Data Storytelling: Food Production and Supply in Africa (20 2013)

Introduction

Using the food production and consumption/supply dataset, the aim of this project is to craft a story of how the world f problem can be solved using appropriate visuals making each variable in the dataset purposeful without leaving anyth

This analysis will contain visualisation and narratives so as to show:

- Trends in food production and consumption over the years.
- Average and median food production, the outliers, Quartile and interquartile ranges.
- A comparison between average food consumption and production for each year.

In [1]:
```
# Importing libraries

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

Food Production

In [2]:
```
# opening the production dataset and viewing the first 5 rows

foodprod = pd.read_csv('Africa Food Production.csv')
foodprod.head()
```

Out[2]:

	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

Data Cleaning and Wrangling

In [3]:
```
# information on the food production dataset

foodprod.info()

# The dataset contains 23110 rows amd 4 columns
# There are no null or missing values in any of its rows.
```

```
<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: 541.7+ KB
```

In [4]:
```
# Checking for the number of unique items produced by all countries

print(foodprod.Item.nunique())

foodprod.Item.unique()

# There are 94 unique items produced by different African countries
```

```
94
```

Out[4]:
```
array(['Wheat and products', 'Rice (Milled Equivalent)',
       'Barley and products', 'Maize and products', 'Oats',
       'Sorghum and products', 'Cereals, Other', 'Potatoes and products',
       'Sugar beet', 'Sugar (Raw Equivalent)', 'Honey', 'Beans', 'Peas',
       'Pulses, Other and products', 'Nuts and products',
       'Groundnuts (Shelled Eq)', 'Sunflower seed',
       'Rape and Mustardseed', 'Cottonseed',
       'Olives (including preserved)', 'Groundnut Oil',
       'Sunflowerseed Oil', 'Rape and Mustard Oil', 'Olive Oil',
       'Oilcrops Oil, Other', 'Tomatoes and products', 'Onions',
       'Vegetables, Other', 'Oranges, Mandarines',
       'Lemons, Limes and products', 'Grapefruit and products',
       'Citrus, Other', 'Bananas', 'Apples and products', 'Dates',
       'Grapes and products (excl wine)', 'Fruits, Other', 'Pimento',
       'Wine', 'Beer', 'Beverages, Alcoholic', 'Bovine Meat',
       'Mutton & Goat Meat', 'Pigmeat', 'Poultry Meat', 'Meat, Other',
       'Offals, Edible', 'Butter, Ghee', 'Fats, Animals, Raw', 'Eggs',
       'Milk - Excluding Butter', 'Freshwater Fish', 'Demersal Fish',
       'Pelagic Fish', 'Marine Fish, Other', 'Crustaceans', 'Cephalopods',
       'Molluscs, Other', 'Millet and products', 'Cassava and products',
       'Sweet potatoes', 'Sugar cane', 'Soyabeans', 'Sesame seed',
       'Palm kernels', 'Oilcrops, Other', 'Cottonseed Oil',
       'Palmkernel Oil', 'Palm Oil', 'Pineapples and products',
       'Coffee and products', 'Cocoa Beans and products',
       'Beverages, Fermented', 'Fish, Body Oil', 'Yams', 'Roots, Other',
       'Coconuts - Incl Copra', 'Soyabean Oil', 'Coconut Oil', 'Pepper',
       'Aquatic Animals, Others', 'Sweeteners, Other', 'Cream',
       'Fish, Liver Oil', 'Sesameseed Oil', 'Spices, Other',
       'Aquatic Plants', 'Plantains', 'Tea (including mate)',
       'Rye and products', 'Alcohol, Non-Food', 'Sugar non-centrifugal',
       'Maize Germ Oil', 'Cloves'], dtype=object)
```

In [5]:
```
# Descriptive statistics of the food production dataset

foodprod.describe()
```

Out[5]:

	Year	Value
count	23110.000000	23110.000000
mean	2008.498269	327.785201
std	2.871740	1607.940343
min	2004.000000	0.000000
25%	2006.000000	3.000000
50%	2008.000000	18.000000
75%	2011.000000	108.000000
max	2013.000000	54000.000000

Data Exploration and Visualisation

In [6]:
```
# grouping by country, year and item to get the total production per item

foodprod_sum = foodprod.groupby(['Country', 'Year', 'Item'])['Value'].sum()
foodprod_sum
```

Out[6]:
```
Country   Year  Item
Algeria   2004  Apples and products       165
                Bananas                    0
                Barley and products     1212
                Beans                      2
                Beer                     110
                                         ...
Zimbabwe  2013  Tea (including mate)       19
                Tomatoes and products      24
                Vegetables, Other         203
                Wheat and products         25
                Wine                        2
Name: Value, Length: 23110, dtype: int64
```

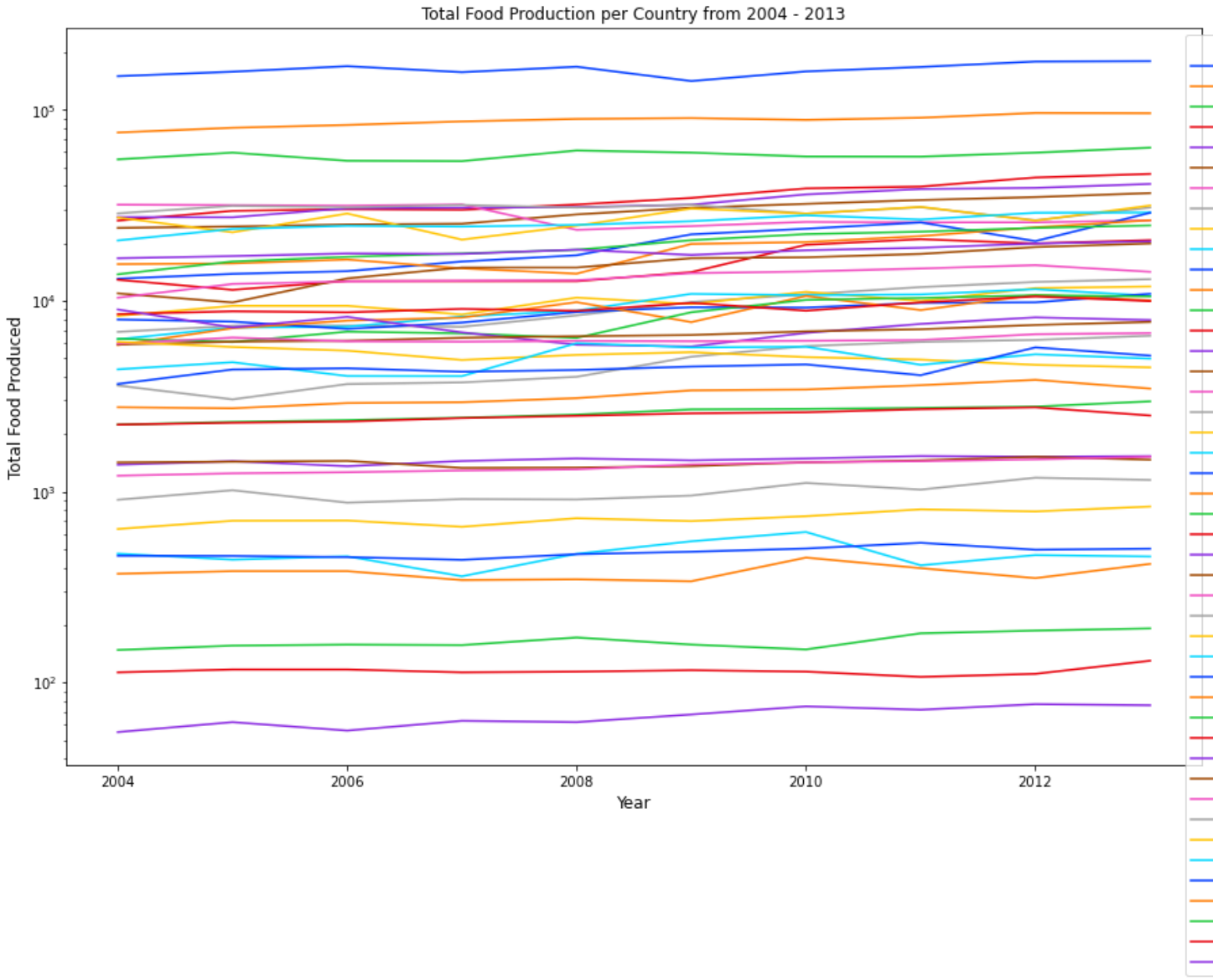
In [7]:
```
# grouping by country and year to get the total food production per year

foodprod_tot = pd.DataFrame(foodprod.groupby(['Country', 'Year'], as_index = False)['Value'].su
foodprod_tot

# Algeria is seen to be the Country with the highest food production per year
```

Out[7]:

	Country	Year	Value
0	Algeria	2004	15536
1	Algeria	2005	15667
2	Algeria	2006	16417
3	Algeria	2007	14763
4	Algeria	2008	13841
...	...	...	...
445	Zimbabwe	2009	5754
446	Zimbabwe	2010	6777
447	Zimbabwe	2011	7551
448	Zimbabwe	2012	8173
449	Zimbabwe	2013	7914

450 rows × 3 columns

In [8]:
```
# Sorting by total values in descending order
foodprod_grouped = foodprod_tot.sort_values(by = 'Value', ascending=False)

# Visualising with a line chart the trend  from 2004 - 2013 of total items produced per country
plt.subplots(figsize=(15,10))
lineplt = sns.lineplot(x=foodprod_grouped['Year'], y= foodprod_grouped['Value'], hue=foodprod_
lineplt.set_yscale('log')
lineplt.legend(loc='upper right',bbox_to_anchor= (1.2, 1))

# Defining the title of the plot, xaxis and yaxis.
plt.title('Total Food Production per Country from 2004 - 2013')
plt.xlabel('Year', fontsize = 12)
plt.ylabel('Total Food Produced', fontsize = 12);
```



It can be observed that Nigeria, Egypt and South Africa are the highest food producing countries between 20(can also be seen that Cabo Verde, Sao Tome and Principe, Djibouti are the lowest food producing countries b 2004 - 2013.

To effectively observe the change in the food production trend per country between 2004-2013, the data will b using a map. To do this;

- A country code dataset will be imported and merged with the food production dataset.
- This newly merged dataset will then be used to create a map visualization using the `plotly.express` library.

In [9]:
```
# Importing the country code dataset

code = pd.read_csv('2014_world_gdp_with_codes.txt')
code.head()
```

Out[9]:

	COUNTRY	GDP (BILLIONS)	CODE
0	Afghanistan	21.71	AFG
1	Albania	13.40	ALB
2	Algeria	227.80	DZA
3	American Samoa	0.75	ASM
4	Andorra	4.80	AND

In [10]:
```
# Dropping and renaming columns in the country code dataset to fit the column names in the food

code.columns = ['Country', 'Gdp(billions)','Country Code']

# dropping unwanted column
code.drop('Gdp(billions)', axis=1, inplace=True)
code.head()
```

Out[10]:

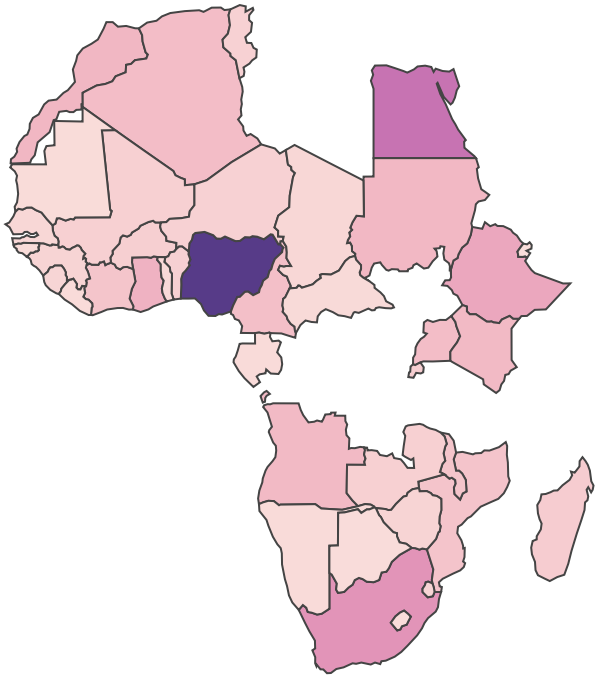
	Country	Country Code
0	Afghanistan	AFG
1	Albania	ALB
2	Algeria	DZA
3	American Samoa	ASM
4	Andorra	AND

In [11]:
```
# Merging the country code dataset with the food production dataset

merged = pd.merge(foodprod_grouped, code, on ='Country', how = 'inner')
merged.head()
```

Out[11]:

	Country	Year	Value	Country Code
0	Nigeria	2013	179631	NGA
1	Nigeria	2012	178816	NGA
2	Nigeria	2006	168987	NGA
3	Nigeria	2008	167935	NGA
4	Nigeria	2011	167403	NGA

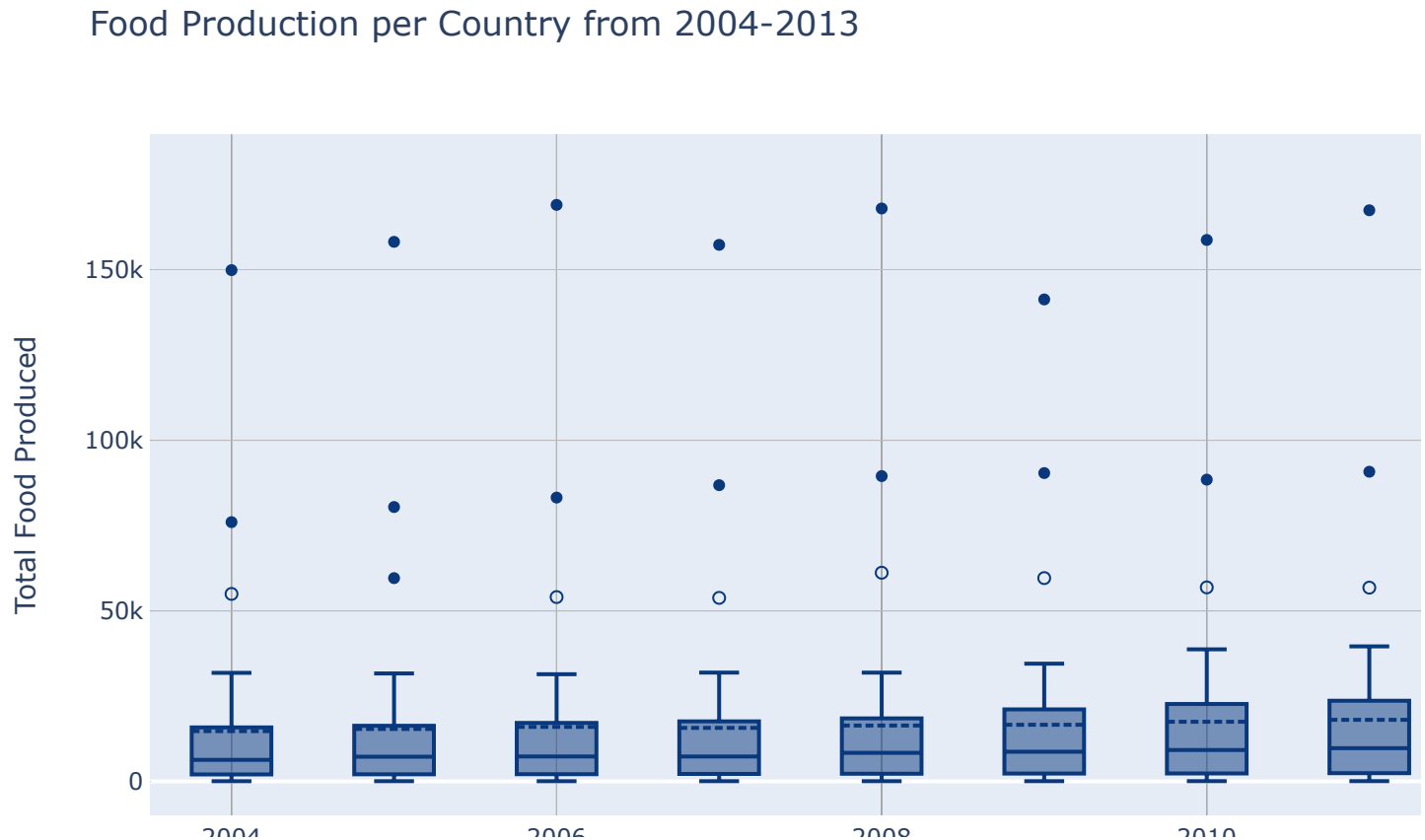
In [12]:
```
# Creating the map visualisation for the food production of African countries between 2004 and

fig = px.choropleth(merged, locations="Country Code",
                    color='Value',
                    hover_name="Country",
                    animation_frame='Year',
                    scope="africa",
                    color_continuous_scale = px.colors.sequential.Purpor
                   )
fig.update_geos(fitbounds="locations", visible=False)
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```



Year=2013

Statistics

In [13]:
```
# Creating a boxplot for the food production per country from 2004-2013

import plotly.graph_objects as go
fig = go.Figure()
fig.add_trace(go.Box(y = foodprod_grouped.Value,
                     x= foodprod_grouped.Year,
                     boxpoints = "suspectedoutliers",
                     boxmean = True,
                     marker_color='rgb(9,56,125)',
                     line_color='rgb(9,56,125)'
))
fig.update_layout(title_text="Food Production per Country from 2004-2013", yaxis_title= "Total
fig.show()
```



> From the boxplot above, it can be observed that there exists a number of outliers in the food produced in all th
> between 2004 and 2013 with 2012 having the largest outlier. The dataset for the year 2012 will be explored to
> outlier and other statistics including the mean, median, first quartile, third quartile and interquartile range.

In [14]:
```
# Extracting the food production dataset for 2012 to be analysed

foodprod_12 = foodprod.query('Year == 2012')
foodprod_12.shape
```

Out[14]:
```
(2307, 4)
```

In [15]:
```
# To find the mean of the food production in 2012

foodprod_12m = foodprod_12.Value.mean()
foodprod_12m
```

Out[15]:
```
364.8318162115301
```

In [16]:
```
# To find the median of the food consumption in 2012
foodprod_12med = foodprod_12.Value.median()
foodprod_12med
```

Out[16]:
```
21.0
```

In [17]:
```
# Using numpy to compute the quartile and interquartile range

Q_3, Q_1 = np.percentile(foodprod_12.Value, [75 ,25])
Q_3, Q_1
```

Out[17]:
```
(122.0, 3.0)
```

In [18]:
```
# Computing the interquartile range

IQR_ = np.subtract(*np.percentile(foodprod_12.Value, [75 ,25]))
IQR_
```

Out[18]: 119.0

In [19]:
```
# Computing the upper and lower fence

lower_fence = Q_1 - 1.5*IQR_
upper_fence = Q_3 + 1.5*IQR_
upper_fence, lower_fence
```

Out[19]: (300.5, -175.5)

In [20]:
```
# Computing the outlier present in the total food consumption in 2012

foodprod_12outliers = foodprod_12[(foodprod_12["Value"] > upper_fence) | (foodprod_12['Value']
foodprod_12outliers
```

Out[20]:

	Country	Item	Year	Value
8	Algeria	Wheat and products	2012	3432
28	Algeria	Barley and products	2012	1592
78	Algeria	Potatoes and products	2012	4219
198	Algeria	Olives (including preserved)	2012	394
258	Algeria	Tomatoes and products	2012	797
...	...	...	...	...
22228	Zambia	Vegetables, Other	2012	351
22508	Zimbabwe	Maize and products	2012	968
22598	Zimbabwe	Sugar cane	2012	3929
22608	Zimbabwe	Sugar (Raw Equivalent)	2012	501
23088	Zimbabwe	Milk - Excluding Butter	2012	410

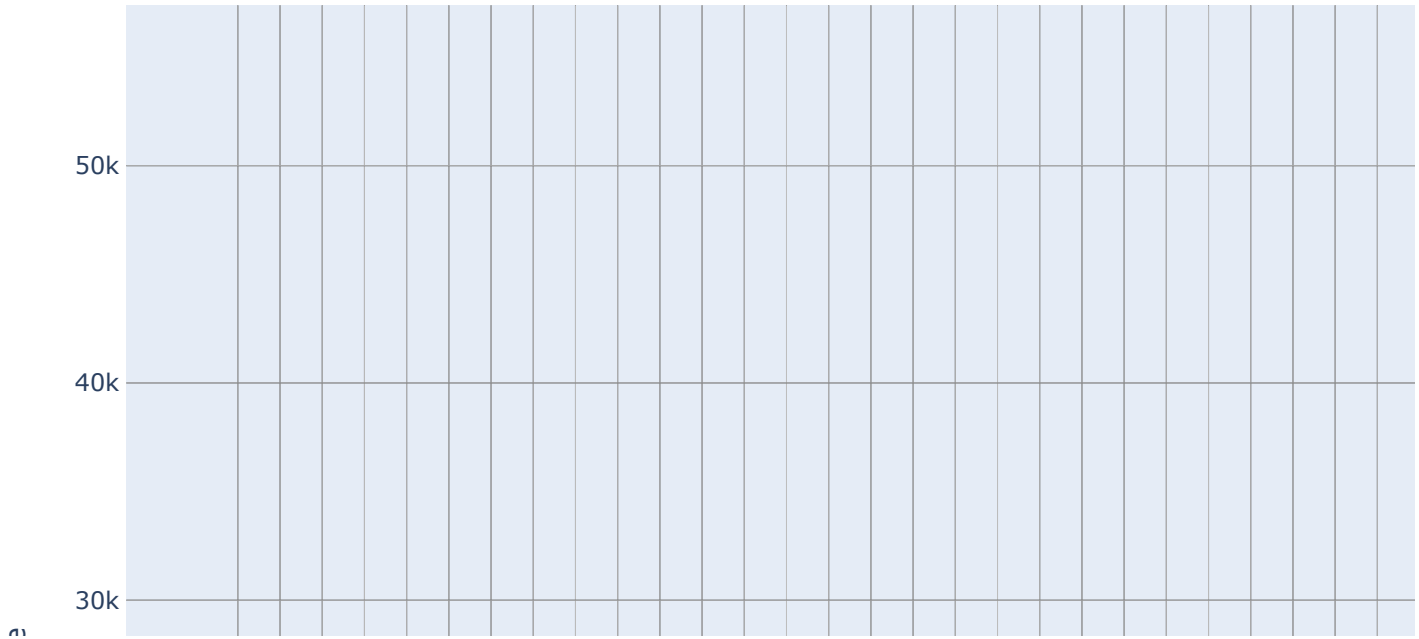
368 rows × 4 columns

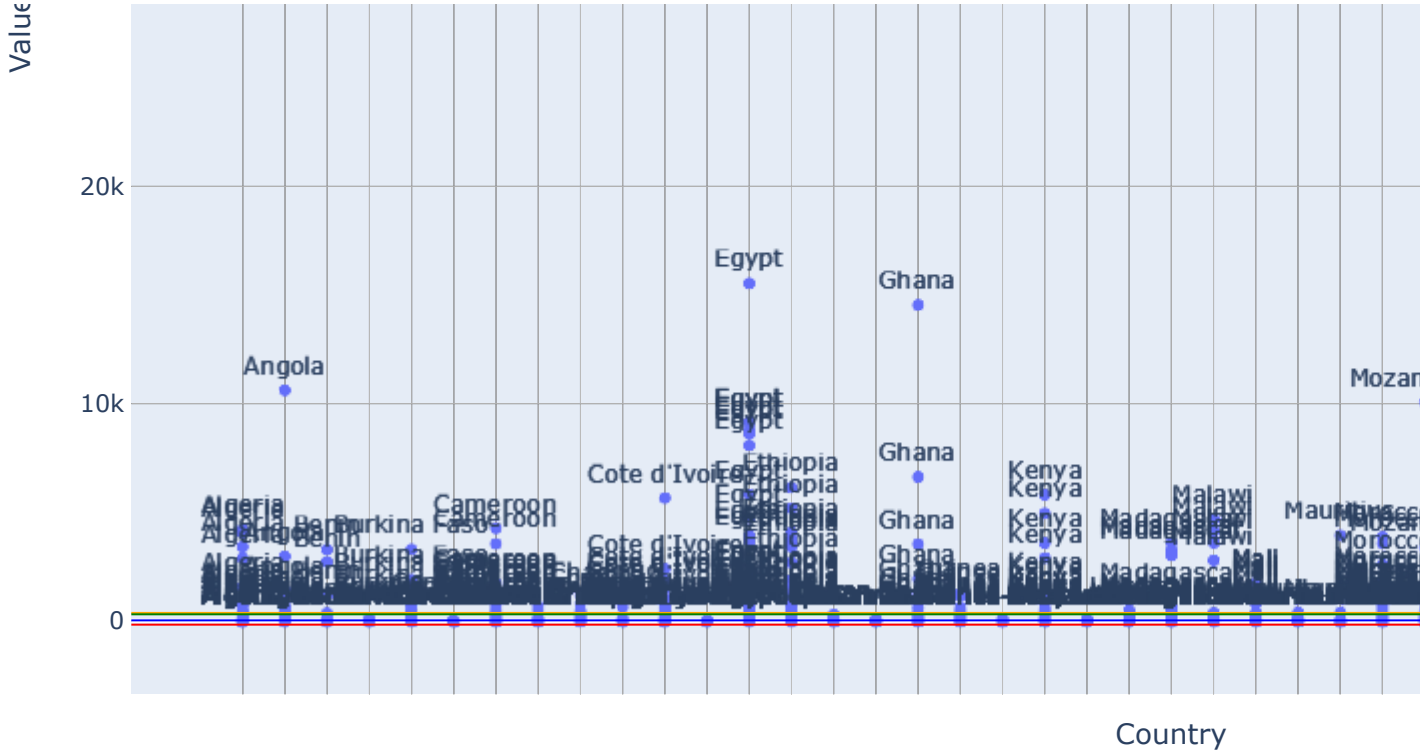
In [21]:
```
# Computing the number of countries that present an outlier
foodprod_12outliers.Country.nunique()
```

Out[21]: 38

> It can be observed that for the food production in African countries in 2012, there are approximately 368 outlie
> different countries.

In [22]:
```
# Visualizing with a scatterplot to see how the food production varied per country in 2012

fig = px.scatter(foodprod_12, x="Country", y="Value", text ="Country")

# adding horizontal lines to represent the mean, median, upper and lower fence.
fig.update_layout(width=1200,
                  height = 800,
                  autosize=False, shapes=[
                      #add lower fence, mean, median and upper fence lines
                      dict(type= "line", xref= "paper", x0 = 0, x1 = 1, yref = "y", y0 = lower_
                      dict(type= "line", xref= "paper", x0 = 0, x1 = 1, yref = "y", y0 = foodpr
                      dict(type= "line", xref= "paper", x0 = 0, x1 = 1, yref = "y", y0 = foodpr
                      dict(type= "line", xref= "paper", x0 = 0, x1 = 1, yref = "y", y0 = upper_
                  ], showlegend=False
                  )

fig.update_traces(textposition = "top center")
fig.update_xaxes(showticklabels = False)
fig.show()
```





From the scatterplot above, it can be observed that for 2012, indeed many outliers lie above the upper fence v represented by the green line.

Food Supply

In [23]:
```
# Opening and viewing the first 5 rows of the food supply dataset

foodsup = pd.read_csv('Africa Food Supply.csv')
foodsup.head()
```

Out[23]:

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

Data Wrangling

In [24]:
```
# Computing information about the food supply dataset

foodsup.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: 8.9+ KB
```

In [25]:
```
# Computing the unique number of countries in the food supply dataset

foodsup.Country.nunique()
```

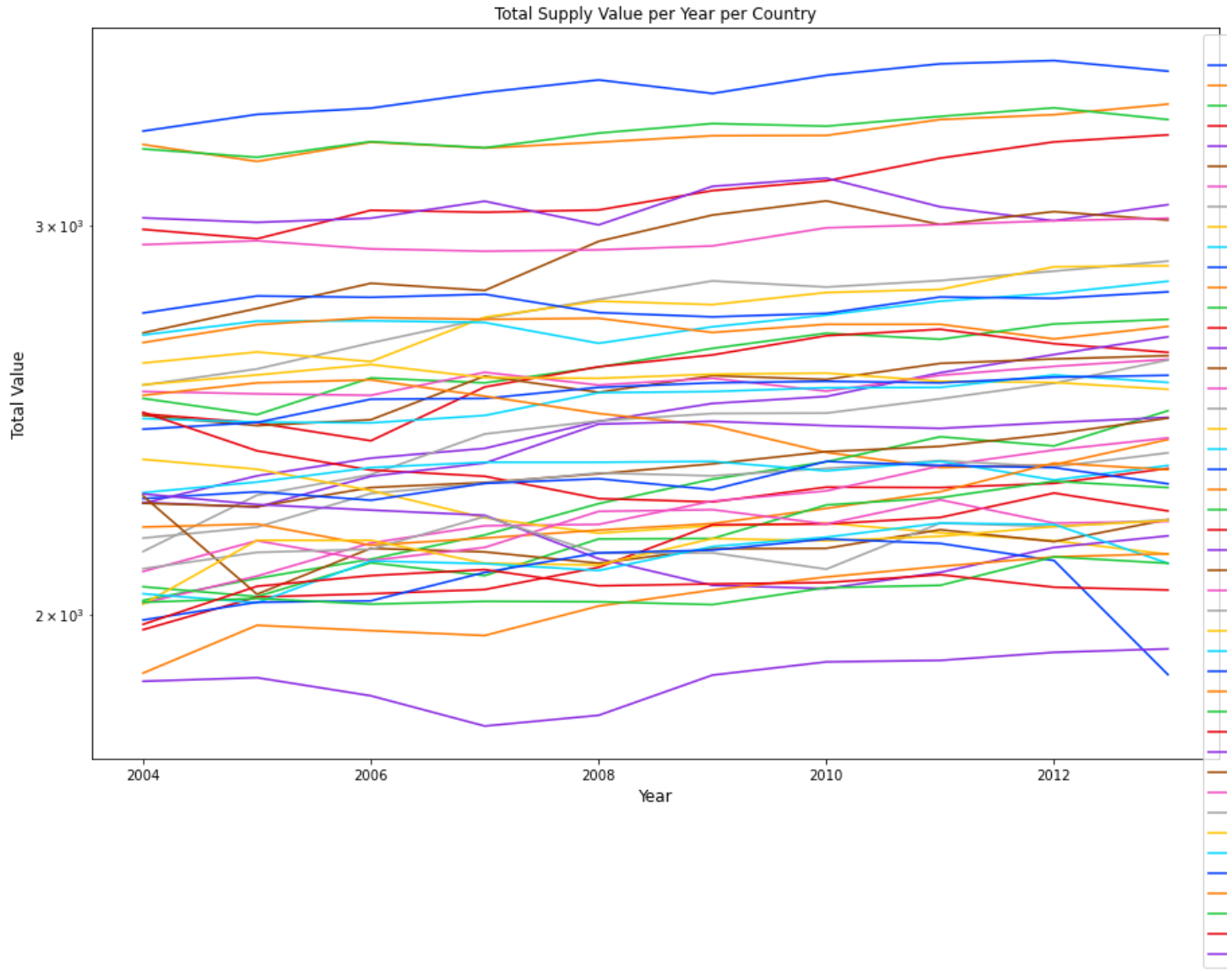
Out[25]: 45

In [26]:
```
# Grouping the supply dataset by country and year to get the total value of items supplied

foodsup_tot = pd.DataFrame(foodsup.groupby(['Country', 'Year'], as_index = False).sum())
foodsup_tot.head()
```

Out[26]:

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

In [27]:
```
# Sorting by total values supplied in descending order
foodsup_grouped = foodsup_tot.sort_values(by='Value', ascending=False)

# Visualising with a line chart the trend  from 2004 - 2013 of total items supplied to each cou
plt.subplots(figsize=(15,10))
lineplt1 = sns.lineplot(x=foodsup_grouped['Year'], y= foodsup_grouped['Value'], hue=foodsup_gr
lineplt1.set_yscale('log')
lineplt1.legend(loc='upper right',bbox_to_anchor= (1.2, 1))

# Defining the title of the plot, xaxis and yaxis.
plt.title('Total Supply Value per Year per Country')
plt.xlabel('Year', fontsize = 12)
plt.ylabel('Total Value', fontsize = 12);
```



It can be observed that Egypt, Morocco and Tunisia are the countries with the highest food supply between 2
It can also be seen that Chad, Madagascar, Zambia are the lowest food producing countries between 2004 -

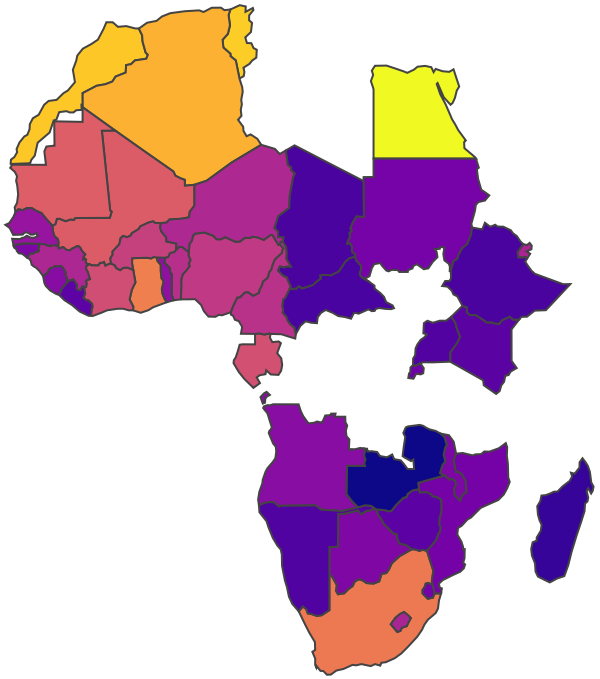
To effectively observe the change in the food supply trend per country between 2004-2013, the data will be vi using a map. To do this; ** A country code dataset will be imported and merged with the food supply dataset. newly merged dataset will then be used to create a map visualization using the `plotly.express` library.

In [28]:
```
# Merging the country code dataset with the food supply dataset

merged1 = pd.merge(foodsup_grouped, code, on ='Country', how = 'inner')
merged1.head()
```

Out[28]:

	Country	Year	Value	Country Code
0	Egypt	2012	3561	EGY
1	Egypt	2011	3549	EGY
2	Egypt	2013	3522	EGY
3	Egypt	2010	3507	EGY
4	Egypt	2008	3490	EGY

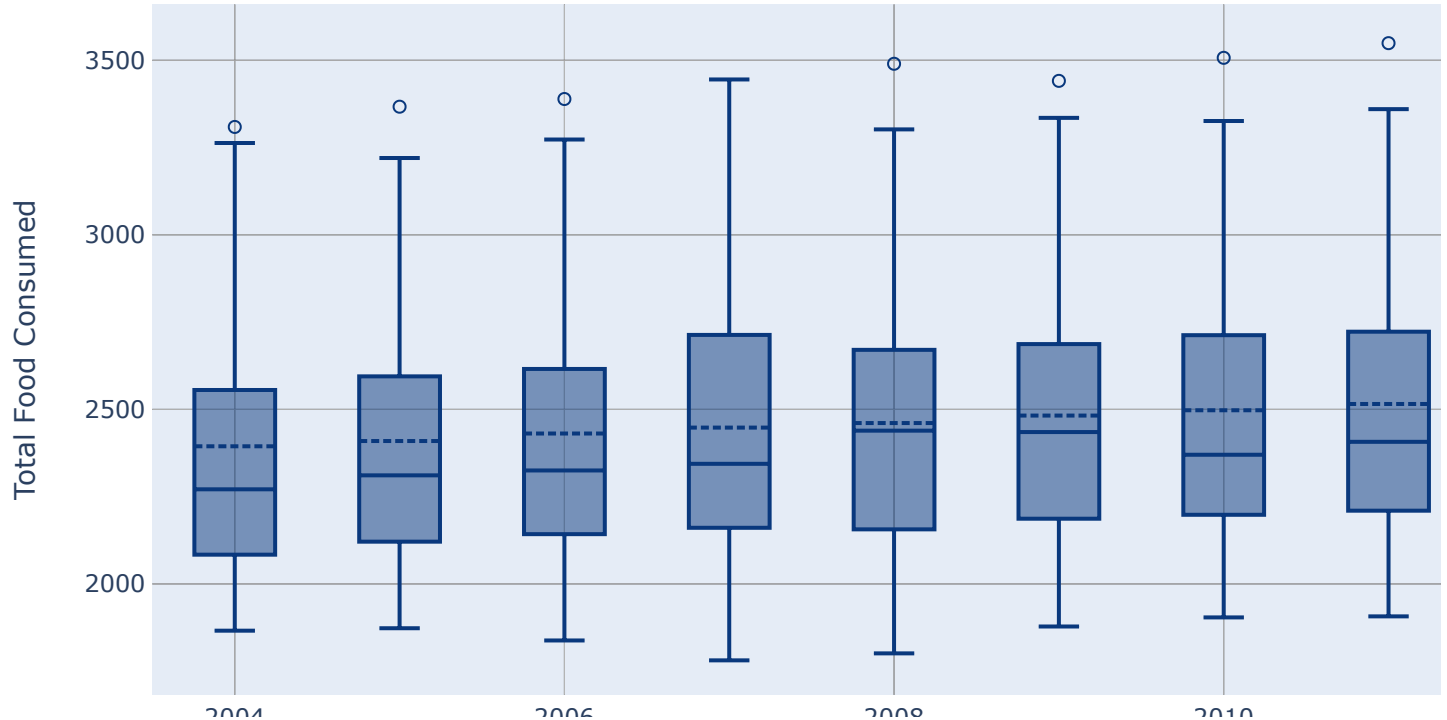
In [29]:
```
# Creating the map visualisation for the food supply in African countries between 2004 and 201

fig = px.choropleth(merged1, locations="Country Code",
                    color='Value',
                    hover_name="Country",
                    animation_frame='Year',
                    scope="africa"
                   )
fig.update_geos(fitbounds="locations", visible=False)
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```



Year=2012

Statistic

In [30]:
```
# Creating a boxplot for the food consumption per country from 2004-2013

import plotly.graph_objects as go
fig = go.Figure()
fig.add_trace(go.Box(y = foodsup_grouped.Value,
                     x= foodsup_grouped.Year,
                     boxpoints = "suspectedoutliers",
                     boxmean = True,
                     marker_color='rgb(9,56,125)',
                     line_color='rgb(9,56,125)'
))
fig.update_layout(title_text="Food Consumption per Country from 2004-2013", yaxis_title= "Total
fig.show()
```



> It can be observed that an outlier exists in the total food consumed for all the years except 2007 and 2013. Th
> for the year 2012 will be explored to identify the outlier and other statistics including the mean, median, first q
> quartile and interquartile range. As with the production dataset, 2012 is the year with the largest outlier. Henc
> 2012 will be explored and the outlier will be analysed

In [31]:
```
# Extracting the food consumption dataset for 2012 to be analysed

foodsup_12 = foodsup.query('Year == 2012')
```

In [32]:
```
# To find the mean of the food consumption in 2012

foodsup_12m = foodsup_12.Value.mean()
foodsup_12m
```

Out[32]: 2527.6444444444446

In [33]:
```
# To find the median of the food consumption in 2012
foodsup_12med = foodsup_12.Value.median()
foodsup_12med
```

Out[33]: 2414.0

In [34]:
```
# Using numpy to compute the quartile and interquartile range

Q3, Q1 = np.percentile(foodsup_12.Value, [75 ,25])
Q3, Q1
```

Out[34]: (2707.0, 2200.0)

In [35]:
```
# Computing the interquartile range

IQR = np.subtract(*np.percentile(foodsup_12.Value, [75 ,25]))
IQR
```

Out[35]: 507.0

In [36]:
```
# Computing the upper and lower fence

lowerfence = Q1 - 1.5*IQR
upperfence = Q3 + 1.5*IQR
upperfence, lowerfence
```

Out[36]: (3467.5, 1439.5)

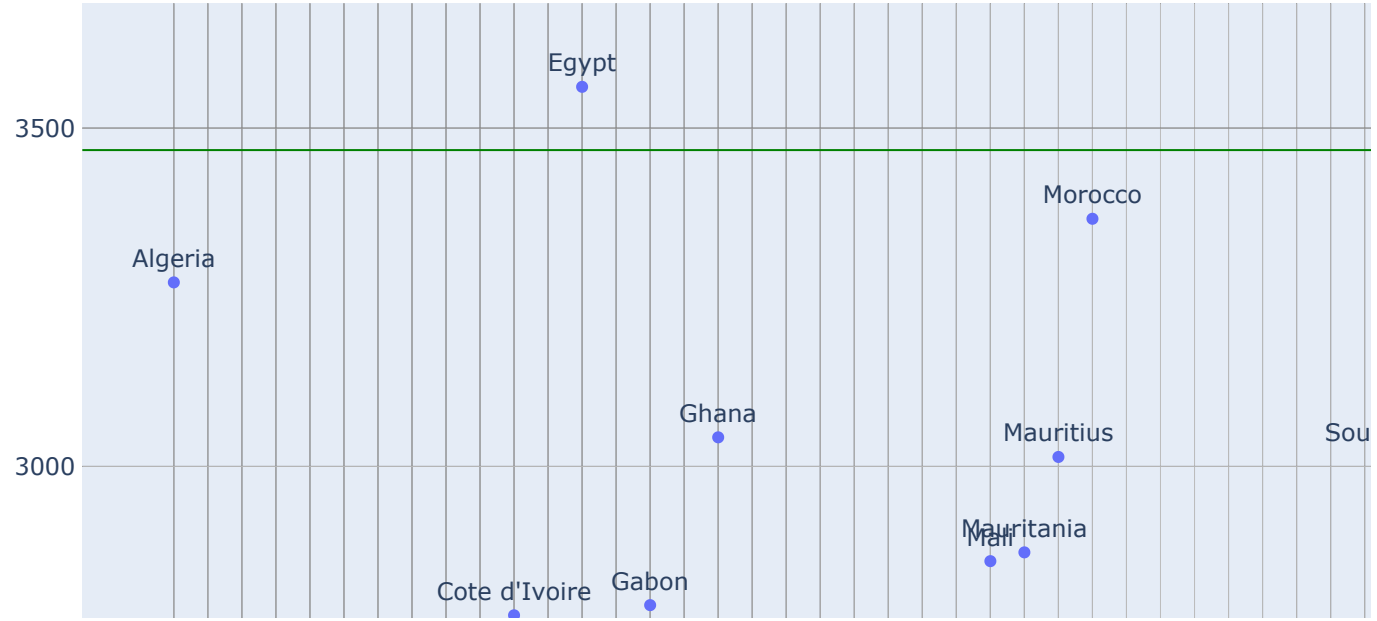
In [37]:
```
# Computing the outlier present in the total food consumption in 2012

foodsup_12outliers = foodsup_12[(foodsup_12["Value"] > upperfence) | (foodsup_12['Value'] < lou
foodsup_12outliers
```

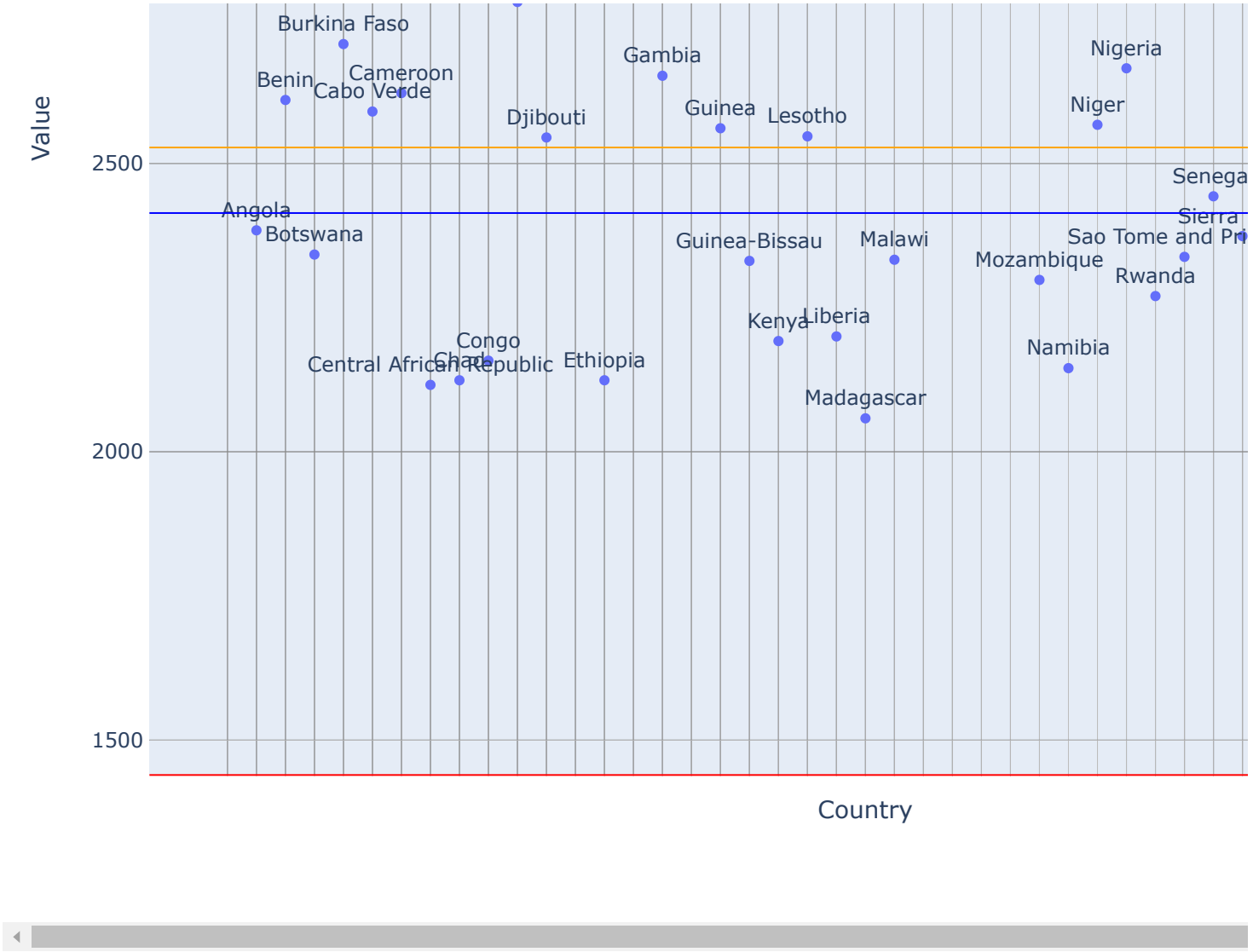
Out[37]:

	Country	Year	Value
128	Egypt	2012	3561

The only outlier present in the food consumption for 2012 is Egypt with a total value of ** consumed items.

In [38]:
```
# Visualizing with a scatterplot to see how the food consumption in Egypt varied from other cou

fig = px.scatter(foodsup_12, x="Country", y="Value", text ="Country")

# adding horizontal lines to represent the mean, median, upper and lower fence.
fig.update_layout(width=1000,
                  height = 900,
                  autosize=False, shapes=[
                      #add lower fence, mean, median and upper fence lines
                      dict(type= "line", xref= "paper", x0 = 0, x1 = 1, yref = "y", y0 = lowerfe
                      dict(type= "line", xref= "paper", x0 = 0, x1 = 1, yref = "y", y0 = foodsup
                      dict(type= "line", xref= "paper", x0 = 0, x1 = 1, yref = "y", y0 = foodsup
                      dict(type= "line", xref= "paper", x0 = 0, x1 = 1, yref = "y", y0 = upperfe
                  ], showlegend=False
                  )

fig.update_traces(textposition = "top center")
fig.update_xaxes(showticklabels = False)
fig.show()
```





From the analysis of both the food production and food supply/consumption data it can be observed that: Alth
Nigeria, Egypt and South Africa are the highest food producing countries, of these 3 countries only Egypt is re
have one of the highest food supply between 2004 and 2013.

Although correlation does not imply causation, it would be expected that the more food a country produces, th

Rice Production & Population

The high production of food in some African countries between 2004 and 2013 may be influenced by its popu
Although this is in no way certain, this idea may be measured using hypothesis testing. To understand and te
hypothesis, a particular food common to most countries will be used as a measure. In this case, `Rice (Mil`
`Equivalent)` will be used.

The dataset containing `Rice (Milled Equivalent)` production for all countries between 2004 and 2013 w
extracted from the `food production` dataset, then merged with `country code` dataset (contains the cou
per country) and the `World Bank population` dataset (contains the measured population per country over
years) to create the test data to be used in measuring the proposed hypoyhesis.

For the hypothesis testing the proposed X and Y variables are:

- X = The amount of rice produced in a year in African countries.
- Y = The average population of African countries in a year.

The proposed null and alternative hypothesis test statements to be used are given as:

Null Hypothesis: There is no significant correlation between the amount of rice produced in a year in African c the mean population of African countries in that yearthe mean population of African countries in that year. i.e and Y are not correlated.

$$H_0 : U_{corr} =< 0$$

- where H_0 is the notation for null hypothesis.
- U_{corr} is the notation for correlation between X and Y.

Alternative Hypothesis: There is a significant correlation between the amount of rice produced in a year in Afr countries and the mean population of African countries in that year. i.e There is a corrrelation between variabl

$$H_1 : U_{corr} > 0$$

- where H_1 is the notation for alternative hypothesis.
- U_{corr} is the notation for correlation between X and Y.

In [39]:
```
# Extracting the dataset for rice production in African countries over the years
rice_df = foodprod.query("Item == 'Rice (Milled Equivalent)'")

# Finding the sum of values per country per year
rice_tot = pd.DataFrame(rice_df.groupby(['Country', 'Year'], as_index = False)['Value'].sum())
rice_tot
```

Out[39]:

	Country	Year	Value
0	Algeria	2004	0
1	Algeria	2005	0
2	Algeria	2006	0
3	Algeria	2007	0
4	Algeria	2008	0
...	...	...	...
375	Zimbabwe	2009	0
376	Zimbabwe	2010	0
377	Zimbabwe	2011	0
378	Zimbabwe	2012	0
379	Zimbabwe	2013	0

380 rows × 3 columns

In [40]:
```
# Merging the rice dataset with the country code dataset
rice_prod = pd.merge(rice_tot, code, on ='Country', how = 'inner')

# Renaming the Value column in the rice production data
rice_prod.rename(columns = {"Value":"Rice Production (Kilotonnes)"}, inplace=True)
rice_prod.head()
```

Out[40]:

	Country	Year	Rice Production (Kilotonnes)	Country Code
0	Algeria	2004	0	DZA
1	Algeria	2005	0	DZA
2	Algeria	2006	0	DZA
3	Algeria	2007	0	DZA
4	Algeria	2008	0	DZA

In [41]:
```
# Importing the population dataset

pop_data = pd.read_csv("Population data.csv", skiprows = 4)
pop_data.head()
```

Out[41]:

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965
0	Aruba	ABW	Population, total	SP.POP.TOTL	54211.0	55438.0	56225.0	56695.0	57032.0	57360.0
1	Afghanistan	AFG	Population, total	SP.POP.TOTL	8996973.0	9169410.0	9351441.0	9543205.0	9744781.0	9956320.0
2	Angola	AGO	Population, total	SP.POP.TOTL	5454933.0	5531472.0	5608539.0	5679458.0	5735044.0	5770570.0
3	Albania	ALB	Population, total	SP.POP.TOTL	1608800.0	1659800.0	1711319.0	1762621.0	1814135.0	1864791.0
4	Andorra	AND	Population, total	SP.POP.TOTL	13411.0	14375.0	15370.0	16412.0	17469.0	18549.0

5 rows × 65 columns

In [42]:
```
# Dropping unwanted columns
pop_data.drop(['Indicator Name', 'Indicator Code'], axis = 1, inplace =True)
pop_data.head()

# Extracting only the population recorded between 2004 and 2013
pop_data = pop_data[['Country Name', 'Country Code', '2004', '2005', '2006', '2007', '2008', '
pop_data.head()
```

Out[42]:

	Country Name	Country Code	2004	2005	2006	2007	2008	2009	2010	
0	Aruba	ABW	98737.0	100031.0	100834.0	101222.0	101358.0	101455.0	101669.0	102
1	Afghanistan	AFG	24726684.0	25654277.0	26433049.0	27100536.0	27722276.0	28394813.0	29185507.0	30117
2	Angola	AGO	18758145.0	19433602.0	20149901.0	20905363.0	21695634.0	22514281.0	23356246.0	24220
3	Albania	ALB	3026939.0	3011487.0	2992547.0	2970017.0	2947314.0	2927519.0	2913021.0	2905
4	Andorra	AND	76244.0	78867.0	80993.0	82684.0	83862.0	84463.0	84449.0	83

In [43]:
```
# Cleaning the population dataset using the melt function to convert year rows to columns

pop_df = pd.melt(
    pop_data,
    id_vars=["Country Name", "Country Code"],
    var_name = "Year",
    value_name = "Population"
)
pop_df.head()
```

Out[43]:

	Country Name	Country Code	Year	Population
0	Aruba	ABW	2004	98737.0
1	Afghanistan	AFG	2004	24726684.0
2	Angola	AGO	2004	18758145.0
3	Albania	ALB	2004	3026939.0
4	Andorra	AND	2004	76244.0

In [44]:
```
# Checking the datatype for columns in the newly cleaned population dataset
pop_df.dtypes
```

Out[44]:
```
Country Name     object
Country Code     object
Year             object
Population      float64
dtype: object
```

In [45]:
```
# Converting the datatype in the Year column of the population dataset from object to integer

pop_df["Year"] = pop_df["Year"].astype(int)
```

In [46]:
```
# Merging the population and rice production dataset using the country code and year
pop_riceprod = pd.merge(rice_prod, pop_df, on= ["Country Code", "Year"], how = "inner")

# Dropping thr country name column
pop_riceprod.drop("Country Name", axis = 1, inplace = True)
pop_riceprod.head()
```

Out[46]:

	Country	Year	Rice Production (Kilotonnes)	Country Code	Population
0	Algeria	2004	0	DZA	32692163.0
1	Algeria	2005	0	DZA	33149724.0
2	Algeria	2006	0	DZA	33641002.0
3	Algeria	2007	0	DZA	34166972.0
4	Algeria	2008	0	DZA	34730608.0

In [47]:
```
# Computing the average rice production and popopulation per year

test_data = pd.DataFrame(pop_riceprod.groupby("Year", as_index=False).mean())
test_data
```

Out[47]:

	Year	Rice Production (Kilotonnes)	Population
0	2004	333.971429	2.130359e+07
1	2005	356.028571	2.181938e+07
2	2006	387.885714	2.235149e+07
3	2007	366.657143	2.290022e+07
4	2008	428.457143	2.346734e+07
5	2009	413.657143	2.405480e+07
6	2010	444.057143	2.466383e+07
7	2011	449.228571	2.529508e+07
8	2012	488.371429	2.594786e+07
9	2013	488.371429	2.661992e+07

In [48]:
```
# Performing the Pearson Correlation Test to measure the correlation between the average rice p

from scipy import stats
x = np.array(test_data['Rice Production (Kilotonnes)'])
y = np.array(test_data['Population'])

stats.pearsonr(x, y)
```

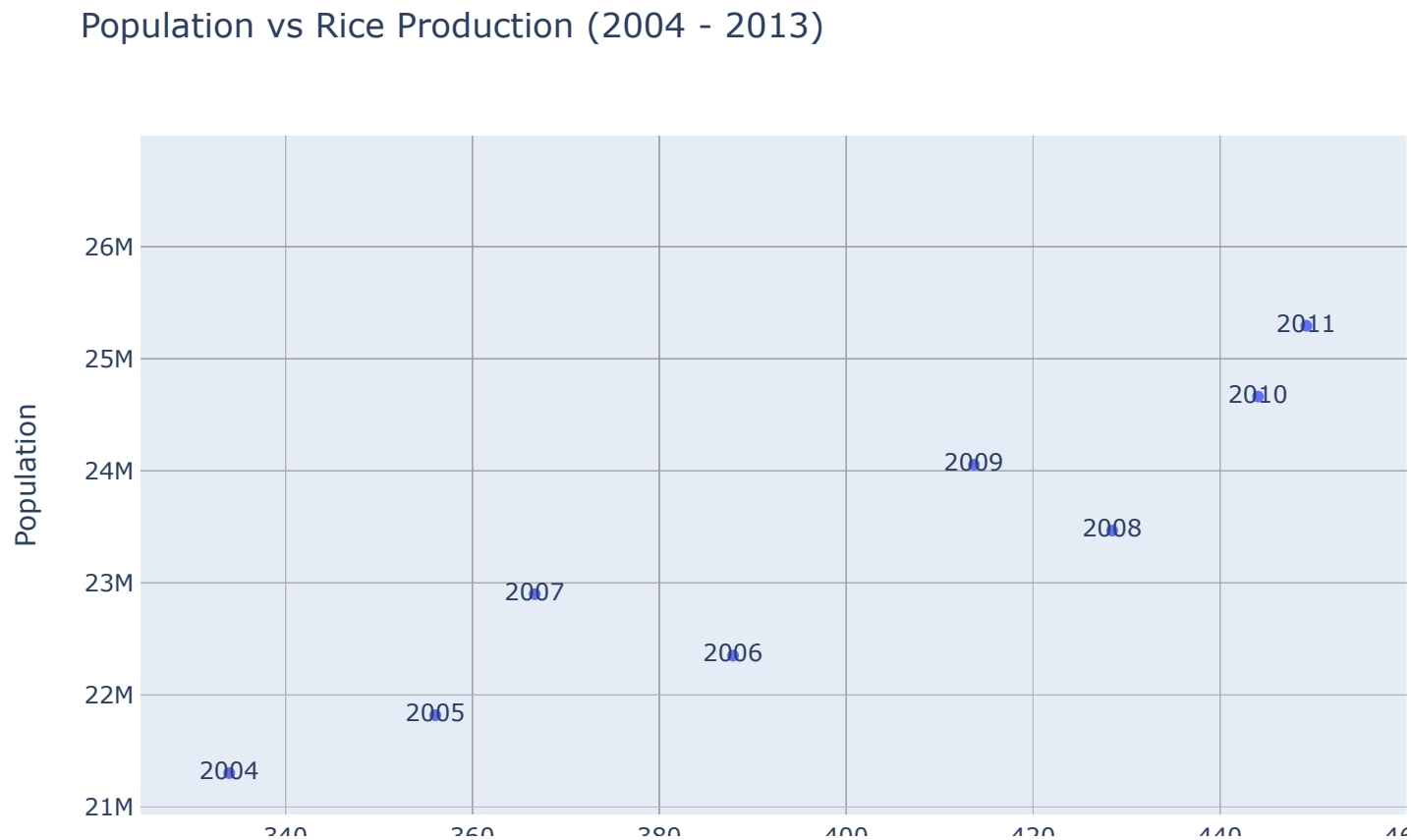
Out[48]:
```
(0.9669903743060018, 4.991516647534302e-06)
```

> The results above show a positive correlation of 0.97and a p-value of 4.99 x 10e-6. This means that there's a positive correlation between X and Y, andsuch a small p-value suggests that that H_0 is unlikely to be true. Th reject the null hypothesis at the 5% level based on the Pearson Correlation test.

It can therefore be concluded that the correlation coefficient (0.97) is significantly far from 0 i.e., there is a sig
correlation between mean rice production in a year and the mean population of African countries for that year

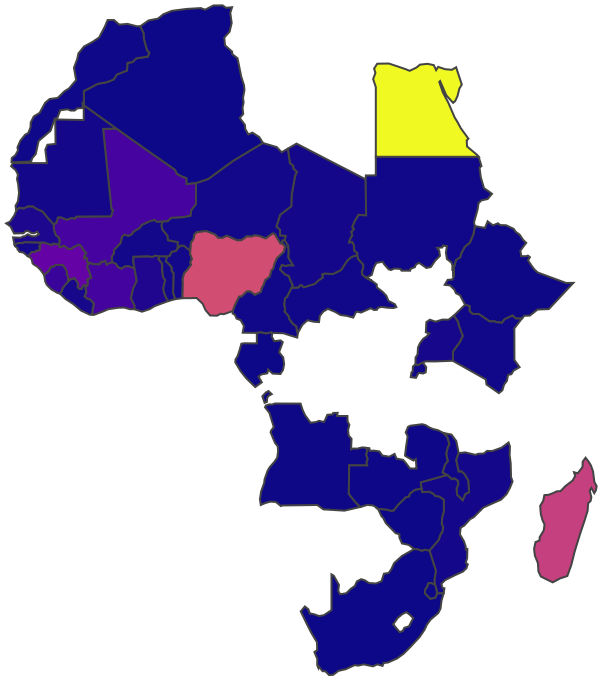
It is also very important to state that correlation does not imply causation. Although there exists a strong corre
between population and rice production, this doesn't necessarily mean that the production of rice drives popu
vice versa.

In [49]:
```
# visualising the test data with a scatter plot

fig = px.scatter(test_data, x = "Rice Production (Kilotonnes)", y = "Population", text = "Year
fig.show();
```



The scatterplot above shows a positive corrrelation between rice production and the population in African cou
the years. As the population increases, the average amount of rice produced tends to increase.

In [50]:
```
# Creating a map plot of Africa showing how the production of rice varies per country over the

fig = px.choropleth(pop_riceprod, locations="Country Code",
                    color='Rice Production (Kilotonnes)',
                    hover_name="Country",
                    animation_frame='Year',
                    scope="africa"
                   )
fig.update_geos(fitbounds="locations", visible=False)
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```



Year=2004

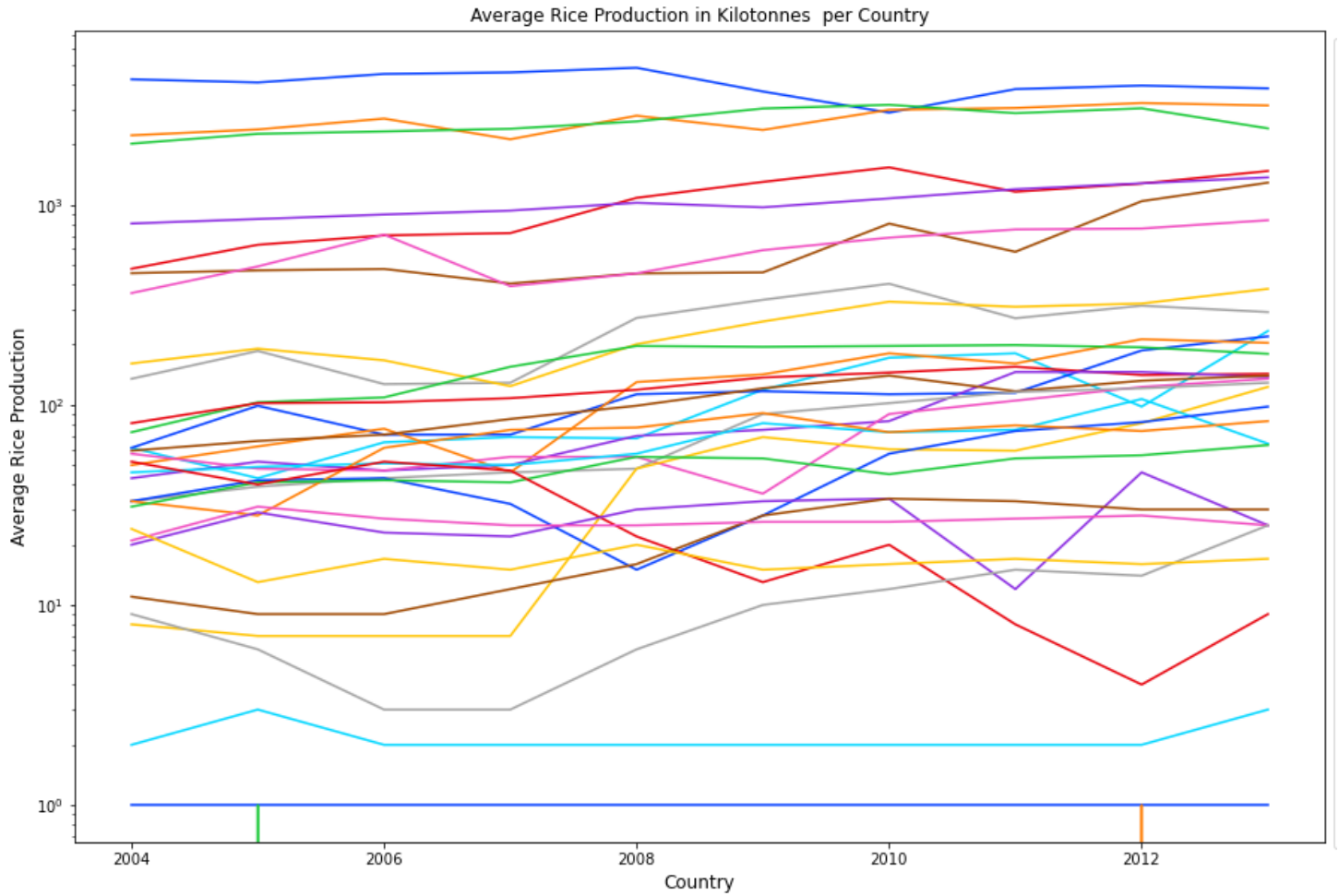
In [51]:
```
# To obtain the top countries with the highest rice production in kilotonnes over the years

top = pd.DataFrame(pop_riceprod.groupby(['Country', 'Year'], as_index = False)['Rice Productio
top.sort_values(by = 'Rice Production (Kilotonnes)', ascending= False, inplace =True)
top
```

Out[51]:

	Country	Year	Rice Production (Kilotonnes)
84	Egypt	2008	4838
83	Egypt	2007	4587
82	Egypt	2006	4506
80	Egypt	2004	4237
81	Egypt	2005	4086
...	...	...	...
304	Swaziland	2008	0
305	Swaziland	2009	0
306	Swaziland	2010	0
307	Swaziland	2011	0
349	Zimbabwe	2013	0

350 rows × 3 columns

In [52]:
```
# Visualising in a plot the top countries with the highest rice production in kilotonnes over t

plt.subplots(figsize=(15,10))
lineplt = sns.lineplot(x=top['Year'], y= top['Rice Production (Kilotonnes)'], hue=top['Country
lineplt.set_yscale('log')
lineplt.legend(loc='upper right',bbox_to_anchor= (1.2, 1))

plt.title('Average Rice Production in Kilotonnes  per Country')
plt.xlabel('Country', fontsize = 12)
plt.ylabel('Average Rice Production', fontsize = 12);
```



It can be observed that `Egypt`, `Nigeria` and `Madagascar` are the highest `Rice (Milled Equivalent)` countries between 2004 and 2013. To compare if these 3 countries have any other products in common i.e Eg and Madagascar, a dataframe will be created for each country with their total produce. Then the items in each dataframes will be compared to see what they have in common.

In [53]:
```
# From the original production dataset, 3 dataframes for Egypt, Nigeria and Madagascar are extr

Egypt = foodprod.query('Country == "Egypt"')
Nigeria = foodprod.query('Country == "Nigeria"')
Madagascar = foodprod.query('Country == "Madagascar"')
```

In [54]:
```
# Comparing between Egypt and Nigeria to find other simmilar produced items between them

Egy_Nig = Egypt['Item'][Egypt['Item'].isin(Nigeria['Item'])].unique()
Egy_Nig
```

Out[54]:
```
array(['Wheat and products', 'Rice (Milled Equivalent)',
       'Maize and products', 'Sorghum and products',
       'Potatoes and products', 'Sweet potatoes', 'Roots, Other',
       'Sugar cane', 'Sugar (Raw Equivalent)',
       'Pulses, Other and products', 'Nuts and products', 'Soyabeans',
       'Groundnuts (Shelled Eq)', 'Cottonseed', 'Sesame seed',
       'Oilcrops, Other', 'Soyabean Oil', 'Groundnut Oil',
       'Cottonseed Oil', 'Oilcrops Oil, Other', 'Tomatoes and products',
       'Onions', 'Vegetables, Other', 'Citrus, Other', 'Fruits, Other',
       'Pimento', 'Spices, Other', 'Beer', 'Bovine Meat',
       'Mutton & Goat Meat', 'Pigmeat', 'Poultry Meat', 'Meat, Other',
       'Offals, Edible', 'Butter, Ghee', 'Fats, Animals, Raw', 'Eggs',
       'Milk - Excluding Butter', 'Freshwater Fish', 'Demersal Fish',
       'Pelagic Fish', 'Marine Fish, Other', 'Crustaceans', 'Cephalopods',
       'Molluscs, Other'], dtype=object)
```

In [55]:
```
# To compute the total number of similar items produced by both Egypt and Nigeria

len(Egy_Nig)
```

Out[55]:
```
45
```

In [56]:
```
# Comparing between Egypt and Madagascar to find other simmilar produced items between them

Egy_Mad= Egypt['Item'][Egypt['Item'].isin(Madagascar['Item'])].unique()
Egy_Mad
```

Out[56]:
```
array(['Wheat and products', 'Rice (Milled Equivalent)',
       'Maize and products', 'Sorghum and products',
       'Potatoes and products', 'Sweet potatoes', 'Roots, Other',
       'Sugar cane', 'Sugar (Raw Equivalent)', 'Sweeteners, Other',
       'Honey', 'Beans', 'Peas', 'Pulses, Other and products',
       'Nuts and products', 'Soyabeans', 'Groundnuts (Shelled Eq)',
       'Cottonseed', 'Oilcrops, Other', 'Groundnut Oil', 'Cottonseed Oil',
       'Oilcrops Oil, Other', 'Tomatoes and products', 'Onions',
       'Vegetables, Other', 'Oranges, Mandarines',
       'Lemons, Limes and products', 'Grapefruit and products', 'Bananas',
       'Apples and products', 'Grapes and products (excl wine)',
       'Fruits, Other', 'Pimento', 'Spices, Other', 'Wine', 'Beer',
       'Beverages, Alcoholic', 'Bovine Meat', 'Mutton & Goat Meat',
       'Pigmeat', 'Poultry Meat', 'Meat, Other', 'Offals, Edible',
       'Fats, Animals, Raw', 'Fish, Body Oil', 'Eggs',
       'Milk - Excluding Butter', 'Freshwater Fish', 'Demersal Fish',
       'Pelagic Fish', 'Marine Fish, Other', 'Crustaceans', 'Cephalopods',
       'Molluscs, Other', 'Aquatic Animals, Others', 'Aquatic Plants'],
      dtype=object)
```

In [57]:
```
# To compute the total number of similar items produced by both Egypt and Madagascar

len(Egy_Mad)
```

Out[57]:
```
56
```

In [58]:
```
# Comparing between Nigeria and Madagascar to find other simmilar produced items between them

Nig_Mad= Nigeria['Item'][Nigeria['Item'].isin(Madagascar['Item'])].unique()
Nig_Mad
```

Out[58]:
```
array(['Wheat and products', 'Rice (Milled Equivalent)',
       'Maize and products', 'Sorghum and products',
       'Cassava and products', 'Potatoes and products', 'Sweet potatoes',
       'Roots, Other', 'Sugar cane', 'Sugar (Raw Equivalent)',
       'Pulses, Other and products', 'Nuts and products', 'Soyabeans',
       'Groundnuts (Shelled Eq)', 'Cottonseed', 'Coconuts - Incl Copra',
       'Palm kernels', 'Oilcrops, Other', 'Groundnut Oil',
       'Cottonseed Oil', 'Palmkernel Oil', 'Palm Oil', 'Coconut Oil',
       'Oilcrops Oil, Other', 'Tomatoes and products', 'Onions',
       'Vegetables, Other', 'Pineapples and products', 'Fruits, Other',
       'Coffee and products', 'Cocoa Beans and products', 'Pimento',
       'Spices, Other', 'Beer', 'Bovine Meat', 'Mutton & Goat Meat',
       'Pigmeat', 'Poultry Meat', 'Meat, Other', 'Offals, Edible',
       'Fats, Animals, Raw', 'Eggs', 'Milk - Excluding Butter',
       'Freshwater Fish', 'Demersal Fish', 'Pelagic Fish',
       'Marine Fish, Other', 'Crustaceans', 'Cephalopods',
       'Molluscs, Other'], dtype=object)
```

In [59]:
```
# To compute the total number of similar items produced by both Nigeria and Madagascar

len(Nig_Mad)
```

Out[59]:
```
50
```

Conclusion

From the analysis carried out above for the food produced and food supplied in African Countries between 20 can be concluded that:

- There is a strong positive correlation between the average production of rice and the population between 2013. Although, this does not imply that an increase in population drives an increase in rice production, it the relationship between the two variables.
- The countries with the highest food production dont have high food supply i.e the amount of food supplie years is relatively smaller compared to the amount of food being produced. Although there is a high amou production and population over the years, there is a low amount smaller food supply when compared to t of food produced. It can be deduced that there is a possible food shortage in African countries between 2 2013.