In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import ydata_profiling as pp
```

In [2]:
```
#Fetching the food production data
df=pd.read_csv(r"C:\Users\Chinenye Claire\Desktop\Hamoye Internship\Africa Food Production.csv")
```

In [3]:
```
df.head()
```

Out[3]:

	Country	Item	Year	Value
0	Algeria	Wheat and products	2004	2731
1	Algeria	Wheat and products	2005	2415
2	Algeria	Wheat and products	2006	2688
3	Algeria	Wheat and products	2007	2319
4	Algeria	Wheat and products	2008	1111

In [4]:
```
#Fetching the food consumption/supply data
df1=pd.read_csv(r"C:\Users\Chinenye Claire\Desktop\Hamoye Internship\Africa Food Supply.csv")
```

In [5]:
```
df1.head()
```

Out[5]:

	Country	Year	Value
0	Algeria	2004	2987
1	Algeria	2005	2958
2	Algeria	2006	3047
3	Algeria	2007	3041
4	Algeria	2008	3048

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23110 entries, 0 to 23109
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   Country  23110 non-null  object
 1   Item     23110 non-null  object
 2   Year     23110 non-null  int64 
 3   Value    23110 non-null  int64 
dtypes: int64(2), object(2)
memory usage: 722.3+ KB
```

In [7]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 450 entries, 0 to 449
Data columns (total 3 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   Country  450 non-null    object
 1   Year     450 non-null    int64 
 2   Value    450 non-null    int64 
dtypes: int64(2), object(1)
memory usage: 10.7+ KB
```

In [8]:
```
#Checking for missing values in the food production report
df.isnull().sum()
```

Out[8]:
```
Country    0
Item       0
Year       0
Value      0
dtype: int64
```

In [9]:
```
#Checking for missing values in the consumption report
df1.isnull().sum()
```

Out[9]:
```
Country    0
Year       0
Value      0
dtype: int64
```

In [10]:
```
#Total production quantities for each item over the 10-year period
go=df.loc[:, df.columns!='Year']
gk=go.groupby(['Item']).sum()
print(gk)
```

```
                          Value
Item                           
Alcohol, Non-Food          3652
Apples and products       21706
Aquatic Animals, Others      13
Aquatic Plants             1378
Bananas                  153785
...                         ...
Tomatoes and products    172192
Vegetables, Other        410403
Wheat and products       229875
Wine                      11219
Yams                     511523

[94 rows x 1 columns]

C:\Users\Chinenye Claire\AppData\Local\Temp\ipykernel_9192\3939341071.py:3: FutureWarning: The
default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a future version, nume
ric_only will default to False. Either specify numeric_only or select only columns which shoul
d be valid for the function.
  gk=go.groupby(['Item']).sum()
```

In [11]:
```
#Kinds of food items produced by countries over the 10-year period
go=df.loc[:, df.columns!='Year']
gq=go.groupby(['Country'])['Item'].unique()
print(gq)
```

```
Country
Algeria                          [Wheat and products, Rice (Milled Equivalent),...
Angola                           [Wheat and products, Rice (Milled Equivalent),...
Benin                            [Rice (Milled Equivalent), Maize and products,...
Botswana                         [Wheat and products, Maize and products, Mille...
Burkina Faso                     [Rice (Milled Equivalent), Maize and products,...
Cabo Verde                       [Maize and products, Cassava and products, Pot...
Cameroon                         [Wheat and products, Rice (Milled Equivalent),...
Central African Republic         [Rice (Milled Equivalent), Maize and products,...
Chad                             [Wheat and products, Rice (Milled Equivalent),...
Congo                            [Rice (Milled Equivalent), Maize and products,...
Cote d'Ivoire                    [Rice (Milled Equivalent), Maize and products,...
Djibouti                         [Maize and products, Sugar cane, Sugar (Raw Eq...
Egypt                            [Wheat and products, Rice (Milled Equivalent),...
Ethiopia                         [Wheat and products, Rice (Milled Equivalent),...
Gabon                            [Rice (Milled Equivalent), Maize and products,...
Gambia                           [Rice (Milled Equivalent), Maize and products,...
Ghana                            [Rice (Milled Equivalent), Maize and products,...
Guinea                           [Rice (Milled Equivalent), Maize and products,...
Guinea-Bissau                    [Rice (Milled Equivalent), Maize and products,...
Kenya                            [Wheat and products, Rice (Milled Equivalent),...
Lesotho                          [Wheat and products, Barley and products, Maiz...
Liberia                          [Rice (Milled Equivalent), Cassava and product...
Madagascar                       [Wheat and products, Rice (Milled Equivalent),...
Malawi                           [Wheat and products, Rice (Milled Equivalent),...
Mali                             [Wheat and products, Rice (Milled Equivalent),...
Mauritania                       [Wheat and products, Rice (Milled Equivalent),...
Mauritius                        [Rice (Milled Equivalent), Maize and products,...
Morocco                          [Wheat and products, Rice (Milled Equivalent),...
Mozambique                       [Wheat and products, Rice (Milled Equivalent),...
Namibia                          [Wheat and products, Maize and products, Mille...
Niger                            [Wheat and products, Rice (Milled Equivalent),...
Nigeria                          [Wheat and products, Rice (Milled Equivalent),...
Rwanda                           [Wheat and products, Rice (Milled Equivalent),...
Sao Tome and Principe            [Maize and products, Cassava and products, Yam...
Senegal                          [Rice (Milled Equivalent), Maize and products,...
Sierra Leone                     [Rice (Milled Equivalent), Maize and products,...
South Africa                     [Wheat and products, Rice (Milled Equivalent),...
Sudan                            [Wheat and products, Rice (Milled Equivalent),...
Swaziland                        [Wheat and products, Rice (Milled Equivalent),...
Togo                             [Rice (Milled Equivalent), Maize and products,...
Tunisia                          [Wheat and products, Barley and products, Oats...
Uganda                           [Wheat and products, Rice (Milled Equivalent),...
United Republic of Tanzania     [Wheat and products, Rice (Milled Equivalent),...
Zambia                           [Wheat and products, Rice (Milled Equivalent),...
Zimbabwe                         [Wheat and products, Rice (Milled Equivalent),...
Name: Item, dtype: object
```

In [12]:
```
#Total value of food production by countries for each year
gl=df.groupby(['Country', 'Year']).sum()
print(gl)
```

```
                Value
Country  Year
Algeria  2004  15536
         2005  15667
         2006  16417
         2007  14763
         2008  13841
...              ...
Zimbabwe 2009   5754
         2010   6777
         2011   7551
         2012   8173
         2013   7914

[450 rows x 1 columns]

C:\Users\Chinenye Claire\AppData\Local\Temp\ipykernel_9192\1833989290.py:2: FutureWarning: The
default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a future version, nume
ric_only will default to False. Either specify numeric_only or select only columns which shoul
d be valid for the function.
  gl=df.groupby(['Country', 'Year']).sum()
```

In [13]:
```
#Total Production per country over the 10 year period
go=df.loc[:, df.columns!='Year']
gm=go.groupby(['Country']).sum()
print(gm)
```

```
                              Value
Country
Algeria                      188650
Angola                       195499
Benin                         88072
Botswana                       4808
Burkina Faso                 100129
Cabo Verde                     1658
Cameroon                     197670
Central African Republic      24921
Chad                          45078
Congo                         25791
Cote d'Ivoire                182486
Djibouti                       666
Egypt                        877498
Ethiopia                     350693
Gabon                         13566
Gambia                         4699
Ghana                        295004
Guinea                        66848
Guinea-Bissau                  7296
Kenya                        256872
Lesotho                        3785
Liberia                       14657
Madagascar                   133071
Malawi                       153717
Mali                          92292
Mauritania                    10039
Mauritius                     51785
Morocco                      271821
Mozambique                   157584
Namibia                       14197
Niger                         86808
Nigeria                     1628030
Rwanda                        95174
Sao Tome and Principe          1152
Senegal                       49414
Sierra Leone                  47737
South Africa                 579592
Sudan                        301584
Swaziland                     62665
Togo                          32155
Tunisia                       92713
Uganda                       278421
United Republic of Tanzania  332802
Zambia                        82701
Zimbabwe                      73316

C:\Users\Chinenye Claire\AppData\Local\Temp\ipykernel_9192\1573185959.py:3: FutureWarning: The
default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a future version, nume
ric_only will default to False. Either specify numeric_only or select only columns which shoul
d be valid for the function.
  gm=go.groupby(['Country']).sum()
```

In [14]:
```
#Total value of food production in Africa Annually
gn=df.groupby(['Year']).sum()
print(gn)
```

```
       Value
Year
2004  663006
2005  691257
2006  718602
2007  705659
2008  736804
2009  746870
2010  786466
2011  812214
2012  841667
2013  872571
```

```
C:\Users\Chinenye Claire\AppData\Local\Temp\ipykernel_9192\71804020.py:2: FutureWarning: The d
efault value of numeric_only in DataFrameGroupBy.sum is deprecated. In a future version, numer
ic_only will default to False. Either specify numeric_only or select only columns which should
be valid for the function.
  gn=df.groupby(['Year']).sum()
```

In [15]:
```
#Total consumption by countries for each year
gh=df1.groupby(['Country', 'Year']).sum()
print(gh)
```

```
               Value
Country  Year
Algeria  2004   2987
         2005   2958
         2006   3047
         2007   3041
         2008   3048
...              ...
Zimbabwe 2009   2147
         2010   2168
         2011   2200
         2012   2197
         2013   2110

[450 rows x 1 columns]
```

In [16]:
```
#Total value of food consumption per country over the 10 year period
gz=df1.loc[:, df1.columns!='Year']
gi=gz.groupby(['Country']).sum()
print(gi)
```

```
                            Value
Country
Algeria                     31118
Angola                      22556
Benin                       25378
Botswana                    22263
Burkina Faso                26072
Cabo Verde                  25514
Cameroon                    24603
Central African Republic    20719
Chad                        20511
Congo                       21530
Cote d'Ivoire               27666
Djibouti                    24165
Egypt                       34580
Ethiopia                    20292
Gabon                       27299
Gambia                      25695
Ghana                       29180
Guinea                      25180
Guinea-Bissau               22963
Kenya                       21453
Lesotho                     25588
Liberia                     21827
Madagascar                  20608
Malawi                      22925
Mali                        27502
Mauritania                  27443
Mauritius                   30543
Morocco                     32967
Mozambique                  21702
Namibia                     21602
Niger                       25024
Nigeria                     26988
Rwanda                      21309
Sao Tome and Principe       24462
Senegal                     23784
Sierra Leone                22291
South Africa                29629
Sudan                       23238
Swaziland                   23171
Togo                        23339
Tunisia                     33055
Uganda                      22205
United Republic of Tanzania 21550
Zambia                      18701
Zimbabwe                    21209
```

In [17]:
```
#Total value of food consumption in Africa each year
gj=df1.groupby(['Year']).sum()
print(gj)
```

```
       Value
Year
2004  107740
2005  108418
2006  109386
2007  110149
2008  110734
2009  111700
2010  112383
2011  113194
2012  113744
2013  113951
```

```
C:\Users\Chinenye Claire\AppData\Local\Temp\ipykernel_9192\725304963.py:2: FutureWarning: The
default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a future version, nume
ric_only will default to False. Either specify numeric_only or select only columns which shoul
d be valid for the function.
  gj=df1.groupby(['Year']).sum()
```

In [18]:
```
#Top producers of selected food items
#What countries have produced the highest total quantity of Beverages, Alcoholic from 2004 to 2
df7=df.query("Item == 'Cassava and products'")
df7.nlargest(25, 'Value')
```

Out[18]:

	Country	Item	Year	Value
15448	Nigeria	Cassava and products	2012	54000
15449	Nigeria	Cassava and products	2013	53000
15447	Nigeria	Cassava and products	2011	52403
15442	Nigeria	Cassava and products	2006	45721
15444	Nigeria	Cassava and products	2008	44582
15443	Nigeria	Cassava and products	2007	43410
15446	Nigeria	Cassava and products	2010	42533
15441	Nigeria	Cassava and products	2005	41565
15440	Nigeria	Cassava and products	2004	38845
15445	Nigeria	Cassava and products	2009	36822
639	Angola	Cassava and products	2013	16412
7779	Ghana	Cassava and products	2013	15990
7778	Ghana	Cassava and products	2012	14547
637	Angola	Cassava and products	2011	14334
7777	Ghana	Cassava and products	2011	14241
636	Angola	Cassava and products	2010	13859
7776	Ghana	Cassava and products	2010	13504
635	Angola	Cassava and products	2009	12828
7775	Ghana	Cassava and products	2009	12231
7774	Ghana	Cassava and products	2008	11351
638	Angola	Cassava and products	2012	10636
7773	Ghana	Cassava and products	2007	10218
13957	Mozambique	Cassava and products	2011	10094
634	Angola	Cassava and products	2008	10057
13958	Mozambique	Cassava and products	2012	10051

In [19]:
```
#Top producers of selected food items
#What countries have produced the highest total quantity of Beverages, Alcoholic from 2004 to 2
df5=df.query("Item == 'Beverages, Alcoholic'")
df5.nlargest(25, 'Value')
```

Out[19]:

	Country	Item	Year	Value
18273	South Africa	Beverages, Alcoholic	2007	427
18272	South Africa	Beverages, Alcoholic	2006	396
18275	South Africa	Beverages, Alcoholic	2009	392
18270	South Africa	Beverages, Alcoholic	2004	376
18271	South Africa	Beverages, Alcoholic	2005	374
18277	South Africa	Beverages, Alcoholic	2011	374
18279	South Africa	Beverages, Alcoholic	2013	371
18278	South Africa	Beverages, Alcoholic	2012	356
18276	South Africa	Beverages, Alcoholic	2010	350
18274	South Africa	Beverages, Alcoholic	2008	335
22968	Zimbabwe	Beverages, Alcoholic	2012	72
22969	Zimbabwe	Beverages, Alcoholic	2013	72
22960	Zimbabwe	Beverages, Alcoholic	2004	62
22961	Zimbabwe	Beverages, Alcoholic	2005	62
22962	Zimbabwe	Beverages, Alcoholic	2006	56
22963	Zimbabwe	Beverages, Alcoholic	2007	54
22964	Zimbabwe	Beverages, Alcoholic	2008	54
22967	Zimbabwe	Beverages, Alcoholic	2011	51
22965	Zimbabwe	Beverages, Alcoholic	2009	50
6139	Egypt	Beverages, Alcoholic	2013	47
6137	Egypt	Beverages, Alcoholic	2011	46
6138	Egypt	Beverages, Alcoholic	2012	46
22966	Zimbabwe	Beverages, Alcoholic	2010	46
6130	Egypt	Beverages, Alcoholic	2004	41
6136	Egypt	Beverages, Alcoholic	2010	39

In [20]:
```
#Top producers of selected food items
#What countries have produced the highest total quantity of Oats from 2004 to 2013?
df4=df.query("Item == 'Oats'")
df4.nlargest(25, 'Value')
```

Out[20]:

	Country	Item	Year	Value
49	Algeria	Oats	2013	113
48	Algeria	Oats	2012	110
45	Algeria	Oats	2009	96
13187	Morocco	Oats	2011	96
43	Algeria	Oats	2007	92
40	Algeria	Oats	2004	89
42	Algeria	Oats	2006	89
46	Algeria	Oats	2010	88
41	Algeria	Oats	2005	78
47	Algeria	Oats	2011	67
17817	South Africa	Oats	2011	66
6409	Ethiopia	Oats	2013	60
6401	Ethiopia	Oats	2005	57
17818	South Africa	Oats	2012	57
13188	Morocco	Oats	2012	56
6407	Ethiopia	Oats	2011	49
6406	Ethiopia	Oats	2010	48
13189	Morocco	Oats	2013	46
6408	Ethiopia	Oats	2012	44
17812	South Africa	Oats	2006	44
17813	South Africa	Oats	2007	42
6402	Ethiopia	Oats	2006	40
6400	Ethiopia	Oats	2004	39
6404	Ethiopia	Oats	2008	37
13186	Morocco	Oats	2010	37

In [21]:
```
#Top producers of selected food items
#What countries have produced the highest total quantity of rice from 2004 to 2013?
df2=df.query("Item == 'Rice (Milled Equivalent)'")
df2.nlargest(25, 'Value')
```

Out[21]:

	Country	Item	Year	Value
5674	Egypt	Rice (Milled Equivalent)	2008	4838
5673	Egypt	Rice (Milled Equivalent)	2007	4587
5672	Egypt	Rice (Milled Equivalent)	2006	4506
5670	Egypt	Rice (Milled Equivalent)	2004	4237
5671	Egypt	Rice (Milled Equivalent)	2005	4086
5678	Egypt	Rice (Milled Equivalent)	2012	3943
5679	Egypt	Rice (Milled Equivalent)	2013	3818
5677	Egypt	Rice (Milled Equivalent)	2011	3785
5675	Egypt	Rice (Milled Equivalent)	2009	3682
15398	Nigeria	Rice (Milled Equivalent)	2012	3224
10656	Madagascar	Rice (Milled Equivalent)	2010	3160
15399	Nigeria	Rice (Milled Equivalent)	2013	3135
15397	Nigeria	Rice (Milled Equivalent)	2011	3046
10658	Madagascar	Rice (Milled Equivalent)	2012	3035
10655	Madagascar	Rice (Milled Equivalent)	2009	3028
15396	Nigeria	Rice (Milled Equivalent)	2010	2983
5676	Egypt	Rice (Milled Equivalent)	2010	2888
10657	Madagascar	Rice (Milled Equivalent)	2011	2868
15394	Nigeria	Rice (Milled Equivalent)	2008	2787
15392	Nigeria	Rice (Milled Equivalent)	2006	2696
10654	Madagascar	Rice (Milled Equivalent)	2008	2611
10659	Madagascar	Rice (Milled Equivalent)	2013	2408
10653	Madagascar	Rice (Milled Equivalent)	2007	2398
15391	Nigeria	Rice (Milled Equivalent)	2005	2379
15395	Nigeria	Rice (Milled Equivalent)	2009	2365

In [22]:
```
#Annual production data for selected countries
#Nigeria's production from 2004 to 2013?
df9=df.query("Country == 'Nigeria'")
Data=df9.groupby(['Year']).sum()
print(Data)
```

```
       Value
Year
2004  149857
2005  158149
2006  168987
2007  157273
2008  167935
2009  141270
2010  158709
2011  167403
2012  178816
2013  179631

C:\Users\Chinenye Claire\AppData\Local\Temp\ipykernel_9192\2680246833.py:4: FutureWarning: The
default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a future version, nume
ric_only will default to False. Either specify numeric_only or select only columns which shoul
d be valid for the function.
  Data=df9.groupby(['Year']).sum()
```

In [54]:
```
#Item production values for selected country (Swaziland)
df20=df.query("Country == 'Swaziland'")
get=df20.loc[:, df20.columns!='Year']
Ref=get.groupby(['Item']).sum()
print(Ref)
```

```
                            Value
Item
Alcohol, Non-Food            1074
Bananas                        48
Beans                          15
Beer                          228
Beverages, Fermented          579
Bovine Meat                   155
Butter, Ghee                    0
Citrus, Other                   0
Coconut Oil                     0
Cottonseed                     14
Cottonseed Oil                  1
Crustaceans                     0
Dates                           0
Eggs                           10
Fats, Animals, Raw             10
Freshwater Fish                 0
Fruits, Other                  63
Grapefruit and products       370
Groundnuts (Shelled Eq)        28
Lemons, Limes and products      9
Maize and products            714
Milk - Excluding Butter       409
Mutton & Goat Meat             20
Nuts and products               9
Offals, Edible                 20
Oilcrops Oil, Other             0
Oranges, Mandarines           421
Pigmeat                        12
Pineapples and products       270
Potatoes and products          71
Poultry Meat                   60
Pulses, Other and products     21
Rice (Milled Equivalent)        0
Roots, Other                  526
Sorghum and products            0
Sugar (Raw Equivalent)       6405
Sugar cane                  50850
Sweet potatoes                 24
Sweeteners, Other             109
Tomatoes and products          43
Vegetables, Other              74
Wheat and products              3
```

```
C:\Users\Chinenye Claire\AppData\Local\Temp\ipykernel_9192\608748177.py:3: FutureWarning: The
default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a future version, nume
ric_only will default to False. Either specify numeric_only or select only columns which shoul
d be valid for the function.
  Ref=get.groupby(['Item']).sum()
```

In [23]:
```
#Annual production data for selected countries
#Djibouti's production from 2004 to 2013?
df11=df.query("Country == 'Djibouti'")
D_Data=df11.groupby(['Year']).sum()
print(D_Data)
```

```
      Value
Year
2004     55
2005     62
2006     56
2007     63
2008     62
2009     68
2010     75
2011     72
2012     77
2013     76

C:\Users\Chinenye Claire\AppData\Local\Temp\ipykernel_9192\3269569980.py:4: FutureWarning: The
default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a future version, nume
ric_only will default to False. Either specify numeric_only or select only columns which shoul
d be valid for the function.
  D_Data=df11.groupby(['Year']).sum()
```

In [24]:
```
#Annual consumption data for selected countries
#Nigeria's consumption from 2004 to 2013?
df10=df1.query("Country == 'Nigeria'")
C_Data=df10.groupby(['Year']).sum()
print(C_Data)
```

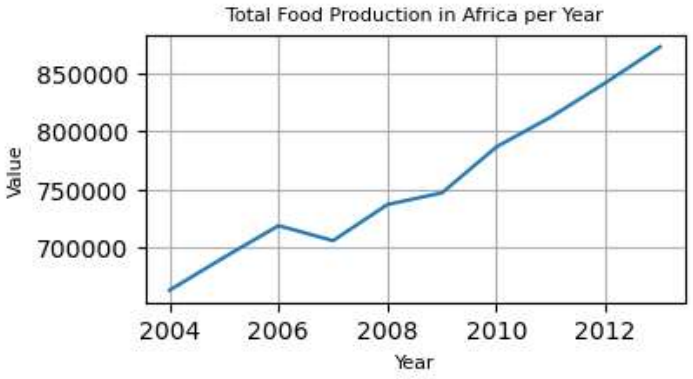
```
      Value
Year
2004   2655
2005   2705
2006   2725
2007   2720
2008   2723
2009   2683
2010   2706
2011   2706
2012   2665
2013   2700

C:\Users\Chinenye Claire\AppData\Local\Temp\ipykernel_9192\3720268402.py:4: FutureWarning: The
default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a future version, nume
ric_only will default to False. Either specify numeric_only or select only columns which shoul
d be valid for the function.
  C_Data=df10.groupby(['Year']).sum()
```

In [25]:
```
#Annual consumption data for selected countries
#Djibouti's consumption from 2004 to 2013?
df12=df1.query("Country == 'Djibouti'")
C2_Data=df12.groupby(['Year']).sum()
print(C2_Data)
```

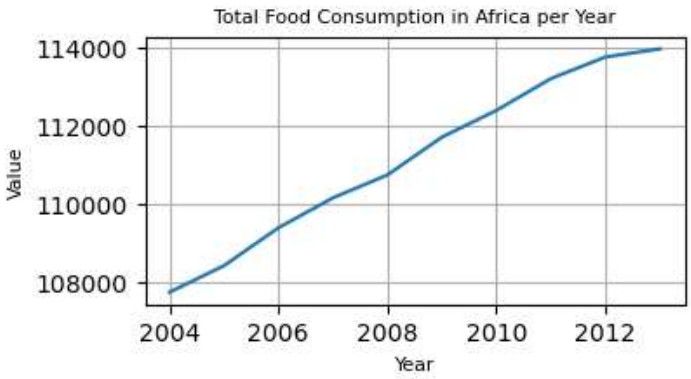
```
      Value
Year
2004   2136
2005   2265
2006   2314
2007   2414
2008   2447
2009   2466
2010   2467
2011   2504
2012   2545
2013   2607
```

```
C:\Users\Chinenye Claire\AppData\Local\Temp\ipykernel_9192\2058101065.py:4: FutureWarning: The
default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a future version, nume
ric_only will default to False. Either specify numeric_only or select only columns which shoul
d be valid for the function.
  C2_Data=df12.groupby(['Year']).sum()
```

In [46]:
```
#Create Visualizations
plt.figure(figsize=(4, 2))
plt.plot(gn)
plt.title('Total Food Production in Africa per Year', fontsize=8)
plt.xlabel('Year', fontsize=8)
plt.ylabel('Value', fontsize=8)
plt.grid(True)
plt.show()
```

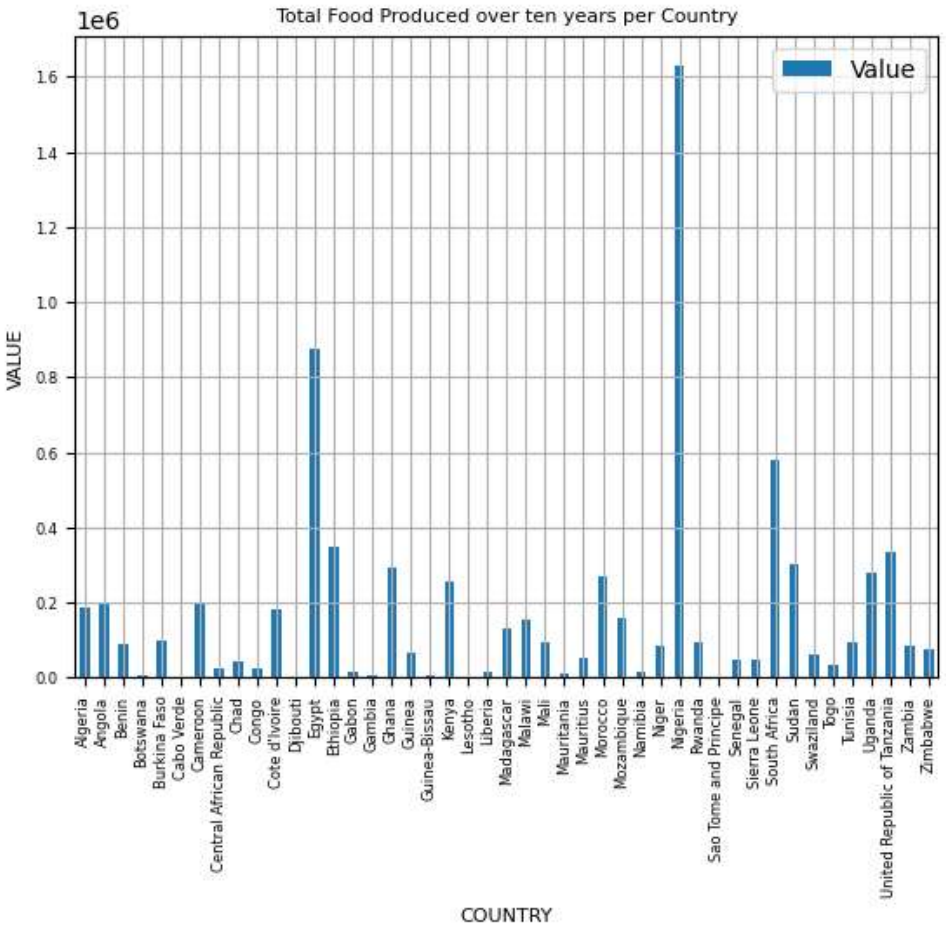


In [45]:
```
#Create Visualizations
plt.figure(figsize=(4, 2))
plt.plot(gj)
plt.title('Total Food Consumption in Africa per Year', fontsize=8)
plt.xlabel('Year', fontsize=8)
plt.ylabel('Value', fontsize=8)
plt.grid(True)
plt.show()
```



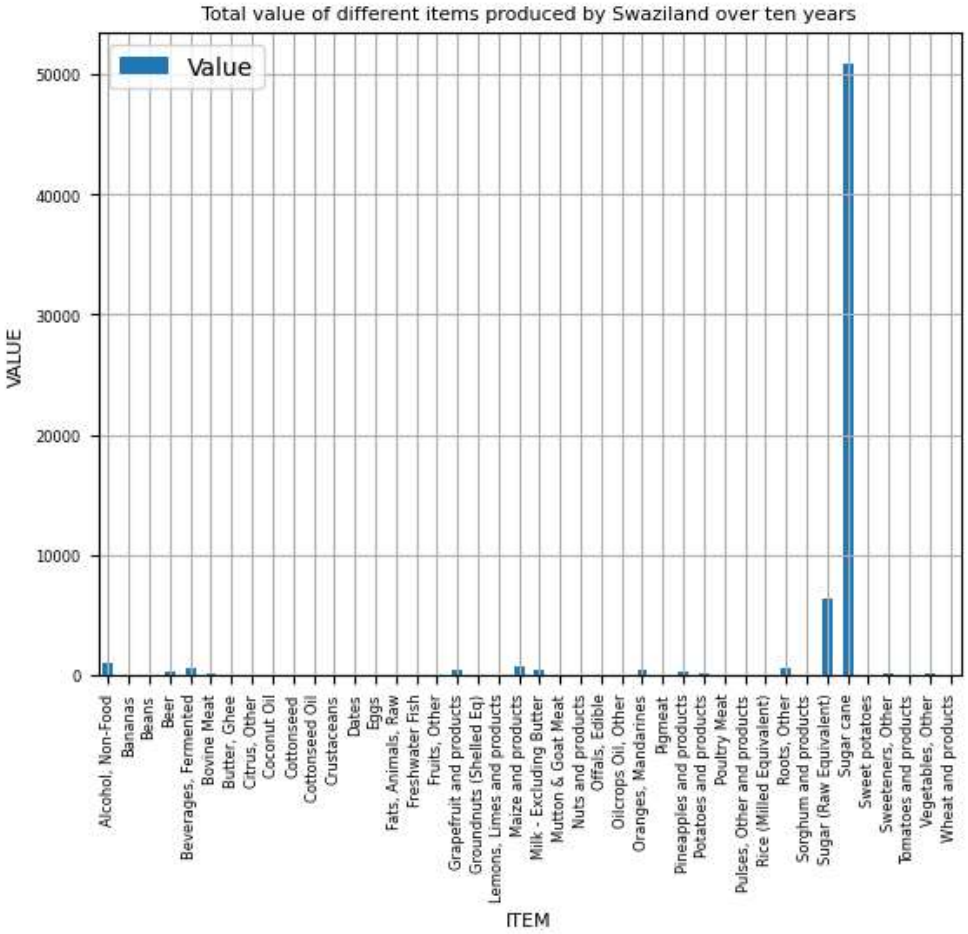
In [29]:
```
#Create Visualizations
plt.figure(figsize=(12, 8))
gm.plot(kind='bar')
plt.title('Total Food Produced over ten years per Country', fontsize=8)
plt.xlabel('COUNTRY', fontsize=8)
plt.ylabel('VALUE', fontsize=8)
plt.xticks(rotation=90, fontsize=6)
plt.yticks(fontsize=6)
plt.grid(True)
plt.show()
```

```
<Figure size 1200x800 with 0 Axes>
```

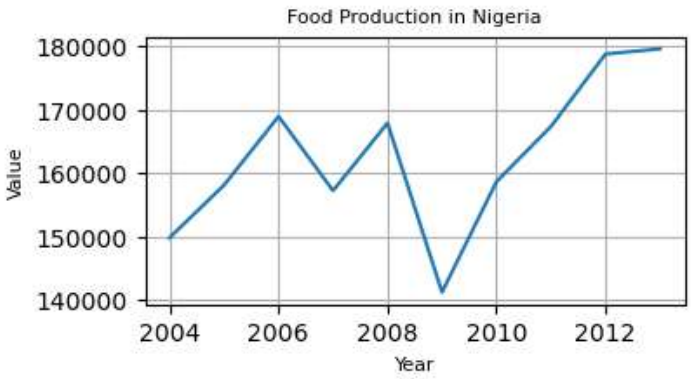


In [59]:
```
#Create Visualizations
plt.figure(figsize=(6, 4))
Ref.plot(kind='bar')
plt.title('Total value of different items produced by Swaziland over ten years', fontsize=8)
plt.xlabel('ITEM', fontsize=8)
plt.ylabel('VALUE', fontsize=8)
plt.xticks(rotation=90, fontsize=6)
plt.yticks(fontsize=6)
plt.grid(True)
plt.show()
```

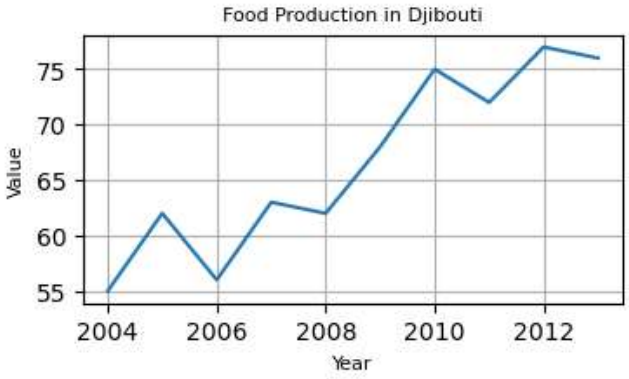
```
<Figure size 600x400 with 0 Axes>
```



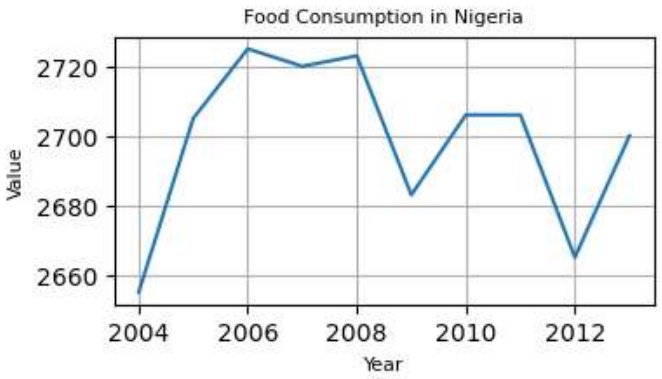
In [30]:
```
#Create Visualizations
plt.figure(figsize=(4, 2))
plt.plot(Data)
plt.title('Food Production in Nigeria', fontsize=8)
plt.xlabel('Year', fontsize=8)
plt.ylabel('Value', fontsize=8)
plt.grid(True)
plt.show()
```



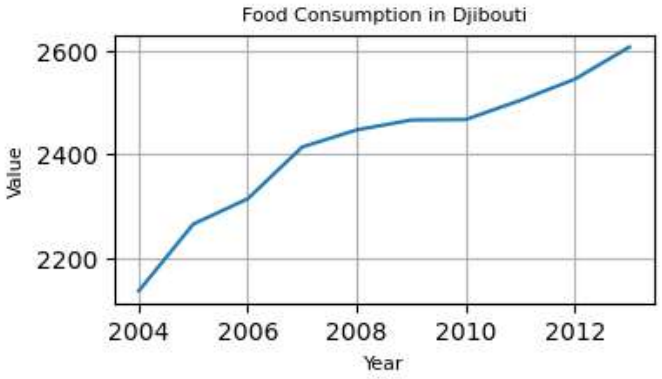
In [31]:
```
#Create Visualizations
plt.figure(figsize=(4, 2))
plt.plot(D_Data)
plt.title('Food Production in Djibouti', fontsize=8)
plt.xlabel('Year', fontsize=8)
plt.ylabel('Value', fontsize=8)
plt.grid(True)
plt.show()
```



In [32]:
```
#Create Visualizations
plt.figure(figsize=(4, 2))
plt.plot(C_Data)
plt.title('Food Consumption in Nigeria', fontsize=8)
plt.xlabel('Year', fontsize=8)
plt.ylabel('Value', fontsize=8)
plt.grid(True)
plt.show()
```

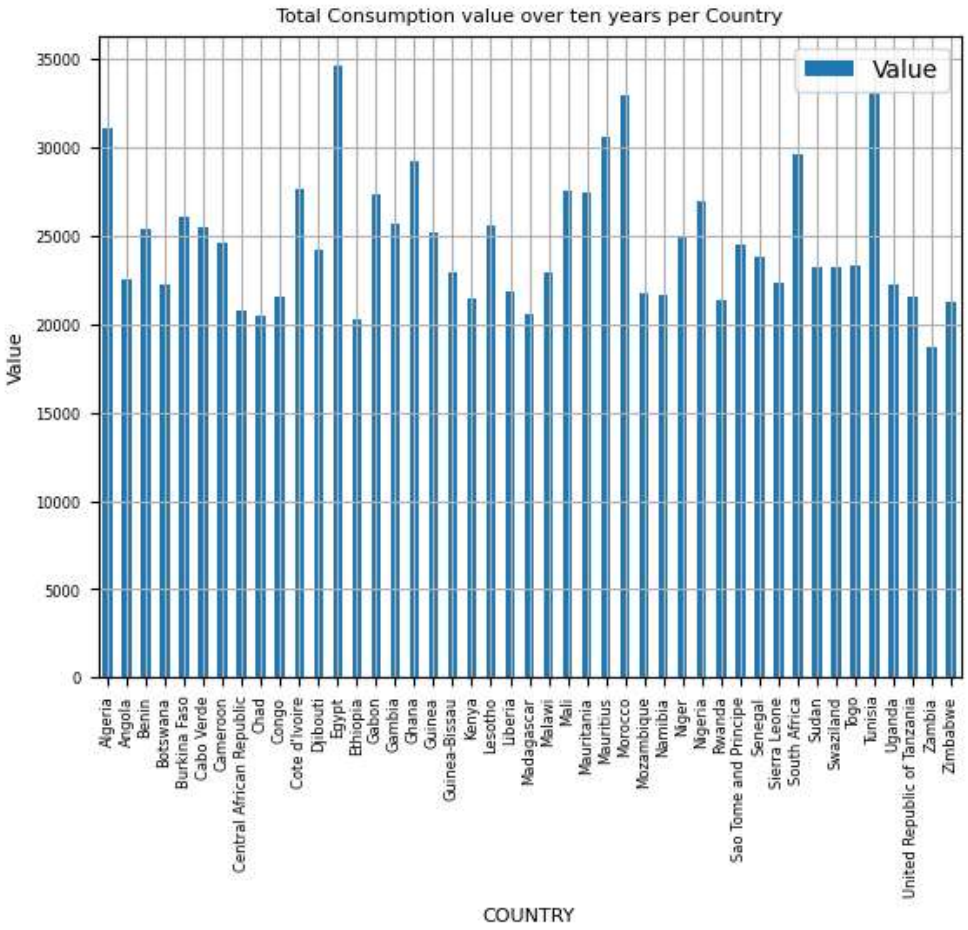


In [33]:
```
#Create Visualizations
plt.figure(figsize=(4, 2))
plt.plot(C2_Data)
plt.title('Food Consumption in Djibouti', fontsize=8)
plt.xlabel('Year', fontsize=8)
plt.ylabel('Value', fontsize=8)
plt.grid(True)
plt.show()
```



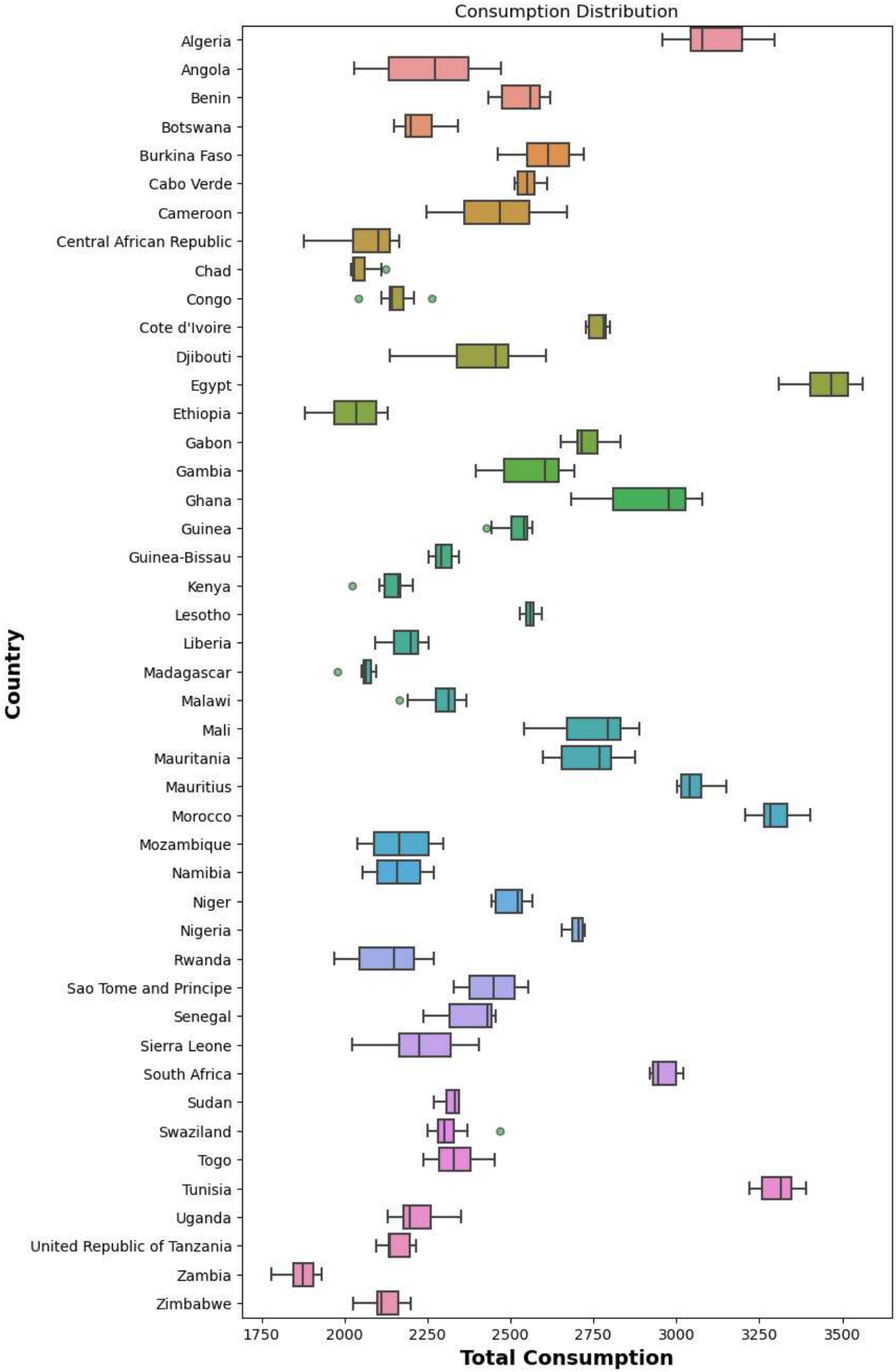
In [35]:
```
#Create Visualizations
plt.figure(figsize=(12, 8))
gi.plot(kind='bar')
plt.title('Total Consumption value over ten years per Country', fontsize=8)
plt.xlabel('COUNTRY', fontsize=8)
plt.ylabel('Value', fontsize=8)
plt.xticks(rotation=90, fontsize=6)
plt.yticks(fontsize=6)
plt.grid(True)
plt.show()
```

```
<Figure size 1200x800 with 0 Axes>
```



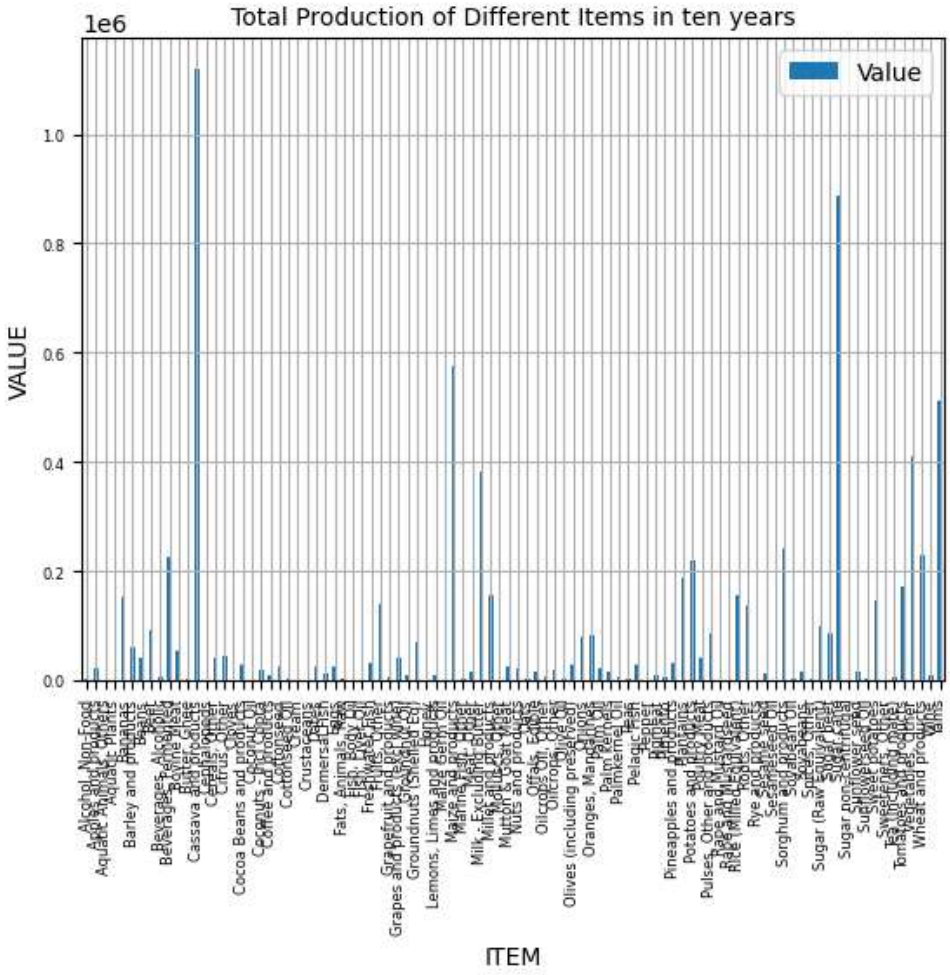
In [36]:
```
#Create Visualizations
#Consumption Distribution by country
plt.figure(figsize=[8, 16])
flierprops = dict(marker='o', markersize=5, markeredgecolor='black', markerfacecolor='green', a
p = sns.boxplot(y=df1['Country'], x=df1['Value'], flierprops=flierprops)
p.set_xlabel('Total Consumption', fontsize= 14, fontweight='bold')
p.set_ylabel('Country', fontsize= 14, fontweight='bold')
p.set_title('Consumption Distribution')
```

Out[36]:
```
Text(0.5, 1.0, 'Consumption Distribution')
```

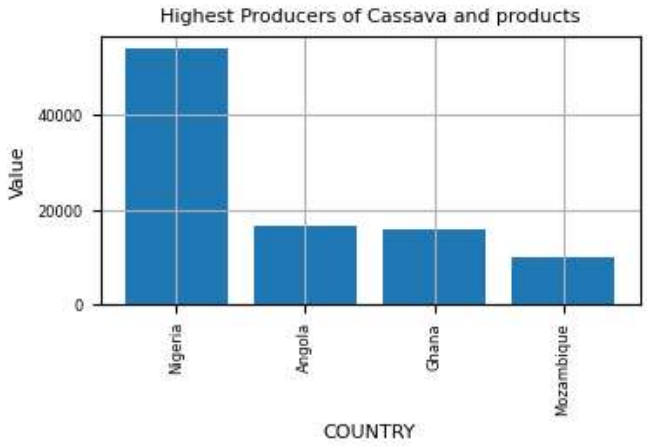

Consumption Distribution
Algeria
Angola
Benin
Botswana
Burkina Faso
Cabo Verde
Cameroon
Central African Republic
Chad
Congo
Cote d'Ivoire
Djibouti
Egypt
Ethiopia
Gabon
Gambia
Ghana
Guinea
Guinea-Bissau
Kenya
Lesotho
Liberia
Madagascar
Malawi
Mali
Mauritania
Mauritius
Morocco
Mozambique
Namibia
Niger
Nigeria
Rwanda
Sao Tome and Principe
Senegal
Sierra Leone
South Africa
Sudan
Swaziland
Togo
Tunisia
Uganda
United Republic of Tanzania
Zambia
Zimbabwe
Country
1750
2000
2250
2500
2750
3000
3250
3500
Total Consumption

In [44]:
```
#Create Visualizations
plt.figure(figsize=(20, 5))
gk.plot(kind='bar')
plt.title('Total Production of Different Items in ten years', fontsize=10)
plt.xlabel('ITEM', fontsize=10)
plt.ylabel('VALUE', fontsize=10)
plt.xticks(rotation=90, fontsize=6)
plt.yticks(fontsize=6)
plt.grid(True)
plt.show()
```

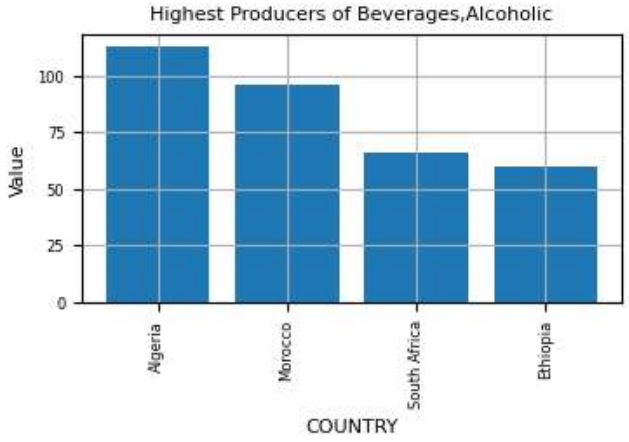
```
<Figure size 2000x500 with 0 Axes>
```



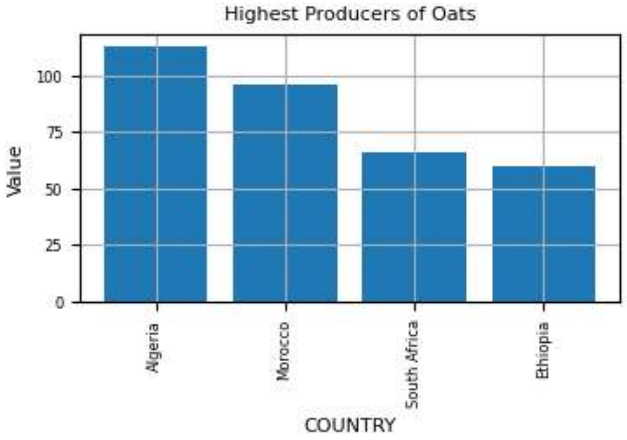
In [65]:
```
#Create Visualizationsd
f8=df7.nlargest(25, 'Value')
plt.figure(figsize=(4, 2))
Country = df8['Country'].head(25)
Value = df8['Value'].head(25)
plt.bar(Country[0:24],Value[0:24])
plt.title('Highest Producers of Cassava and products', fontsize=8)
plt.xlabel('COUNTRY', fontsize=8)
plt.ylabel('Value', fontsize=8)
plt.xticks(rotation=90, fontsize=6)
plt.yticks(fontsize=6)
plt.grid(True)
plt.show()
```



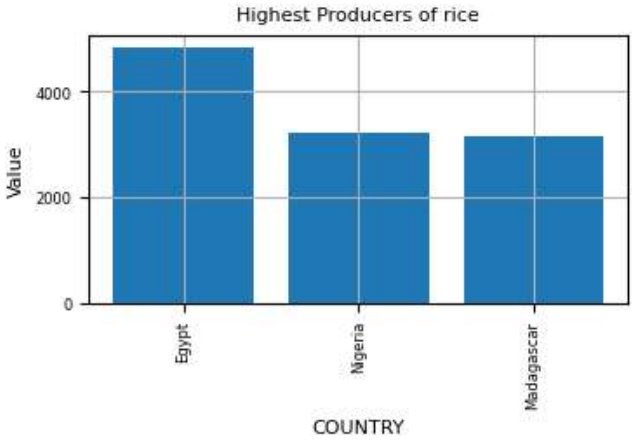
In [66]:
```
df6=df5.nlargest(25, 'Value')
plt.figure(figsize=(4, 2))
Country = df6['Country'].head(25)
Value = df6['Value'].head(25)
plt.bar(Country[0:24],Value[0:24])
plt.title('Highest Producers of Beverages,Alcoholic', fontsize=8)
plt.xlabel('COUNTRY', fontsize=8)
plt.ylabel('Value', fontsize=8)
plt.xticks(rotation=90, fontsize=6)
plt.yticks(fontsize=6)
plt.grid(True)
plt.show()
```



In [67]:
```
df5=df4.nlargest(25, 'Value')
plt.figure(figsize=(4, 2))
Country = df5['Country'].head(25)
Value = df5['Value'].head(25)
plt.bar(Country[0:24],Value[0:24])
plt.title('Highest Producers of Oats', fontsize=8)
plt.xlabel('COUNTRY', fontsize=8)
plt.ylabel('Value', fontsize=8)
plt.xticks(rotation=90, fontsize=6)
plt.yticks(fontsize=6)
plt.grid(True)
plt.show()
```



In [68]:
```
df3=df2.nlargest(25, 'Value')
plt.figure(figsize=(4, 2))
Country = df3['Country'].head(25)
Value = df3['Value'].head(25)
plt.bar(Country[0:24],Value[0:24])
plt.title('Highest Producers of rice', fontsize=8)
plt.xlabel('COUNTRY', fontsize=8)
plt.ylabel('Value', fontsize=8)
plt.xticks(rotation=90, fontsize=6)
plt.yticks(fontsize=6)
plt.grid(True)
plt.show()
```



In [42]:
```
#Summary of the total annual production report for Africa
gn.describe()
```

Out[42]:

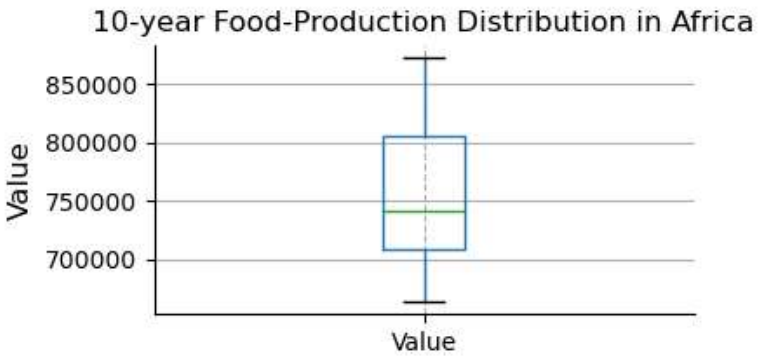
	Value
count	10.000000
mean	757511.600000
std	68489.402767
min	663006.000000
25%	708894.750000
50%	741837.000000
75%	805777.000000
max	872571.000000

In [43]:
```
#summary of the total annual consumption report for Africa
gj.describe()
```

Out[43]:

	Value
count	10.000000
mean	111139.900000
std	2209.897406
min	107740.000000
25%	109576.750000
50%	111217.000000
75%	112991.250000
max	113951.000000

In [61]:
```
plt.figure(figsize=(4, 2))
gn.boxplot()
plt.title('10-year Food-Production Distribution in Africa', fontsize=12)
plt.ylabel('Value', fontsize=12)
plt.grid(axis='x', linestyle='--')
sns.despine()
plt.show()
```



In [64]:
```
plt.figure(figsize=(4, 2))
gj.boxplot(column='Value')
plt.title('10-year Food-Consumption Distribution in Africa', fontsize=12)
plt.xlabel('Value', fontsize=12)
plt.grid(axis='x', linestyle='--')
sns.despine()
plt.show()
```