EDA ON REFUGE DATA BY CHUKWUEMEKA

June 30, 2021

1 GLOBAL FOOD PRICE PROJECT PROPOSAL

1.1 INTRODUCTION

Global Food Price is described as the average price of food commodities across countries, regions, and the global level. The level of food price usually depends on the food production process which includes food marketing and distribution. Fluctuations in the price of commodities can be multifactorial ranging from availability of natural resources for agriculture, market demand, energy cost, cost of production, exchange rate, government policy, and weather events amongst all. These factors have both positive and negative effects.

Starting from 2007-08 there was a surge in food prices, particularly in developing countries. This led to a global crisis causing political and economic instability as well as social unrest between poor and developed countries. The trend dropped and increased again in 2009 and 2010, reaching new heights in 2011 & 2012. Over the years prices dropped significantly reaching a lower point in March 2016 with a reduced Food and Agricultural Organization(FAO) food price index. The FAO Food Price Index (FFPI) is a measure of the monthly change in international prices of a basket of food commodities. In recent times, global food prices rose in March 2021, which marked the 10th consecutive monthly increase with products like vegetable oil and dairy products leading the rise.

It is needful to say that at this point, the impact of food prices not only provides an indicator of the balance of agricultural produce and market demands but also has an impact on the cost of living, food policies, and migration. While the producers benefit from the high food prices, consumers only benefit when the food prices are low. By implication, food prices now have an impact on food affordability, quality of a diet, undernourishment, and hunger.

In line with the United Nations Development Programme's (UNDP) sustainable development goal 2, i.e., zero hunger we will be taking a closer look at the trends in global food prices, possible causes and effects of increased global food prices and offer our solution.

1.2 PROBLEM STATEMENT

It can be observed from the previous discussion that global food price fluctuations can cause famine and large population shift. Hence, Identifying the drivers of global food prices and predicting future changes in global food prices, could help in understanding food prices and its causal effects.

1.3 AIM

Our research aims to understand and analyze fluctuations in global food prices and pair the outcome with currency fluctuations, weather patterns, and refugee movements. This will help us to build an end-to-end analysis and a food price prediction engine that will help the Government make better

decisions on food policy adjustments, International bodies with planning of food aid programmes, Individuals with planning and productivity in the advent of a potential food price crisis…

1.4 OBJECTIVES

To achieve the above aim, we will: Analyze available datasets to observe and make inferences about changing food prices, fluctuations, and the trend they follow. Attempt to compare their correlation with factors such as currency fluctuation, weather patterns, and refugee movements. Investigate which food item controls the trends of the majority of the food markets. Use the best-performed model in predicting food prices and deploying it in a web application that can predict food prices.

1.5 REVIEW OF PAST LITERATURE

Most of the recent research on Global Food Prices has centered around policy-making across nations and countries in addressing the issue. An article by ALNAP, it cited IFPRI/CGIAR, 2008 where it was stated that factors that have contributed to the global food price crisis are either cyclical, structural, or unique. Various World Organizations like WFP, UNOCHA/CERF, UNICEF, IMF, WORLD BANK, NEPAD, ADB, AU, WTO, etc have championed different policies towards mitigating the menace of the Global Food Prices crisis, especially through financial aids. Notable among them are FAO's Procurement and distribution of seeds, fertilizers, and other inputs which have been carried out in 54 countries under the Food and Agriculture Organisation (FAO) Initiative on Soaring Food Prices (ISFP). FAO is also urging governments and the International community to implement measures in support of poor countries hard hit by food price increases, specifically to provide small farmers with improved access to inputs like seeds and fertilizers to increase local crop production (RHVP/Wahenga brief, 2008).

From a micro perspective, Nigeria as a country has had several policies both in present and in the past regarding mitigation of the food prices crisis. Policies like Operation Feed the Nation, Green Revolution, and presently FADAMA programs. These policies and programs have contributed little or none to solving the challenge of the food price crisis.

With regards to predictive modeling technique, Artificial Neural Network(ANN) algorithm and Time Series Forecasting algorithms like ARIMA have been used recently by researchers in this Global Food prices crisis domain. A Machine Learning Approach to Forecasting Consumer Food Prices, J. Jay Harris(2017), applied ANN in modeling Global Food Prices, which was significantly insightful. In this project, we shall also be exploring predictive modeling techniques as well as time series models in forecasting food prices.

1.6 DATA COLLECTION

Data related to global food prices will be collected from the Open source database compiled by the World Food Programme and distributed by the Humanitarian Data Exchange. Data on currency fluctuations will be gotten from the World Bank's open-source database on official exchange rates. Data on Refugee movements will be extracted from the Refugee statistics of the United Nations High Commissioner for Refugees. Data on Weather patterns will be excerpts from the World Meteorological Organization.

1.7 MACHINE LEARNING WORKFLOW

Data Volumes ↓ Data Ingestion ↓ Data Wrangling ↓ Data Cleaning ↓ Data preprocessing→ Stationarity check→ Time series modeling→forecasting ↓ Predictive modeling

1.8 WEB APPLICATION DEVELOPMENT FOR THE MODEL

The end product of this Global food prediction engine will be in the form of a web app that can be accessed from anywhere as long as there is an Internet connection, It will have a drop-down list to select the food categories, and a graph showing the trend of the price fluctuations over the years and the prediction over the next couple of months. The web app will be built using the streamlit service which makes deploying models quick and easy. The model which would have been worked on and perfected is saved as a pickle file and a python script is created for the usage of the model, then using streamlit, the interface stated above is created in python, then connected and deployed for use.

1.9 References:

"World Food Situation". FAO. Archived from the original on 29 April 2011. Retrieved 24 April 2011. How do Food Prices Affect Producers and Consumers in Developing Countries?, ICTSD, Information Note Number 10, September 2009 UN Food and Agriculture Organization (2009). The State of Food Insecurity in the World 2009. Rome. Rahman, M. Mizanur (11 August 2011). "Food price inflation: Global and national problem". The Daily Star. "FAO Food Price Index". FAO. Retrieved 2 May 2017.

1.10 Group Trailblazers:

- Abiona Oluwafemi
- Roqeebat Olanrewaju
- Omeh Chukwuemeka
- Habeebullah Agbaje

1.11 Terms

1.11.1 Who is a refugee?

Refugees are people who have fled war, violence, conflict or persecution and have crossed an international border to find safety in another country.They often have had to flee with little more than the clothes on their back, leaving behind homes, possessions, jobs and loved ones. Refugees are defined and protected in international law. The 1951 Refugee Convention is a key legal document and defines a refugee as:"someone who is unable or unwilling to return to their country of origin owing to a well-founded fear of being persecuted for reasons of race, religion, nationality, membership of a particular social group, or political opinion." link

A refugee is a person who has fled their own country because they are at risk of serious human rights violations and persecution there. The risks to their safety and life were so great that they felt they had no choice but to leave and seek safety outside their country because their own government cannot or will not protect them from those dangers. Refugees have a right to international protection.

1.11.2 Who is an asylum-seeker?

An asylum-seeker is a person who has left their country and is seeking protection from persecution and serious human rights violations in another country, but who hasn't yet been legally recognized as a refugee and is waiting to receive a decision on their asylum claim. Seeking asylum is a human right. This means everyone should be allowed to enter another country to seek asylum.

1.11.3 Who is a migrant?

There is no internationally accepted legal definition of a migrant. Like most agencies and organizations, we at Amnesty International understand migrants to be people staying outside their country of origin, who are not asylum-seekers or refugees.

Some migrants leave their country because they want to work, study or join family, for example. Others feel they must leave because of poverty, political unrest, gang violence, natural disasters or other serious circumstances that exist there.

2 Import Libraries

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy
```

3 Setting display layout

[2]:
```
pd.set_option("display.max_column", None)
pd.set_option("display.max_colwidth", None)
pd.set_option("display.max_row", None)
pd.set_option("display.float_format", lambda x: "%.2f" %x)
plt.style.use('ggplot')
plt.rcParams['font.size'] = 10
```

[3]:
```
temp = pd.read_csv("https://raw.githubusercontent.com/Hab-eeb/Global_Food_Price/
↪main/Datasets/GlobalTemperatures.csv")
print(f"The number of rows is: {temp.shape[0]} and numbers of columns is: {temp.
↪shape[1]}")
```

```
The number of rows is: 3192 and numbers of columns is: 9
```

[4]:
```
temp.head()
```

[4]:

	dt	LandAverageTemperature	LandAverageTemperatureUncertainty	\
0	1750-01-01	3.03	3.57	
1	1750-02-01	3.08	3.70	
2	1750-03-01	5.63	3.08	

```
3  1750-04-01                    8.49                                2.45
4  1750-05-01                   11.57                                2.07

   LandMaxTemperature  LandMaxTemperatureUncertainty  LandMinTemperature  \
0                 NaN                            NaN                 NaN
1                 NaN                            NaN                 NaN
2                 NaN                            NaN                 NaN
3                 NaN                            NaN                 NaN
4                 NaN                            NaN                 NaN

   LandMinTemperatureUncertainty  LandAndOceanAverageTemperature  \
0                            NaN                             NaN
1                            NaN                             NaN
2                            NaN                             NaN
3                            NaN                             NaN
4                            NaN                             NaN

   LandAndOceanAverageTemperatureUncertainty
0                                        NaN
1                                        NaN
2                                        NaN
3                                        NaN
4                                        NaN
```

[5]:
```
ref = pd.read_csv("populations_countries.csv",delimiter = ',')
print(f"The number of rows is: {ref.shape[0]} and numbers of columns is: {ref.
 ↪shape[1]}")
```

```
The number of rows is: 90004 and numbers of columns is: 11
```

[6]:
```
ref.head()
```

[6]:
```
   Year                    Country of origin Country of origin (ISO)  \
0  2000                          Afghanistan                      AFG
1  2000                                 Iraq                      IRQ
2  2000  Serbia and Kosovo: S/RES/1244 (1999)                      SRB
3  2000                               Turkey                      TUR
4  2000                                 Chad                      TCD

  Country of asylum Country of asylum (ISO)  Refugees under UNHCR's mandate  \
0       Afghanistan                     AFG                               0
1           Albania                     ALB                               9
2           Albania                     ALB                             507
3           Albania                     ALB                               5
4           Algeria                     DZA                              20

   Asylum-seekers  IDPs of concern to UNHCR  Venezuelans displaced abroad  \
```

```
0                0                      758625                          NaN
1                0                           0                          NaN
2                5                           0                          NaN
3                0                           0                          NaN
4               19                           0                          NaN

   Stateless persons  Others of concern
0                  0                  0
1                  0                  0
2                  0                  0
3                  0                  0
4                  0                  0
```

[7]:
```
"""
=====No duplicate Entries====
"""
dp = ref.duplicated()
print(f"The number of duplicated row and column are: {ref[dp].shape[0]} and␣
 ↪{ref[dp].shape[1]} respectively")
ref.duplicated().sum()
```

```
The number of duplicated row and column are: 0 and 11 respectively
```

[7]:
```
0
```

[8]:
```
ref.columns
```

[8]:
```
Index(['Year', 'Country of origin', 'Country of origin (ISO)',
       'Country of asylum', 'Country of asylum (ISO)',
       'Refugees under UNHCR's mandate', 'Asylum-seekers',
       'IDPs of concern to UNHCR', 'Venezuelans displaced abroad',
       'Stateless persons', 'Others of concern'],
      dtype='object')
```

[9]:
```
ref.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 90004 entries, 0 to 90003
Data columns (total 11 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Year                            90004 non-null  int64
 1   Country of origin               90004 non-null  object
 2   Country of origin (ISO)         88989 non-null  object
 3   Country of asylum               90004 non-null  object
 4   Country of asylum (ISO)         90004 non-null  object
 5   Refugees under UNHCR's mandate  90004 non-null  int64
 6   Asylum-seekers                  90004 non-null  int64
```

```
 7   IDPs of concern to UNHCR        90004 non-null  int64
 8   Venezuelans displaced abroad    59 non-null     float64
 9   Stateless persons               90004 non-null  int64
 10  Others of concern               90004 non-null  int64
dtypes: float64(1), int64(6), object(4)
memory usage: 7.6+ MB
```

[10]:
```
ref.dtypes
```

[10]:
```
Year                                int64
Country of origin                  object
Country of origin (ISO)            object
Country of asylum                  object
Country of asylum (ISO)            object
Refugees under UNHCR's mandate      int64
Asylum-seekers                      int64
IDPs of concern to UNHCR            int64
Venezuelans displaced abroad      float64
Stateless persons                   int64
Others of concern                   int64
dtype: object
```

[11]:
```
ref_obj = ref.select_dtypes(include = 'object')
ref_int = ref.select_dtypes(include = 'int')
ref_float = ref.select_dtypes(include = 'float')
ref_obj.shape, ref_int.shape, ref_float.shape
```

[11]:
```
((90004, 4), (90004, 6), (90004, 1))
```

[12]:
```
ref.nunique().sort_values(ascending = False)
```

[12]:
```
Refugees under UNHCR's mandate    6996
Asylum-seekers                    3915
Stateless persons                  675
Others of concern                  640
IDPs of concern to UNHCR           454
Country of origin                  212
Country of origin (ISO)            211
Country of asylum                  189
Country of asylum (ISO)            189
Venezuelans displaced abroad        59
Year                                21
dtype: int64
```

[13]:
```
ref.describe(include = 'all')
```

[13]:

	Year	Country of origin	Country of origin (ISO)
count	90004.00	90004	88989
unique	NaN	212	211
top	NaN	Somalia	SOM
freq	NaN	1990	1990
mean	2011.07	NaN	NaN
std	6.02	NaN	NaN
min	2000.00	NaN	NaN
25%	2006.00	NaN	NaN
50%	2012.00	NaN	NaN
75%	2016.00	NaN	NaN
max	2020.00	NaN	NaN

	Country of asylum	Country of asylum (ISO)
count	90004	90004
unique	189	189
top	United States of America	USA
freq	3572	3572
mean	NaN	NaN
std	NaN	NaN
min	NaN	NaN
25%	NaN	NaN
50%	NaN	NaN
75%	NaN	NaN
max	NaN	NaN

	Refugees under UNHCR's mandate	Asylum-seekers
count	90004.00	90004.00
unique	NaN	NaN
top	NaN	NaN
freq	NaN	NaN
mean	3077.74	393.77
std	46313.22	5586.88
min	0.00	0.00
25%	5.00	0.00
50%	14.00	6.00
75%	103.00	41.00
max	3641370.00	940668.00

	IDPs of concern to UNHCR	Venezuelans displaced abroad
count	90004.00	59.00
unique	NaN	NaN
top	NaN	NaN
freq	NaN	NaN
mean	4881.33	170025.02
std	124509.91	358009.88
min	0.00	11.00

```
25%                        0.00                      6568.00
50%                        0.00                     25686.00
75%                        0.00                    142565.50
max                  8252788.00                   1771237.00

        Stateless persons  Others of concern
count            90004.00           90004.00
unique                NaN                NaN
top                   NaN                NaN
freq                  NaN                NaN
mean               732.52             362.10
std              26468.49           16814.69
min                  0.00               0.00
25%                  0.00               0.00
50%                  0.00               0.00
75%                  0.00               0.00
max            3500000.00         2351313.00
```

[14]:
```
plt.figure(figsize = (15,8))
ref_corr = ref.corr()
sns.heatmap(ref_corr, annot = True, square = True)
plt.show()
```

[15]:
```
ref["Country of origin"].unique()
```

[15]:
```
array(['Afghanistan', 'Iraq', 'Serbia and Kosovo: S/RES/1244 (1999)',
       'Turkey', 'Chad', 'Cameroon', 'Congo', 'Dem. Rep. of the Congo',
       'Palestinian', 'Guinea', 'Liberia', 'Libya', 'Mali', 'Morocco',
       'Nigeria', 'Rwanda', 'Sierra Leone', 'Somalia', 'Sudan',
       'Syrian Arab Rep.', 'Western Sahara', 'Unknown ', 'Angola',
       'Burundi', 'Comoros', 'Guinea-Bissau', 'United Rep. of Tanzania',
       'Zambia', 'Djibouti', 'Eritrea', 'Ethiopia', 'Russian Federation',
       'Yemen', 'Stateless', 'Albania', 'Algeria', 'Armenia', 'Benin',
       'Bangladesh', 'Bosnia and Herzegovina', 'Bulgaria', 'Chile',
       'Colombia', 'Cuba', 'Dominican Rep.', 'Ecuador', 'Estonia',
       'Georgia', 'Ghana', 'Haiti', 'India', 'Iran (Islamic Rep. of)',
       'Kazakhstan', 'Kyrgyzstan', "Lao People's Dem. Rep.", 'Lebanon',
       'Sri Lanka', 'Nicaragua', 'Pakistan', 'Paraguay', 'Peru',
       'Romania', 'Senegal', 'Viet Nam', 'Tunisia', 'Ukraine',
```

```
'Azerbaijan', 'Egypt', 'Argentina', 'Austria', 'Bahrain',
'Belarus', 'Bolivia (Plurinational State of)', 'Brazil',
'Cambodia', 'China', 'Cyprus', 'Czechia', 'Fiji', 'France',
'United Kingdom of Great Britain and Northern Ireland', 'Germany',
'Guatemala', 'China, Hong Kong SAR', 'Croatia', 'Hungary',
'Indonesia', 'Israel', 'Italy', 'Jordan', 'Kenya', 'Rep. of Korea',
"Dem. People's Rep. of Korea", 'Kuwait', 'Lithuania', 'Latvia',
'North Macedonia', 'Rep. of Moldova', 'Mexico', 'Malaysia',
'Mongolia', 'Mauritius', 'Myanmar', 'Nepal', 'Niger',
'Philippines', 'Poland', 'Portugal', 'South Africa', 'El Salvador',
'Saudi Arabia', 'Singapore', 'Solomon Islands', 'Slovenia',
'Thailand', 'Timor-Leste', 'Tonga', 'United Arab Emirates',
'Uganda', 'Uruguay', 'Uzbekistan', 'Zimbabwe', "Cote d'Ivoire",
'Tajikistan', 'Togo', 'Bhutan', 'Burkina Faso', 'Mauritania',
'Central African Rep.', 'Equatorial Guinea', 'Madagascar',
'Namibia', 'China, Macao SAR', 'Honduras', 'Antigua and Barbuda',
'Barbados', 'Belgium', 'Botswana', 'Costa Rica', 'Denmark',
'Dominica', 'Gabon', 'Gambia', 'Greece', 'Grenada', 'Guyana',
'Iceland', 'Ireland', 'Jamaica', 'Japan', 'Saint Lucia', 'Malawi',
'Mozambique', 'Malta', 'Netherlands', 'Oman', 'Panama', 'Qatar',
'Spain', 'Slovakia', 'Eswatini', 'Sweden', 'Switzerland',
'Trinidad and Tobago', 'United States of America',
'Saint Vincent and the Grenadines',
'Venezuela (Bolivarian Republic of)', 'Tibetan', 'Turkmenistan',
'Seychelles', 'Sao Tome and Principe', 'Papua New Guinea',
'Suriname', 'Tuvalu', 'Canada', 'Belize', 'Australia', 'Bahamas',
'Cabo Verde', 'Finland', 'Nauru', 'San Marino',
'Saint Kitts and Nevis', 'Samoa', 'Lesotho', 'Andorra',
'New Zealand', 'Norway', 'Micronesia (Federated States of)',
'Gibraltar', 'Turks and Caicos Islands', 'Kiribati', 'Maldives',
'Bermuda', 'Brunei Darussalam', 'New Caledonia', 'Monaco',
'Montenegro', 'Holy See', 'South Sudan', 'Niue', 'Palau',
'Cayman Islands', 'Marshall Islands', 'Curacao ', 'Guadeloupe',
'Vanuatu', 'French Guiana', 'Luxembourg', 'Liechtenstein',
'Anguilla', 'Martinique'], dtype=object)
```

4 Feature Engineering

- To create a continent and sub-region features of the country's of origin or country of asyluumns

[16]:
```
cont = pd.read_csv("countryContinent.csv")
cont.shape
```

[16]:
```
(249, 9)
```

[17]:
```
cont.head()
```

[17]:
```
           country code_2 code_3  country_code     iso_3166_2 continent  \
0      Afghanistan     AF     AFG             4  ISO 3166-2:AF      Asia
1    land Islands     AX     ALA           248  ISO 3166-2:AX    Europe
2          Albania     AL     ALB             8  ISO 3166-2:AL    Europe
3          Algeria     DZ     DZA            12  ISO 3166-2:DZ    Africa
4   American Samoa     AS     ASM            16  ISO 3166-2:AS   Oceania

         sub_region  region_code  sub_region_code
0     Southern Asia       142.00            34.00
1  Northern Europe       150.00           154.00
2  Southern Europe       150.00            39.00
3  Northern Africa         2.00            15.00
4        Polynesia         9.00            61.00
```

[18]:
```
cont.nunique()
```

[18]:
```
country           249
code_2            248
code_3            249
country_code      249
iso_3166_2        249
continent           5
sub_region         22
region_code         5
sub_region_code    22
dtype: int64
```

[19]:
```
cont.continent.value_counts()
```

[19]:
```
Africa      58
Americas    55
Asia        51
Europe      51
Oceania     25
Name: continent, dtype: int64
```

[20]:
```
cont.sub_region.value_counts()
```

[20]:
```
Caribbean                  28
Eastern Africa             20
Western Asia               18
Western Africa             17
Northern Europe            16
Southern Europe            16
South America              14
South-Eastern Asia         11
Eastern Europe             10
```

```
Polynesia                    10
Southern Asia                 9
Western Europe                9
Middle Africa                 9
Central America               8
Eastern Asia                  8
Micronesia                    7
Northern Africa               7
Central Asia                  5
Southern Africa               5
Melanesia                     5
Northern America              5
Australia and New Zealand     3
Name: sub_region, dtype: int64
```

[21]:
```
ref["Country of origin"].unique()
```

[21]:
```
array(['Afghanistan', 'Iraq', 'Serbia and Kosovo: S/RES/1244 (1999)',
       'Turkey', 'Chad', 'Cameroon', 'Congo', 'Dem. Rep. of the Congo',
       'Palestinian', 'Guinea', 'Liberia', 'Libya', 'Mali', 'Morocco',
       'Nigeria', 'Rwanda', 'Sierra Leone', 'Somalia', 'Sudan',
       'Syrian Arab Rep.', 'Western Sahara', 'Unknown ', 'Angola',
       'Burundi', 'Comoros', 'Guinea-Bissau', 'United Rep. of Tanzania',
       'Zambia', 'Djibouti', 'Eritrea', 'Ethiopia', 'Russian Federation',
       'Yemen', 'Stateless', 'Albania', 'Algeria', 'Armenia', 'Benin',
       'Bangladesh', 'Bosnia and Herzegovina', 'Bulgaria', 'Chile',
       'Colombia', 'Cuba', 'Dominican Rep.', 'Ecuador', 'Estonia',
       'Georgia', 'Ghana', 'Haiti', 'India', 'Iran (Islamic Rep. of)',
       'Kazakhstan', 'Kyrgyzstan', "Lao People's Dem. Rep.", 'Lebanon',
       'Sri Lanka', 'Nicaragua', 'Pakistan', 'Paraguay', 'Peru',
       'Romania', 'Senegal', 'Viet Nam', 'Tunisia', 'Ukraine',
       'Azerbaijan', 'Egypt', 'Argentina', 'Austria', 'Bahrain',
       'Belarus', 'Bolivia (Plurinational State of)', 'Brazil',
       'Cambodia', 'China', 'Cyprus', 'Czechia', 'Fiji', 'France',
       'United Kingdom of Great Britain and Northern Ireland', 'Germany',
       'Guatemala', 'China, Hong Kong SAR', 'Croatia', 'Hungary',
       'Indonesia', 'Israel', 'Italy', 'Jordan', 'Kenya', 'Rep. of Korea',
       "Dem. People's Rep. of Korea", 'Kuwait', 'Lithuania', 'Latvia',
       'North Macedonia', 'Rep. of Moldova', 'Mexico', 'Malaysia',
       'Mongolia', 'Mauritius', 'Myanmar', 'Nepal', 'Niger',
       'Philippines', 'Poland', 'Portugal', 'South Africa', 'El Salvador',
       'Saudi Arabia', 'Singapore', 'Solomon Islands', 'Slovenia',
       'Thailand', 'Timor-Leste', 'Tonga', 'United Arab Emirates',
       'Uganda', 'Uruguay', 'Uzbekistan', 'Zimbabwe', "Cote d'Ivoire",
       'Tajikistan', 'Togo', 'Bhutan', 'Burkina Faso', 'Mauritania',
       'Central African Rep.', 'Equatorial Guinea', 'Madagascar',
       'Namibia', 'China, Macao SAR', 'Honduras', 'Antigua and Barbuda',
```

```
       'Barbados', 'Belgium', 'Botswana', 'Costa Rica', 'Denmark',
       'Dominica', 'Gabon', 'Gambia', 'Greece', 'Grenada', 'Guyana',
       'Iceland', 'Ireland', 'Jamaica', 'Japan', 'Saint Lucia', 'Malawi',
       'Mozambique', 'Malta', 'Netherlands', 'Oman', 'Panama', 'Qatar',
       'Spain', 'Slovakia', 'Eswatini', 'Sweden', 'Switzerland',
       'Trinidad and Tobago', 'United States of America',
       'Saint Vincent and the Grenadines',
       'Venezuela (Bolivarian Republic of)', 'Tibetan', 'Turkmenistan',
       'Seychelles', 'Sao Tome and Principe', 'Papua New Guinea',
       'Suriname', 'Tuvalu', 'Canada', 'Belize', 'Australia', 'Bahamas',
       'Cabo Verde', 'Finland', 'Nauru', 'San Marino',
       'Saint Kitts and Nevis', 'Samoa', 'Lesotho', 'Andorra',
       'New Zealand', 'Norway', 'Micronesia (Federated States of)',
       'Gibraltar', 'Turks and Caicos Islands', 'Kiribati', 'Maldives',
       'Bermuda', 'Brunei Darussalam', 'New Caledonia', 'Monaco',
       'Montenegro', 'Holy See', 'South Sudan', 'Niue', 'Palau',
       'Cayman Islands', 'Marshall Islands', 'Curacao ', 'Guadeloupe',
       'Vanuatu', 'French Guiana', 'Luxembourg', 'Liechtenstein',
       'Anguilla', 'Martinique'], dtype=object)
```

[22]:
```
ref["continent"] = ref["Country of origin"].map({'Afghanistan': 'Asia',
 'Iraq': 'Asia',
 'Serbia and Kosovo: S/RES/1244 (1999)': "Europe",
 'Turkey': 'Asia',
 'Chad': 'Africa',
 'Cameroon': 'Africa',
 'Congo': 'Africa',
 'Dem. Rep. of the Congo': 'Africa',
 'Palestinian': 'Asia',
 'Guinea': 'Africa',
 'Liberia': 'Africa',
 'Libya': 'Africa',
 'Mali': 'Africa',
 'Morocco': 'Africa',
 'Nigeria': 'Africa',
 'Rwanda': 'Africa',
 'Sierra Leone': 'Africa',
 'Somalia': 'Africa',
 'Sudan': 'Africa',
 'Syrian Arab Rep.': "Asia",
 'Western Sahara': 'Africa',
 'Unknown ':"Unknown Continent",
 'Angola': 'Africa',
 'Burundi': 'Africa',
 'Comoros': 'Africa',
 'Guinea-Bissau': 'Africa',
 'United Rep. of Tanzania': 'Africa',
```

```
'Zambia': 'Africa',
'Djibouti': 'Africa',
'Eritrea': 'Africa',
'Ethiopia': 'Africa',
'Russian Federation': "Europe",
'Yemen': "Asia",
'Stateless': "Asia",
'Albania': "Europe",
'Algeria': 'Africa',
'Armenia': 'Asia',
'Benin': 'Africa',
'Bangladesh': 'Asia',
'Bosnia and Herzegovina': "Europe",
'Bulgaria': "Europe",
'Chile': "Americas",
'Colombia': "Americas",
'Cuba': "Americas",
'Dominican Rep.': "Americas",
'Ecuador': "Americas",
'Estonia': "Europe",
'Georgia':'Europe',
'Ghana': 'Africa',
'Haiti':'Asia',
'India':'Asia',
'Iran (Islamic Rep. of)': 'Asia',
'Kazakhstan':"Asia",
'Kyrgyzstan': "Asia",
"Lao People's Dem. Rep.": "Asia",
'Lebanon': "Asia",
'Sri Lanka': 'Asia',
'Nicaragua': "Americas",
'Pakistan': 'Asia',
'Paraguay': "Americas",
'Peru': "Americas",
'Romania': "Europe",
'Senegal': 'Africa',
'Viet Nam': "Asia",
'Tunisia': 'Africa',
'Ukraine': "Europe",
'Azerbaijan': 'Asia',
'Egypt': 'Africa',
'Argentina': "Americas",
'Austria': "Europe",
'Bahrain': 'Asia',
'Belarus': "Europe",
'Bolivia (Plurinational State of)': "Americas",
'Brazil': "Americas",
```

```
'Cambodia': 'Asia',
'China': 'Asia',
'Cyprus': 'Asia',
'Czechia':"Europe",
'Fiji': "Oceania",
'France': "Europe",
'United Kingdom of Great Britain and Northern Ireland': "Europe",
'Germany': "Europe",
'Guatemala': "Americas",
'China': 'Asia',
'Hong Kong SAR': 'Asia',
'Croatia': "Europe",
'Hungary': "Europe",
'Indonesia': 'Asia',
'Israel': 'Asia',
'Italy': "Europe",
'Jordan': 'Asia',
'Kenya': 'Africa',
'Rep. of Korea': 'Asia',
"Dem. People's Rep. of Korea": 'Asia',
'Kuwait': "Asia",
'Lithuania':"Europe",
'Latvia':"Europe",
'North Macedonia':"Europe",
'Rep. of Moldova':"Europe",
'Mexico': "Americas",
'Malaysia': 'Asia',
'Mongolia': 'Asia',
'Mauritius':"Africa",
'Myanmar': 'Asia',
'Nepal': 'Asia',
'Niger': 'Africa',
'Philippines': 'Asia',
'Poland': "Europe",
'Portugal': "Europe",
'South Africa': 'Africa',
'El Salvador': "Americas",
'Saudi Arabia': 'Asia',
'Singapore': 'Asia',
'Solomon Islands':"Oceania",
'Slovenia':"Europe",
'Thailand': "Asia",
'Timor-Leste': "Asia",
'Tonga':"Oceania",
'United Arab Emirates': 'Asia',
'Uganda': 'Africa',
'Uruguay': "Americas",
```

```
'Uzbekistan':"Asia",
'Zimbabwe': 'Africa',
"Cote d'Ivoire": 'Africa',
'Tajikistan':"Asia",
'Togo': 'Africa',
'Bhutan':"Asia",
'Burkina Faso': 'Africa',
'Mauritania': "Africa",
'Central African Rep.': 'Africa',
'Equatorial Guinea': 'Africa',
'Madagascar': 'Africa',
'Namibia': 'Africa',
'China, Macao SAR': "Asia",
'Honduras':"Americas",
'Antigua and Barbuda':"Americas",
'Barbados':"Americas",
'Belgium': "Europe",
'Botswana': 'Africa',
'Costa Rica':"Americas",
'Denmark': "Europe",
'Dominica':"Americas",
'Gabon': 'Africa',
'Gambia': 'Africa',
'Greece':"Europe",
'Grenada':"Americas",
'Guyana':"Americas",
'Iceland': "Europe",
'Ireland': "Europe",
'Jamaica':"Americas",
'Japan': 'Asia',
'Saint Lucia':"Americas",
'Malawi': "Africa",
'Mozambique': 'Africa',
'Malta':"Europe",
'Netherlands':"Europe",
'Oman': "Asia",
'Panama':"Americas",
'Qatar': 'Asia',
'Spain': "Europe",
'Slovakia':"Europe",
'Eswatini':"Africa",
'Sweden': "Europe",
'Switzerland': "Europe",
'Trinidad and Tobago': "Americas",
'United States of America': "Americas",
'Saint Vincent and the Grenadines': "Americas",
'Venezuela (Bolivarian Republic of)': "Americas",
```

```
 'Tibetan': "Asia",
 'Turkmenistan': "Asia",
 'Seychelles': 'Africa',
 'Sao Tome and Principe': 'Africa',
 'Papua New Guinea': "Oceania",
 'Suriname': "Americas",
 'Tuvalu':"Oceania",
 'Canada':"Americas",
 'Belize':"Americas",
 'Australia':"Oceania",
 'Bahamas':"Americas",
 'Cabo Verde': "Africa",
 'Finland': "Europe",
 'Nauru':"Oceania",
 'San Marino':"Europe",
 'Saint Kitts and Nevis':"Americas",
 'Samoa':"Oceania",
 'Lesotho': 'Africa',
 'Andorra':"Europe",
 'New Zealand': "Europe",
 'Norway': "Europe",
 'Micronesia (Federated States of)':"Oceania",
 'Gibraltar':"Europe",
 'Turks and Caicos Islands':"Americas",
 'Kiribati':"Oceania",
 'Maldives': "Asia",
 'Bermuda':"Americas",
 'Brunei Darussalam': "Asia",
 'New Caledonia':"Oceania",
 'Monaco':"Europe",
 'Montenegro':"Europe",
 'Holy See':"Europe",
 'South Sudan': "Africa",
 'Niue':"Oceania",
 'Palau':"Oceania",
 'Cayman Islands':"Americas",
 'Marshall Islands':"Oceania",
 'Curacao ':"Americas",
 'Guadeloupe':"Americas",
 'Vanuatu':"Oceania",
 'French Guiana':"Americas",
 'Luxembourg':"Europe",
 'Liechtenstein':"Europe",
 'Anguilla':"Americas",
 'Martinique':"Americas"})
ref["continent"].head()
```

[22]:
```
0      Asia
1      Asia
2    Europe
3      Asia
4    Africa
Name: continent, dtype: object
```

[23]:
```
ref.head()
```

[23]:
```
   Year                     Country of origin Country of origin (ISO)  \
0  2000                           Afghanistan                      AFG
1  2000                                  Iraq                      IRQ
2  2000  Serbia and Kosovo: S/RES/1244 (1999)                      SRB
3  2000                                Turkey                      TUR
4  2000                                  Chad                      TCD

  Country of asylum Country of asylum (ISO)  Refugees under UNHCR's mandate  \
0       Afghanistan                     AFG                               0
1           Albania                     ALB                               9
2           Albania                     ALB                             507
3           Albania                     ALB                               5
4           Algeria                     DZA                              20

   Asylum-seekers  IDPs of concern to UNHCR  Venezuelans displaced abroad  \
0               0                    758625                           NaN
1               0                         0                           NaN
2               5                         0                           NaN
3               0                         0                           NaN
4              19                         0                           NaN

   Stateless persons  Others of concern continent
0                  0                   0      Asia
1                  0                   0      Asia
2                  0                   0    Europe
3                  0                   0      Asia
4                  0                   0    Africa
```

5 Handling Missing Values

[24]:
```
ref.isnull().sum()
```

[24]:
```
Year                            0
Country of origin               0
Country of origin (ISO)      1015
Country of asylum               0
Country of asylum (ISO)         0
```

```
Refugees under UNHCR's mandate        0
Asylum-seekers                        0
IDPs of concern to UNHCR              0
Venezuelans displaced abroad      89945
Stateless persons                     0
Others of concern                     0
continent                            61
dtype: int64
```

[25]:
```
ref.notnull().sum()
```

[25]:
```
Year                              90004
Country of origin                 90004
Country of origin (ISO)           88989
Country of asylum                 90004
Country of asylum (ISO)           90004
Refugees under UNHCR's mandate    90004
Asylum-seekers                    90004
IDPs of concern to UNHCR          90004
Venezuelans displaced abroad         59
Stateless persons                 90004
Others of concern                 90004
continent                         89943
dtype: int64
```

[26]:
```
ref["Country of origin"].fillna(value = ref["Country of origin"].mode(),␣
 ↪inplace = True)
ref.isnull().sum()
```

[26]:
```
Year                                  0
Country of origin                     0
Country of origin (ISO)            1015
Country of asylum                     0
Country of asylum (ISO)               0
Refugees under UNHCR's mandate        0
Asylum-seekers                        0
IDPs of concern to UNHCR              0
Venezuelans displaced abroad      89945
Stateless persons                     0
Others of concern                     0
continent                            61
dtype: int64
```

[27]:
```
ref["continent"].value_counts()
```

[27]:
```
Africa              38918
Asia                30089
```

```
Europe                11467
Americas               8063
Unknown Continent      1015
Oceania                 391
Name: continent, dtype: int64
```

[28]:
```
ref["continent"].value_counts(normalize = True)*100
```

[28]:
```
Africa              43.27
Asia                33.45
Europe              12.75
Americas             8.96
Unknown Continent    1.13
Oceania              0.43
Name: continent, dtype: float64
```

5.1 Creating Continent and sub-region

[29]:
```
# continent1 = (cont.continent == "Africa")
# Africa_Country = cont.country[continent1]
# for countries1 in Africa_Country:
#     if countries1 in [cout for cout in ref["Country of origin"]]:
#         ref["continent"] = "Africa"
#     else:
#         pass

# continent2 = (cont.continent == "Asia")
# Asia_country = cont.country[continent2]
# for countries2 in Asia_country:
#     if countries2 in [cout for cout in ref["Country of origin"]]:
#         ref["continent"].append(continent2 = "Asia")
#     else:
#         pass

# continent3 = (cont.continent == "Europe")
# Europe_country = cont.country[continent3]
# for countries3 in Europe_country:
#     if countries3 in [cout for cout in ref["Country of origin"]]:
#         ref["continent"].append(continent3 = "Europe")
#     else:
#         pass

# continent4 = (cont.continent == "Americas")
# American_country = cont.country[continent4]
# for countries4 in American_country:
#     if countries4 in [cout for cout in ref["Country of origin"]]:
#         ref["continent"].append(continent4 = "Americas")
```

```
#     else:
#         pass
# continent5 = (cont.continent == "Oceania")
# Oceania_country = cont.country[continent5]
# for countries5 in Oceania_country:
#     if countries5 in [cout for cout in ref["Country of origin"]]:
#         ref["continent"].append(continent5 = "Oceania")
#     else:
#         pass
# ref.head()
```

[30]:
```
ref.groupby("continent")["Country of origin"].value_counts().head()
```

[30]:
```
continent  Country of origin
Africa     Somalia                   1990
           Dem. Rep. of the Congo    1892
           Sudan                     1813
           Ethiopia                  1587
           Nigeria                   1455
Name: Country of origin, dtype: int64
```

[31]:
```
ref.continent.unique()
```

[31]:
```
array(['Asia', 'Europe', 'Africa', 'Unknown Continent', 'Americas',
       'Oceania', nan], dtype=object)
```

6 Exploring column by column

6.1 Year

- The years range from 2000 - 2020
- The year 2020 has highest frequency
- The year 2000 has the least frequency

[32]:
```
ref.Year.describe(include = 'all')
```

[32]:
```
count   90004.00
mean     2011.07
std         6.02
min      2000.00
25%      2006.00
50%      2012.00
75%      2016.00
max      2020.00
Name: Year, dtype: float64
```

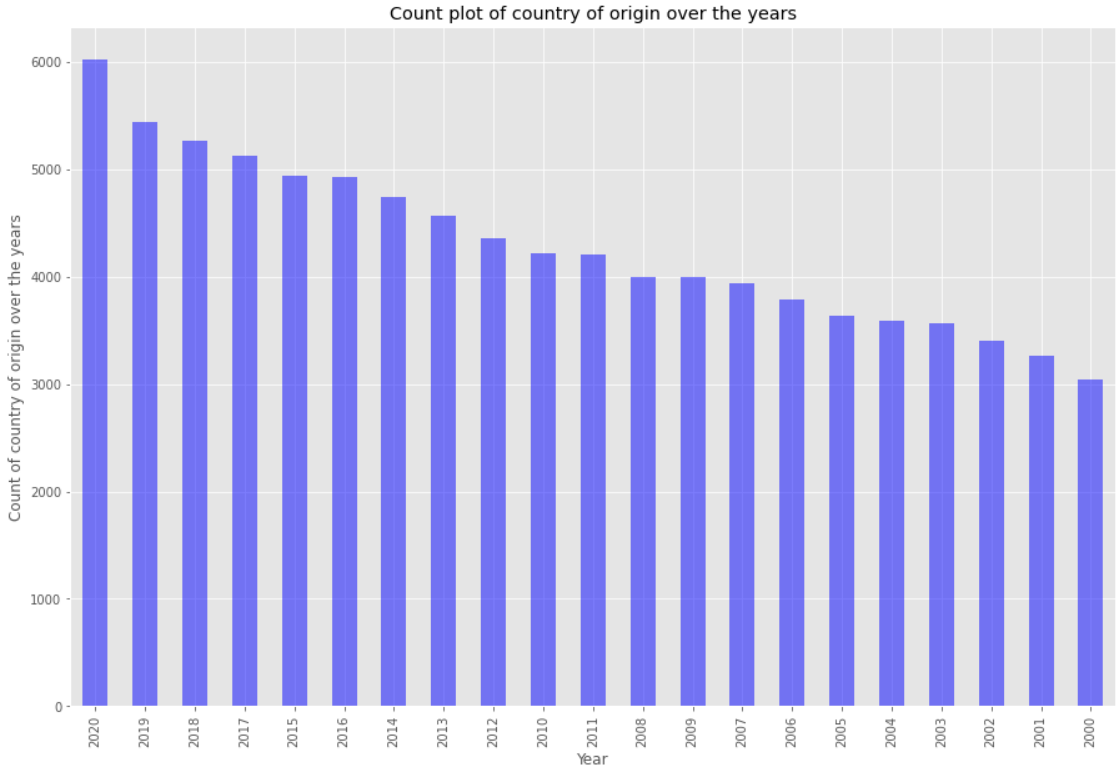
[33]:
```
ref.Year.unique()
```

[33]:
```
array([2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010,
       2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020])
```

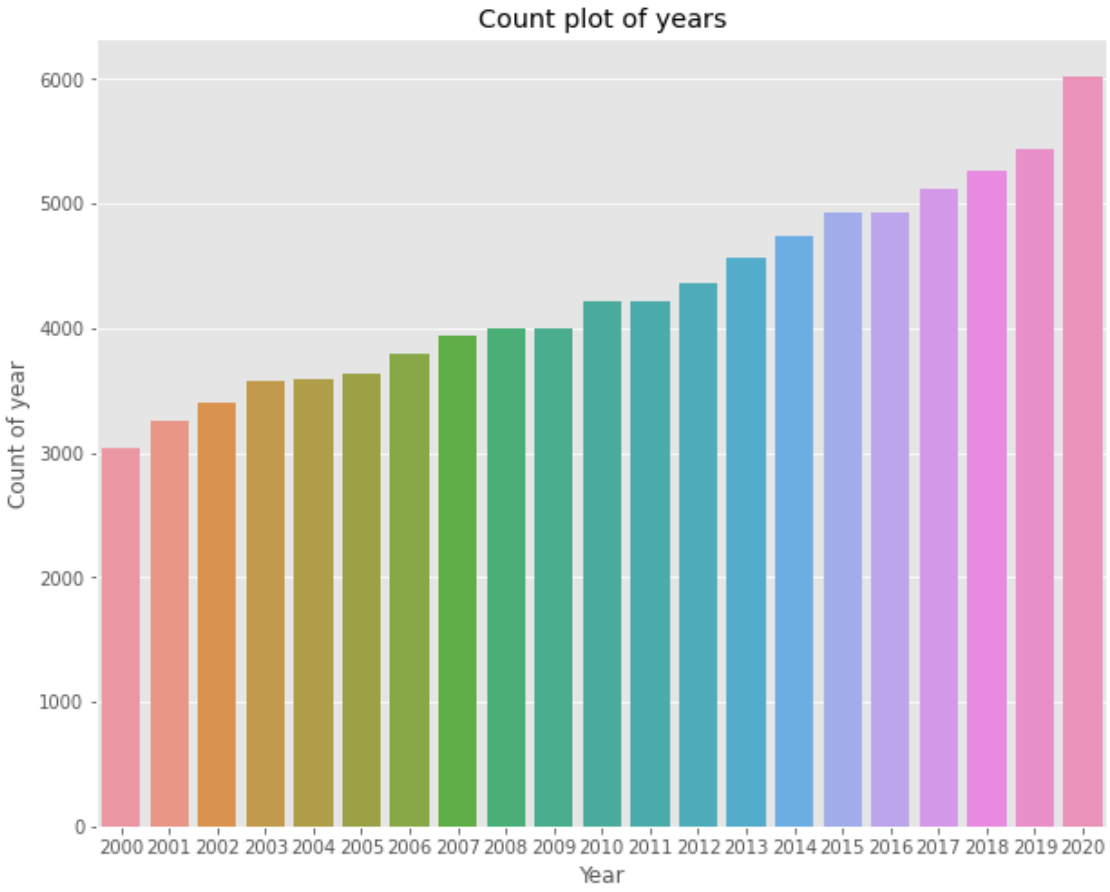
[34]:
```
ref.Year.value_counts()
```

[34]:
```
2020    6024
2019    5438
2018    5263
2017    5120
2015    4933
2016    4923
2014    4744
2013    4561
2012    4356
2010    4212
2011    4208
2008    4001
2009    4000
2007    3937
2006    3789
2005    3640
2004    3586
2003    3570
2002    3403
2001    3259
2000    3037
Name: Year, dtype: int64
```

[35]:
```
ref.groupby("Year")["Country of origin"].count().sort_values(ascending =False).
↪plot.bar(figsize = (15,10), alpha = 0.5,
                                        color = 'blue', ylabel = "Count of␣
↪country of origin over the years",
                                        title = "Count plot of country of␣
↪origin over the years");
```



[36]:
```
plt.figure(figsize = (10,8))
sns.countplot(x = "Year", data = ref)
plt.ylabel("Count of year")
plt.title("Count plot of years")
plt.show()
```



6.2 Country of origin

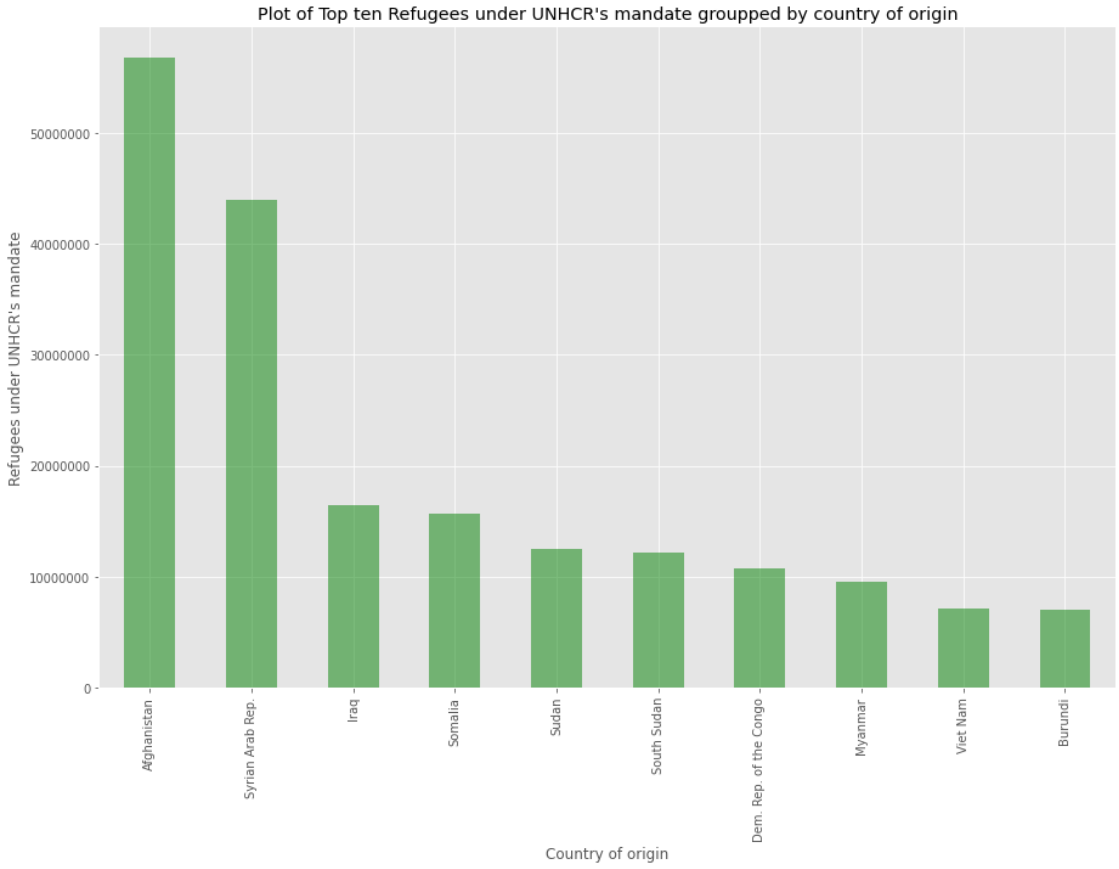
*The country of origin feature represents the country where those seeking refugee are coming from.

- Somalia is country with the highest number in terms of refugees country's of origin.
- There are 212 countries where refugees originated.
- The data didn't give any information why people from these countries are seeking refugee and asyluum in other countries, but we know that most of these countries are in war. Countries like Somalia, Afghanistan, Syria etc.
- Refugees from Afghanistan as country of origin has the highest number of Refugees under UNHCR's mandate

[37]:
```
ref["Country of origin"].describe(include = 'all')
```

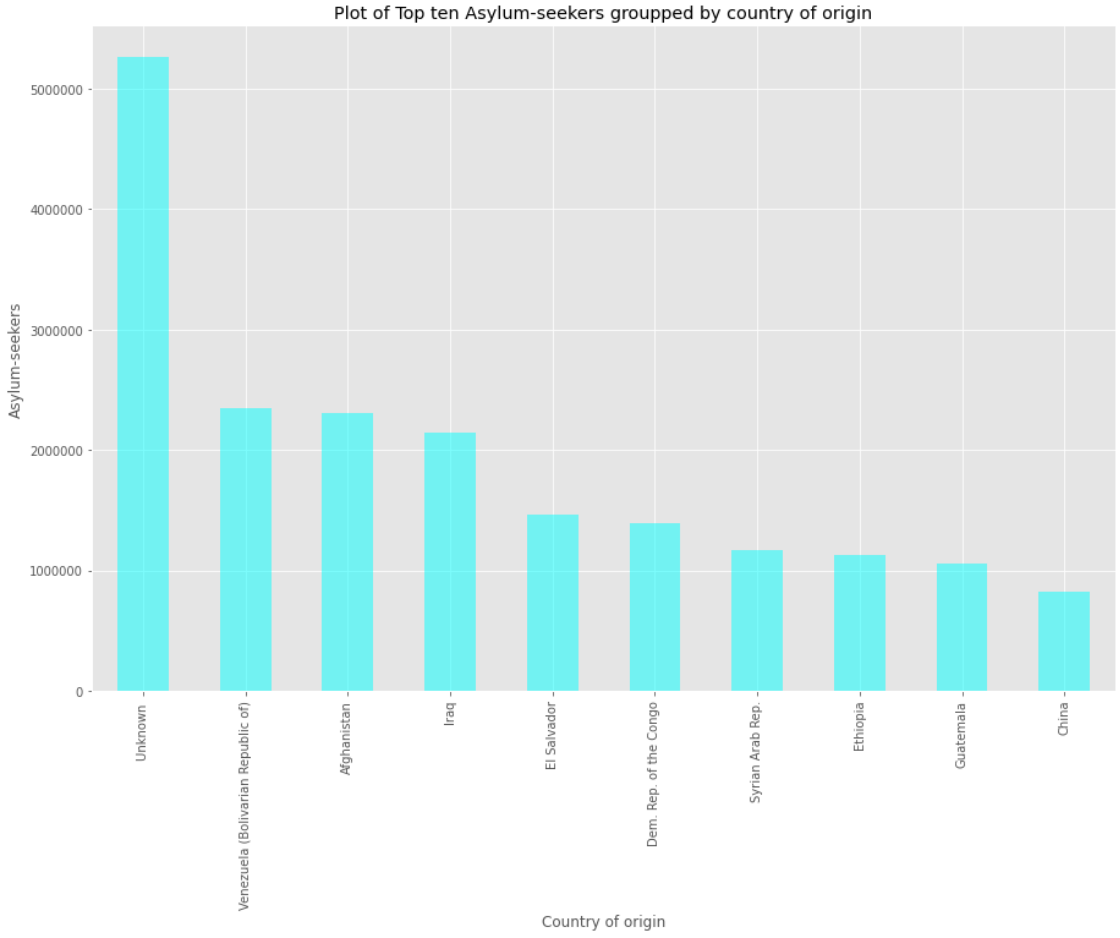
[37]:
```
count       90004
unique        212
top       Somalia
freq         1990
Name: Country of origin, dtype: object
```

[38]:
```
"""
Afghanistan has the number of Refugees under UNHCR's mandate
"""
plt.ticklabel_format(style='plain',useOffset=False)
ref.groupby("Country of origin")["Refugees under UNHCR's mandate"].sum().
↪sort_values(ascending =False).head(10).plot.bar(figsize = (15,10), alpha = 0.
↪5,
                              color = 'green', ylabel = "Refugees under UNHCR's␣
↪mandate",
                                                          title="Plot of Top ten␣
↪Refugees under UNHCR's mandate groupped by country of origin");
```

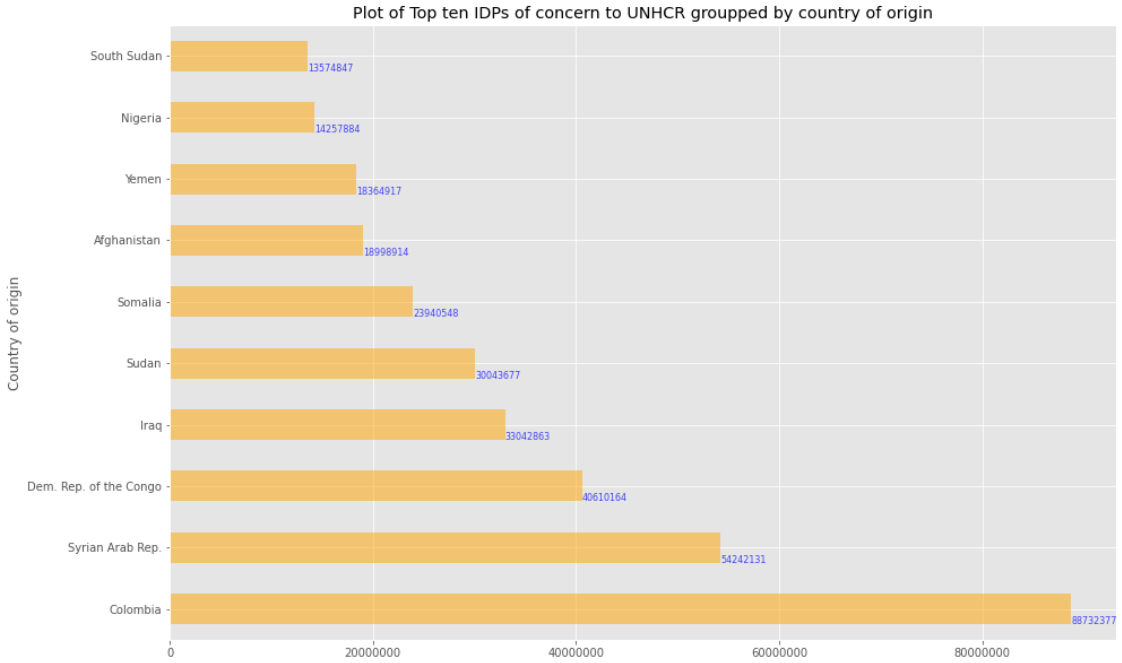


[39]:
```
"""
Asyluum seekers from unknown countries are highest followed by Venezuela
"""
plt.ticklabel_format(style='plain',useOffset=False)
ref.groupby("Country of origin")["Asylum-seekers"].sum().sort_values(ascending␣
↪= False).head(10).plot.bar(figsize = (15,10), alpha = 0.5,
                              color = 'cyan', ylabel = "Asylum-seekers",
```

```
                                                    title="Plot of Top ten Asylum-seekers␣
↪groupped by country of origin");
```



[41]:
```
"""
Colombia has the highest number of IDPs of concern to UNHCR based on country of␣
↪origin
"""
plt.ticklabel_format(style='plain',useOffset=False)
ax2 = ref.groupby("Country of origin")["IDPs of concern to UNHCR"].sum().
↪sort_values(ascending = False).head(10).plot.barh(figsize = (15,10), alpha =␣
↪0.5,
                                color = 'orange', ylabel = "IDPs of concern to␣
↪UNHCR",
                                                    title="Plot of Top ten IDPs of␣
↪concern to UNHCR groupped by country of origin");
for i in ax2.patches:
    ax2.text(i.get_width()+0.005, i.get_y(), str(int(round(i.get_width(),2))),␣
↪fontsize=8, color='b', alpha=0.7);
```



[42]:
```
ref.columns
```

[42]:
```
Index(['Year', 'Country of origin', 'Country of origin (ISO)',
       'Country of asylum', 'Country of asylum (ISO)',
       'Refugees under UNHCR's mandate', 'Asylum-seekers',
       'IDPs of concern to UNHCR', 'Venezuelans displaced abroad',
       'Stateless persons', 'Others of concern', 'continent'],
      dtype='object')
```

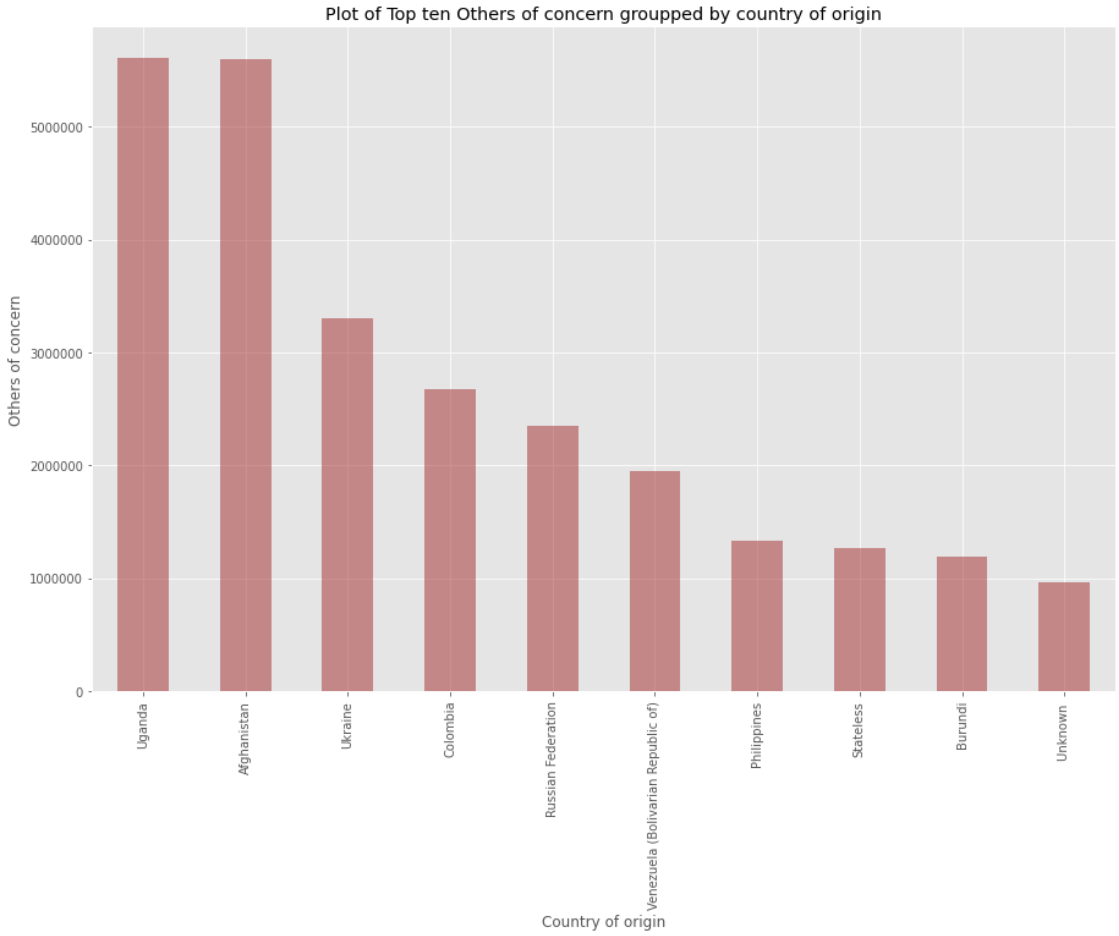
[43]:
```
ref.groupby("Country of origin")["Venezuelans displaced abroad"].sum().
↪sort_values(ascending = False).head(10)
```

[43]:
```
Country of origin
Venezuela (Bolivarian Republic of)   10031476.00
Afghanistan                                 0.00
Panama                                      0.00
New Zealand                                 0.00
Nicaragua                                   0.00
Niger                                       0.00
Nigeria                                     0.00
Niue                                        0.00
North Macedonia                             0.00
Norway                                      0.00
Name: Venezuelans displaced abroad, dtype: float64
```

[44]:
```
ref.groupby("Country of origin")["Stateless persons"].sum().
 ↪sort_values(ascending = False).head(10)
```

[44]:
```
Country of origin
Stateless          65929313
Afghanistan               0
New Caledonia             0
Nicaragua                 0
Niger                     0
Nigeria                   0
Niue                      0
North Macedonia           0
Norway                    0
Oman                      0
Name: Stateless persons, dtype: int64
```

[45]:
```
plt.ticklabel_format(style='plain',useOffset=False)
ref.groupby("Country of origin")["Others of concern"].sum().
 ↪sort_values(ascending = False).head(10).plot.bar(figsize = (15,10), alpha =␣
 ↪0.5,
                            color = 'brown', ylabel = "Others of concern",
                            title="Plot of Top ten Others of concern groupped␣
 ↪by country of origin");
```



6.3 Country of asylum & Asylum-seekers

The Country of asylum represent the country where people from various countries of origin are seeking for asyluum outside their home country. There are 189 country of asyluum seekers * United States of America has the highest number of asyluum seekers with 3572 frequency.

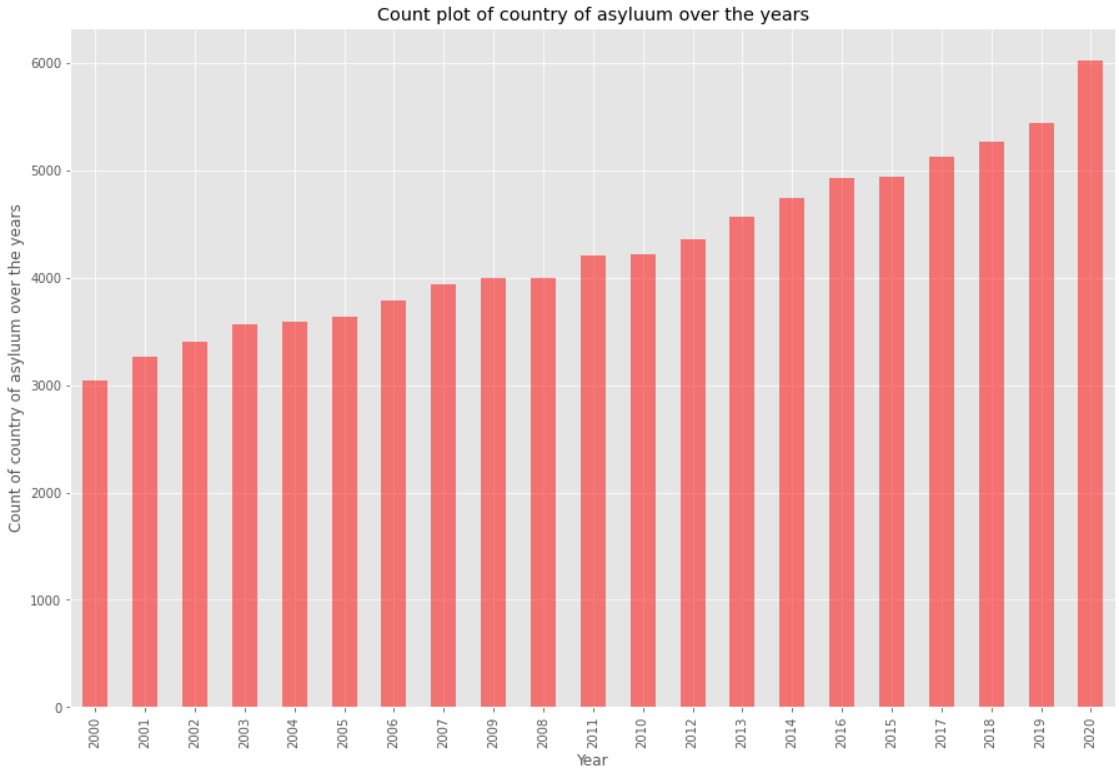
[46]:
```
ref[["Country of asylum", "Asylum-seekers"]].describe(include = 'all')
```

[46]:

	Country of asylum	Asylum-seekers
count	90004	90004.00
unique	189	NaN
top	United States of America	NaN
freq	3572	NaN
mean	NaN	393.77
std	NaN	5586.88
min	NaN	0.00
25%	NaN	0.00
50%	NaN	6.00

```
75%                             NaN          41.00
max                             NaN      940668.00
```

[47]:
```
"""
Year 2020 has the highest number of people seeking asyluum in various countries␣
 ↪represented
"""
ref.groupby("Year")["Country of asylum"].count().sort_values(ascending =True).
 ↪plot.bar(figsize = (15,10), alpha = 0.5,
                                          color = 'red', ylabel = "Count of␣
 ↪country of asyluum over the years",
                                              title="Count plot of country of␣
 ↪asyluum over the years");
```

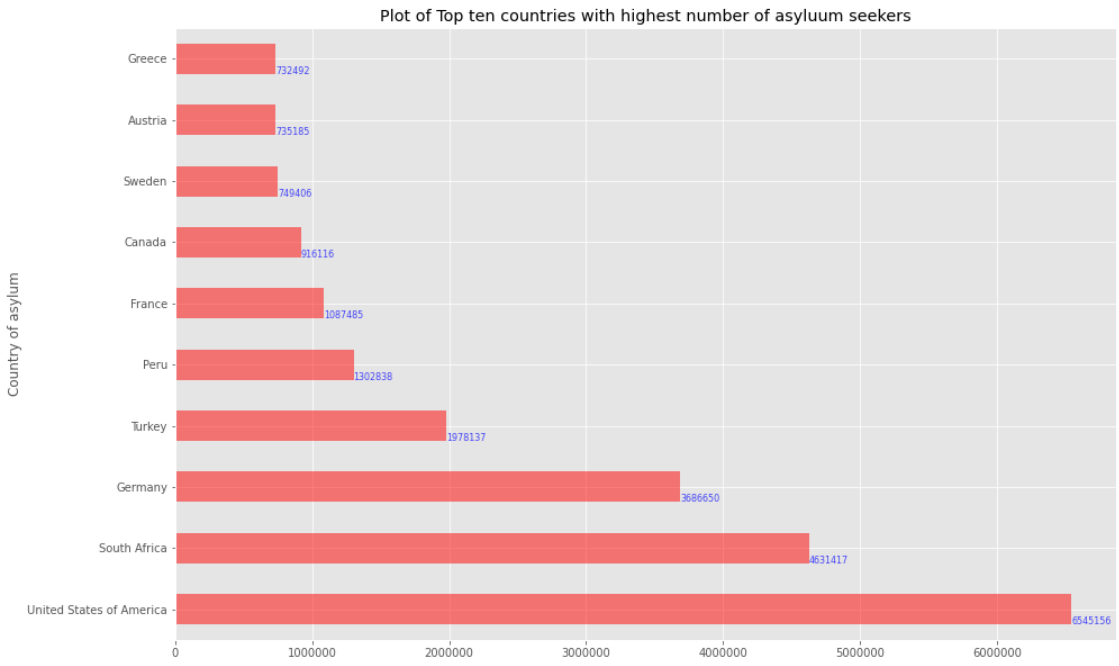


[48]:
```
ref.groupby("Country of asylum")["Asylum-seekers"].sum().sort_values(ascending␣
 ↪= False).head(10)
```

[48]:
```
Country of asylum
United States of America    6545156
South Africa                4631417
Germany                     3686650
Turkey                      1978137
```

```
Peru                          1302838
France                        1087485
Canada                         916116
Sweden                         749406
Austria                        735185
Greece                         732492
Name: Asylum-seekers, dtype: int64
```

[49]:
```
plt.ticklabel_format(style='plain',useOffset=False)
ax1 = ref.groupby("Country of asylum")["Asylum-seekers"].sum().
 ↪sort_values(ascending = False).head(10).plot.barh(figsize = (15,10), alpha =␣
 ↪0.5,
                        color = 'red', ylabel = "Asyluum seekers",
                                          title="Plot of Top ten countries with␣
 ↪highest number of asyluum seekers");
for i in ax1.patches:
    ax1.text(i.get_width()+0.005, i.get_y(), str(int(round(i.get_width(),2))),␣
 ↪fontsize=8, color='b', alpha=0.7);
```



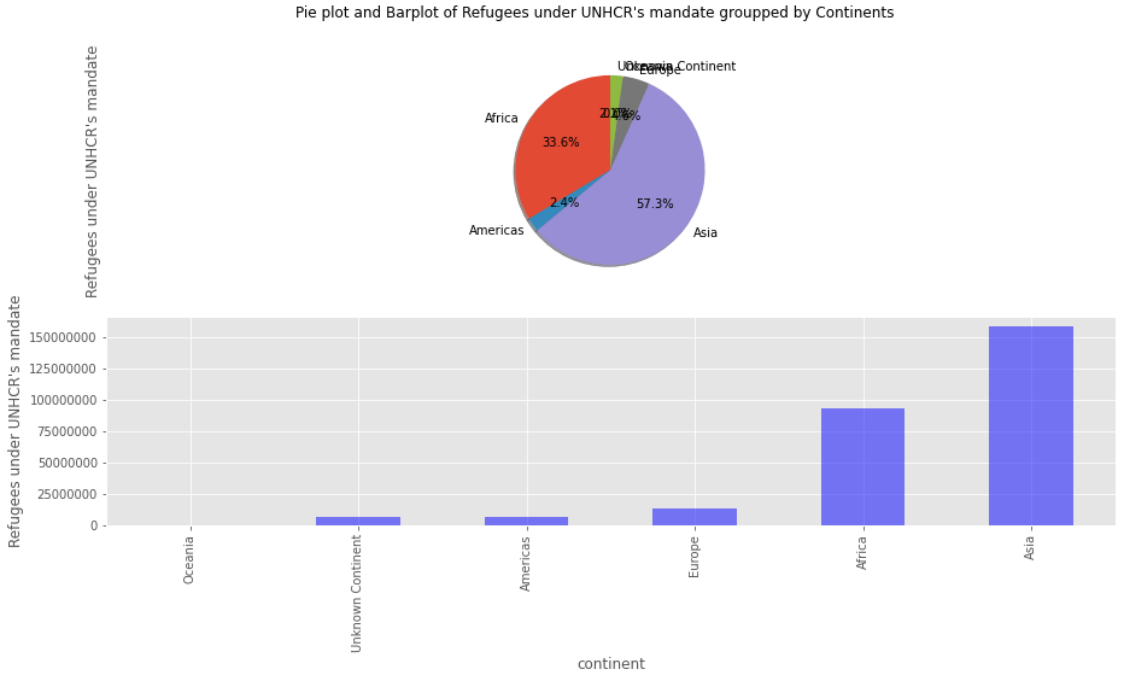
6.4 Refugees under UNHCR's mandate

[50]:
```
ref["Refugees under UNHCR's mandate"].describe(include = 'all')
```

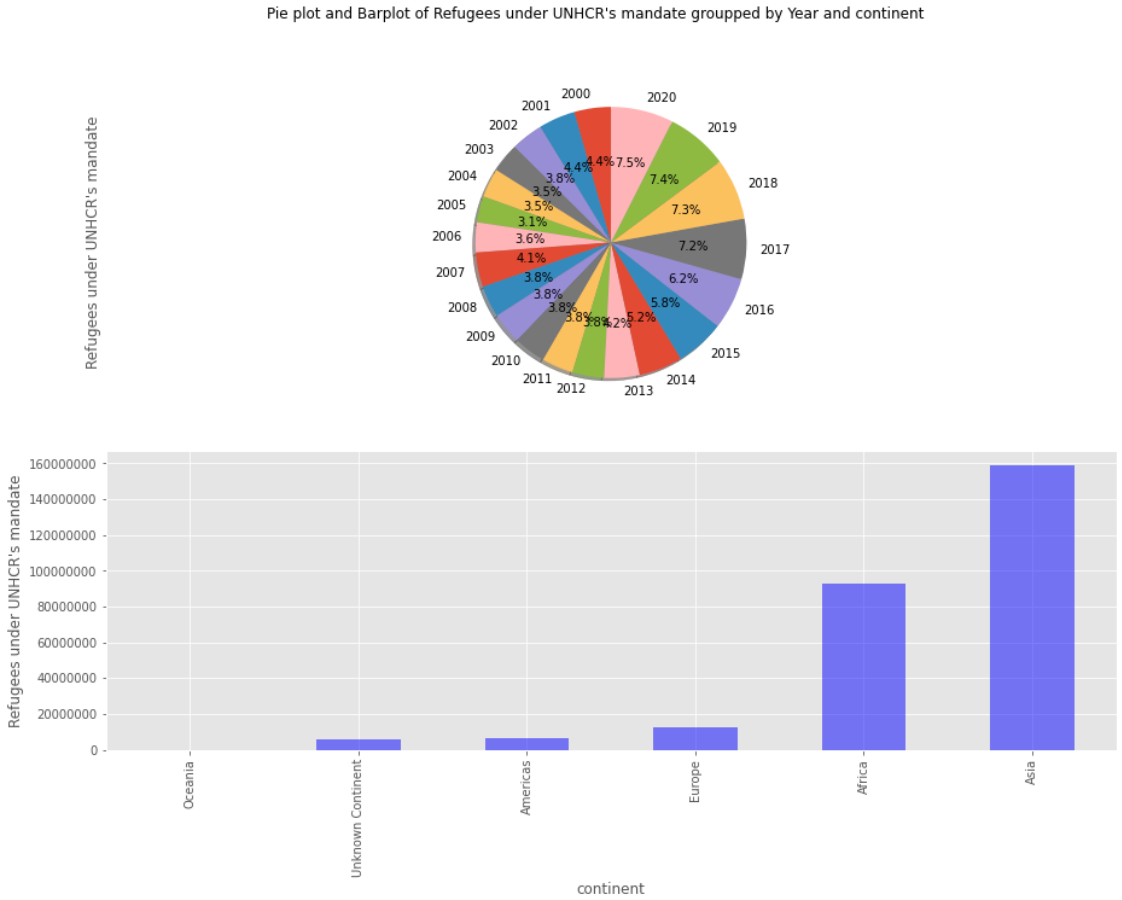
[50]:
```
count     90004.00
mean       3077.74
```

```
std       46313.22
min           0.00
25%           5.00
50%          14.00
75%         103.00
max     3641370.00
Name: Refugees under UNHCR's mandate, dtype: float64
```

[51]:
```
plt.suptitle("Pie plot and Barplot of Refugees under UNHCR's mandate groupped␣
 ↪by Continents")
plt.subplot(2,1,1)
ax = ref.groupby('continent')["Refugees under UNHCR's mandate"].sum().plot.
 ↪pie(autopct = "%1.1f%%",
                                                    shadow = True, startangle =␣
 ↪90, figsize = (15,7))
ax.axis("equal")
plt.subplot(2,1,2)
plt.ticklabel_format(style = 'plain', useOffset=False)
ax1 = ref.groupby('continent')["Refugees under UNHCR's mandate"].sum().
 ↪sort_values(ascending = True).plot.bar(figsize = (15,7),
                                  alpha = 0.5,color = 'blue', ylabel = "Refugees␣
 ↪under UNHCR's mandate");
plt.show()
```



[52]:
```
plt.suptitle("Pie plot and Barplot of Refugees under UNHCR's mandate groupped␣
 ↪by Year and continent")
plt.subplot(2,1,1)
ax = ref.groupby('Year')["Refugees under UNHCR's mandate"].sum().plot.
 ↪pie(autopct = "%1.1f%%", shadow = True, startangle = 90, figsize = (15,10))
ax.axis("equal")
plt.subplot(2,1,2)
plt.ticklabel_format(style = 'plain', useOffset=False)
ax1 = ref.groupby('continent')["Refugees under UNHCR's mandate"].sum().
 ↪sort_values(ascending = True).plot.bar(figsize = (15,10), alpha = 0.5,
                                         color = 'blue', ylabel = "Refugees␣
 ↪under UNHCR's mandate");
plt.show()
```



[53]:
```
ref.columns
```

[53]:
```
Index(['Year', 'Country of origin', 'Country of origin (ISO)',
       'Country of asylum', 'Country of asylum (ISO)',
```

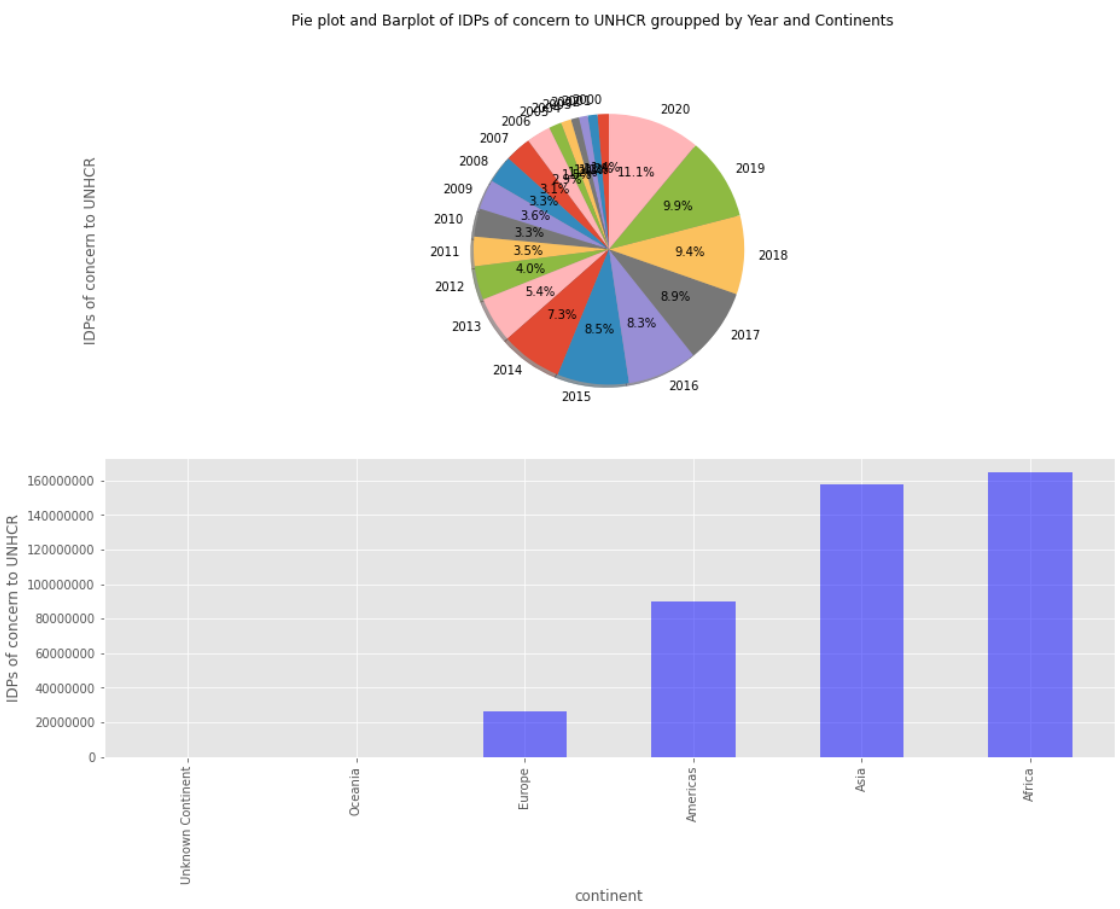
```
       'Refugees under UNHCR's mandate', 'Asylum-seekers',
       'IDPs of concern to UNHCR', 'Venezuelans displaced abroad',
       'Stateless persons', 'Others of concern', 'continent'],
      dtype='object')
```

6.5 IDPs of concern to UNHCR

[54]:
```
ref["IDPs of concern to UNHCR"].describe(include = 'all')
```

[54]:
```
count     90004.00
mean       4881.33
std      124509.91
min           0.00
25%           0.00
50%           0.00
75%           0.00
max     8252788.00
Name: IDPs of concern to UNHCR, dtype: float64
```

[55]:
```
plt.suptitle("Pie plot and Barplot of IDPs of concern to UNHCR groupped by Year␣
 ↪and Continents")
plt.subplot(2,1,1)
ax = ref.groupby('Year')["IDPs of concern to UNHCR"].sum().plot.pie(autopct =␣
 ↪"%1.1f%%",
                                                    shadow = True, startangle =␣
 ↪90, figsize = (15,10))
ax.axis("equal")
plt.subplot(2,1,2)
plt.ticklabel_format(style = 'plain', useOffset=False)
ax1 = ref.groupby('continent')["IDPs of concern to UNHCR"].sum().
 ↪sort_values(ascending = True).plot.bar(figsize = (15,10),
                                  alpha = 0.5,color = 'blue', ylabel = "IDPs of␣
 ↪concern to UNHCR");
plt.show()
```



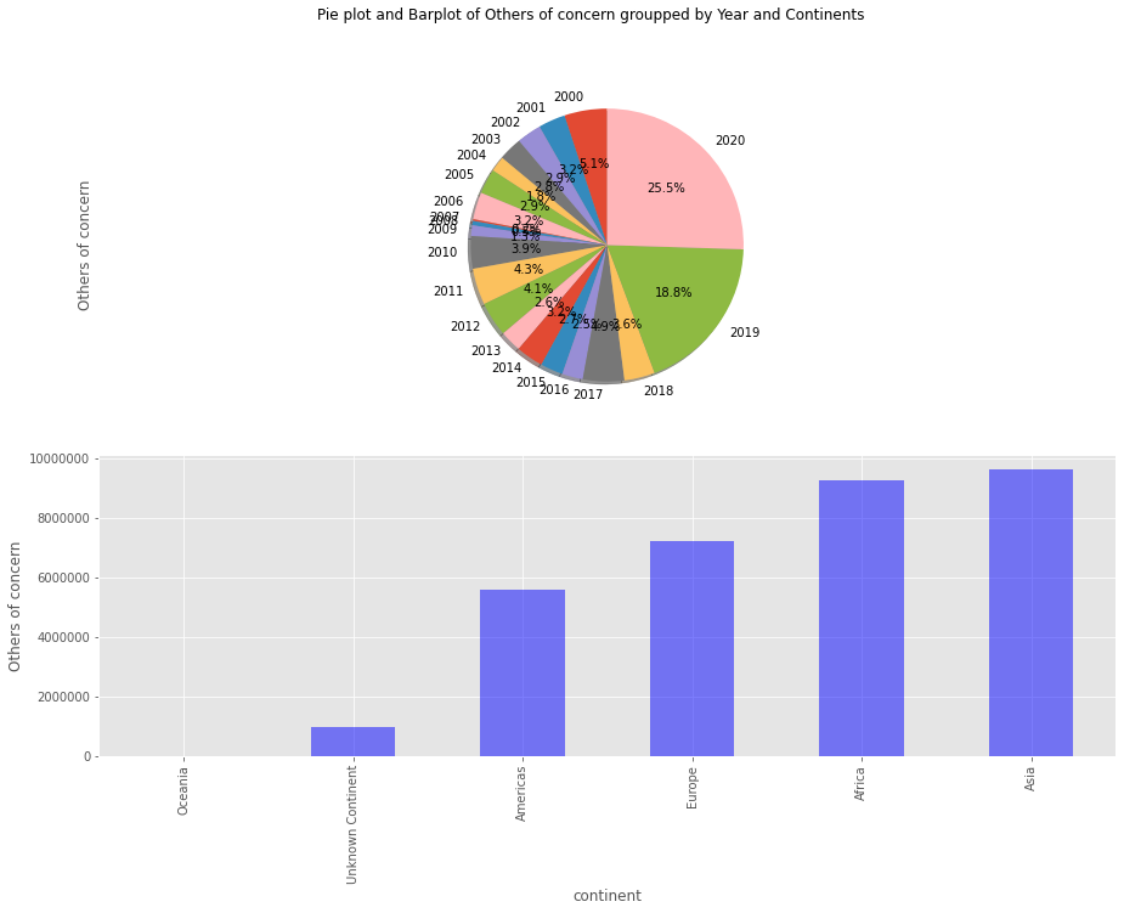
6.6 Others of concern

[56]:
```
ref["Others of concern"].describe(include = 'all')
```

[56]:
```
count     90004.00
mean        362.10
std       16814.69
min           0.00
25%           0.00
50%           0.00
75%           0.00
max     2351313.00
Name: Others of concern, dtype: float64
```

[57]:
```
plt.suptitle("Pie plot and Barplot of Others of concern groupped by Year and␣
 ↪Continents")
plt.subplot(2,1,1)
ax = ref.groupby('Year')["Others of concern"].sum().plot.pie(autopct = "%1.
 ↪1f%%",
```

```
                                                        shadow = True, startangle =␣
 ↪90, figsize = (15,10))
ax.axis("equal")
plt.subplot(2,1,2)
plt.ticklabel_format(style = 'plain', useOffset=False)
ax1 = ref.groupby('continent')["Others of concern"].sum().sort_values(ascending␣
 ↪= True).plot.bar(figsize = (15,10),
                                  alpha = 0.5,color = 'blue', ylabel = "Others of␣
 ↪concern");
plt.show()
```



7 Conclusion

The EDA shows a true picture of what we see in our world. The Asia world battling a lot of wars comes out top in term of refugees and asyluum seekers. Though, there was no exact data information on relation of global food prices to refuee movement, but war, civil unrest comes with hunger. This makes people leave their country of origin to another country as a refugee or asylum seeker. Therefore, I can categorically say there is a positive correlation with refugee movement to

food prices. If there conflicts or war, there will be lack of food and even if there is food, it will scarce and costly. This will encourage migration of people to avoid dying of starvation.