Quite a higher peak in the front and then decrease drastically.

```
plt.subplots(figsize=(10,5))
sns.lineplot(x='Income Group',y='Sub-region',data=merged_df,color='g')
plt.show()
```

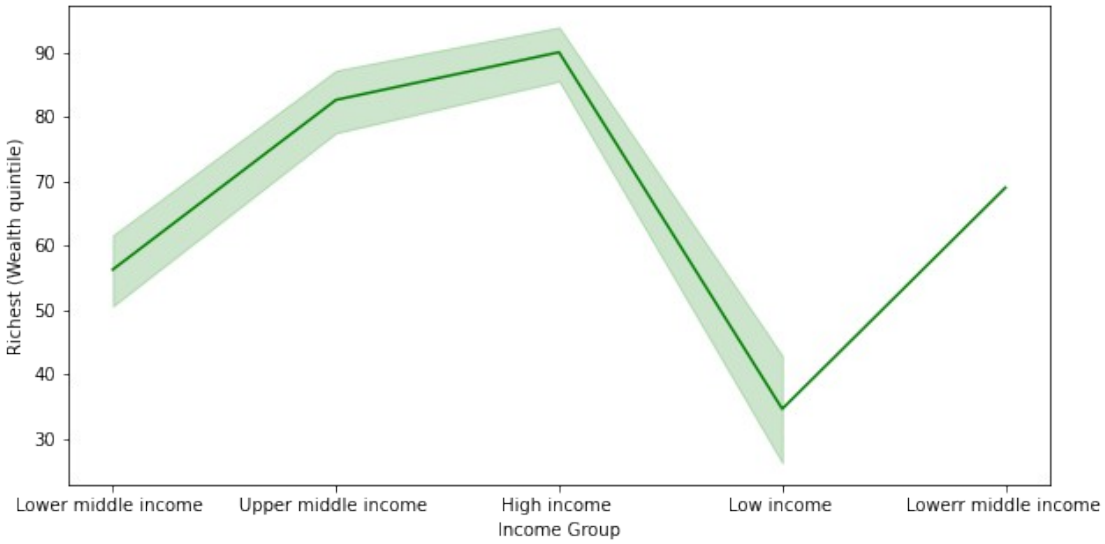



**Inference:**

1. The plot displays the information about the 2 categorical features that is Income Group and Sub-Region.

2. This shows that how the Income Group and sub-region are related to each other and therefore we can there is Quite a higher peak in the front and then decrease drastically and then there is sudden peak.
3. There is lesser count for the High Income Group.
4. The count for the upper Middle Income and Lower Middle Income are more.

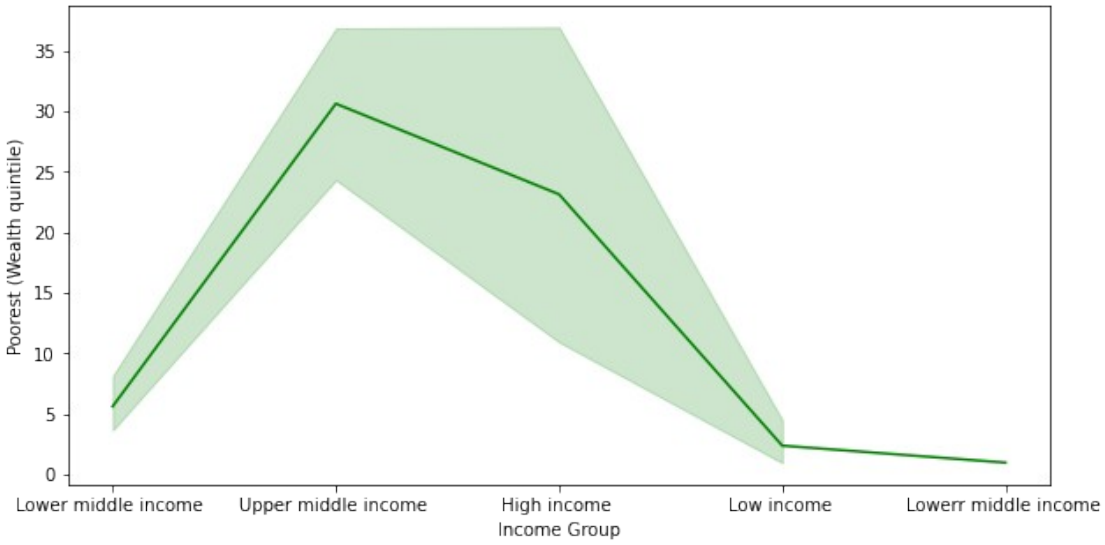
```
plt.subplots(figsize=(10,5))
sns.lineplot(x='Income Group',y='Richest (Wealth
quintile)',data=merged_df,color='g')
plt.show()
```




**Inference:**

1. The plot displays the information about the 1 categorical features that is Income Group and 1 Numerical Feature Richest.
2. The Richest has more count in the High Income and then it drops for the Low Income Category.
3. The Lower Income and Lower Middle Income are in the same portion of the Richest (Wealth Quantile)

```
plt.subplots(figsize=(10,5))
sns.lineplot(x='Income Group',y='Poorest (Wealth
quintile)',data=merged_df,color='g')
plt.show()
```




**Inference:**

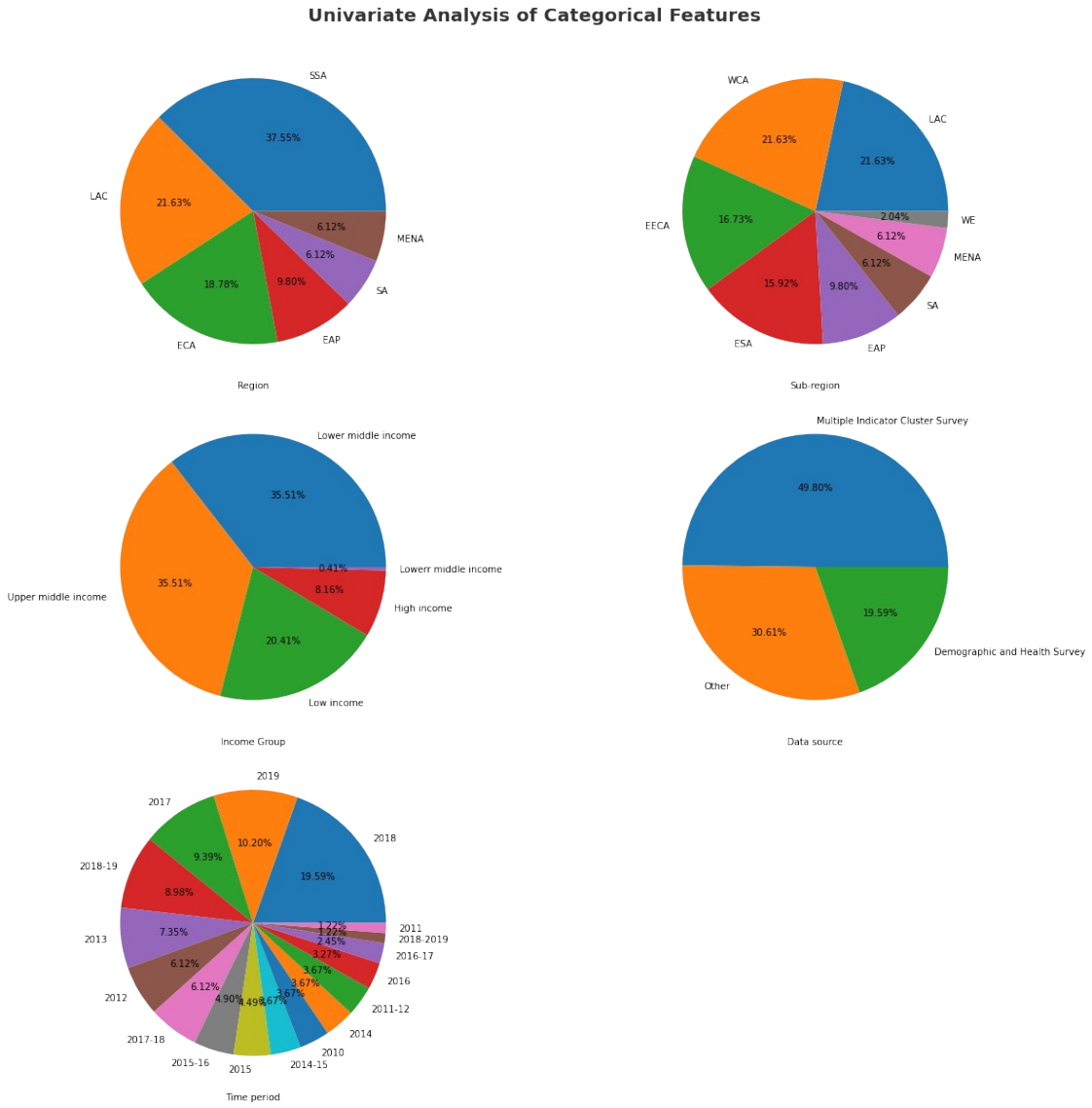
1. The plot displays the information about the 1 categorical features that is Income Group and 1 Numerical Feature Poorest.
2. The Poorest has more count in the Upper Middle Income and then it drops for the Low Income Category and even get worsed in the Lower Middle Income.
3. The Lower Income and Lower Middle Income are in the same portion of the Poorest (Wealth Quantile)
4. The Upper Middle Income are most in this case.

```
merged_df.head()
```

```
  Region Sub-region          Income Group  Total  Rural (Residence)  \
0    SSA        ESA  Lower middle income    15.0                2.0
1    LAC        LAC  Upper middle income    39.0               13.5
2    ECA       EECA  Upper middle income    81.0               69.0
3     SA         SA  Lower middle income    34.0               30.0
4    LAC        LAC          High income    63.0               54.0

   Urban (Residence)  Poorest (Wealth quintile)  Richest (Wealth
quintile)  \
0               22.0                        0.0
61.0
1               44.0                        3.0
73.0
2               89.0                       46.0
99.0
3               49.0                        7.0
75.0
4               68.0                        9.0
97.0

                        Data source Time period
```

```
0      Demographic and Health Survey     2015-16
1  Multiple Indicator Cluster Survey     2011-12
2      Demographic and Health Survey     2015-16
3  Multiple Indicator Cluster Survey        2019
4  Multiple Indicator Cluster Survey        2012
```

```
plt.figure(figsize=(17, 27))
plt.suptitle('Univariate Analysis of Categorical Features',
fontsize=20, fontweight='bold', alpha=0.8, y=1.)
for i in range(0, len(categorical_features)):
    plt.subplot(5, 2, i+1)

plt.pie(merged_df[categorical_features[i]].value_counts(),labels=merge
d_df[categorical_features[i]].value_counts().index,autopct='%.2f%%')
    plt.xlabel(categorical_features[i])
    plt.xticks(rotation=45)
    plt.tight_layout()
```

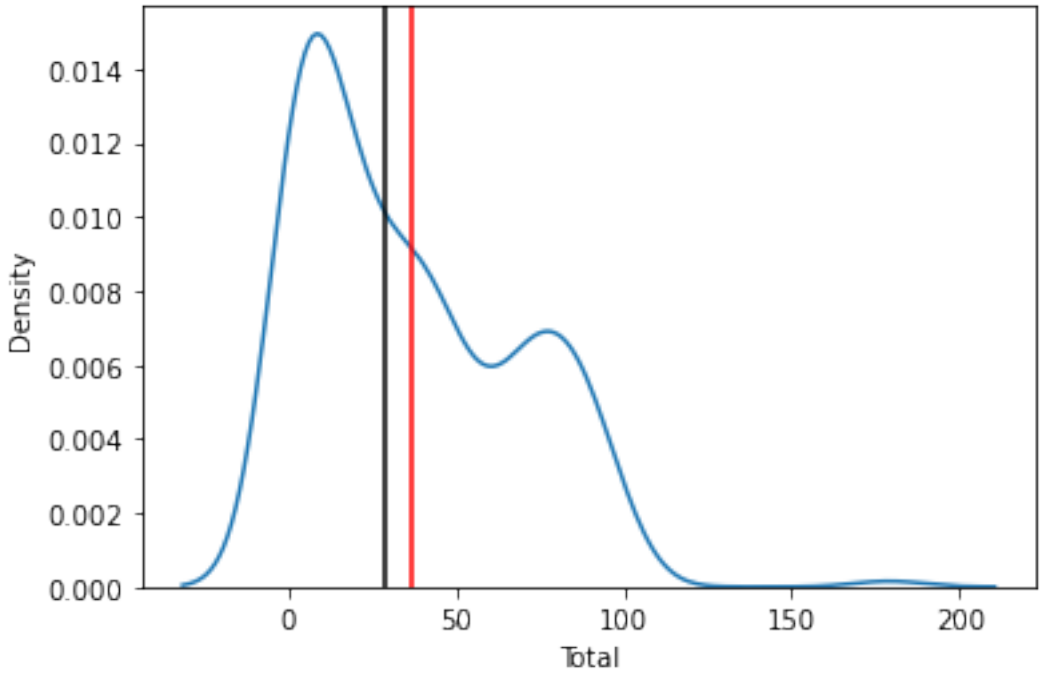



**Inference:**

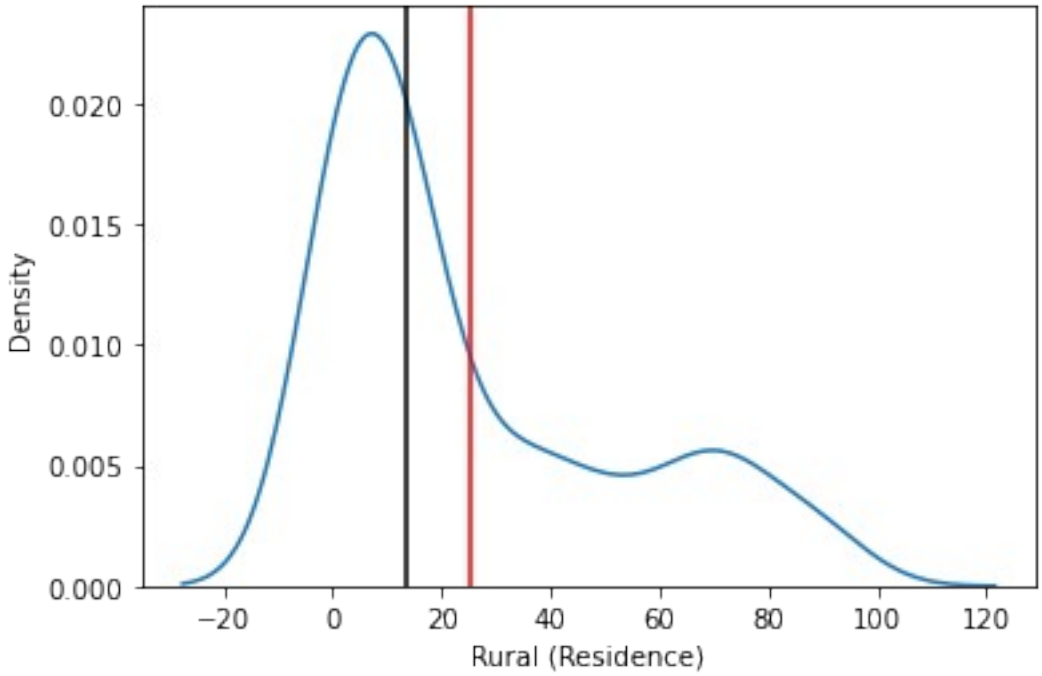
1. Therefore we have categorical columns with their percentage of the contribution.
2. This piechart shows us that In Region SSA is more than 37% and In case Income Group we can see that upper Middle Income and Lower Middle Income are similar in their respective Percentage.

```
for i in merged_df.select_dtypes(include=np.number):
    sns.kdeplot(x= merged_df[i])
    plt.axvline(merged_df[i].mean(),color='red')
    plt.axvline(merged_df[i].median(),color='black')
    print('Column Name:',i,'Skewness:',merged_df[i].skew())
    plt.show()
```

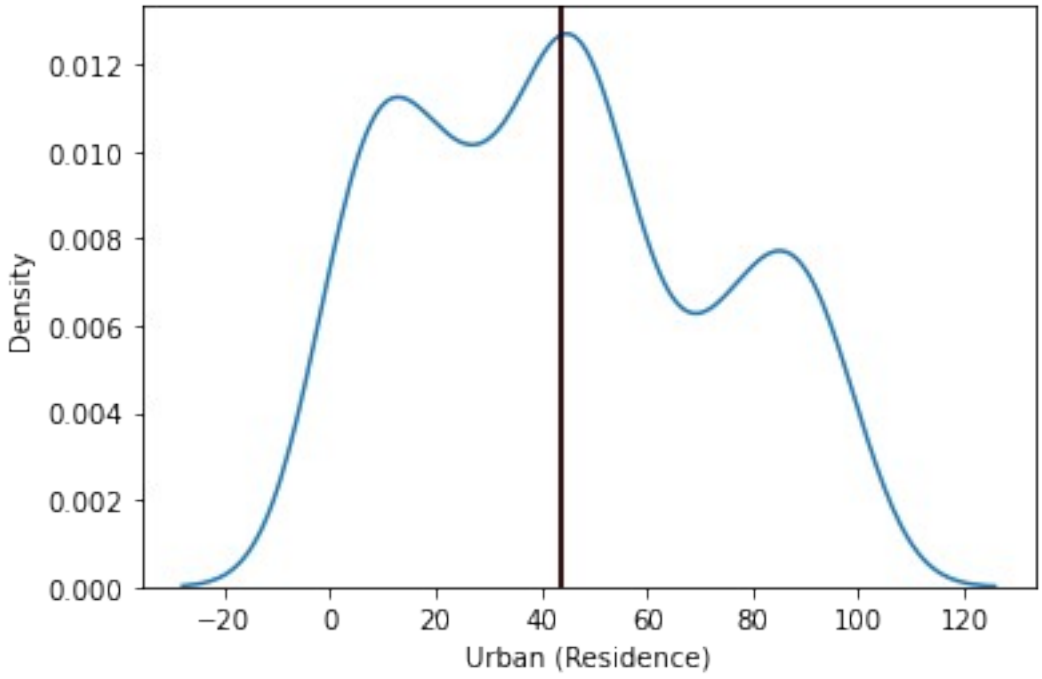
```
Column Name: Total Skewness: 0.8033210884474662
```



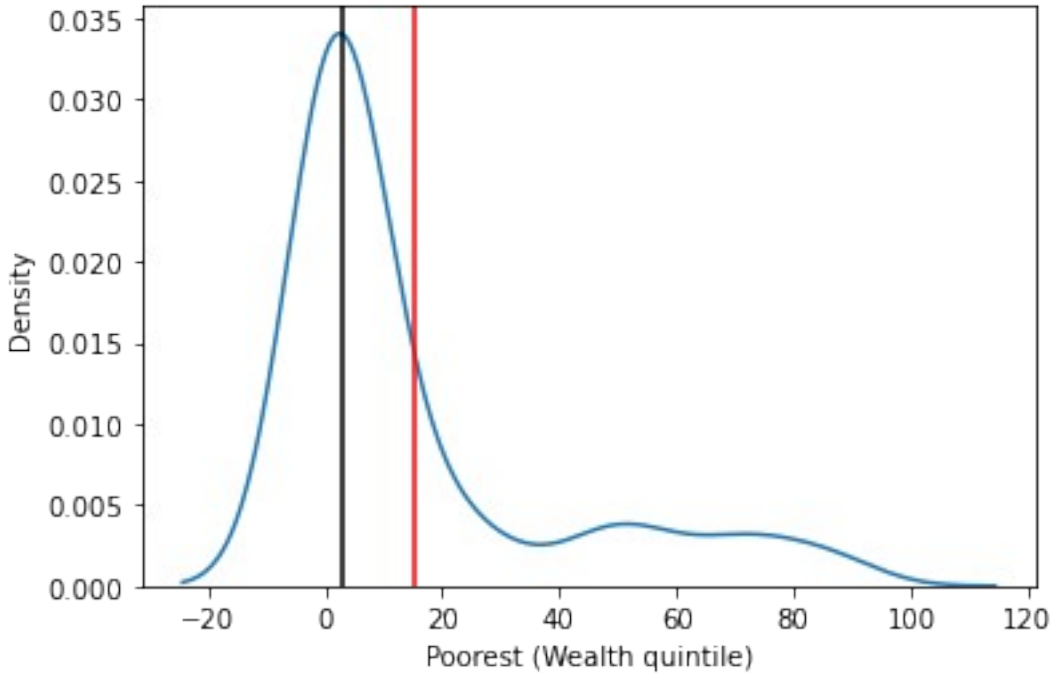

Column Name: Rural (Residence) Skewness: 1.0333740029855834



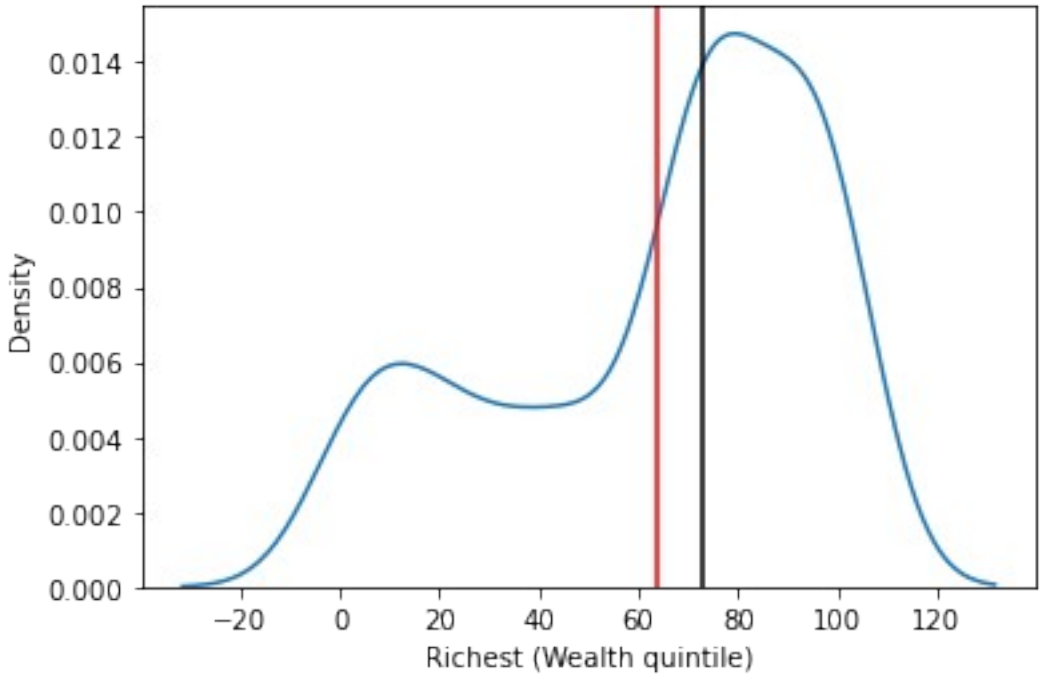

Column Name: Urban (Residence) Skewness: 0.2689144778739272




Column Name: Poorest (Wealth quintile) Skewness: 1.7224357933981174



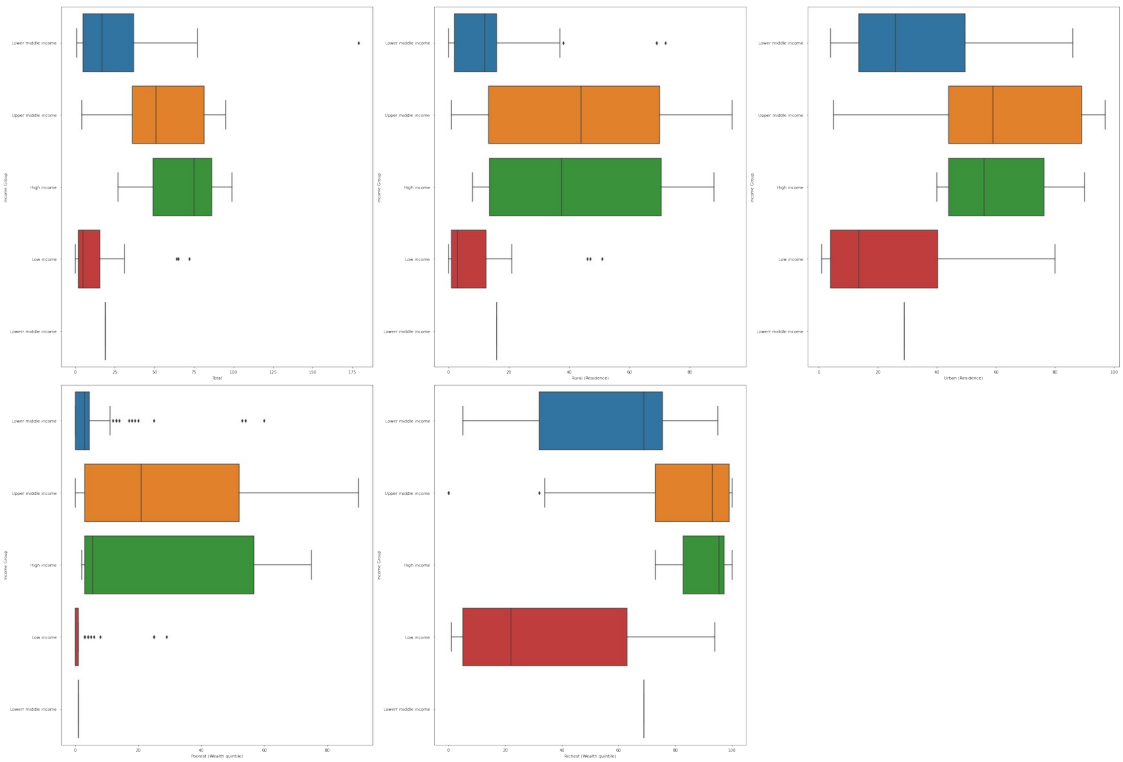

Column Name: Richest (Wealth quintile) Skewness: -0.7018318093887127




**Inference:**

1. Therefore we check the Overall Skewness for the Merged DataSet.
2. There are some skewness Present and therefore we can say we will do the required transformation.

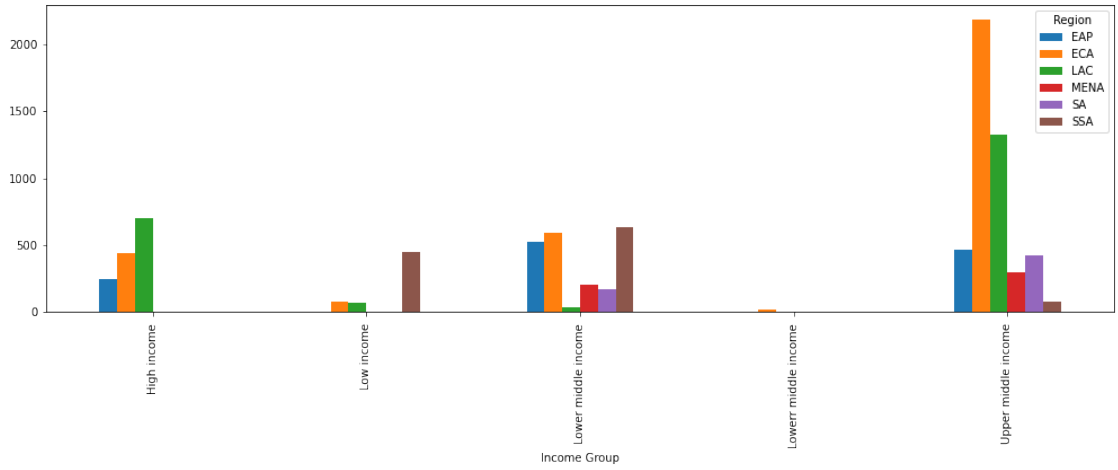
```
it=1
plt.figure(figsize=(37,25))
for i in merged_df.select_dtypes(include=np.number):
    plt.subplot(2,3,it)
    sns.boxplot(x=merged_df[i],y=merged_df['Income Group'])
    it=it+1
plt.tight_layout()
plt.show()
```



**Inference:**

1. These are the boxplot with the Income Group for Numerical column.

```
pd.crosstab(columns=merged_df['Region'],index=merged_df['Income Group'],values=merged_df['Total'],aggfunc='sum').plot(kind='bar',figsize=(17,5))
plt.show()
```

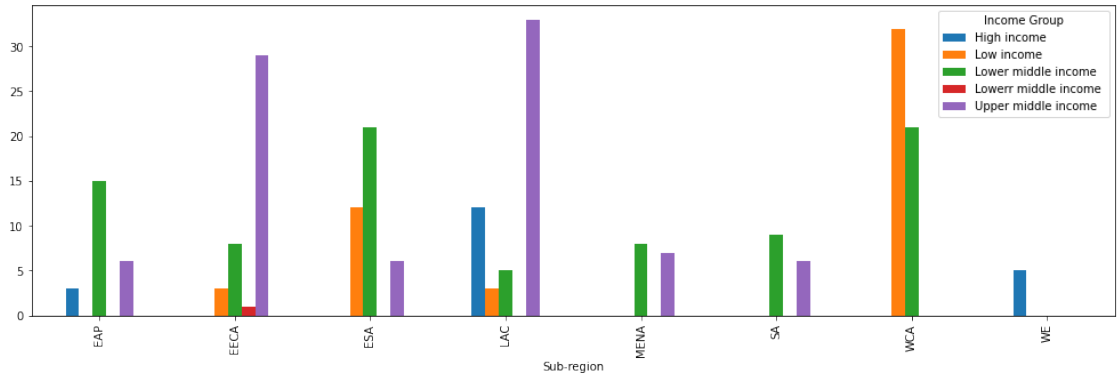



**Inference:**

1. This is a crosstab creation with the Merged Dataset
2. The Lower Middle Income doesn't have any count and there contributions to the final dataset is very less.

3. In-case of the Upper Middle Income we can have good contributions from the all Region.

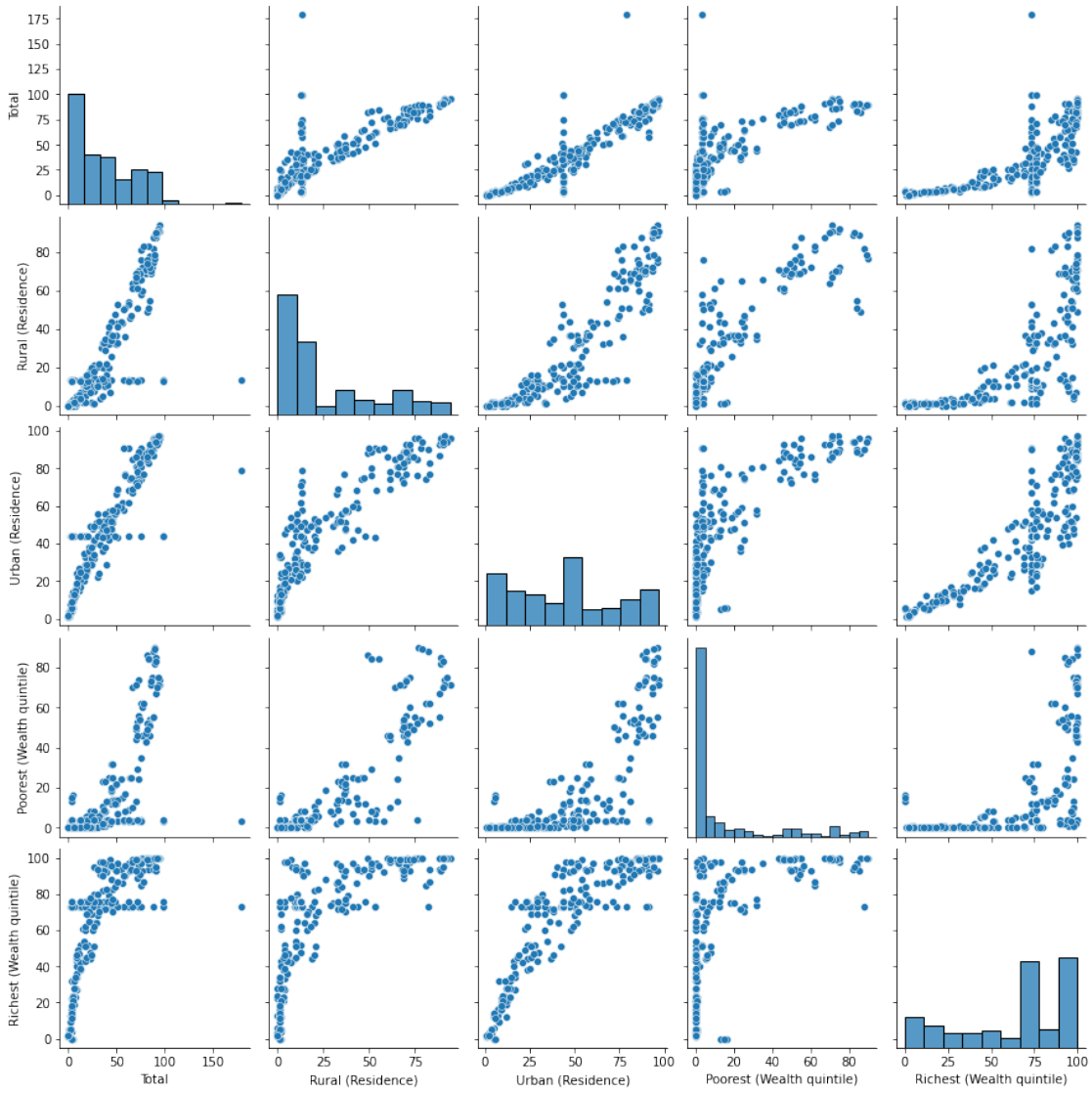
```
pd.crosstab(columns=merged_df['Income Group'],index=merged_df['Sub-region']).plot(kind='bar',figsize=(17,5))
plt.show()
```




**Inference:**

1. Therefore from the crosstab we can see that Lower Middle Income is more in all the sub-regions except the WE.
2. The EECA sub-regions all the Income Group.
3. In LAC sub-region we have higher number of Upper Middle Income.
4. In WCA sub-region we have Low Income count more in this region

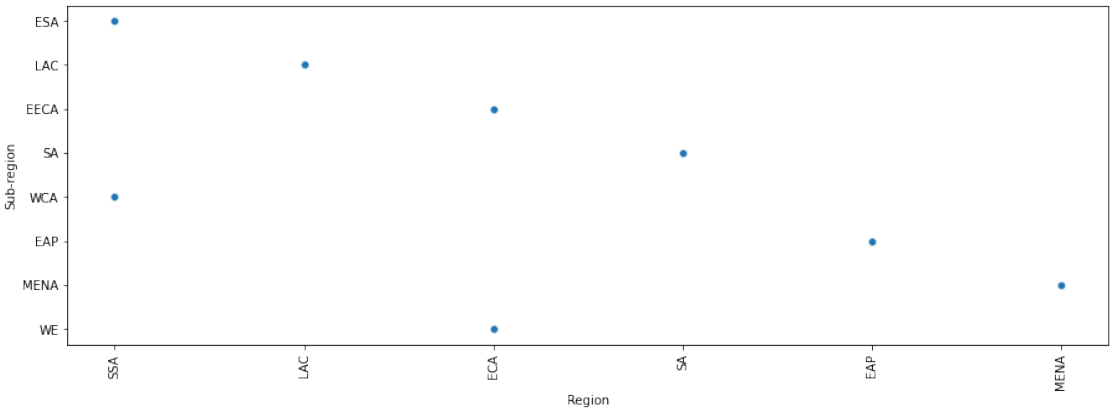
```
sns.pairplot(data=merged_df)
plt.show()
```

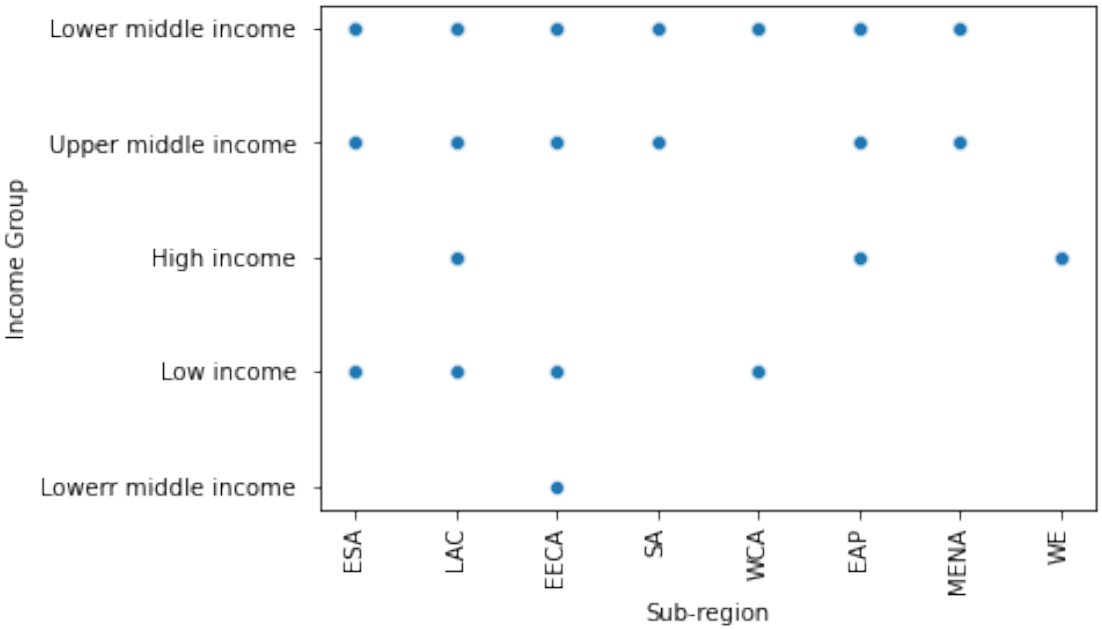


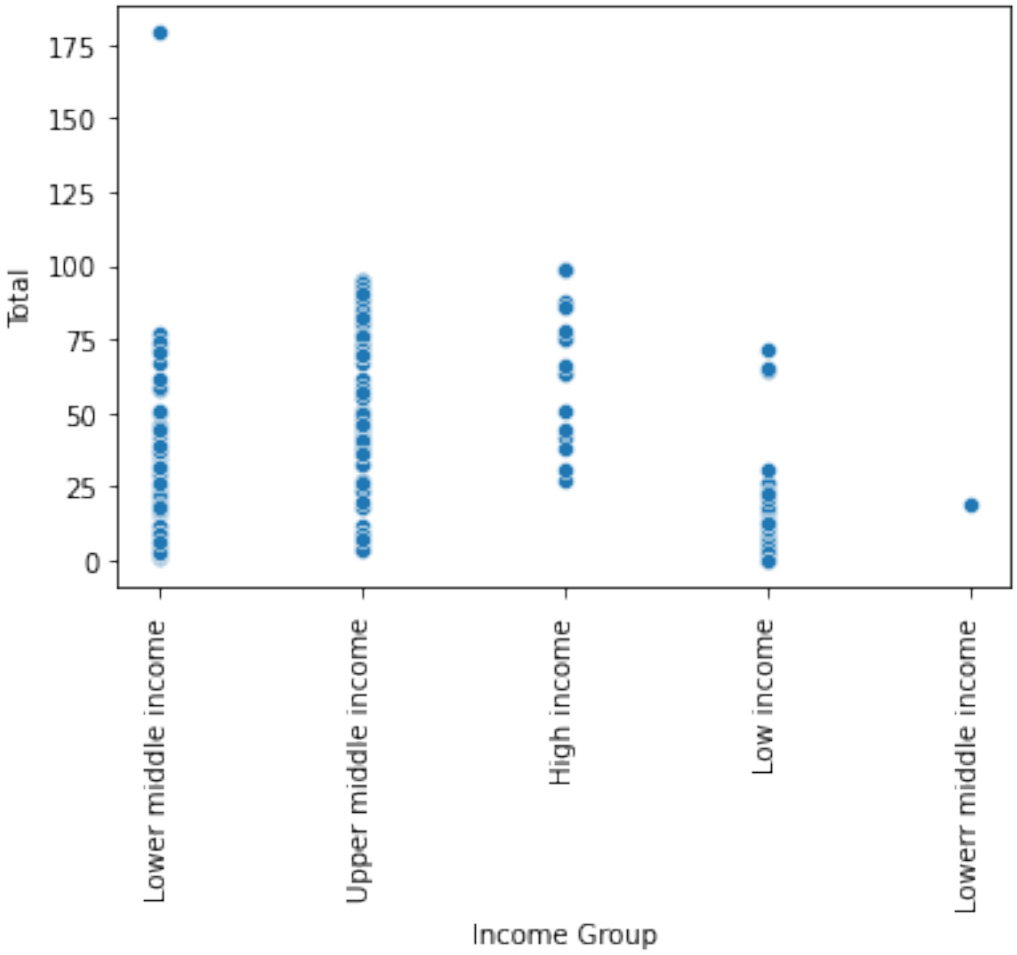

**Inference:**

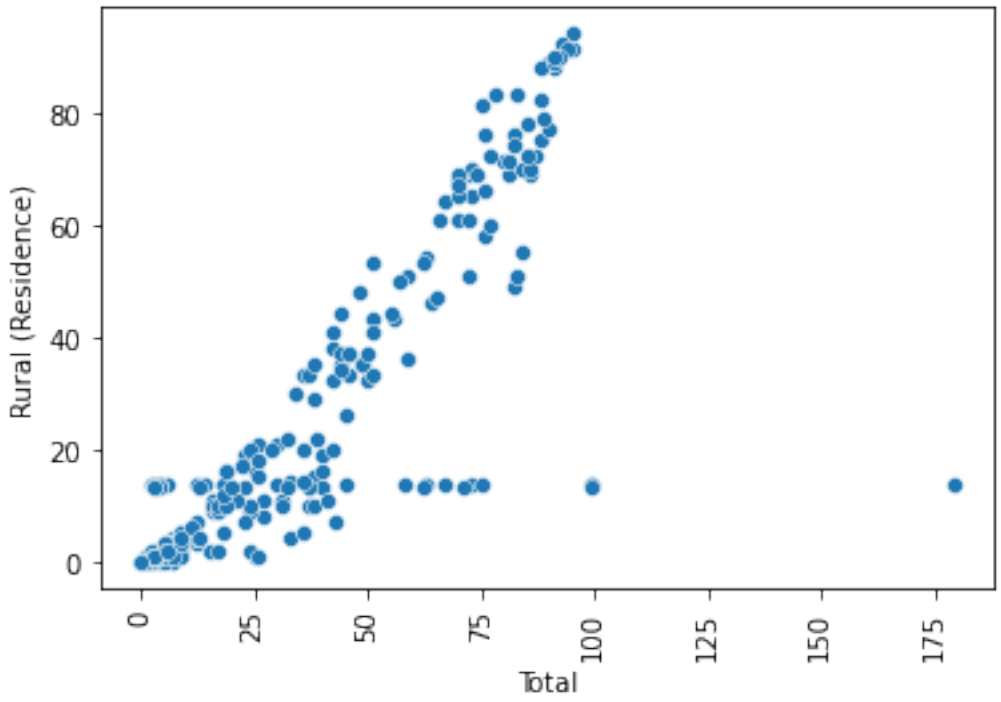
1. Looking in this pairplot we can say all are scattered.
2. There is some Multicollinearity presence.
3. The pairplot in the diagonal shows the Histogram of the same column.

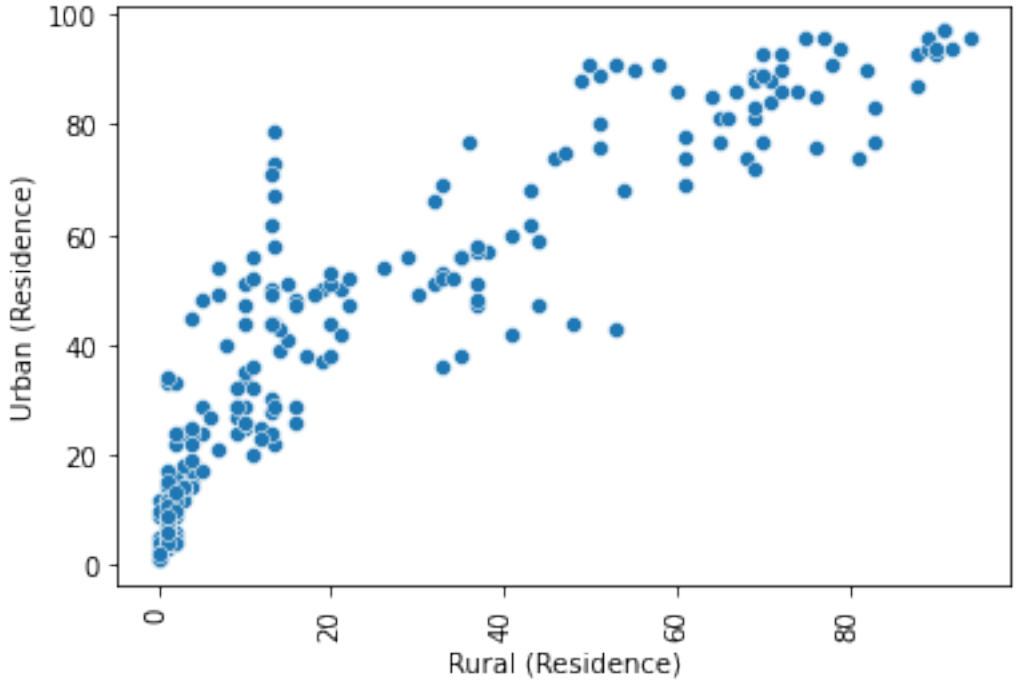
```
plt.figure(figsize=(15,5))
for i in range(merged_df.shape[1]-1):

sns.scatterplot(data=merged_df,x=merged_df.columns[i],y=merged_df.colu
mns[i+1])
    plt.xticks(rotation=90)
    plt.show()
```

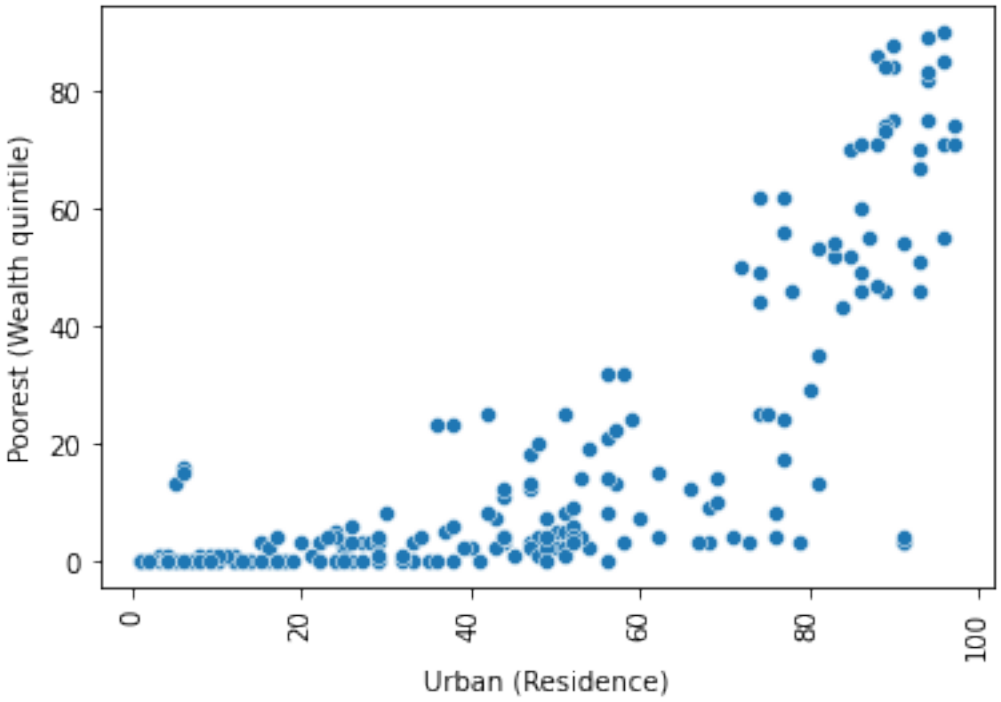

ESA
LAC
EECA
SA
WCA
EAP
MENA
WE
Sub-region
SSA
LAC
ECA
SA
EAP
MENA
Region


Lower middle income
Upper middle income
High income
Low income
Lowerr middle income
Income Group
ESA
LAC
EECA
SA
WCA
EAP
MENA
WE
Sub-region

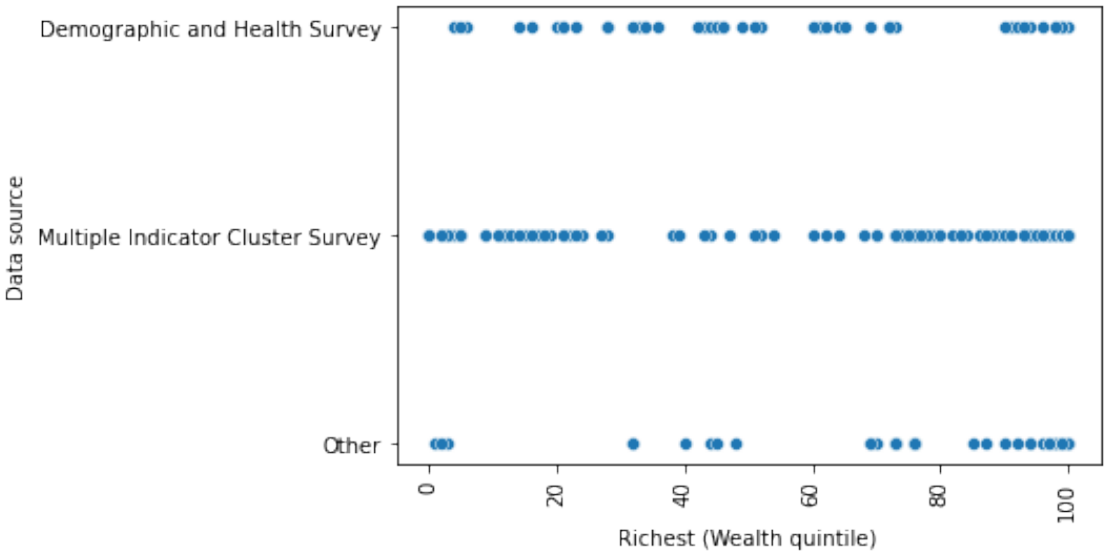

Total
175
150
125
100
75
50
25
0
Lower middle income
Upper middle income
High income
Low income
Lowerr middle income
Income Group

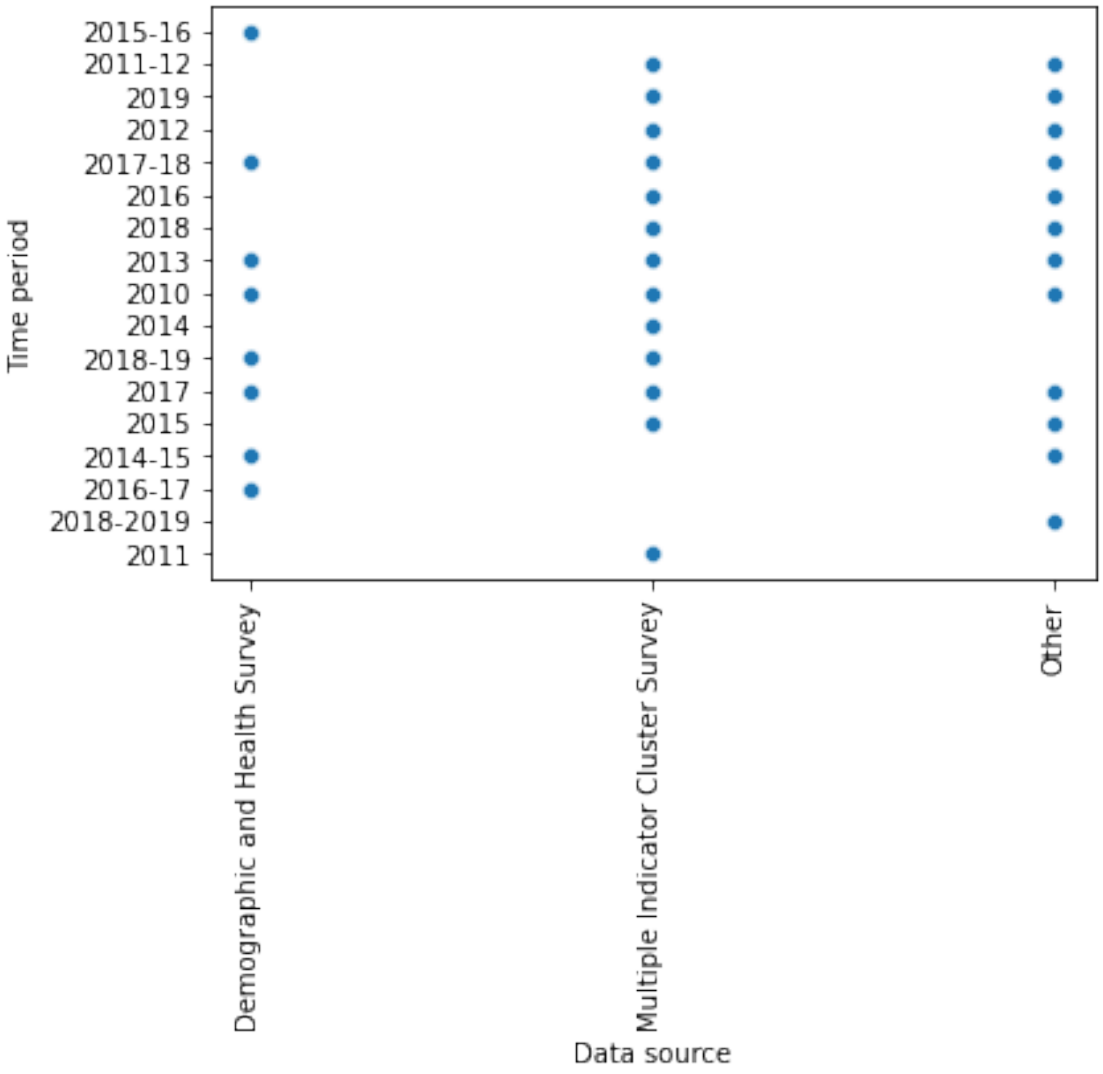

Rural (Residence)
80
60
40
20
0
0
25
50
75
100
125
150
175
Total


Urban (Residence)
Rural (Residence)
0
20
40
60
80
100


Poorest (Wealth quintile)
Urban (Residence)
0
20
40
60
80
100


Richest (Wealth quintile)
Poorest (Wealth quintile)
0
20
40
60
80
100


Data source
Demographic and Health Survey
Multiple Indicator Cluster Survey
Other
Richest (Wealth quintile)
0
20
40
60
80
100




**Inference:**

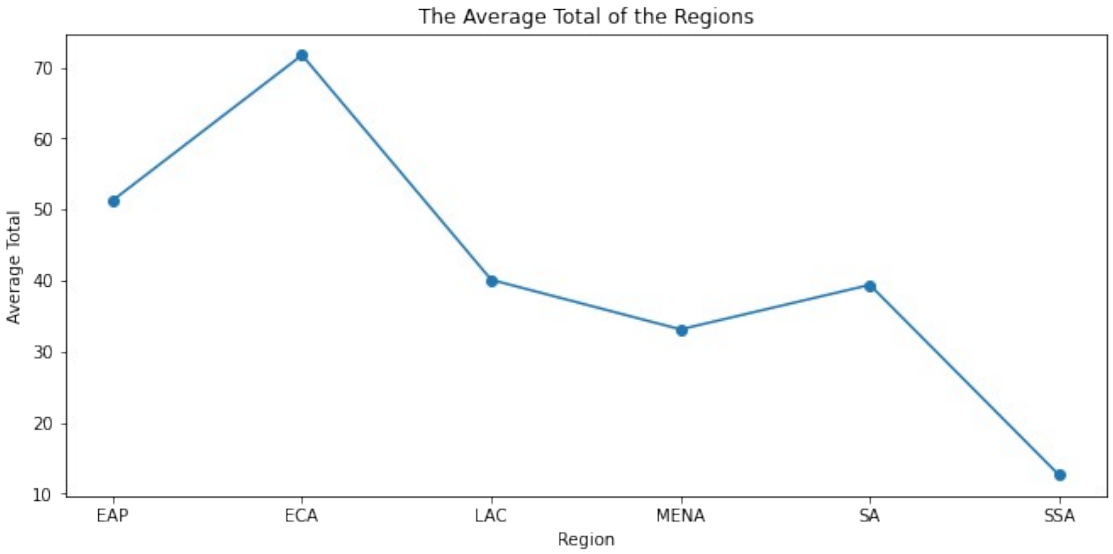
1. I have plotted a scatter plot with the current column and the next column.
2. All the datapoints are scattered there is no pattern seen.

```
plt.figure(figsize=(11,7))
sns.heatmap(merged_df.corr()[np.abs(merged_df.corr())>0.5],annot=True)
plt.show()
```


**inference:**

1. All the variables are Highly correlated to each other.
2. This shows that there are chances of one influence the other.
3. Possible of Multicollinearity.

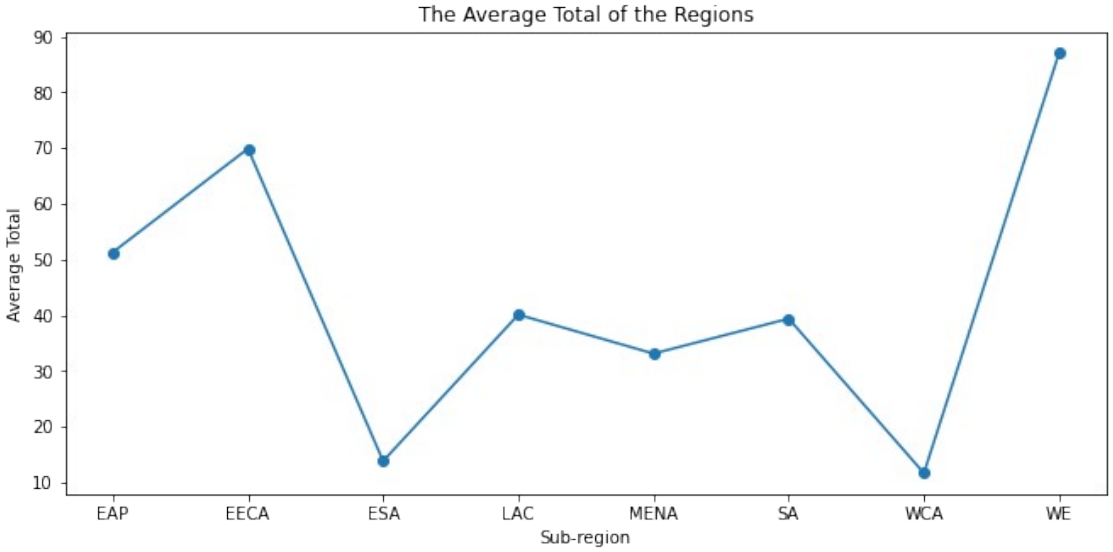
```
plt.figure(figsize=(11,5))
merged_df.groupby('Region')
['Total'].mean().plot(kind='line',marker='o')
plt.ylabel('Average Total')
plt.title('The Average Total of the Regions')
plt.show()
```




**Inference:**

1. The value of Average total and the Region is plotted. We can see that ECA region has the Higher Total value and the least is the SSA with the total of lesser than 10.

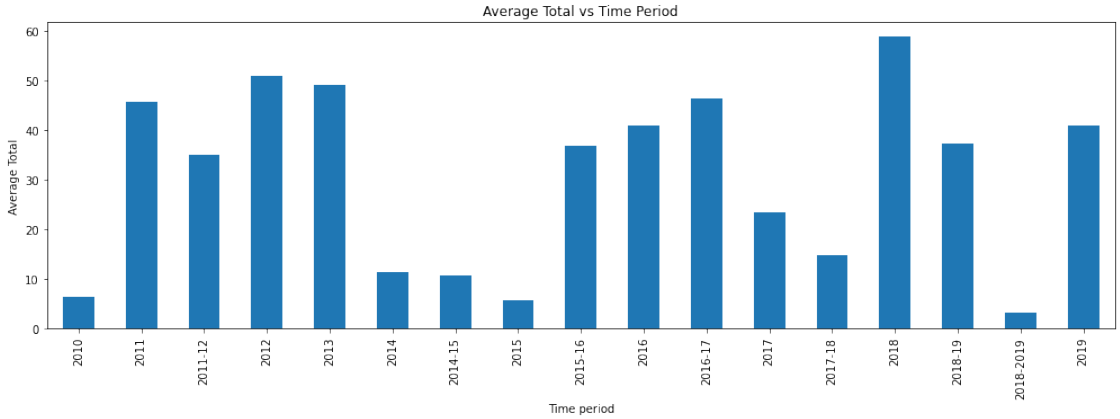
```
plt.figure(figsize=(11,5))
merged_df.groupby('Sub-region')
['Total'].mean().plot(kind='line',marker='o')
plt.ylabel('Average Total')
plt.title('The Average Total of the Regions')
plt.show()
```




**Inference:**

1. The value of Average total and the sub-Region is plotted. We can see that WE sub-region has the Higher Total value and the least is the ESA with the total of lesser than 10.

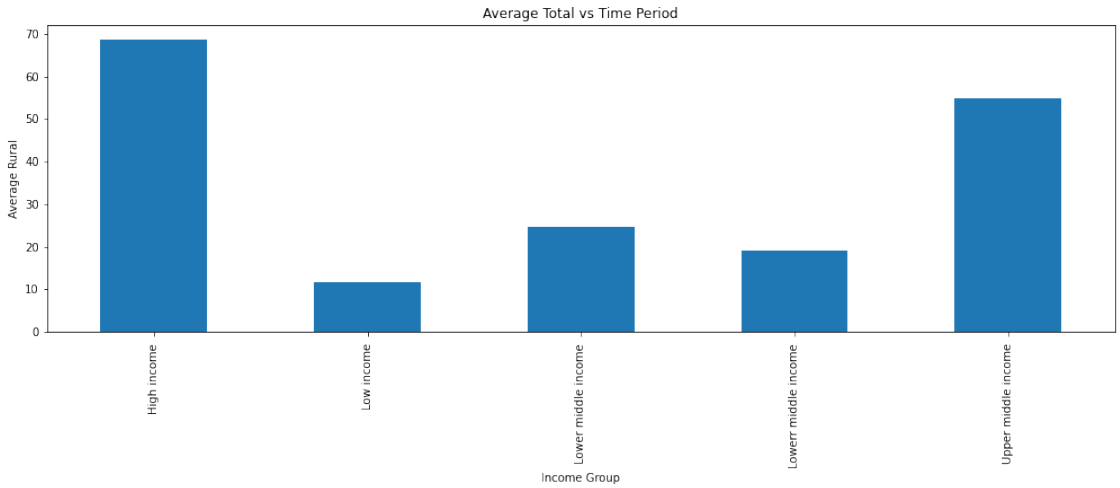
```
plt.figure(figsize=(17,5))
merged_df.groupby('Time period')['Total'].mean().plot(kind='bar')
plt.ylabel('Average Total')
plt.title('Average Total vs Time Period',color='black')
plt.show()
```




**Inference:**

1. The 2018 Time period is more in the Average total and therefore we can say that the contribution in 2018 is more than the other Time period.
2. The Least Time period is 2018-2019 with the total average is lesser than 10.

```
plt.figure(figsize=(17,5))
merged_df.groupby('Income Group')['Total'].mean().plot(kind='bar')
plt.ylabel('Average Rural')
plt.title('Average Total vs Time Period',color='black')
plt.show()
```

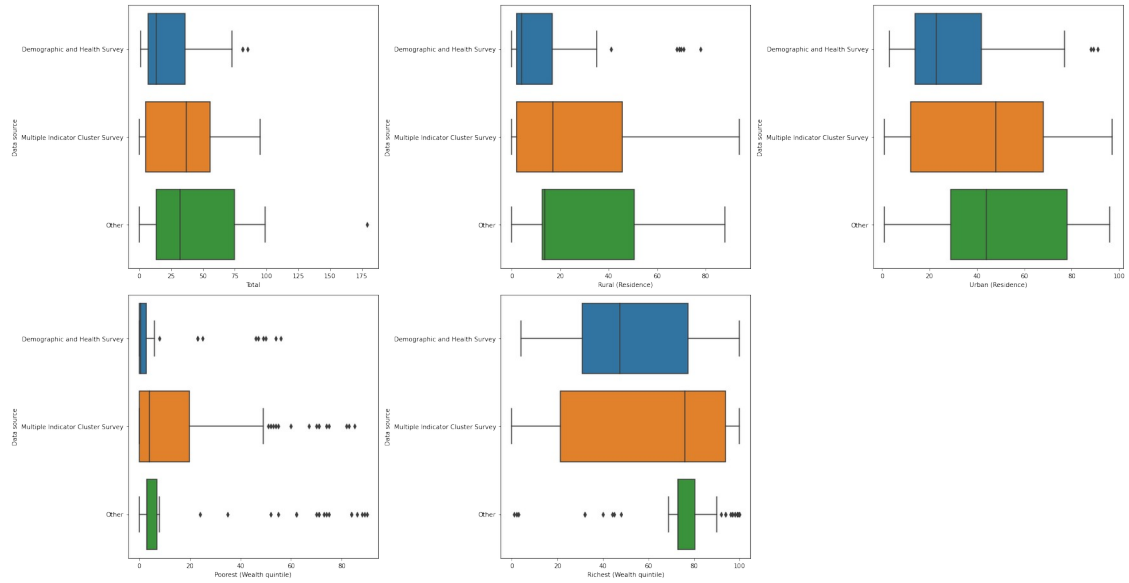



**Inference:**

1. The contribution is more for the category High Income.
2. As we known that the total will be more the High Income people and that is what seen over here.

3. The least is the Low Income and thus total is less than what we expect to have.

```
it=1
plt.figure(figsize=(25,13))
for i in merged_df.select_dtypes(include=np.number):
    plt.subplot(2,3,it)
    sns.boxplot(x=merged_df[i],y=merged_df['Data source'])
    it=it+1
plt.tight_layout()
plt.show()
```




**Inference:**

1. These are the box plot of the Numerical Column and then we have the DataSource as the Hue over here.
2. As we can there are some Outliers present in the DataSource we have.
3. The other catrgory has excessive amount of outliers.

## Outlier Treatment

```
print('Before:',merged_df.shape)
```

```
Before: (245, 10)
```

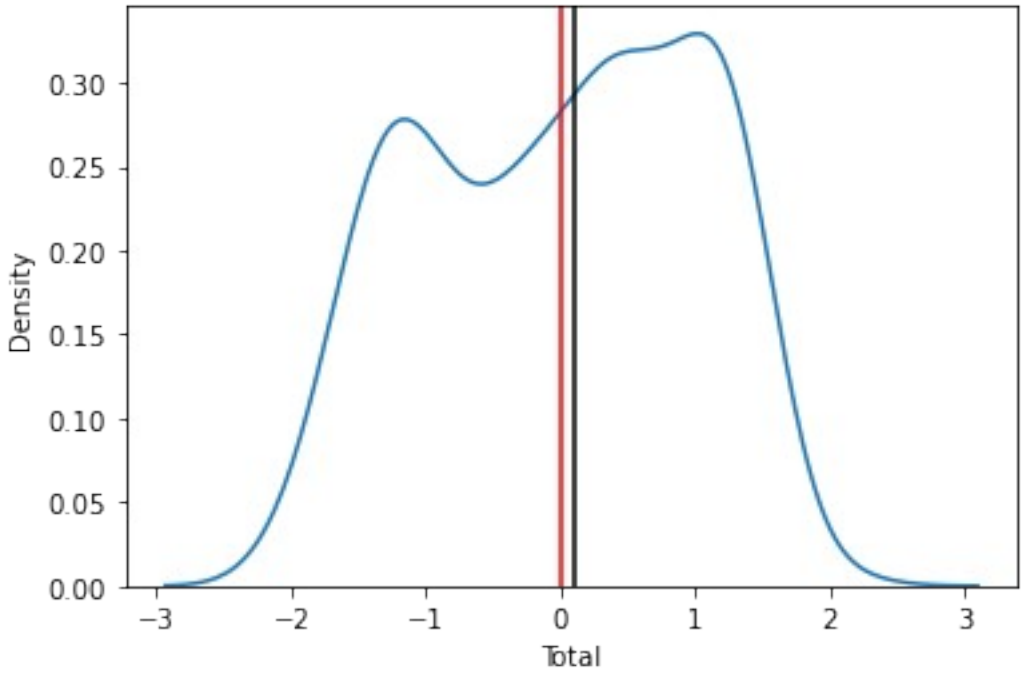
```
for i in merged_df.select_dtypes(include=np.number):
    q1 = merged_df[i].quantile(0.25)
    q3 = merged_df[i].quantile(0.75)
    iqr = q3 - q1
    ul = q3 + 1.5*iqr
    ll = q1 - 1.5*iqr
    merged_df[i] = np.where(merged_df[i]>ul,ul,merged_df[i])
    merged_df[i] = np.where(merged_df[i]<ll,ll,merged_df[i])

print('After:',merged_df.shape)
```

```
After: (245, 10)
```

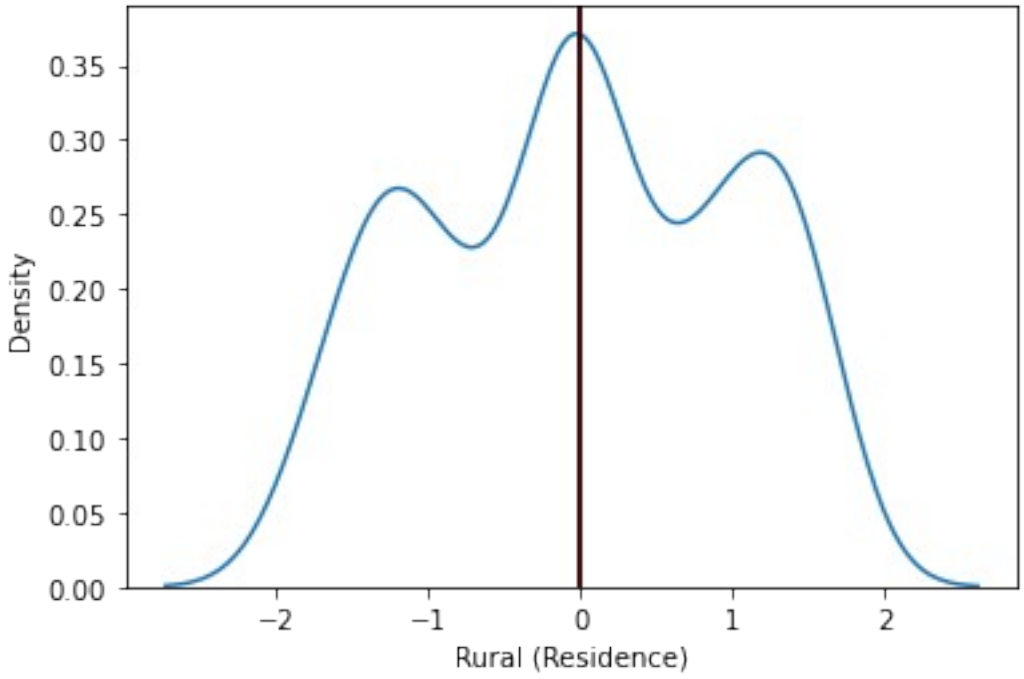
### Using Power Transformation for Reducing the Skewness

```
from sklearn.preprocessing import PowerTransformer
pt = PowerTransformer()
for i in merged_df:
    if i in merged_df.skew()[np.abs(merged_df.skew())>0.5]:
        merged_df[i] = pt.fit_transform(merged_df[[i]])

for i in merged_df.select_dtypes(include=np.number):
    sns.kdeplot(x= merged_df[i])
    plt.axvline(merged_df[i].mean(),color='red')
    plt.axvline(merged_df[i].median(),color='black')
    print('Column Name:',i,'Skewness:',merged_df[i].skew())
    plt.show()
```

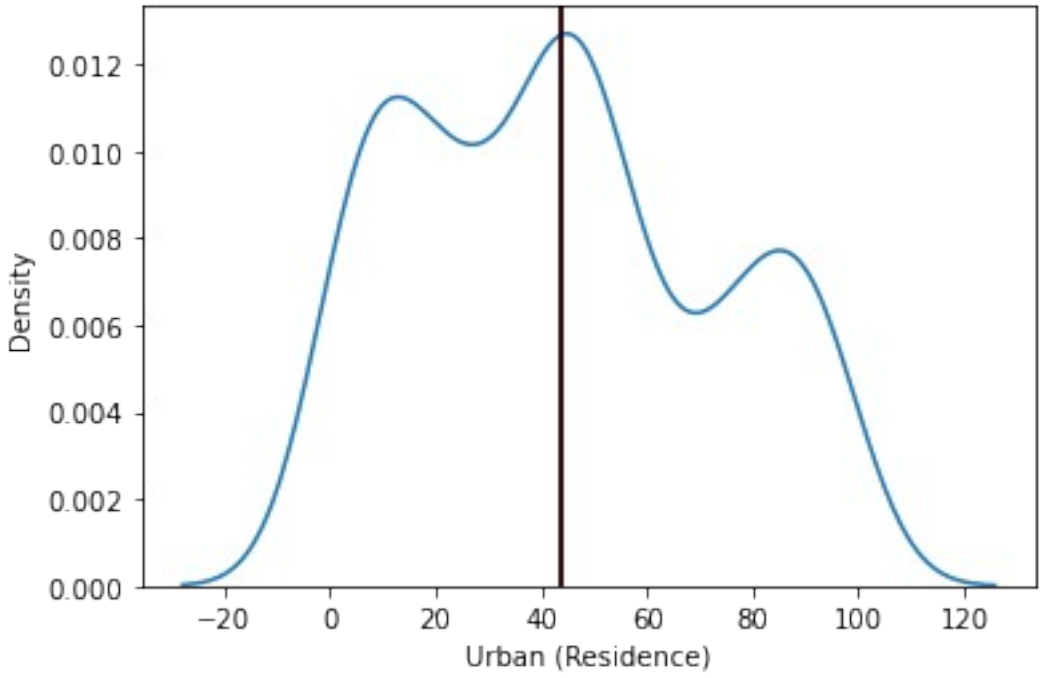
```
Column Name: Total Skewness: -0.15891329896503723
```



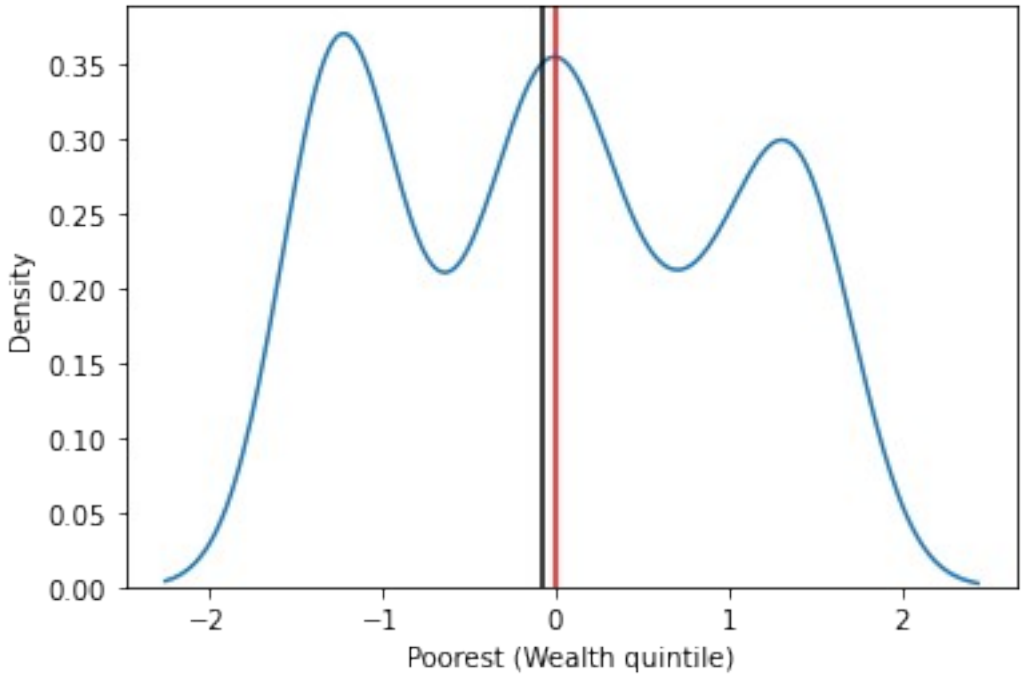

```
Column Name: Rural (Residence) Skewness: -0.06634763396158129
```



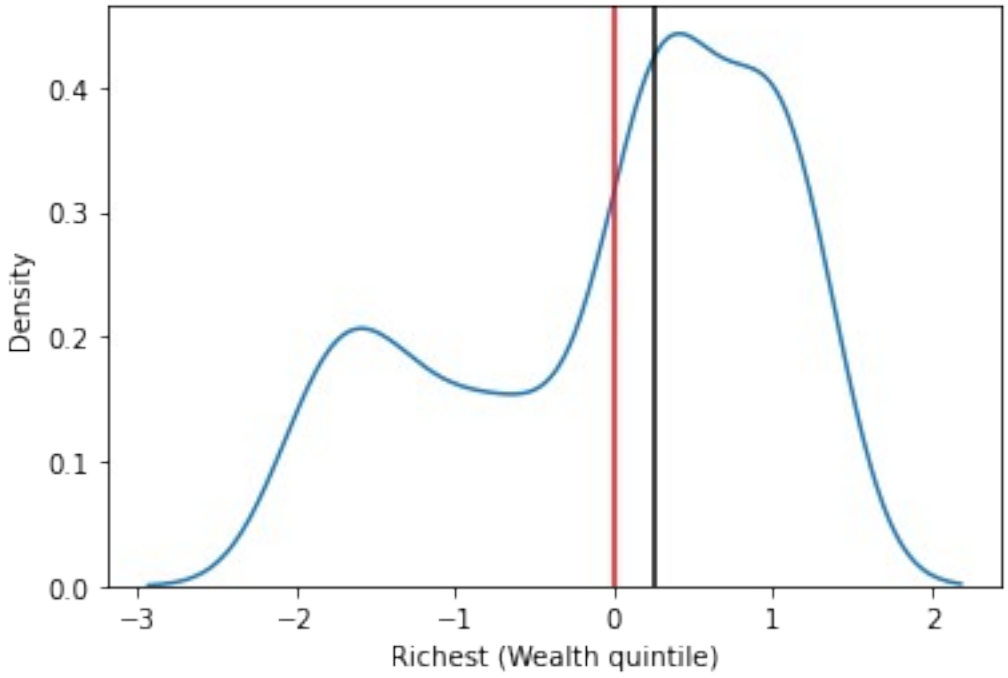

Column Name: Urban (Residence) Skewness: 0.2689144778739272




Column Name: Poorest (Wealth quintile) Skewness: 0.09424306762754271




```
Column Name: Richest (Wealth quintile) Skewness: -0.6141789551779578
```




**Inference:**

1. The skewness are reduced a therefore we can proceed with this further for Model build.

## Converting this Final dataset into a CSV

```
merged_df.to_csv('final.csv')
```