```
import pandas as pd
df=pd.read_excel('Goal9.xlsx',sheet_name='data')
df
```

```
       Goal Target Indicator     SeriesCode  \
0         9    9.1     9.1.1    SP_ROD_R2KM
1         9    9.1     9.1.1    SP_ROD_R2KM
2         9    9.1     9.1.1    SP_ROD_R2KM
3         9    9.1     9.1.1    SP_ROD_R2KM
4         9    9.1     9.1.1    SP_ROD_R2KM
...     ...    ...       ...            ...
51245     9    9.c     9.c.1  IT_MOB_4GNTWK
51246     9    9.c     9.c.1  IT_MOB_4GNTWK
51247     9    9.c     9.c.1  IT_MOB_4GNTWK
51248     9    9.c     9.c.1  IT_MOB_4GNTWK
51249     9    9.c     9.c.1  IT_MOB_4GNTWK

                                      SeriesDescription  GeoAreaCode
\
0      Proportion of the rural population who live wi...           50

1      Proportion of the rural population who live wi...           51

2      Proportion of the rural population who live wi...          108

3      Proportion of the rural population who live wi...          231

4      Proportion of the rural population who live wi...          368

...                                                  ...          ...

51245  Proportion of population covered by at least a...          894

51246  Proportion of population covered by at least a...          894

51247  Proportion of population covered by at least a...          894

51248  Proportion of population covered by at least a...          894

51249  Proportion of population covered by at least a...          894


      GeoAreaName  TimePeriod  Value  Time_Detail  ...  Nature  \
0      Bangladesh        2015  86.70         2015  ...       E
1         Armenia        2017  66.00         2017  ...       E
2         Burundi        2016  24.90         2016  ...       E
3        Ethiopia        2015  21.60         2015  ...       E
4            Iraq        2018  63.40         2018  ...       E
...           ...         ...    ...          ...  ...     ...
51245      Zambia        2016   5.77         2016  ...       C
51246      Zambia        2017  43.41         2017  ...       C
```

```
51247      Zambia        2018  43.41         2018  ...       C
51248      Zambia        2019  49.10         2019  ...       C
51249      Zambia        2020  88.37         2020  ...       C

       Sampling Stations  Cities  Level of requirement Quantile  IHR
Capacity  \
0                     NaN     NaN                    NaN      NaN
NaN
1                     NaN     NaN                    NaN      NaN
NaN
2                     NaN     NaN                    NaN      NaN
NaN
3                     NaN     NaN                    NaN      NaN
NaN
4                     NaN     NaN                    NaN      NaN
NaN
...                   ...     ...                    ...      ...
...
51245                 NaN     NaN                    NaN      NaN
NaN
51246                 NaN     NaN                    NaN      NaN
NaN
51247                 NaN     NaN                    NaN      NaN
NaN
51248                 NaN     NaN                    NaN      NaN
NaN
51249                 NaN     NaN                    NaN      NaN
NaN

      Hazard type  Migratory status  Population Group  \
0             NaN               NaN               NaN
1             NaN               NaN               NaN
2             NaN               NaN               NaN
3             NaN               NaN               NaN
4             NaN               NaN               NaN
...           ...               ...               ...
51245         NaN               NaN               NaN
51246         NaN               NaN               NaN
51247         NaN               NaN               NaN
51248         NaN               NaN               NaN
51249         NaN               NaN               NaN

       Name of international institution
0                                    NaN
1                                    NaN
2                                    NaN
3                                    NaN
4                                    NaN
...                                  ...
51245                                NaN
```

```
51246                                    NaN
51247                                    NaN
51248                                    NaN
51249                                    NaN

[51250 rows x 65 columns]
```

```
df.head()
```

```
   Goal Target Indicator   SeriesCode  \
0     9    9.1     9.1.1  SP_ROD_R2KM
1     9    9.1     9.1.1  SP_ROD_R2KM
2     9    9.1     9.1.1  SP_ROD_R2KM
3     9    9.1     9.1.1  SP_ROD_R2KM
4     9    9.1     9.1.1  SP_ROD_R2KM

                                  SeriesDescription  GeoAreaCode
GeoAreaName  \
0  Proportion of the rural population who live wi...           50
Bangladesh
1  Proportion of the rural population who live wi...           51
Armenia
2  Proportion of the rural population who live wi...          108
Burundi
3  Proportion of the rural population who live wi...          231
Ethiopia
4  Proportion of the rural population who live wi...          368
Iraq

   TimePeriod  Value  Time_Detail  ...  Nature  Sampling Stations
Cities  \
0        2015   86.7         2015  ...       E                NaN
NaN
1        2017   66.0         2017  ...       E                NaN
NaN
2        2016   24.9         2016  ...       E                NaN
NaN
3        2015   21.6         2015  ...       E                NaN
NaN
4        2018   63.4         2018  ...       E                NaN
NaN

   Level of requirement Quantile  IHR Capacity Hazard type  Migratory
status  \
0                   NaN      NaN           NaN         NaN
NaN
1                   NaN      NaN           NaN         NaN
NaN
2                   NaN      NaN           NaN         NaN
NaN
```

```
3                    NaN      NaN            NaN          NaN
NaN
4                    NaN      NaN            NaN          NaN
NaN

   Population Group  Name of international institution
0               NaN                                NaN
1               NaN                                NaN
2               NaN                                NaN
3               NaN                                NaN
4               NaN                                NaN

[5 rows x 65 columns]

df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51250 entries, 0 to 51249
Data columns (total 65 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   Goal                                    51250 non-null  int64
 1   Target                                  51250 non-null  object
 2   Indicator                               51250 non-null  object
 3   SeriesCode                              51250 non-null  object
 4   SeriesDescription                       51250 non-null  object
 5   GeoAreaCode                             51250 non-null  int64
 6   GeoAreaName                             51250 non-null  object
 7   TimePeriod                              51250 non-null  int64
 8   Value                                   51034 non-null  float64
 9   Time_Detail                             51250 non-null  int64
 10  TimeCoverage                            0 non-null      float64
 11  UpperBound                              0 non-null      float64
 12  LowerBound                              0 non-null      float64
 13  BasePeriod                              18020 non-null  float64
 14  Source                                  51250 non-null  object
 15  GeoInfoUrl                              0 non-null      float64
 16  FootNote                                8650 non-null   object
 17  Substance use disorders                 0 non-null      float64
 18  Type of renewable technology            0 non-null      float64
 19  Severity of price levels                0 non-null      float64
 20  Sex                                     0 non-null      float64
 21  Service Attribute                       0 non-null      float64
 22  Deviation Level                         0 non-null      float64
 23  Mountain Elevation                      0 non-null      float64
 24  Parliamentary committees                0 non-null      float64
 25  Mode of transportation                  6508 non-null   object
 26  Level_of_government                     0 non-null      float64
 27  Fiscal intervention stage               0 non-null      float64
 28  Type of support                         0 non-null      float64
```

```
 29  Type of speed                                0 non-null      float64
 30  Policy instruments                           0 non-null      float64
 31  Policy Domains                               0 non-null      float64
 32  Counterpart                                  0 non-null      float64
 33  Type of skill                                0 non-null      float64
 34  Education level                              0 non-null      float64
 35  Type of waste treatment                      0 non-null      float64
 36  Custom Breakdown                             0 non-null      float64
 37  Location                                     26 non-null     object
 38  Food Waste Sector                            0 non-null      float64
 39  Reporting Type                               51250 non-null  object
 40  Freq                                         0 non-null      float64
 41  Type of product                              0 non-null      float64
 42  Cause of death                               0 non-null      float64
 43  Report Ordinal                               0 non-null      float64
 44  Government_Name                              0 non-null      float64
 45  Observation Status                           20635 non-null  object
 46  Type of occupation                           0 non-null      float64
 47  Name of non-communicable disease             0 non-null      float64
 48  Grounds of discrimination                    0 non-null      float64
 49  Units                                        51250 non-null  object
 50  Level/Status                                 0 non-null      float64
 51  Age                                          0 non-null      float64
 52  Disability status                            0 non-null      float64
 53  Frequency of Chlorophyll-a concentration     0 non-null      float64
 54  Activity                                     28689 non-null  object
 55  Nature                                       51250 non-null  object
 56  Sampling Stations                            0 non-null      float64
 57  Cities                                       0 non-null      float64
 58  Level of requirement                         0 non-null      float64
 59  Quantile                                     0 non-null      float64
 60  IHR Capacity                                 0 non-null      float64
 61  Hazard type                                  0 non-null      float64
 62  Migratory status                             0 non-null      float64
 63  Population Group                             0 non-null      float64
 64  Name of international institution            0 non-null      float64
dtypes: float64(47), int64(4), object(14)
memory usage: 25.4+ MB
```

From the above information,it is clear that columns TimeCoverage,UpperBound,LowerBound,GeoInfoUrl,Substance use disorders,Type of renewable technology ,Severity of price levels ,Sex ,Service Attribute,Deviation Level,Mountain Elevation,Parliamentary committees,Level_of_government,Fiscal intervention stage ,Type of support ,Type of speed,Policy instruments ,Policy Domains,Counterpart,Type of skill,Education level ,Type of waste treatment,Custom Breakdown ,Food Waste Sector,Freq ,Type of product,Cause of death,Report Ordinal ,Government_Name ,Type of occupation ,Name of non-communicable disease ,Grounds of discrimination,Level/Status,Age , Disability status,Frequency of Chlorophyll-a concentration ,Sampling Stations,Cities ,Level of requirement ,Quantile ,IHR

Capacity , Hazard type ,Migratory status,Population Group,Name of international institution are empty ,so it can be dropped using drop function.

```
df.drop(['TimeCoverage','UpperBound','LowerBound','GeoInfoUrl','Substa
nce use disorders','Type of renewable technology' ,'Severity of price
levels','Sex' ,'Service Attribute','Deviation Level','Mountain
Elevation','Parliamentary committees','Level_of_government','Fiscal
intervention stage' ,'Type of support' ,'Type of speed','Policy
instruments','Policy Domains','Counterpart','Type of skill','Education
level' ,'Type of waste treatment','Custom Breakdown' ,'Food Waste
Sector','Freq'  ,'Type of product','Cause of death','Report
Ordinal' ,'Government_Name' ,'Type of occupation' ,'Name of non-
communicable disease' ,'Grounds of
discrimination','Level/Status','Age'  , 'Disability status','Frequency
of Chlorophyll-a concentration' ,'Sampling Stations','Cities' ,'Level
of requirement' ,'Quantile' ,'IHR Capacity'  , 'Hazard
type' ,'Migratory status','Population Group','Name of international
institution'],axis=1,inplace=True)

#checking whther the columns are dropped.
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51250 entries, 0 to 51249
Data columns (total 20 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Goal                    51250 non-null  int64
 1   Target                  51250 non-null  object
 2   Indicator               51250 non-null  object
 3   SeriesCode              51250 non-null  object
 4   SeriesDescription       51250 non-null  object
 5   GeoAreaCode             51250 non-null  int64
 6   GeoAreaName             51250 non-null  object
 7   TimePeriod              51250 non-null  int64
 8   Value                   51034 non-null  float64
 9   Time_Detail             51250 non-null  int64
 10  BasePeriod              18020 non-null  float64
 11  Source                  51250 non-null  object
 12  FootNote                8650 non-null   object
 13  Mode of transportation  6508 non-null   object
 14  Location                26 non-null     object
 15  Reporting Type          51250 non-null  object
 16  Observation Status      20635 non-null  object
 17  Units                   51250 non-null  object
 18  Activity                28689 non-null  object
 19  Nature                  51250 non-null  object
dtypes: float64(2), int64(4), object(14)
memory usage: 7.8+ MB
```

```
#checking for null values
df.isnull().sum()
```

```
Goal                            0
Target                          0
Indicator                       0
SeriesCode                      0
SeriesDescription               0
GeoAreaCode                     0
GeoAreaName                     0
TimePeriod                      0
Value                         216
Time_Detail                     0
BasePeriod                  33230
Source                          0
FootNote                    42600
Mode of transportation      44742
Location                    51224
Reporting Type                  0
Observation Status          30615
Units                           0
Activity                    22561
Nature                          0
dtype: int64
```

```
import numpy as np
df1=df.replace(np.nan,0)
df1.tail()
df1['Target'].unique()
```

```
array(['9.1', '9.2', '9.3', '9.4', '9.5', '9.a', '9.b', '9.c'],
      dtype=object)
```

```
#to check there is no null value present
df1.isnull().sum()
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51250 entries, 0 to 51249
Data columns (total 20 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Goal                    51250 non-null  int64
 1   Target                  51250 non-null  object
 2   Indicator               51250 non-null  object
 3   SeriesCode              51250 non-null  object
 4   SeriesDescription       51250 non-null  object
 5   GeoAreaCode             51250 non-null  int64
 6   GeoAreaName             51250 non-null  object
 7   TimePeriod              51250 non-null  int64
```

```
 8   Value                    51250 non-null  float64
 9   Time_Detail              51250 non-null  int64
 10  BasePeriod               51250 non-null  float64
 11  Source                   51250 non-null  object
 12  FootNote                 51250 non-null  object
 13  Mode of transportation   51250 non-null  object
 14  Location                 51250 non-null  object
 15  Reporting Type           51250 non-null  object
 16  Observation Status       51250 non-null  object
 17  Units                    51250 non-null  object
 18  Activity                 51250 non-null  object
 19  Nature                   51250 non-null  object
dtypes: float64(2), int64(4), object(14)
memory usage: 7.8+ MB
```

```
df2=df1[(df1['Target']== '9.a') | (df1['Target']=='9.b') |
(df1['Target']=='9.c')]
df2.shape
df2
```

```
       Goal Target Indicator     SeriesCode  \
38427     9    9.a     9.a.1  DC_TOF_INFRAL
38428     9    9.a     9.a.1  DC_TOF_INFRAL
38429     9    9.a     9.a.1  DC_TOF_INFRAL
38430     9    9.a     9.a.1  DC_TOF_INFRAL
38431     9    9.a     9.a.1  DC_TOF_INFRAL
...     ...    ...       ...            ...
51245     9    9.c     9.c.1  IT_MOB_4GNTWK
51246     9    9.c     9.c.1  IT_MOB_4GNTWK
51247     9    9.c     9.c.1  IT_MOB_4GNTWK
51248     9    9.c     9.c.1  IT_MOB_4GNTWK
51249     9    9.c     9.c.1  IT_MOB_4GNTWK

                                      SeriesDescription  GeoAreaCode
\
38427  Total official flows for infrastructure, by re...            4

38428  Total official flows for infrastructure, by re...            4

38429  Total official flows for infrastructure, by re...            4

38430  Total official flows for infrastructure, by re...            4

38431  Total official flows for infrastructure, by re...            4

...                                                  ...          ...

51245  Proportion of population covered by at least a...          894

51246  Proportion of population covered by at least a...          894
```

```
51247  Proportion of population covered by at least a...          894

51248  Proportion of population covered by at least a...          894

51249  Proportion of population covered by at least a...          894


       GeoAreaName  TimePeriod      Value  Time_Detail  BasePeriod  \
38427  Afghanistan        2000    0.38602         2000      2020.0
38428  Afghanistan        2001    0.38101         2001      2020.0
38429  Afghanistan        2002   25.53323         2002      2020.0
38430  Afghanistan        2003  130.05994         2003      2020.0
38431  Afghanistan        2004  505.92026         2004      2020.0
...            ...         ...        ...          ...         ...
51245       Zambia        2016    5.77000         2016         0.0
51246       Zambia        2017   43.41000         2017         0.0
51247       Zambia        2018   43.41000         2018         0.0
51248       Zambia        2019   49.10000         2019         0.0
51249       Zambia        2020   88.37000         2020         0.0

                                                  Source  \
38427  The Organisation for Economic Co-operation and...
38428  The Organisation for Economic Co-operation and...
38429  The Organisation for Economic Co-operation and...
38430  The Organisation for Economic Co-operation and...
38431  The Organisation for Economic Co-operation and...
...                                                  ...
51245  Zambia Information & Communications Technology...
51246  Zambia Information & Communications Technology...
51247  Zambia Information & Communications Technology...
51248  Zambia Information & Communications Technology...
51249  Zambia Information & Communications Technology...

                                                FootNote  \
38427  Commitments;Based on OECD, CRS databased, 2022...
38428  Commitments;Based on OECD, CRS databased, 2022...
38429  Gross disbursements;Based on OECD, CRS databas...
38430  Gross disbursements;Based on OECD, CRS databas...
38431  Gross disbursements;Based on OECD, CRS databas...
...                                                  ...
51245                                                  0
51246  In 2016 we overlooked population coverage by V...
51247  Statistics from MNO with highest 4G population...
51248  Statistics from MNO with highest 4G population...
51249  Statistics from National Mobile & Broadcast Co...

      Mode of transportation Location Reporting Type Observation
Status  \
38427                      0        0              G
0
```

```
38428                        0        0               G
0
38429                        0        0               G
0
38430                        0        0               G
0
38431                        0        0               G
0
...                        ...      ...             ...                  ..
.
51245                        0        0               G
0
51246                        0        0               G
0
51247                        0        0               G
0
51248                        0        0               G
0
51249                        0        0               G
0

         Units Activity Nature
38427  CON_USD        0      C
38428  CON_USD        0      C
38429  CON_USD        0      C
38430  CON_USD        0      C
38431  CON_USD        0      C
...        ...      ...    ...
51245  PERCENT        0      C
51246  PERCENT        0      C
51247  PERCENT        0      C
51248  PERCENT        0      C
51249  PERCENT        0      C

[12823 rows x 20 columns]
```

Goal 1 : 9.A Facilitate sustainable and resilient infrastructure development in developing countries through enhanced financial, technological and technical support to African countries, least developed countries, landlocked developing countries and small island developing States 18

Analysing top 5 highest and lowest investments receiving countries which belong to African countries, least developed countries, landlocked developing countries and small island developing States 18

Finding average investments over the period of 2000 to 2020

```
df3=df2[(df2['Target']== '9.a')]
Average_investments=df3.groupby('GeoAreaName')
['Value'].mean().reset_index()
```

```
print(Average_investments)
```

```
                    GeoAreaName        Value
0                   Afghanistan   664.048038
1                       Albania   135.981200
2                       Algeria    55.598468
3                        Angola    99.381949
4                      Anguilla     2.187625
..                          ...          ...
152                    Viet Nam  1624.091846
153  Wallis and Futuna Islands     5.878157
154                      Yemen   133.274465
155                     Zambia   125.571542
156                   Zimbabwe    10.636420

[157 rows x 2 columns]
```

```
top_max =Average_investments.nlargest(15, 'Value')
top_min=Average_investments.nsmallest(10,'Value')
print(top_max)
print(top_min)
top_max_countries=list(top_max['GeoAreaName'])
top_min_countries=list(top_min['GeoAreaName'])
print(top_max_countries)
print(top_min_countries)
```

```
      GeoAreaName        Value
61          India  3672.272972
144        Türkiye  2329.390784
28           China  2039.889882
152       Viet Nam  1624.091846
62       Indonesia  1423.777127
43           Egypt  1384.996701
19          Brazil  1256.890173
91         Morocco  1015.429000
10      Bangladesh  1009.765863
86          Mexico   931.695568
103       Pakistan   880.519049
146        Ukraine   809.694844
64            Iraq   723.314948
0      Afghanistan   664.048038
126   South Africa   628.220525
                          GeoAreaName     Value
81                              Malta  0.024100
45                  Equatorial Guinea  0.402919
113             Saint Kitts and Nevis  0.877578
5                 Antigua and Barbuda  1.256256
```

```
139               Trinidad and Tobago  1.279455
142         Turks and Caicos Islands  1.324379
118                     Saudi Arabia  1.503351
100                             Niue  1.631261
115  Saint Vincent and the Grenadines  2.100704
4                            Anguilla  2.187625
['India', 'Türkiye', 'China', 'Viet Nam', 'Indonesia', 'Egypt',
'Brazil', 'Morocco', 'Bangladesh', 'Mexico', 'Pakistan', 'Ukraine',
'Iraq', 'Afghanistan', 'South Africa']
['Malta', 'Equatorial Guinea', 'Saint Kitts and Nevis', 'Antigua and
Barbuda', 'Trinidad and Tobago', 'Turks and Caicos Islands', 'Saudi
Arabia', 'Niue', 'Saint Vincent and the Grenadines', 'Anguilla']

african_countries = [
    "Algeria", "Angola", "Benin", "Botswana", "Burkina Faso",
    "Burundi", "Cabo Verde", "Cameroon", "Central African Republic",
"Chad",
    "Comoros", "Congo (Brazzaville)", "Congo (Kinshasa)", "Cote
d'Ivoire", "Djibouti",
    "Egypt", "Equatorial Guinea", "Eritrea", "Eswatini", "Ethiopia",
    "Gabon", "Gambia", "Ghana", "Guinea", "Guinea-Bissau",
    "Kenya", "Lesotho", "Liberia", "Libya", "Madagascar",
    "Malawi", "Mali", "Mauritania", "Mauritius", "Morocco",
    "Mozambique", "Namibia", "Niger", "Nigeria", "Rwanda",
    "Sao Tome and Principe", "Senegal", "Seychelles", "Sierra Leone",
"Somalia",
    "South Africa", "South Sudan", "Sudan", "Tanzania", "Togo",
    "Tunisia", "Uganda", "Zambia", "Zimbabwe"
]
#least developed countries as of 2020
ldc_countries = [
    "Afghanistan", "Angola", "Bangladesh", "Benin", "Bhutan",
    "Burkina Faso", "Burundi", "Central African Republic", "Chad",
"Comoros",
    "Democratic Republic of the Congo", "Djibouti", "Eritrea",
"Ethiopia", "Gambia",
    "Guinea", "Guinea-Bissau", "Haiti", "Kiribati", "Lao People's
Democratic Republic",
    "Lesotho", "Liberia", "Madagascar", "Malawi", "Mali",
    "Mauritania", "Mozambique", "Myanmar (Burma)", "Nepal", "Niger",
    "Rwanda", "Sao Tome and Principe", "Senegal", "Sierra Leone",
"Solomon Islands",
    "Somalia", "South Sudan", "Sudan", "Timor-Leste", "Togo",
    "Tuvalu", "Uganda", "United Republic of Tanzania", "Vanuatu",
"Yemen",
    "Zambia"
]
#landlocked countries as of 2020
landlocked_countries = [
    "Afghanistan", "Armenia", "Azerbaijan", "Bhutan", "Bolivia",
```

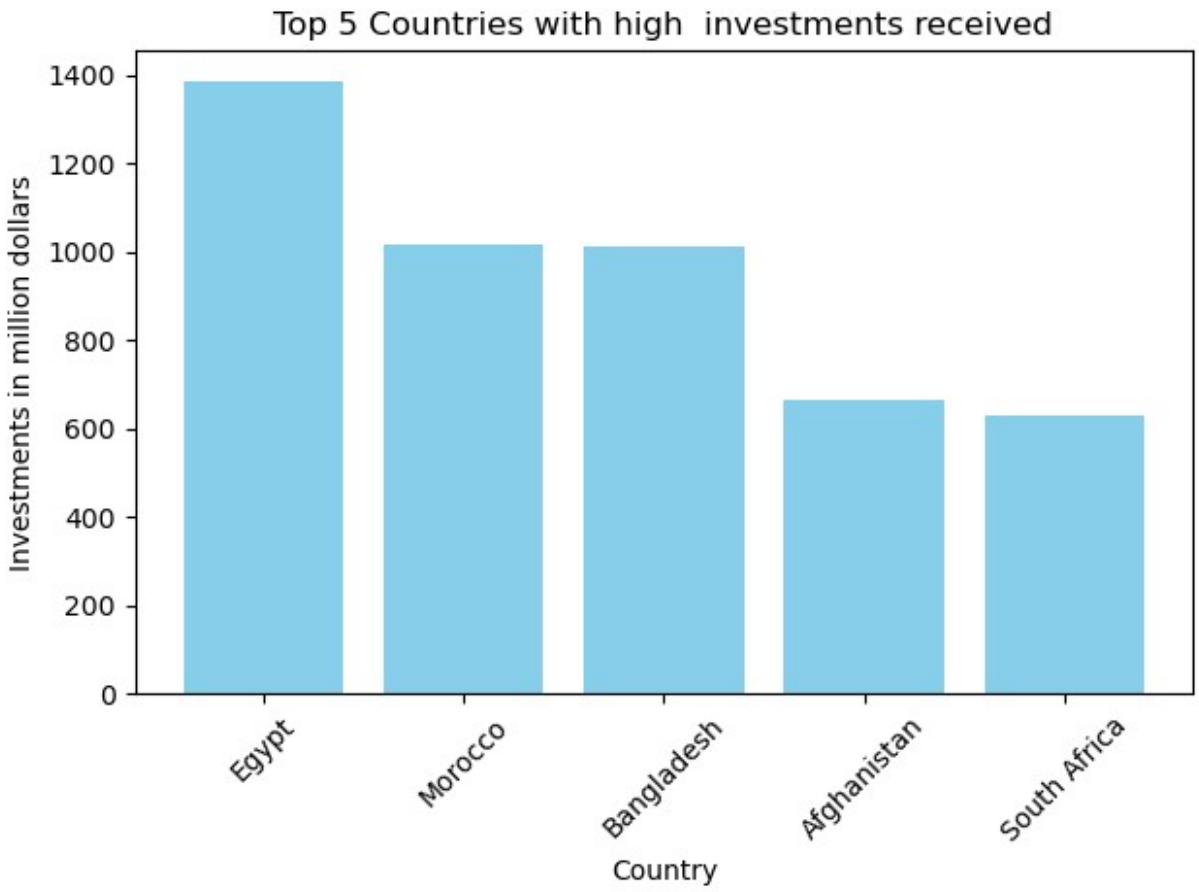
```
    "Botswana", "Burkina Faso", "Burundi", "Central African Republic",
"Chad",
    "Eswatini", "Ethiopia", "Kazakhstan", "Kyrgyzstan", "Lao People's
Democratic Republic",
    "Lesotho", "Malawi", "Mali", "Moldova", "Mongolia",
    "Nepal", "Niger", "Paraguay", "Rwanda", "South Sudan",
    "Sudan", "Tajikistan", "Turkmenistan", "Uganda", "Uzbekistan",
    "Zambia", "Zimbabwe"
]
#small island developing states as of 2020
sids_countries = [
    "Antigua and Barbuda", "The Bahamas", "Barbados", "Cape Verde",
"Comoros",
    "Cuba", "Dominica", "Dominican Republic", "Fiji", "Grenada",
    "Guinea-Bissau", "Guyana", "Haiti", "Jamaica", "Kiribati",
    "Madagascar", "Maldives", "Malta", "Marshall Islands",
"Mauritius",
    "Micronesia (Federated States of)", "Nauru", "Palau", "Papua New
Guinea", "Saint Kitts and Nevis",
    "Saint Lucia", "Saint Vincent and the Grenadines", "Samoa", "Sao
Tome and Principe", "Seychelles",
    "Solomon Islands", "Suriname", "Timor-Leste", "Tonga", "Trinidad
and Tobago",
    "Tuvalu", "Vanuatu"
]


#checking whether the countries  belong to the above categories
Highinvestment=[]
for i in top_max_countries:
    if i in african_countries or i in ldc_countries or i  in
landlocked_countries or i in sids_countries:
        Highinvestment.append(i)
print(Highinvestment)
lowinvestment=[]
for i in top_min_countries:
    if i in african_countries or i in ldc_countries or i  in
landlocked_countries or i in sids_countries:
        lowinvestment.append(i)
print(lowinvestment)


['Egypt', 'Morocco', 'Bangladesh', 'Afghanistan', 'South Africa']
['Malta', 'Equatorial Guinea', 'Saint Kitts and Nevis', 'Antigua and
Barbuda', 'Trinidad and Tobago', 'Saint Vincent and the Grenadines']

import pandas as pd
import matplotlib.pyplot as plt

Hinvestment=top_max[top_max['GeoAreaName'].isin(['Egypt',
```

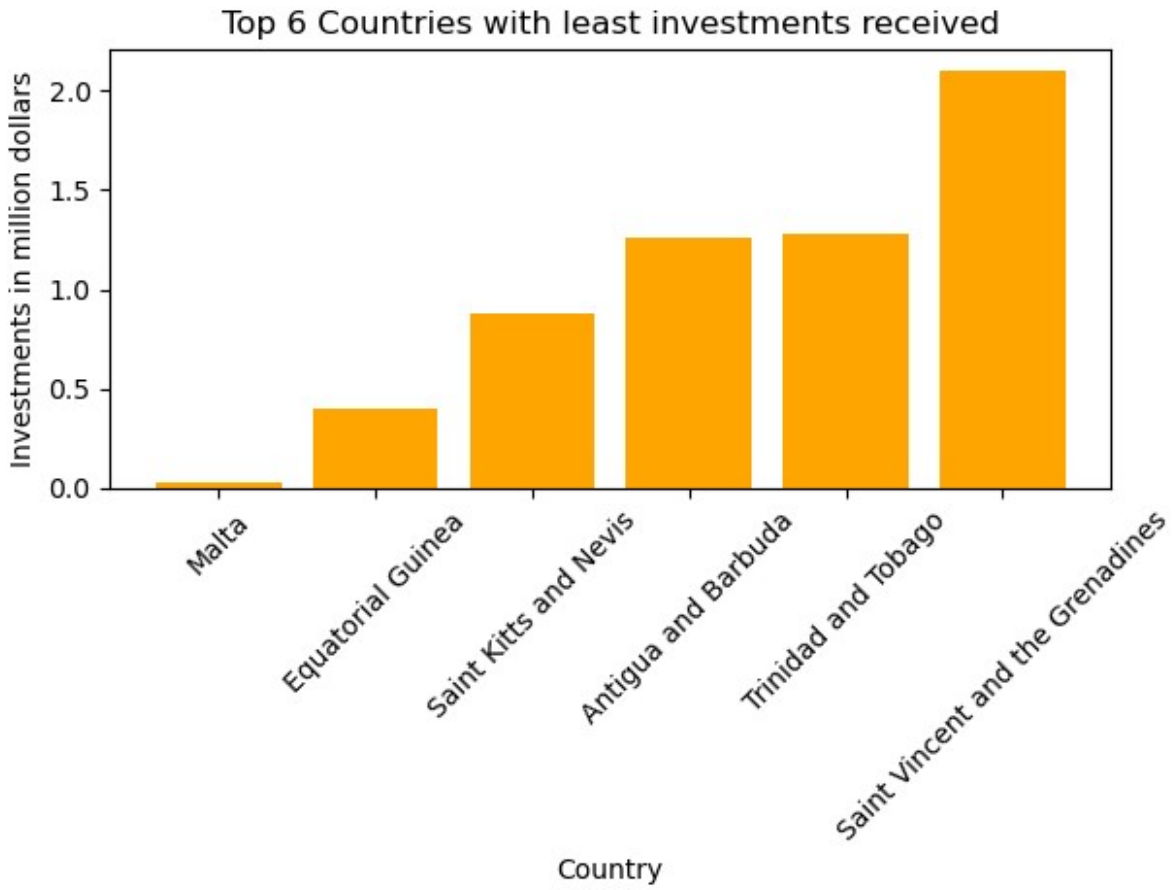
```
'Bangladesh', 'Morocco', 'Afghanistan', 'South Africa'])]
Linvestment=top_min[top_min['GeoAreaName'].isin(['Malta', 'Equatorial
Guinea', 'Saint Kitts and Nevis', 'Antigua and Barbuda', 'Trinidad and
Tobago', 'Saint Vincent and the Grenadines'])]
print(Hinvestment)
print(Linvestment)
```

```
      GeoAreaName        Value
43          Egypt  1384.996701
91         Morocco  1015.429000
10      Bangladesh  1009.765863
0      Afghanistan   664.048038
126   South Africa   628.220525
                          GeoAreaName     Value
81                              Malta  0.024100
45                  Equatorial Guinea  0.402919
113             Saint Kitts and Nevis  0.877578
5                 Antigua and Barbuda  1.256256
139               Trinidad and Tobago  1.279455
115  Saint Vincent and the Grenadines  2.100704
```

```
plt.bar(Hinvestment['GeoAreaName'],Hinvestment['Value'],
color='skyblue')

# Add labels and title
plt.xlabel('Country')
plt.ylabel('Investments in million dollars')
plt.title('Top 5 Countries with high  investments received')

# Rotate the x-axis labels for better readability (optional)
plt.xticks(rotation=45)

# Show the plot
plt.tight_layout()  # Adjust layout for better display

print('''Please note these countries belong to either
african countries or least developed countries or landlocked countries
or small islands''')
plt.show()
plt.bar(Linvestment['GeoAreaName'],Linvestment['Value'],
color='orange')

# Add labels and title
plt.xlabel('Country')
plt.ylabel('Investments in million dollars')
plt.title('Top 6 Countries with least investments received')

# Rotate the x-axis labels for better readability (optional)
plt.xticks(rotation=45)
```

```
# Show the plot
plt.tight_layout()
plt.show()

Please note these countries belong to either
african countries or least developed countries or landlocked countries
or small islands
```

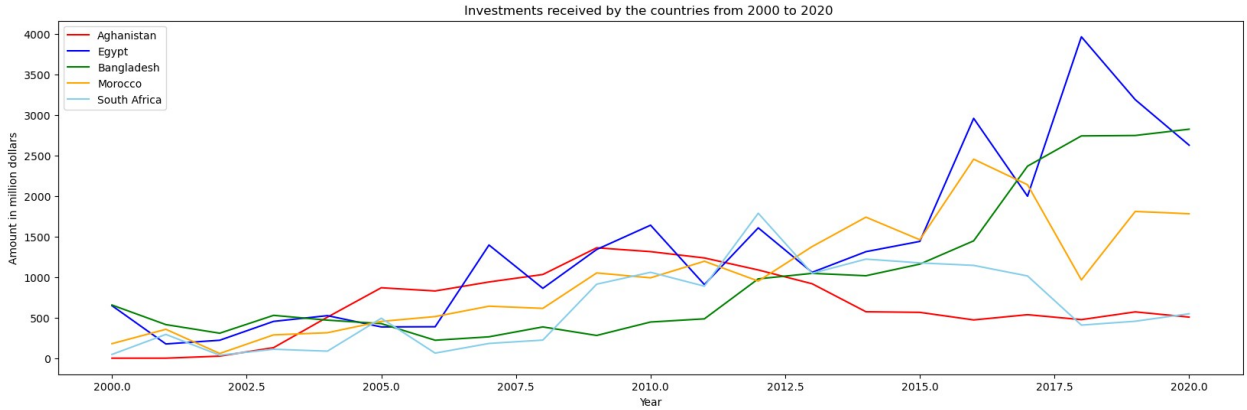




In general, low investments are due to small market size,unfavourable global conditions,sectoral concentration,regulatory environment,global competitiveness,political unstability,poor market accessibility and poor economic performance.

```
#displays the invesments of the countries over the period of 2000 to
2020
t_high=df2[(df2['Indicator']=='9.a.1')]

t_high=t_high[(t_high['GeoAreaName']=='Egypt') |
( t_high['GeoAreaName']=='Bangladesh')|
( t_high['GeoAreaName']=='Morocco')|
(t_high['GeoAreaName']=='Afghanistan')|(t_high['GeoAreaName']=='South
Africa')]

t_high=t_high[['GeoAreaName','TimePeriod','Value']]
af_high=t_high[t_high['GeoAreaName']=='Afghanistan']
eg_high=t_high[t_high['GeoAreaName']=='Egypt']
ba_high=t_high[t_high['GeoAreaName']=='Bangladesh']
Mo_high=t_high[t_high['GeoAreaName']=='Morocco']
sa_high=t_high[t_high['GeoAreaName']=='South Africa']
plt.figure(figsize=(20,6))
plt.plot(af_high['TimePeriod'],af_high['Value'],
```

```
c='red',label='Aghanistan')
plt.plot(eg_high['TimePeriod'],eg_high['Value'],
c='blue',label='Egypt')
plt.plot(ba_high['TimePeriod'],ba_high['Value'],
c='green',label='Bangladesh')
plt.plot(Mo_high['TimePeriod'],Mo_high['Value'],
c='orange',label='Morocco')
plt.plot(sa_high['TimePeriod'],sa_high['Value'],
c='skyblue',label='South Africa')
plt.title('Investments received by the countries from 2000 to 2020 ')
plt.xlabel('Year')
plt.ylabel('Amount in million dollars')
plt.legend()
plt.show()
```



The trend analysis is done for top average investments received for the period of 2000 to 2020 and these countries belong to above categories. Egypt,Bangladesh,Morocco have seen significant growth in investments over the years. Afghanistan received investments till 2010 and since US entered for war ,the investments declined significantly after 2010.

Goal 2: 9.B Support domestic technology development, research and innovation in developing countries, including by ensuring a conducive policy environment for, inter alia, industrial diversification and value addition to commodities

```
Economy=df2[(df2['Indicator']=='9.b.1')]
Economy.head()
```

	Goal	Target	Indicator	SeriesCode	\
41499	9	9.b	9.b.1	NV_IND_TECH	
41500	9	9.b	9.b.1	NV_IND_TECH	
41501	9	9.b	9.b.1	NV_IND_TECH	
41502	9	9.b	9.b.1	NV_IND_TECH	
41503	9	9.b	9.b.1	NV_IND_TECH	

```
                                        SeriesDescription  GeoAreaCode
\
41499  Proportion of medium and high-tech manufacturi...             4

41500  Proportion of medium and high-tech manufacturi...             4

41501  Proportion of medium and high-tech manufacturi...             4

41502  Proportion of medium and high-tech manufacturi...             4

41503  Proportion of medium and high-tech manufacturi...             4


       GeoAreaName  TimePeriod  Value  Time_Detail  BasePeriod  \
41499  Afghanistan        2000  13.59         2000         0.0
41500  Afghanistan        2001  13.59         2001         0.0
41501  Afghanistan        2002  13.59         2002         0.0
41502  Afghanistan        2003  13.59         2003         0.0
41503  Afghanistan        2004  14.28         2004         0.0

                                                 Source FootNote  \
41499  UNIDO CIP 2021 Database. Available at https://...        0
41500  UNIDO CIP 2021 Database. Available at https://...        0
41501  UNIDO CIP 2021 Database. Available at https://...        0
41502  UNIDO CIP 2021 Database. Available at https://...        0
41503  UNIDO CIP 2021 Database. Available at https://...        0

      Mode of transportation Location Reporting Type Observation
Status  \
41499                      0        0              G
A
41500                      0        0              G
A
41501                      0        0              G
A
41502                      0        0              G
A
41503                      0        0              G
A

         Units Activity Nature
41499  PERCENT  ISIC4_C     CA
41500  PERCENT  ISIC4_C     CA
41501  PERCENT  ISIC4_C     CA
41502  PERCENT  ISIC4_C     CA
41503  PERCENT  ISIC4_C     CA

Economy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3080 entries, 41499 to 44578
Data columns (total 20 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Goal                     3080 non-null   int64  
 1   Target                   3080 non-null   object 
 2   Indicator                3080 non-null   object 
 3   SeriesCode               3080 non-null   object 
 4   SeriesDescription        3080 non-null   object 
 5   GeoAreaCode              3080 non-null   int64  
 6   GeoAreaName              3080 non-null   object 
 7   TimePeriod               3080 non-null   int64  
 8   Value                    3080 non-null   float64
 9   Time_Detail              3080 non-null   int64  
 10  BasePeriod               3080 non-null   float64
 11  Source                   3080 non-null   object 
 12  FootNote                 3080 non-null   object 
 13  Mode of transportation   3080 non-null   object 
 14  Location                 3080 non-null   object 
 15  Reporting Type           3080 non-null   object 
 16  Observation Status       3080 non-null   object 
 17  Units                    3080 non-null   object 
 18  Activity                 3080 non-null   object 
 19  Nature                   3080 non-null   object 
dtypes: float64(2), int64(4), object(14)
memory usage: 505.3+ KB
```

```
#To find average value addition of top 10 countries to the economy
Average_contribution=Economy.groupby('GeoAreaName')
['Value'].mean().reset_index()
print(Average_contribution)
```

```
                            GeoAreaName    Value
0                           Afghanistan   9.1570
1                               Albania   9.9195
2                               Algeria   8.1885
3                                Angola   4.2520
4                             Argentina  26.9250
..                                  ...      ...
149  Venezuela (Bolivarian Republic of)  34.2800
150                           Viet Nam  29.3245
151                              Yemen   2.2745
152                             Zambia  10.5135
153                           Zimbabwe  16.0590

[154 rows x 2 columns]
```

```
max_cont =Average_contribution.nlargest(60, 'Value')
min_cont=Average_contribution.nsmallest(10,'Value')
```

```
top_max_con_countries=list(max_cont['GeoAreaName'])
top_min_con_countries=list(min_cont['GeoAreaName'])

##checking whether the countries  belong to the
african,least_developed,land_locked,small_islands categories
Highcontribution=[]
for i in top_max_con_countries:
    if i in african_countries or i in ldc_countries or i  in
landlocked_countries or i in sids_countries:
        Highcontribution.append(i)
print(Highcontribution)
lowcontribution=[]
for i in top_min_con_countries:
    if i in african_countries or i in ldc_countries or i  in
landlocked_countries or i in sids_countries:
        lowcontribution.append(i)
print(lowcontribution)
```

```
['Barbados', 'Trinidad and Tobago', 'Malta', 'Nigeria', 'Cabo Verde']
['Tonga', 'Eswatini', 'Yemen', 'Maldives', 'Tajikistan', 'Burundi',
'Gambia']
```

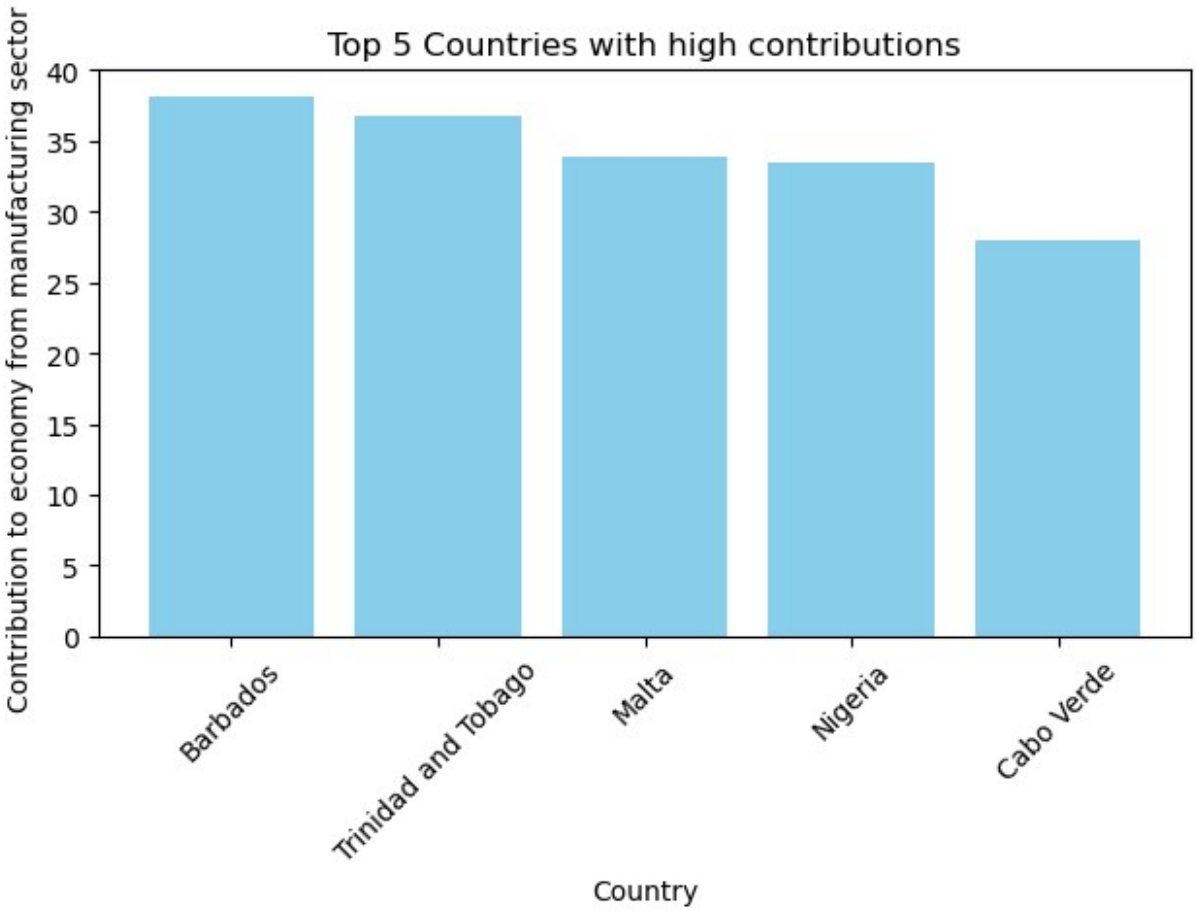
```
Hcontribution=max_cont[max_cont['GeoAreaName'].isin(['Barbados',
'Trinidad and Tobago', 'Malta', 'Nigeria', 'Cabo Verde'])]
Lcontribution=min_cont[min_cont['GeoAreaName'].isin(['Tonga',
'Eswatini', 'Yemen', 'Maldives', 'Tajikistan', 'Burundi', 'Gambia'])]
print(Hcontribution)
print(Lcontribution)
```

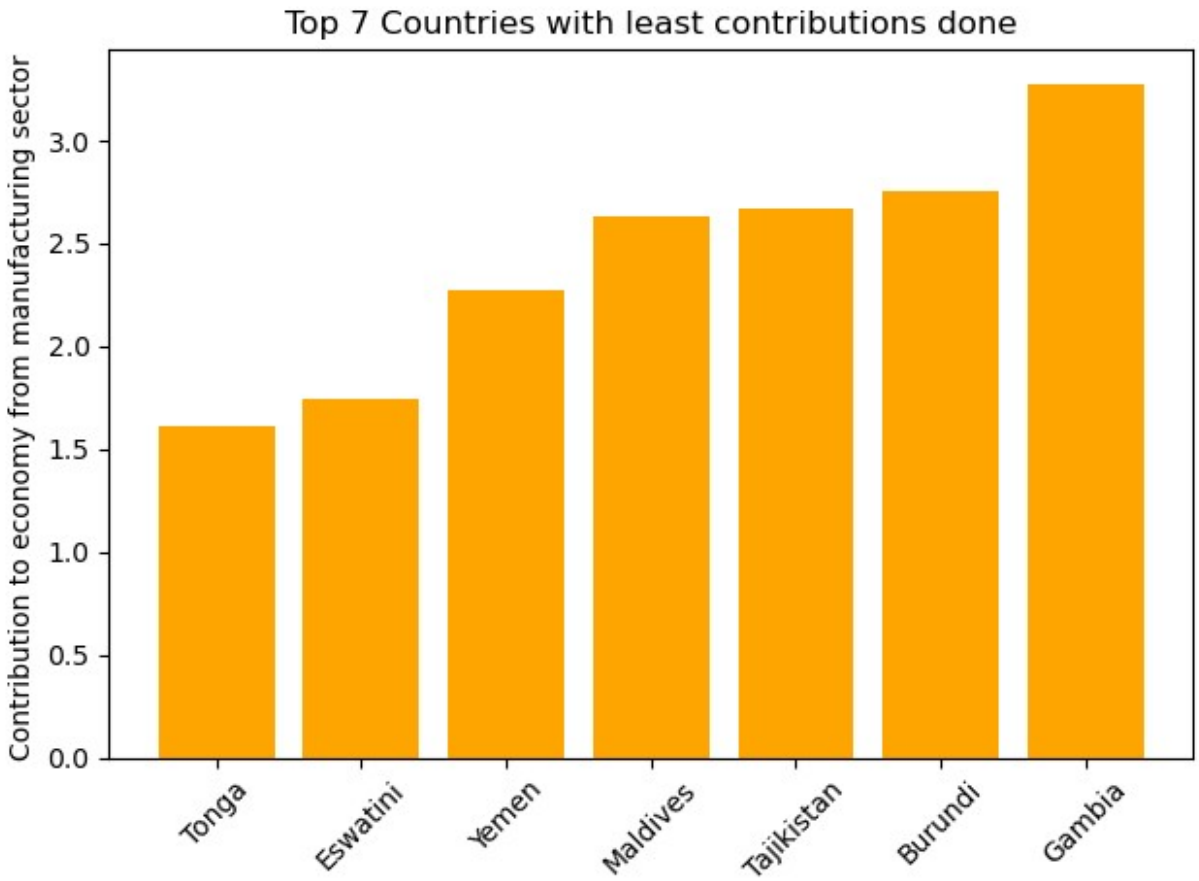
```
             GeoAreaName    Value
12              Barbados  38.1100
138  Trinidad and Tobago  36.7715
87                 Malta  33.8790
100              Nigeria  33.4400
24            Cabo Verde  27.9750
    GeoAreaName   Value
137       Tonga  1.6100
47     Eswatini  1.7445
151       Yemen  2.2745
86     Maldives  2.6300
135  Tajikistan  2.6730
23      Burundi  2.7540
53       Gambia  3.2780
```

```
plt.bar(Hcontribution['GeoAreaName'],Hcontribution['Value'],
color='skyblue')

# Add labels and title
plt.xlabel('Country')
plt.ylabel('Contribution to economy from manufacturing sector ')
```

```
plt.title('Top 5 Countries with high contributions')

# Rotate the x-axis labels for better readability (optional)
plt.xticks(rotation=45)

# Show the plot
plt.tight_layout()  # Adjust layout for better display

print('''Please note these countries belong to either
african countries or least developed countries or landlocked countries
or small islands''')
plt.show()
plt.bar(Lcontribution['GeoAreaName'],Lcontribution['Value'],
color='orange')

# Add labels and title
plt.ylabel('Contribution to economy from manufacturing sector ')
plt.title('Top 7 Countries with least contributions done')

# Rotate the x-axis labels for better readability (optional)
plt.xticks(rotation=45)

# Show the plot
plt.tight_layout()
plt.show()
```

```
Please note these countries belong to either
african countries or least developed countries or landlocked countries
or small islands
```


Top 5 Countries with high contributions
Contribution to economy from manufacturing sector
0
5
10
15
20
25
30
35
40
Barbados
Trinidad and Tobago
Malta
Nigeria
Cabo Verde
Country



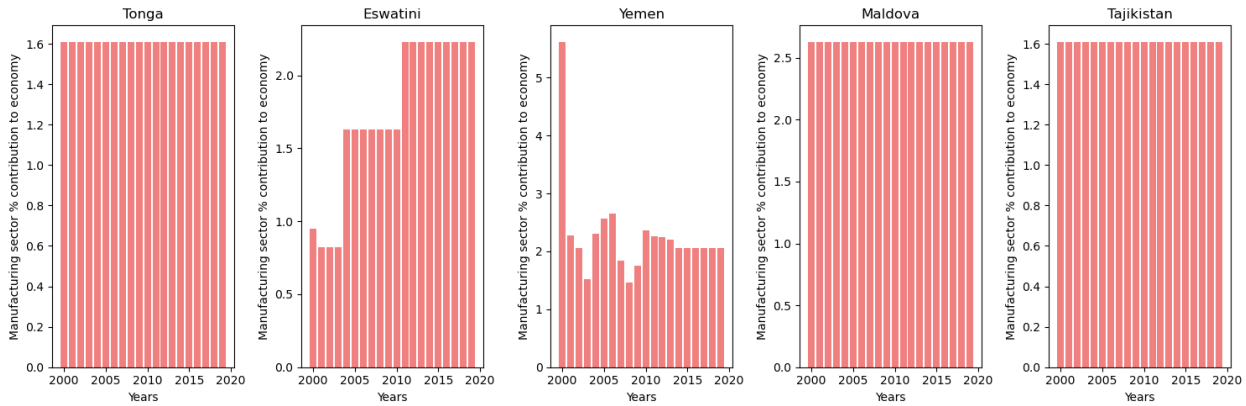
These countries have poor infrastructure and wrong policies related to technology,manufacturing sectors thus leading to poor contribution to economic growth.

```
cont_low=Economy[Economy['GeoAreaName'].isin(['Tonga', 'Eswatini',
'Yemen', 'Maldives', 'Tajikistan'])]
cont_low=cont_low[['GeoAreaName','TimePeriod','Value']]
print(cont_low['GeoAreaName'].unique())
Tong_low=cont_low[cont_low['GeoAreaName']=='Tonga']
Eswat_low=cont_low[cont_low['GeoAreaName']=='Eswatini']
Yeme_low=cont_low[cont_low['GeoAreaName']=='Yemen']
Mald_low=cont_low[cont_low['GeoAreaName']=='Maldives']
Taj_low=cont_low[cont_low['GeoAreaName']=='Tajikistan']
Years=list(cont_low['TimePeriod'].unique())
```

```
['Maldives' 'Eswatini' 'Tajikistan' 'Tonga' 'Yemen']
```

```
values1 = list(Tong_low['Value'])
values2 = list(Eswat_low['Value'])
values3 = list(Yeme_low['Value'])
values4 = list(Mald_low['Value'])
values5 = list(Taj_low['Value'])
```

```
# Create a figure and subplots
fig, axs = plt.subplots(1, 5, figsize=(15, 5))  # 1 row, 2 columns of
subplots

# Create bar chart subplot 1
axs[0].bar(Years, values1, color='lightcoral')
axs[0].set_title('Tonga')
axs[0].set_xlabel('Years')
axs[0].set_ylabel('Manufacturing sector % contribution to economy')

# Create bar chart subplot 2
axs[1].bar(Years, values2, color='lightcoral')
axs[1].set_title('Eswatini')
axs[1].set_xlabel('Years')
axs[1].set_ylabel('Manufacturing sector % contribution to economy')

axs[2].bar(Years, values3, color='lightcoral')
axs[2].set_title('Yemen')
axs[2].set_xlabel('Years')
axs[2].set_ylabel('Manufacturing sector % contribution to economy')

axs[3].bar(Years,values4, color='lightcoral')
axs[3].set_title('Maldova')
axs[3].set_xlabel('Years')
axs[3].set_ylabel('Manufacturing sector % contribution to economy')
axs[4].bar(Years, values1, color='lightcoral')
axs[4].set_title('Tajikistan')
axs[4].set_xlabel('Years')
axs[4].set_ylabel('Manufacturing sector % contribution to economy')
# Adjust layout for subplots
plt.tight_layout()

# Show the plot
plt.show()
```



Some of the countries like Tongo,Eswatini,Yemen ,Maldova,Tajikistan's manufacturing sector contribution to its economy is found.

Interpretation: Eswatini has an upward trend which is a good sign for its economic growth,Yemen has downward trend which is not a good sign and other countries like Tongo,Maldova and Tajikistan have maintained the slope which is not a good sign because they are not trying to increase their growth.

Goal 3- 9.C Significantly increase access to information and communications technology and strive to provide universal and affordable access to the Internet in least developed countries by 2020

```
internet=df[df['Indicator']=='9.c.1']
twoG_internet=internet[internet['SeriesCode']=='IT_MOB_2GNTWK']
twoG_internet=twoG_internet.groupby('GeoAreaName')
['Value'].mean().reset_index()
min_2Gusers=twoG_internet.nsmallest(10,'Value')
print(min_2Gusers)
min_2Guserscountries=list(min_2Gusers['GeoAreaName'])
```

```
                                GeoAreaName      Value
200                                  Tuvalu  27.444444
54   Democratic People's Republic of Korea  36.538462
182                           South Sudan  39.433333
164                          Saint Helena  49.000000
37                Central African Republic  49.032308
55        Democratic Republic of the Congo  50.941176
116                            Madagascar  51.512857
64                                Eritrea  54.850000
142                                 Niger  56.454545
198                            Turkmenistan  57.860251
```

```
##checking whether the countries  belong to the
african,least_developed,land_locked,small_islands categories

low2gusers=[]
for i in min_2Guserscountries:
    if i in african_countries or i in ldc_countries or i  in
landlocked_countries or i in sids_countries:
        low2gusers.append(i)
print(low2gusers)
```

```
['Tuvalu', 'South Sudan', 'Central African Republic', 'Democratic
Republic of the Congo', 'Madagascar', 'Eritrea', 'Niger',
'Turkmenistan']
```

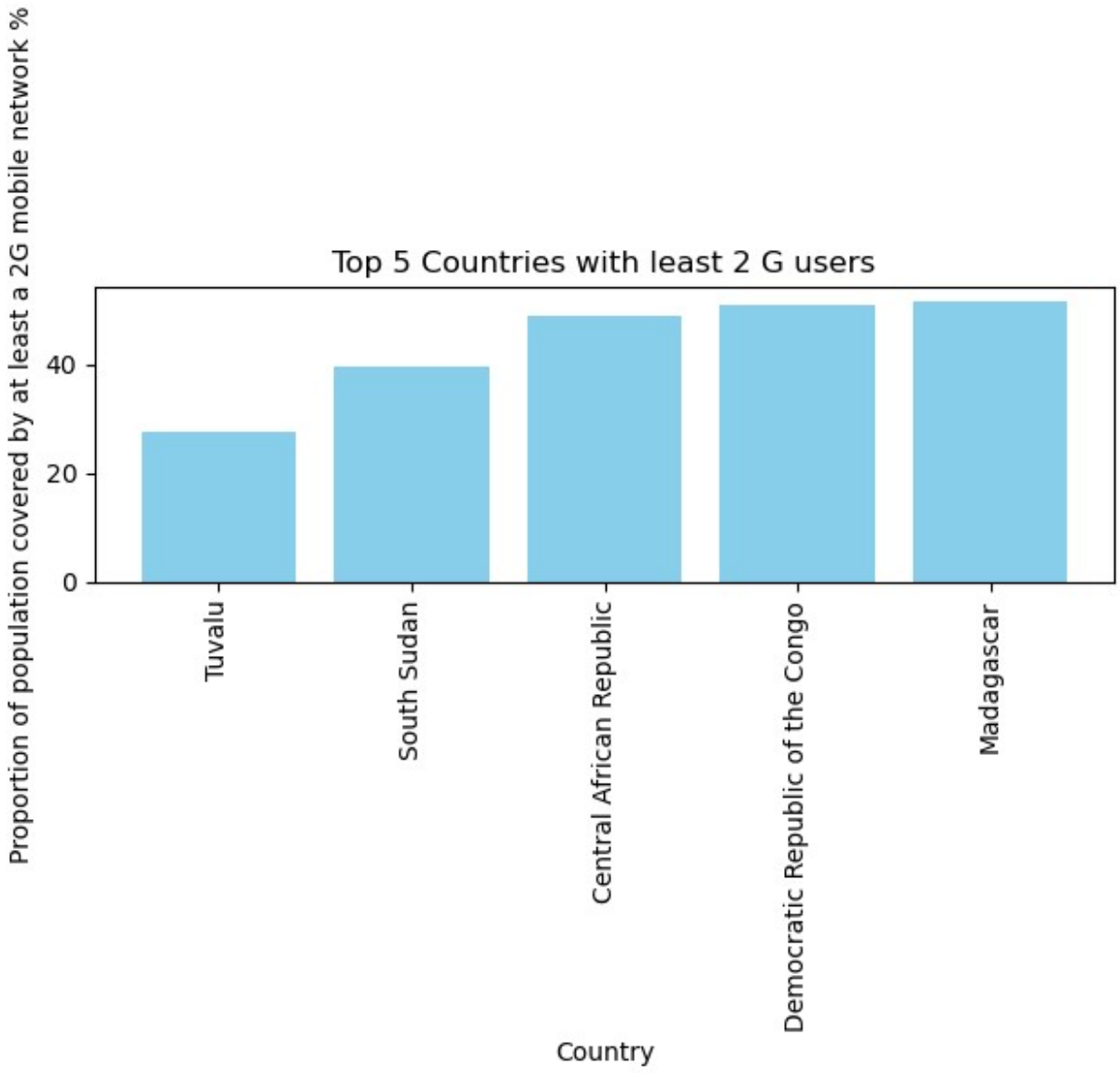
```
Low_2g=min_2Gusers[min_2Gusers['GeoAreaName'].isin(['Tuvalu', 'South
Sudan', 'Central African Republic', 'Democratic Republic of the
Congo', 'Madagascar'])]
print(Low_2g)
```

```
                          GeoAreaName      Value
200                           Tuvalu  27.444444
```

```
182                         South Sudan  39.433333
37            Central African Republic  49.032308
55   Democratic Republic of the Congo  50.941176
116                          Madagascar  51.512857
```

```
plt.bar(Low_2g['GeoAreaName'],Low_2g['Value'], color='skyblue')

# Add labels and title
plt.xlabel('Country')
plt.ylabel('Proportion of population covered by at least a 2G mobile
network %')
plt.title('Top 5 Countries with least 2 G users')

# Rotate the x-axis labels for better readability (optional)
plt.xticks(rotation=90)

# Show the plot
plt.tight_layout()  # Adjust layout for better display

print('''Please note these countries belong to either
african countries or least developed countries or landlocked countries
or small islands''')
plt.show()
```

```
Please note these countries belong to either
african countries or least developed countries or landlocked countries
or small islands
```




```
threeG_internet=internet[internet['SeriesCode']=='IT_MOB_3GNTWK']
three_internet=threeG_internet.groupby('GeoAreaName')
['Value'].mean().reset_index()
min_3Gusers=three_internet.nsmallest(10,'Value')
print(min_3Gusers)
min_3Guserscountries=list(min_3Gusers['GeoAreaName'])
print(min_3Guserscountries)
```

```
                         GeoAreaName      Value
64                           Eritrea   0.000000
146                             Niue   0.000000
129  Micronesia (Federated States of)  12.500000
220         Wallis and Futuna Islands  13.285000
191                 State of Palestine  15.363636
```

```
188                         South Sudan  15.875000
171                         Saint Lucia  15.900000
144                               Niger  17.333333
35                             Cameroon  18.870000
31                           Burkina Faso  19.100000
['Eritrea', 'Niue', 'Micronesia (Federated States of)', 'Wallis and
Futuna Islands', 'State of Palestine', 'South Sudan', 'Saint Lucia',
'Niger', 'Cameroon', 'Burkina Faso']
```

```
low3gusers=[]
for i in min_3Guserscountries:
    if i in african_countries or i in ldc_countries or i  in
landlocked_countries or i in sids_countries:
        low3gusers.append(i)
print(low3gusers)
```

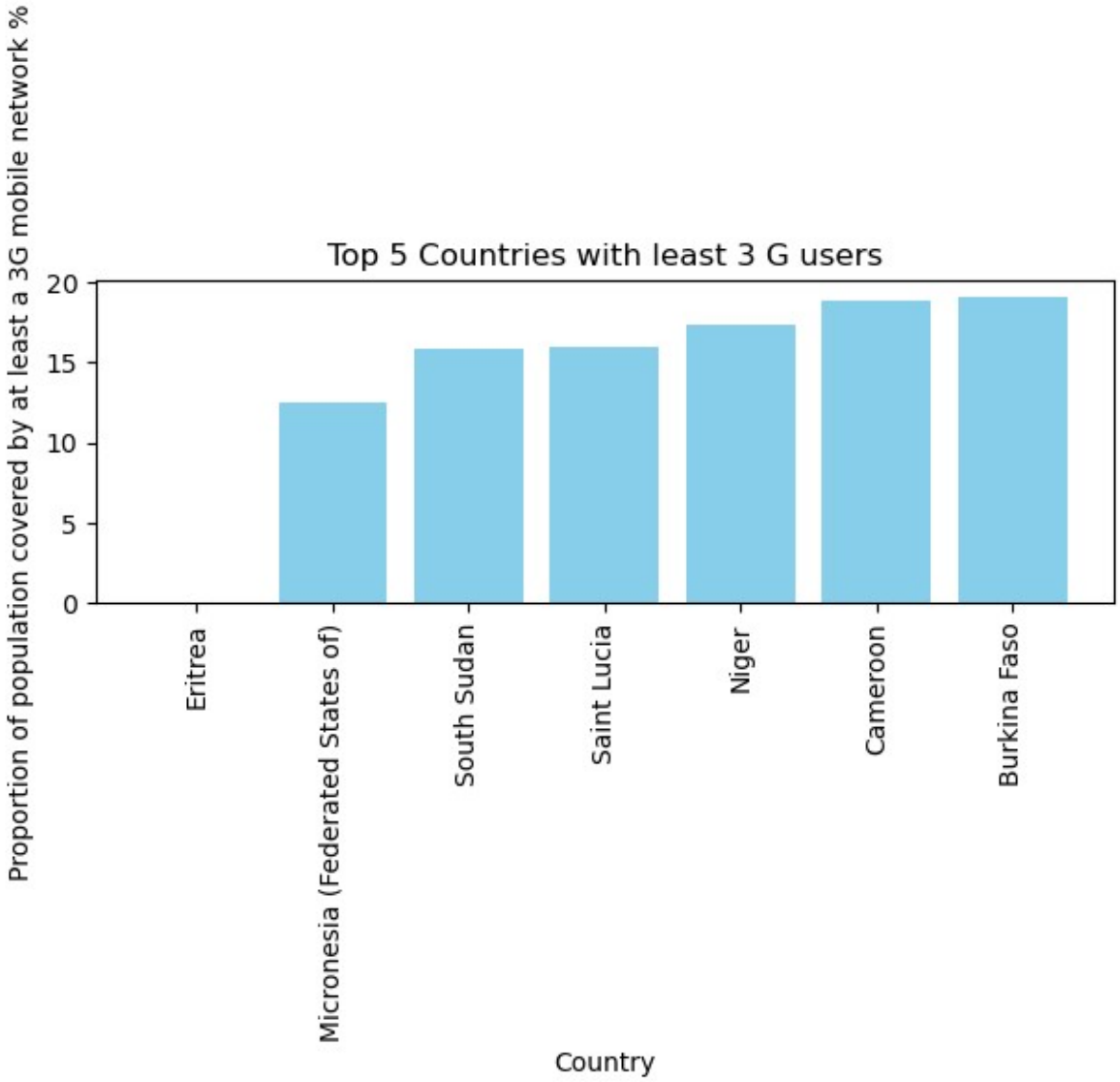
```
['Eritrea', 'Micronesia (Federated States of)', 'South Sudan', 'Saint
Lucia', 'Niger', 'Cameroon', 'Burkina Faso']
```

```
Low_3g=min_3Gusers[min_3Gusers['GeoAreaName'].isin(['Eritrea',
'Micronesia (Federated States of)', 'South Sudan', 'Saint Lucia',
'Niger','Cameroon', 'Burkina Faso'])]
print(Low_3g)
```

```
                           GeoAreaName      Value
64                             Eritrea   0.000000
129  Micronesia (Federated States of)  12.500000
188                         South Sudan  15.875000
171                         Saint Lucia  15.900000
144                               Niger  17.333333
35                             Cameroon  18.870000
31                           Burkina Faso  19.100000
```

```
plt.bar(Low_3g['GeoAreaName'],Low_3g['Value'], color='skyblue')

# Add labels and title
plt.xlabel('Country')
plt.ylabel('Proportion of population covered by at least a 3G mobile
network %')
plt.title('Top 5 Countries with least 3 G users')

# Rotate the x-axis labels for better readability (optional)
plt.xticks(rotation=90)

# Show the plot
plt.tight_layout()  # Adjust layout for better display

print('''Please note these countries belong to either
african countries or least developed countries or landlocked countries
```

```
or small islands''')
plt.show()

Please note these countries belong to either
african countries or least developed countries or landlocked countries
or small islands
```




```
fourG_internet=internet[internet['SeriesCode']=='IT_MOB_4GNTWK']
four_internet=fourG_internet.groupby('GeoAreaName')
['Value'].mean().reset_index()

min_4Gusers=four_internet.nsmallest(10,'Value')
print(min_4Gusers)
```

```
min_4Guserscountries=list(min_4Gusers['GeoAreaName'])
print(min_4Guserscountries)
```

```
                   GeoAreaName      Value
0                   Afghanistan   8.428571
1                       Albania  68.482857
2                       Algeria  24.084444
3                American Samoa   0.000000
4                       Andorra  73.333333
..                          ...        ...
214                    Viet Nam  55.770000
215  Wallis and Futuna Islands  10.000000
216                      Yemen   0.000000
217                     Zambia  33.122857
218                   Zimbabwe  25.051429

[219 rows x 2 columns]
                              GeoAreaName  Value
3                           American Samoa    0.0
54   Democratic People's Republic of Korea    0.0
63                        Equatorial Guinea    0.0
64                                  Eritrea    0.0
88                             Guinea-Bissau    0.0
125                              Mauritania    0.0
128       Micronesia (Federated States of)    0.0
165                   Saint Kitts and Nevis    0.0
170                   Sao Tome and Principe    0.0
186                      State of Palestine    0.0
['American Samoa', "Democratic People's Republic of Korea",
'Equatorial Guinea', 'Eritrea', 'Guinea-Bissau', 'Mauritania',
'Micronesia (Federated States of)', 'Saint Kitts and Nevis', 'Sao Tome
and Principe', 'State of Palestine']
```

```
low4gusers=[]
for i in min_4Guserscountries:
    if i in african_countries or i in ldc_countries or i  in
landlocked_countries or i in sids_countries:
        low4gusers.append(i)
print(low4gusers)
```

```
['Equatorial Guinea', 'Eritrea', 'Guinea-Bissau', 'Mauritania',
'Micronesia (Federated States of)', 'Saint Kitts and Nevis', 'Sao Tome
and Principe']
```

```
Low_4g=min_4Gusers[min_4Gusers['GeoAreaName'].isin(['Equatorial
Guinea', 'Eritrea', 'Guinea-Bissau', 'Mauritania', 'Micronesia
(Federated States of)', 'Saint Kitts and Nevis', 'Sao Tome and
Principe'])]
print('Lowest 4G users in countries which belong to the
```

```
african,least_developed,land_locked,small_islands categories')
print(Low_4g)
```

```
Lowest 4G users in countries  belong to the
african,least_developed,land_locked,small_islands categories
                          GeoAreaName  Value
63                  Equatorial Guinea    0.0
64                            Eritrea    0.0
88                        Guinea-Bissau    0.0
125                          Mauritania    0.0
128  Micronesia (Federated States of)    0.0
165             Saint Kitts and Nevis    0.0
170             Sao Tome and Principe    0.0
```

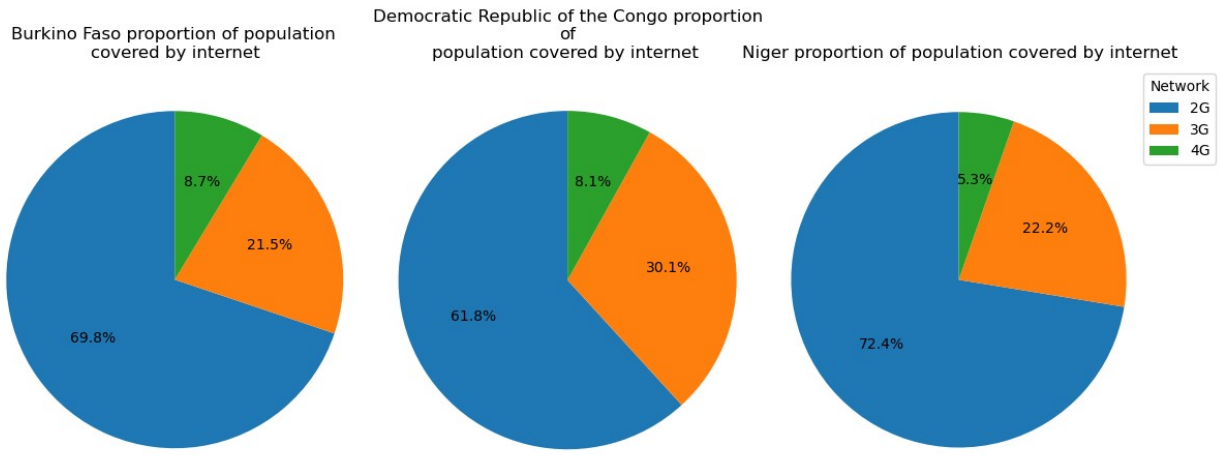
Comparing few countries(Burkina Faso, Niger,Democratic Republic of the Congo)2g,3g,and 4g users in countries which belong to the african,least_developed,land_locked,small_islands categories

```
twog_internet=twoG_internet[twoG_internet['GeoAreaName'].isin(['Burkin
a Faso', 'Niger','Democratic Republic of the Congo'])]
print(twog_internet)
threeg_internet=three_internet[three_internet['GeoAreaName'].isin(['Bu
rkina Faso', 'Niger','Democratic Republic of the Congo'])]
print(threeg_internet)
fourg_internet=four_internet[four_internet['GeoAreaName'].isin(['Burki
na Faso', 'Niger','Democratic Republic of the Congo'])]
print(fourg_internet)
```

```
                          GeoAreaName      Value
30                        Burkina Faso  62.010000
55   Democratic Republic of the Congo  50.941176
142                             Niger  56.454545
                          GeoAreaName      Value
31                        Burkina Faso  19.100000
55   Democratic Republic of the Congo  24.833333
144                             Niger  17.333333
                          GeoAreaName     Value
30                        Burkina Faso  7.691429
55   Democratic Republic of the Congo  6.666667
142                             Niger  4.166667
```

```
values=zip(list(twog_internet['Value']),list(threeg_internet['Value'])
,list(fourg_internet['Value']))
values=list(values)
value1=list(values[0])
value2=list(values[1])
value3=list(values[2])

labels = ['2G', '3G', '4G']
```

```
# Create a figure with subplots
fig, axs = plt.subplots(1, 3, figsize=(12, 5))  # 1 row, 2 columns of
subplots

# Create pie chart subplot 1
axs[0].pie(value1, autopct='%1.1f%%', startangle=90)
axs[0].set_title('''Burkino Faso proportion of population
covered by internet''')

# Create pie chart subplot 2
axs[1].pie(value2,  autopct='%1.1f%%', startangle=90)
axs[1].set_title('''Democratic Republic of the Congo proportion
of
population covered by internet ''')
axs[2].pie(value3, autopct='%1.1f%%', startangle=90)
axs[2].set_title('''Niger proportion of population covered by
internet''')

# Equal aspect ratio ensures that pie is drawn as a circle
axs[0].axis('equal')
axs[1].axis('equal')
axs[2].axis('equal')
plt.legend(labels, title='Network', loc='upper right',
bbox_to_anchor=(1.2, 1))

# Show the plot
plt.tight_layout()
plt.show()
```



Analysis is done on three countries Burkino Faso , Congo , Niger on how their population is covered with 2g,3g and 4g networks separately.

Interpretation: It is found that for all the three countries still the majority of population is not covered by advanced networks.

Conclusion: Network plays an important role in accessing information, communication made faster and easier thus making business,education,scientific sectors to grow faster.