Exploring the relationship between dietary lifestyle and COVID-19 susceptibility

Itu Mukherjee

Abstract

This study is aimed at analyzing if a particular dietary lifestyle have any impact on COVID-19 occurrence. The dataset being used is downloaded from Kaggle and comprises of country wise consumption of different types of food along with number of confirmed cases and deaths of COVID-19. Correlation analysis revealed higher consumption of animal products to be directly associated with COVID-19 related mortality. The results also indicate a possible link between obesity and COVID-19 induced death.

Motivation

COVID-19 has turned into a global pandemic. As biomedical researchers worldwide are working around the clock to come up with a vaccine, there is also an interest in non pharmaceutical intervention strategies.

One of the major factors that can potentially play a role in determining susceptibility towards any decease is diet. With this in mind I wanted to explore links between global dietary lifestyles and COVID-19 susceptibility.

Dataset

I have used the following dataset from Kaggle.

https://www.kaggle.com/mariaren/covid19-healthy-diet-dataset

The dataset contains different categories of food supply quantity in kgs for 170 countries, calculated as the percentage of total population of a country. The dataset also has country-wise numbers of confirmed cases of COVID-19, deaths associated with COVID-19 and individuals recovered from COVID-19.

Data Preparation and Cleaning

The dataset has country-wise supply data for 23 different categories of food. In addition, the dataset also has country-wise statistics of obesity and undernourishment, confirmed cases of COVID-19, deaths associated with COVID-19 and individuals recovered from COVID-19, all showed as percentage of total population.

Quite a few columns have lot of NaN values. In particular, the undernourished column has more than one third missing values. I used pandas dropna() method to drop rows containing missing values. Since the undernourished column had a lot of missing values I decided to drop this column all together from the dataframe using pandas df.drop() method.

The final resulting dataframe had 152 countries as rows and 30 columns, out of which country label was an object while all the other parameters were floats.

Research Questions

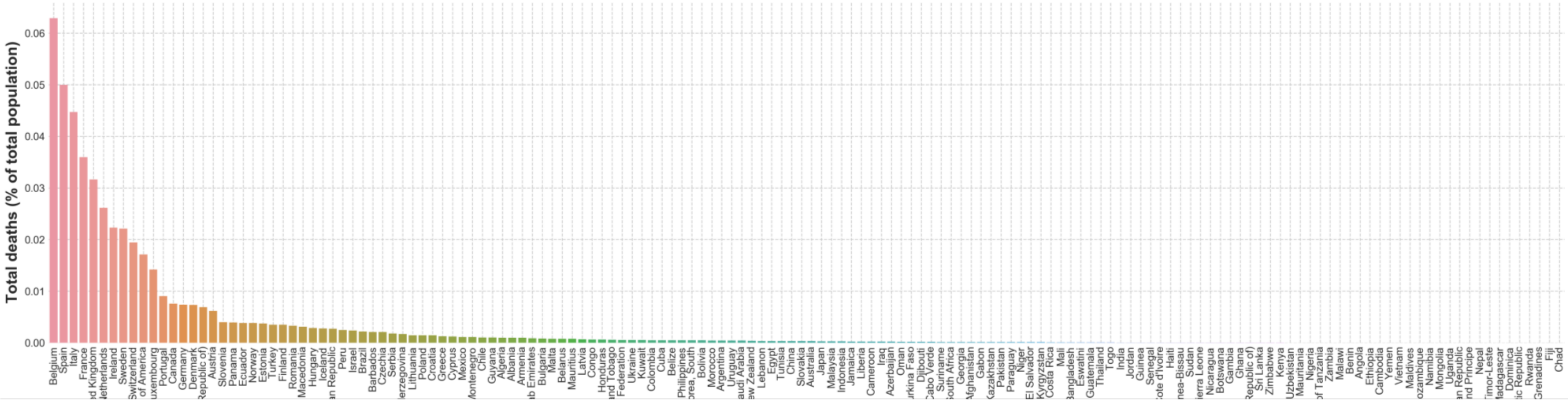
1. What are the country-wise distributions of dietary lifestyles?
2. Does dietary choice have any impact on susceptibility to COVID-19?

Methods

1) For plotting country-wise distributions of diet, I made use of a choropleth plot in which I overlaid country-wise dietary values on a map of the world.

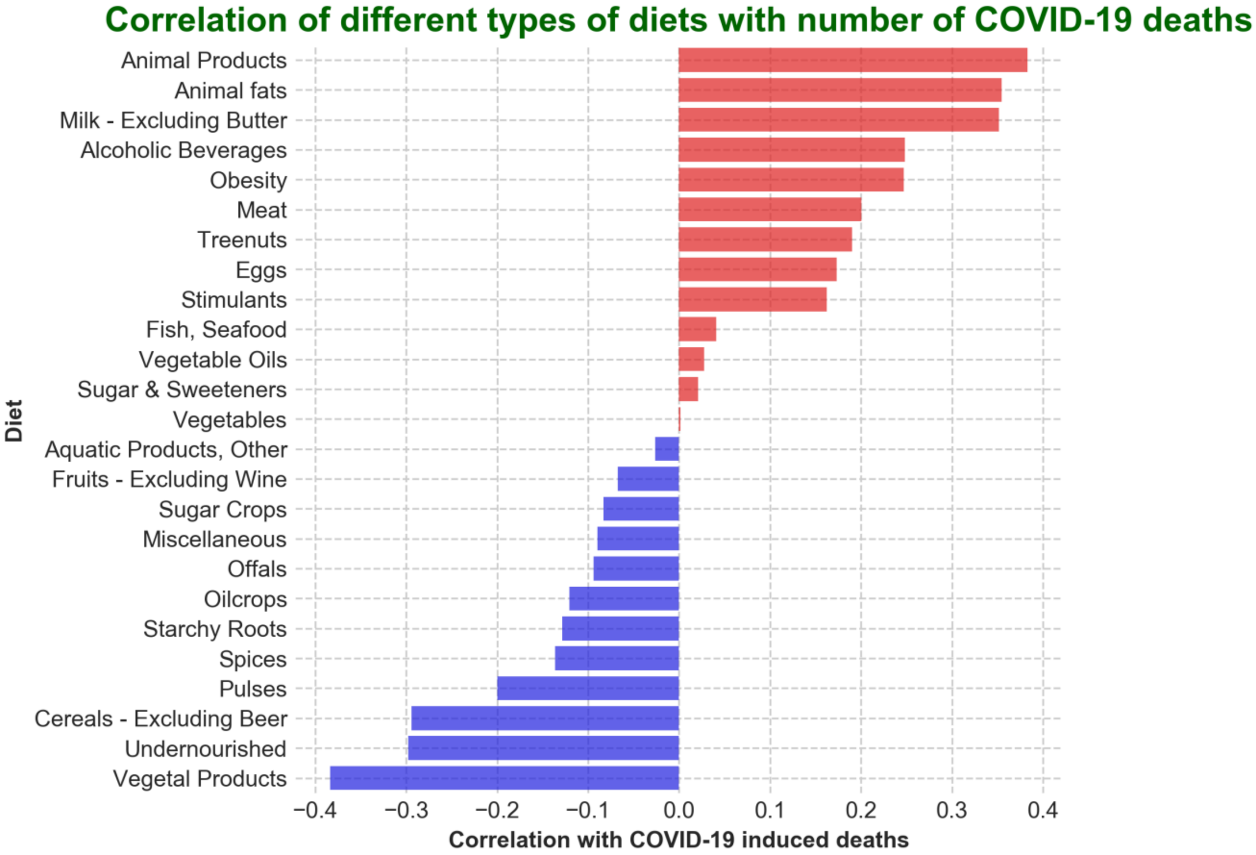
2) To explore links between different dietary choices and susceptibility to COVID-19, I calculated Pearson’s correlation coefficients for the different dietary lifestyles with COVID-19 associated deaths across 152 countries.

Findings



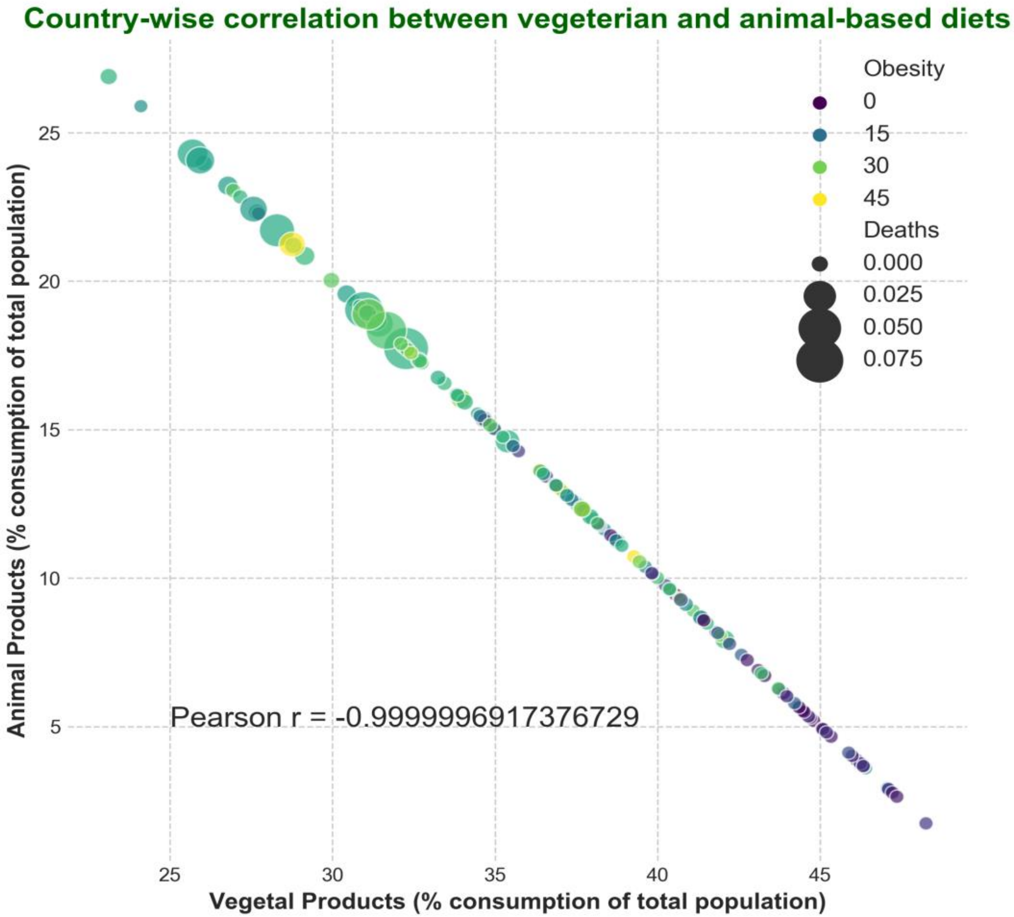
Percentage of total death rates per country

Findings



- The figure shows the correlation of different kinds of diet with COVID-19 associated deaths across 152 counties.
- High positive correlations (shown in red) indicate higher susceptibility to COVID-19 induced death associated with that attribute.
- On the other hand higher negative correlations (shown in blue) suggest a lower risk of COVID-19 induced death.

Findings

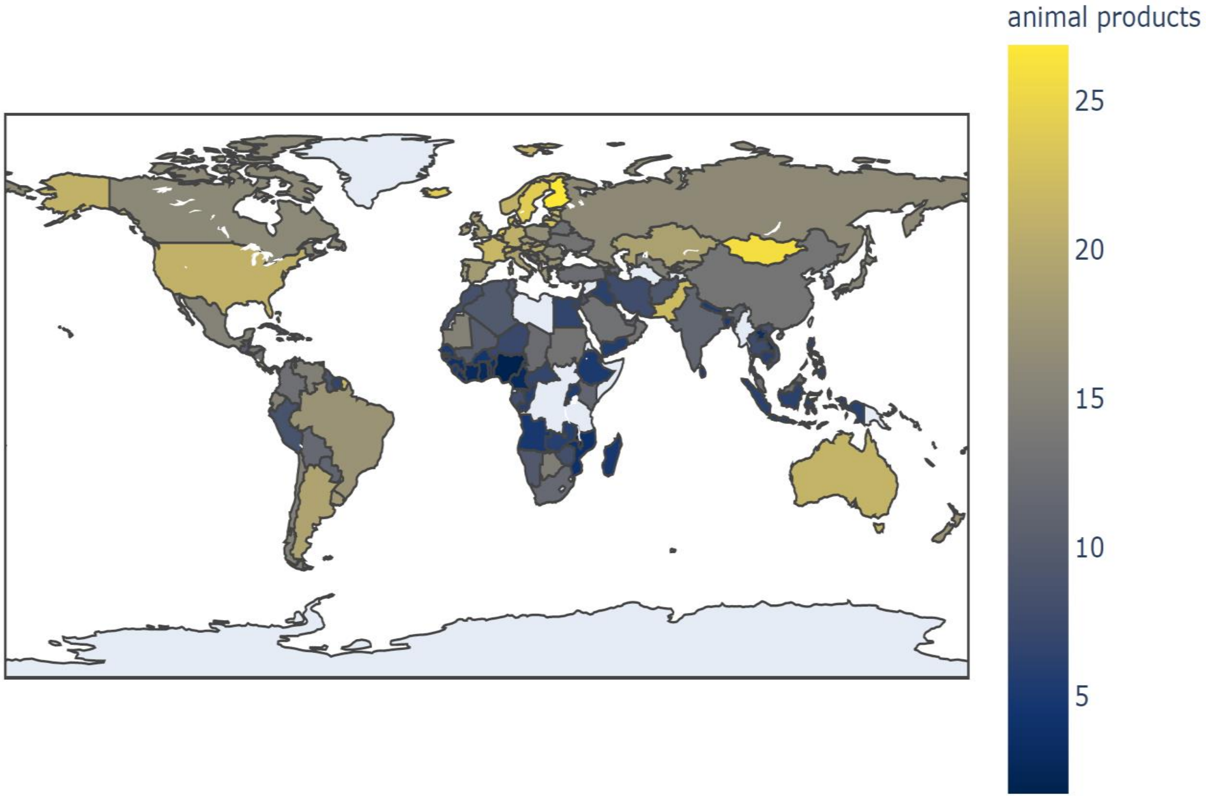


- The figure shows the country-wise correlation between vegetarian and animal-based diets of COVID-19 deaths across 152 counties
- We see a strong negative correlation between vegetal products and animal products across countries of the world.
- The plot also shows that countries with higher percentage of the population on animal products also have higher obesity.
- Therefore, these results indicate a direct association between obesity and COVID-19 induced death.

Findings

- The adjoining figure shows the distribution of animal product consumption in kgs as a percentage of the total population

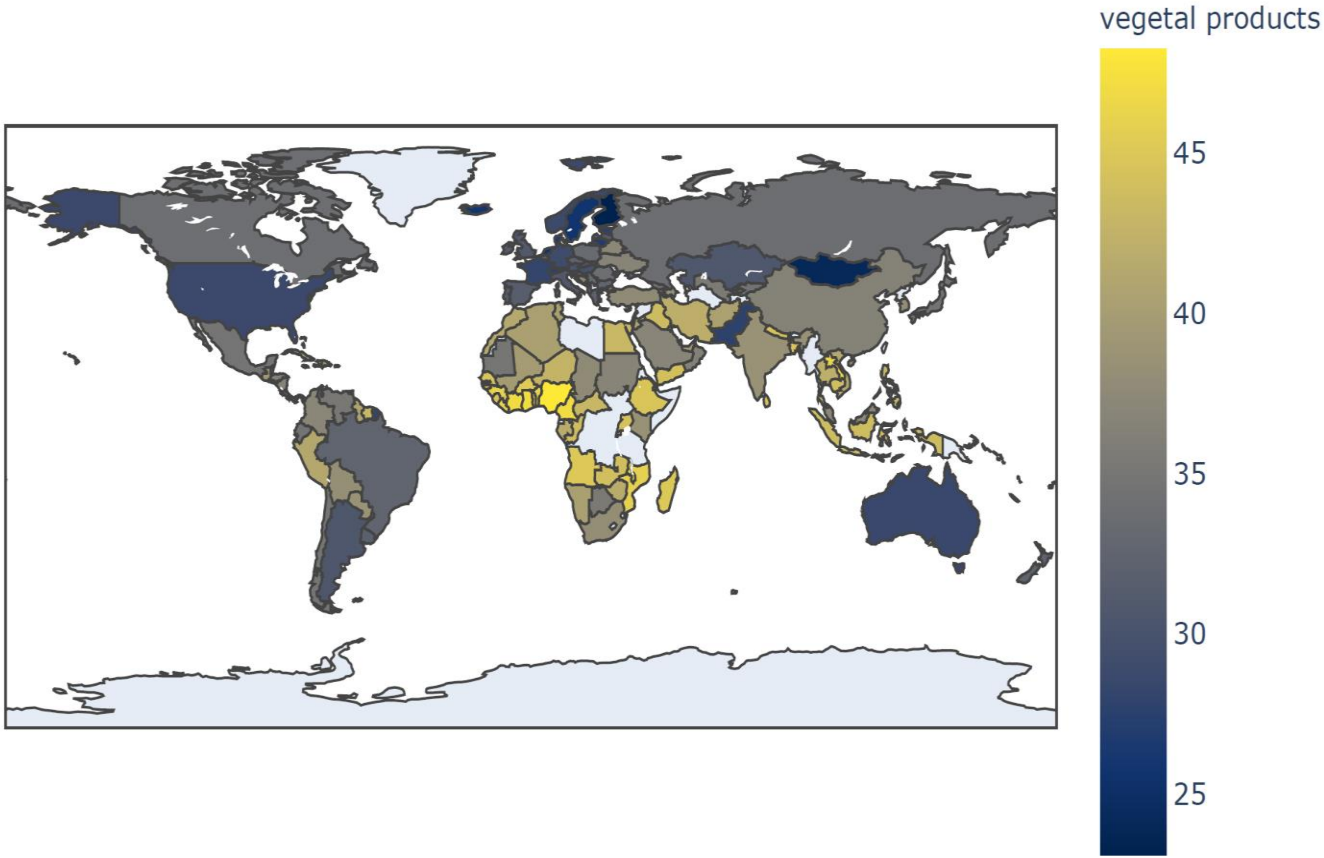
- From this figure we can see that in Western European Countries, Australia and USA the rate of animal product consumption is higher than that of in the African counties and other continents and has a direct correlation with the number of COVID-19 induced deaths



Findings

- The adjoining figure shows the distribution of vegetal product consumption in kgs as a percentage of the total population

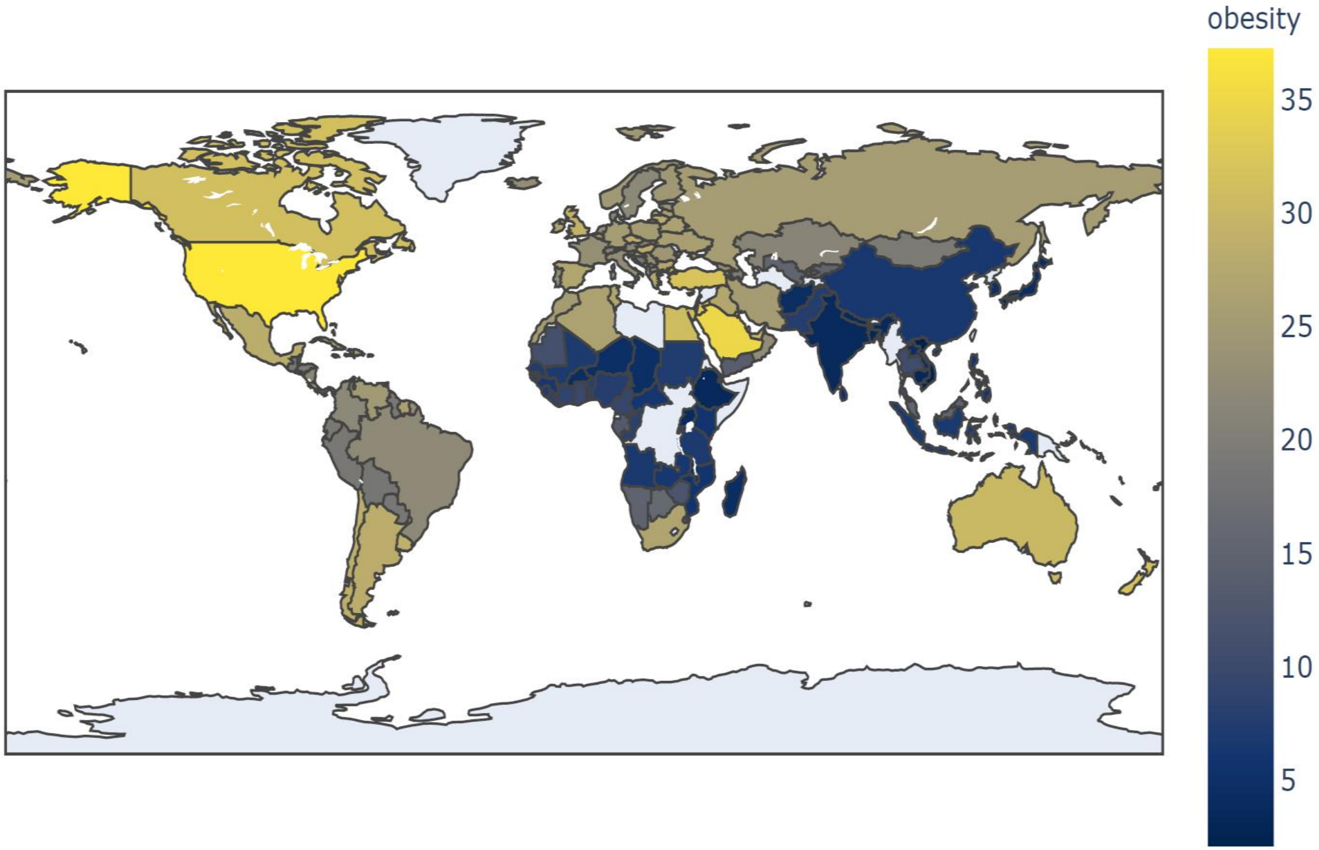
- From this figure we can see that in African Countries and few other parts of Asia the rate of vegetal product consumption is higher than that of in the other countries and having comparatively less death rates.



Findings

- The adjoining figure shows the distribution of obesity as a percentage of the total population

- From this figure we can see that high income countries USA, Canada, Australia and few middle east countries are quite high in obesity which may adversely effect prognosis during COVID-19.



Limitations

- One potential limitation of the present study is that the presence of a lot of missing values led to the truncation of the original dataset which had 170 countries down to 152 countries. Although the loss of information of 18 countries out of a list of 170 does not seem to be a lot, but having the full dataset could possibly add to the strength of the associations observed in this study.

- Although consumption of higher amounts of animal products and obesity are positively associated with susceptibility to COVID-19, it is to be noted that one cannot infer causality from a purely correlational analysis.

Conclusions

- The study reveals global variations in dietary choices with higher income countries showing an increased affinity towards a diet based on animal products whereas most middle to low income countries follow a predominantly vegetal product based diet.

- Direct association of animal product consumption and indirect association of vegetal product consumption with COVID-19 related death indicates that higher consumption of animal products could make an individual susceptible to mortality due to COVID-19.

- The analysis also predicts obesity as a potential risk factor for COVID-19 related mortality

Acknowledgements

- Dataset : Kaggle
- I had insightful discussions with a friend of mine who is a biomedical researcher.

References

- Dataset : www.Kaggle.com
- www.stackoverflow.com
- www.plotly.com : for projections of data on the world map
- www.pandas.pydata.org

Analyzing associations between diet and

susceptibility to COVID-19 induced death

Import libraries

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

import pycountry

from IPython.display import display

import plotly
import plotly.express as px
```

Load data

In [2]:
```
foodsupplyquantity = pd.read_csv('Food_Supply_Quantity_kg_Data.csv')
print(foodsupplyquantity.shape)
display(foodsupplyquantity.head())
```

```
(170, 32)
```

	Country	Alcoholic Beverages	Animal fats	Animal Products	Aquatic Products, Other	Cereals - Excluding Beer	Eggs	Fish, Seafood	Fruits - Excluding Wine	Meat	...	Vegetables	Vegetal Products
0	Afghanistan	0.0014	0.1973	9.4341	0.0	24.8097	0.2099	0.0350	5.3495	1.2020	...	6.7642	40.5645
1	Albania	1.6719	0.1357	18.7684	0.0	5.7817	0.5815	0.2126	6.7861	1.8845	...	11.7753	31.2304
2	Algeria	0.2711	0.0282	9.6334	0.0	13.6816	0.5277	0.2416	6.3801	1.1305	...	11.6484	40.3651
3	Angola	5.8087	0.0560	4.9278	0.0	9.1085	0.0587	1.7707	6.0005	2.0571	...	2.3041	45.0722
4	Antigua and Barbuda	3.5764	0.0087	16.6613	0.0	5.9960	0.2274	4.1489	10.7451	5.6888	...	5.4495	33.3233

5 rows × 32 columns

Data cleaning and preparation for analysis

Check if data file has missing values

In [3]:
```
foodsupplyquantity.isnull().any()
```

Out[3]:
```
Country                          False
Alcoholic Beverages              False
Animal fats                      False
Animal Products                  False
Aquatic Products, Other          False
Cereals - Excluding Beer         False
Eggs                             False
Fish, Seafood                    False
Fruits - Excluding Wine          False
Meat                             False
Milk - Excluding Butter          False
Miscellaneous                    False
Offals                           False
Oilcrops                         False
Pulses                           False
Spices                           False
Starchy Roots                    False
Stimulants                       False
Sugar & Sweeteners               False
Sugar Crops                      False
Treenuts                         False
Vegetable Oils                   False
Vegetables                       False
Vegetal Products                 False
Obesity                           True
Undernourished                    True
Confirmed                         True
Deaths                            True
Recovered                         True
Active                            True
Population                       False
Unit (all except Population)     False
dtype: bool
```

It can be seen that the columns 'Obesity', 'Undernourished', 'Confirmed', 'Deaths', 'Recovered' and 'Active' all have NaN values. These need to be cleaned first before attempting any data analysis. I decided to drop the rows containing missing values/NaN.

In [4]:
```
foodsupplyquantity.dropna(inplace=True)
print(foodsupplyquantity.shape)
foodsupplyquantity.head()
```

```
(152, 32)
```

Out[4]:

	Country	Alcoholic Beverages	Animal fats	Animal Products	Aquatic Products, Other	Cereals - Excluding Beer	Eggs	Fish, Seafood	Fruits - Excluding Wine	Meat	...	Vegetables	Vegetal Products
0	Afghanistan	0.0014	0.1973	9.4341	0.0	24.8097	0.2099	0.0350	5.3495	1.2020	...	6.7642	40.5645
1	Albania	1.6719	0.1357	18.7684	0.0	5.7817	0.5815	0.2126	6.7861	1.8845	...	11.7753	31.2304
2	Algeria	0.2711	0.0282	9.6334	0.0	13.6816	0.5277	0.2416	6.3801	1.1305	...	11.6484	40.3651
3	Angola	5.8087	0.0560	4.9278	0.0	9.1085	0.0587	1.7707	6.0005	2.0571	...	2.3041	45.0722
5	Argentina	4.2672	0.2234	19.3454	0.0	8.4102	0.9979	0.4693	6.0435	7.0421	...	4.3503	30.6559

5 rows × 32 columns

Check the data types of each of the columns

In [5]:
```
foodsupplyquantity.dtypes
```

Out[5]:
```
Country                            object
Alcoholic Beverages               float64
Animal fats                       float64
Animal Products                   float64
Aquatic Products, Other           float64
Cereals - Excluding Beer          float64
Eggs                              float64
Fish, Seafood                     float64
Fruits - Excluding Wine           float64
Meat                              float64
Milk - Excluding Butter           float64
Miscellaneous                     float64
Offals                            float64
Oilcrops                          float64
Pulses                            float64
Spices                            float64
Starchy Roots                     float64
Stimulants                        float64
Sugar & Sweeteners                float64
Sugar Crops                       float64
Treenuts                          float64
Vegetable Oils                    float64
Vegetables                        float64
Vegetal Products                  float64
Obesity                           float64
Undernourished                     object
Confirmed                         float64
Deaths                            float64
Recovered                         float64
Active                            float64
Population                        float64
Unit (all except Population)       object
dtype: object
```

Everything is as expected. The only exception is the 'Undernourished' column which should be a float but it is displayed as an object. Let us inspect this column closely.

In [6]:
```
undernourished = foodsupplyquantity.Undernourished.values
undernourished
```

Out[6]:
```
array(['29.8', '6.2', '3.9', '25', '4.6', '4.3', '<2.5', '<2.5', '<2.5',
       '14.7', '3.9', '<2.5', '<2.5', '7.5', '10.1', '17.1', '<2.5',
       '26.4', '<2.5', '3.6', '20', '12.6', '16.4', '9.9', '<2.5', '59.6',
       '37.5', '2.7', '8.5', '4.8', '40.3', '4.8', '19', '<2.5', '<2.5',
       '5.6', '<2.5', '<2.5', '18.9', '6.2', '9.5', '7.9', '4.5', '9',
       '2.9', '20.6', '20.6', '3.7', '<2.5', '<2.5', '10.5', '10.2',
       '7.9', '<2.5', '5.5', '<2.5', '15.2', '16.5', '28', '8.1', '49.3',
       '12.9', '<2.5', '<2.5', '14.5', '8.3', '4.9', '29', '<2.5', '<2.5',
       '<2.5', '8', '<2.5', '12.2', '<2.5', '29.4', '<2.5', '2.8', '7.1',
       '16.5', '<2.5', '11', '37.2', '<2.5', '<2.5', '44.4', '17.5',
       '2.5', '10.3', '6.3', '<2.5', '10.4', '6.5', '3.6', '13.4', '<2.5',
       '3.4', '27.9', '27.3', '8.7', '<2.5', '<2.5', '17', '16.5', '13.4',
       '3.2', '<2.5', '6.8', '20.3', '10', '10.7', '9.7', '13.3', '<2.5',
       '<2.5', '<2.5', '<2.5', '36.8', '5.7', '7', '7.1', '11.3', '5.7',
       '25.6', '3.4', '<2.5', '6.2', '<2.5', '9', '20.1', '8.5', '<2.5',
       '<2.5', '7.8', '24.9', '16.1', '5.5', '4.3', '<2.5', '41', '3.5',
       '2.6', '<2.5', '30.7', '<2.5', '<2.5', '6.3', '21.2', '9.3',
       '38.9', '46.7', '51.3'], dtype=object)
```

Oops! The column has a lot NaN values (almost 1/3rd). Therefore, I decided to drop this column.

In [7]:
```
foodsupplyquantity.drop(['Undernourished', 'Unit (all except Population)'], axis=1, inplac
e=True)
print(foodsupplyquantity.shape)
foodsupplyquantity.head()
```

```
(152, 30)
```

Out[7]:

	Country	Alcoholic Beverages	Animal fats	Animal Products	Aquatic Products, Other	Cereals - Excluding Beer	Eggs	Fish, Seafood	Fruits - Excluding Wine	Meat	...	Treenuts	Vegetable Oils
0	Afghanistan	0.0014	0.1973	9.4341	0.0	24.8097	0.2099	0.0350	5.3495	1.2020	...	0.0770	0.5345
1	Albania	1.6719	0.1357	18.7684	0.0	5.7817	0.5815	0.2126	6.7861	1.8845	...	0.1515	0.3261
2	Algeria	0.2711	0.0282	9.6334	0.0	13.6816	0.5277	0.2416	6.3801	1.1305	...	0.1152	1.0310
3	Angola	5.8087	0.0560	4.9278	0.0	9.1085	0.0587	1.7707	6.0005	2.0571	...	0.0061	0.6463
5	Argentina	4.2672	0.2234	19.3454	0.0	8.4102	0.9979	0.4693	6.0435	7.0421	...	0.0200	0.9541

5 rows × 30 columns

Now let us check the data types once again to make sure everything is in order.

In [8]:
```
foodsupplyquantity.dtypes
```

Out[8]:
```
Country                      object
Alcoholic Beverages         float64
Animal fats                 float64
Animal Products             float64
Aquatic Products, Other     float64
Cereals - Excluding Beer    float64
Eggs                        float64
Fish, Seafood               float64
Fruits - Excluding Wine     float64
Meat                        float64
Milk - Excluding Butter     float64
Miscellaneous               float64
Offals                      float64
Oilcrops                    float64
Pulses                      float64
Spices                      float64
Starchy Roots               float64
Stimulants                  float64
Sugar & Sweeteners          float64
Sugar Crops                 float64
Treenuts                    float64
Vegetable Oils              float64
Vegetables                  float64
Vegetal Products            float64
Obesity                     float64
Confirmed                   float64
Deaths                      float64
Recovered                   float64
Active                      float64
Population                  float64
dtype: object
```

Excellent! One column (Country) which is an object, everything else are floats. Now the data is ready for further analysis.

Exploratory data analysis

To start with, I wanted to look at the distribution of number of COVID-19 induced deaths across different countries of the world. To do this, I first created a new dataframe containing the data sorted on the 'Deaths' column.

In [9]:
```python
foodquantity_deathsorted = foodsupplyquantity.sort_values(by=['Deaths'], ascending=False)

foodquantity_deathsorted.head()
```

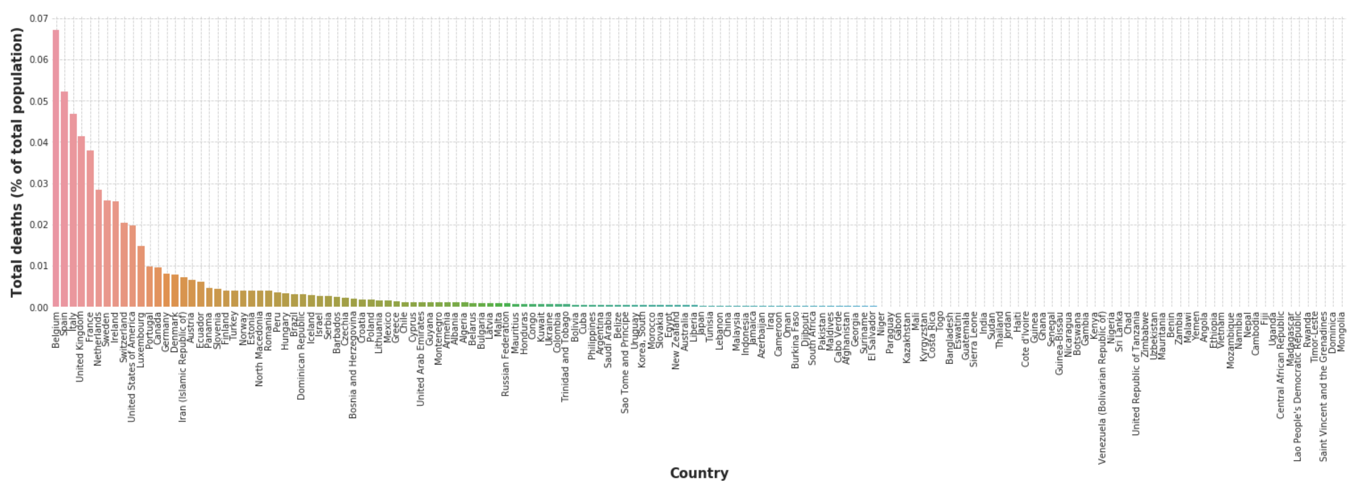
Out[9]:

	Country	Alcoholic Beverages	Animal fats	Animal Products	Aquatic Products, Other	Cereals - Excluding Beer	Eggs	Fish, Seafood	Fruits - Excluding Wine	Meat	...	Treenuts	Vegetable Oils
14	Belgium	5.3730	0.8559	17.7279	0.0010	6.6704	0.6487	1.1325	4.1623	3.2370	...	0.1309	0.5640
141	Spain	5.3152	0.2357	18.3382	0.0011	6.0548	0.7321	2.2646	4.8332	5.3456	...	0.4287	1.5122
74	Italy	3.1892	0.2834	19.0329	0.0005	8.5417	0.6247	1.5816	6.0207	4.2963	...	0.4230	1.4224
159	United Kingdom	5.2632	0.2754	18.8798	0.0006	6.5412	0.6210	1.0911	4.9551	4.4181	...	0.0901	0.7372
51	France	4.1631	0.7907	21.7097	0.0011	6.9951	0.6110	1.8214	4.8780	4.4005	...	0.2093	0.8675

5 rows × 30 columns

Plot the deaths as percentage of total population per country

In [10]:
```python
with plt.style.context('seaborn-whitegrid'):
    fig,ax = plt.subplots(figsize=(26,6))
    ax=sns.barplot(x='Country', y='Deaths', data=foodquantity_deathsorted)
    plt.box(False)
    plt.xticks(rotation=90)
    plt.grid('both', 'both', linestyle='--')
    plt.ylabel('Total deaths (% of total population)', fontsize=15, fontweight='semibold')
    plt.xlabel('Country', fontsize=15, fontweight='semibold')
    plt.savefig('deaths_by_country.png', dpi=300, bbox_iches='tight')
```



Association of food types intake with COVID-19 susceptibility

Create a dataframe of correlations of different food intakes with COVID-19 induced deaths

In [11]:

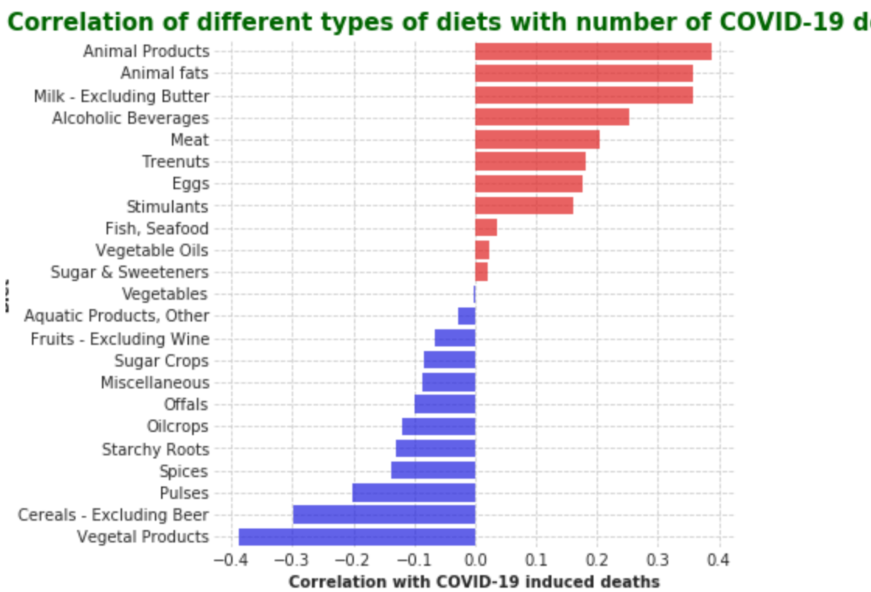
```
deathswithfood = pd.DataFrame((foodsupplyquantity.iloc[:, 1:-6].corrwith(foodsupplyquantit
y.loc[:,'Deaths'],
                                                                                          axis = 0)),
                                    columns=['Deaths'])
deathswithfood.reset_index(inplace=True)

deathswithfood.sort_values(by=['Deaths'], ascending=False, inplace=True)

display(deathswithfood)
```

	index	Deaths
2	Animal Products	0.388432
1	Animal fats	0.356669
9	Milk - Excluding Butter	0.356668
0	Alcoholic Beverages	0.253389
8	Meat	0.204652
19	Treenuts	0.181462
5	Eggs	0.177146
16	Stimulants	0.160374
6	Fish, Seafood	0.037289
20	Vegetable Oils	0.023672
17	Sugar & Sweeteners	0.020309
21	Vegetables	-0.003258
3	Aquatic Products, Other	-0.026893
7	Fruits - Excluding Wine	-0.065977
18	Sugar Crops	-0.083367
10	Miscellaneous	-0.086507
11	Offals	-0.099017
12	Oilcrops	-0.119710
15	Starchy Roots	-0.128933
14	Spices	-0.136672
13	Pulses	-0.202548
4	Cereals - Excluding Beer	-0.298072
22	Vegetal Products	-0.388501

Plot the correlations

In [12]:

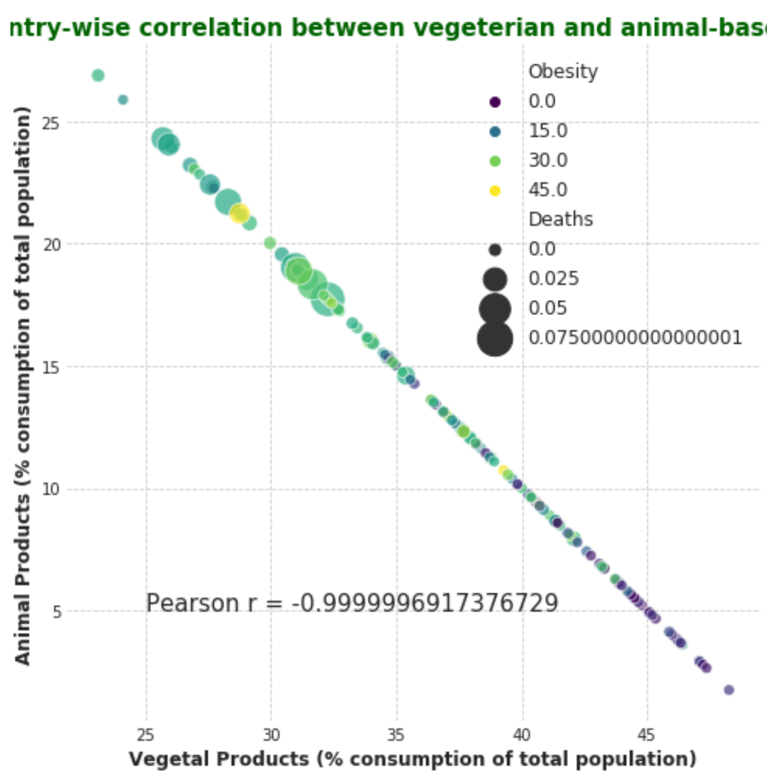
```python
with plt.style.context('seaborn-whitegrid'):
    fig,ax = plt.subplots(figsize=(6, 6))
    sns.barplot(x='Deaths', y = 'index',
                palette = (deathswithfood['Deaths']>0).map({True:'r', False:'b'}),
                data=deathswithfood, alpha=0.7)
    plt.box(False)
    plt.grid('both', 'both', linestyle='--')
    plt.xlabel('Correlation with COVID-19 induced deaths', fontweight='semibold')
    plt.ylabel('Diet', fontweight='semibold')
    plt.title('Correlation of different types of diets with number of COVID-19 deaths',
              color='darkgreen', fontweight='semibold', fontsize=15)
    ax.set_axisbelow(True)
    plt.savefig('deaths_correlation_with_diet.png', dpi=300, bbox_inches='tight')
```



From the correlations plot, it is clear that a diet involving animal products are directly correlated with number of COVID-19 associated deaths, whereas diets primarily involving vegetal products are inversely correlated with COVID-19 associated deaths. Therefore, the next objective was to test if there is a bifurcation between countries based on animal of vegetal products consumption.

In [13]:

```
corr = str(foodsupplyquantity['Animal Products'].corr(foodsupplyquantity['Vegetal Products
']))
with plt.style.context('seaborn-whitegrid'):
    fig,ax=plt.subplots(figsize=(8,8))
    ax=sns.scatterplot(x='Vegetal Products', y='Animal Products', size='Deaths', hue='Obes
ity', palette='viridis',
                       data=foodsupplyquantity, sizes=(50,500),
                       legend='brief', alpha=0.7)
#     ax=sns.regplot(x='Vegetal Products', y='Animal Products',
#                    data=foodsupplyquantity,scatter=False)

#     for i, txt in enumerate(n):
#         ax.annotate(txt, (x[i], y[i]), fontsize=12)
    plt.box(False)
    plt.grid('both', 'both', linestyle='--')
    plt.xlabel('Vegetal Products (% consumption of total population)',
               fontsize=12, fontweight='semibold')
    plt.ylabel('Animal Products (% consumption of total population)',
               fontsize=12, fontweight='semibold')
    plt.title('Country-wise correlation between vegeterian and animal-based diets',
              color='darkgreen', fontweight='semibold', fontsize=15)
    ax.set_axisbelow(True)
    plt.legend(fontsize=12, bbox_to_anchor=(1,1))
    plt.text(25, 5, 'Pearson r = ' + corr, fontsize=15)
    plt.savefig('veg_vs_nonveg.png', dpi=300, bbox_inches='tight')
```



We see a strong negative correlation between vegetal products and animal products across countries of the world. Moreover, the plot also shows that countries with higher percentage of the population on animal products also have higher obesity. Therefore, these results indicate a direct association between obesity and COVID-19 induced death.

Plot the distribution of different diet patterns associated with

COVID-19 susceptibility across all the countries of the world

Create a dataframe with only the values I want to plot.

In [14]:
```
df = foodsupplyquantity.loc[:, ('Country', 'Animal Products', 'Vegetal Products', 'Obesity
', 'Deaths')]
df.columns = ['country', 'animal products', 'vegetal products', 'obesity', 'deaths']
df.head()
```

Out[14]:

	country	animal products	vegetal products	obesity	deaths
0	Afghanistan	9.4341	40.5645	4.5	0.000179
1	Albania	18.7684	31.2304	22.3	0.001085
2	Algeria	9.6334	40.3651	26.6	0.001044
3	Angola	4.9278	45.0722	6.8	0.000006
5	Argentina	19.3454	30.6559	28.5	0.000501

Create a list of 3 letter ISO codes for the countries

In [15]:
```
input_countries = df.country.values.tolist()

countries = {}
for country in pycountry.countries:
    countries[country.name] = country.alpha_3

country_codes = [countries.get(country, 'Unknown code') for country in input_countries]

print(country_codes)
```

```
['AFG', 'ALB', 'DZA', 'AGO', 'ARG', 'ARM', 'AUS', 'AUT', 'AZE', 'BGD', 'BRB', 'BLR', 'BEL', 'BLZ', 'BEN', 'Unknown code', 'BIH', 'B
WA', 'BRA', 'BGR', 'BFA', 'CPV', 'KHM', 'CMR', 'CAN', 'CAF', 'TCD', 'CHL', 'CHN', 'COL', 'COG', 'CRI', 'Unknown code', 'HRV', 'CUB'
, 'CYP', 'CZE', 'DNK', 'DJI', 'DMA', 'DOM', 'ECU', 'EGY', 'SLV', 'EST', 'Unknown code', 'ETH', 'FJI', 'FIN', 'FRA', 'GAB', 'GMB', '
GEO', 'DEU', 'GHA', 'GRC', 'GTM', 'GIN', 'GNB', 'GUY', 'HTI', 'HND', 'HUN', 'ISL', 'IND', 'IDN', 'Unknown code', 'IRQ', 'IRL', 'ISR
', 'ITA', 'JAM', 'JPN', 'JOR', 'KAZ', 'KEN', 'Unknown code', 'KWT', 'KGZ', 'LAO', 'LVA', 'LBN', 'LBR', 'LTU', 'LUX', 'MDG', 'MWI',
'MYS', 'MDV', 'MLI', 'MLT', 'MRT', 'MUS', 'MEX', 'MNG', 'MNE', 'MAR', 'MOZ', 'NAM', 'NPL', 'NLD', 'NZL', 'NIC', 'NER', 'NGA', 'Unkn
own code', 'NOR', 'OMN', 'PAK', 'PAN', 'PRY', 'PER', 'PHL', 'POL', 'PRT', 'ROU', 'RUS', 'RWA', 'VCT', 'STP', 'SAU', 'SEN', 'SRB', '
SLE', 'SVK', 'SVN', 'ZAF', 'ESP', 'LKA', 'SDN', 'SUR', 'SWE', 'CHE', 'THA', 'TLS', 'TGO', 'TTO', 'TUN', 'TUR', 'UGA', 'UKR', 'ARE',
'GBR', 'Unknown code', 'Unknown code', 'URY', 'UZB', 'Unknown code', 'Unknown code', 'YEM', 'ZMB', 'ZWE']
```

Some countries are not named prpperly. Hence, we see the 'Unknown code' pop up in a few places. I will print the list of countries and then manually modify the ISO codes.

In [16]:
```
print(input_countries)
```

```
['Afghanistan', 'Albania', 'Algeria', 'Angola', 'Argentina', 'Armenia', 'Australia', 'Austria', 'Azerbaijan', 'Bangladesh', 'Barbad
os', 'Belarus', 'Belgium', 'Belize', 'Benin', 'Bolivia', 'Bosnia and Herzegovina', 'Botswana', 'Brazil', 'Bulgaria', 'Burkina Faso'
, 'Cabo Verde', 'Cambodia', 'Cameroon', 'Canada', 'Central African Republic', 'Chad', 'Chile', 'China', 'Colombia', 'Congo', 'Costa
Rica', "Cote d'Ivoire", 'Croatia', 'Cuba', 'Cyprus', 'Czechia', 'Denmark', 'Djibouti', 'Dominica', 'Dominican Republic', 'Ecuador',
'Egypt', 'El Salvador', 'Estonia', 'Eswatini', 'Ethiopia', 'Fiji', 'Finland', 'France', 'Gabon', 'Gambia', 'Georgia', 'Germany', 'G
hana', 'Greece', 'Guatemala', 'Guinea', 'Guinea-Bissau', 'Guyana', 'Haiti', 'Honduras', 'Hungary', 'Iceland', 'India', 'Indonesia',
'Iran (Islamic Republic of)', 'Iraq', 'Ireland', 'Israel', 'Italy', 'Jamaica', 'Japan', 'Jordan', 'Kazakhstan', 'Kenya', 'Korea, So
uth', 'Kuwait', 'Kyrgyzstan', "Lao People's Democratic Republic", 'Latvia', 'Lebanon', 'Liberia', 'Lithuania', 'Luxembourg', 'Madag
ascar', 'Malawi', 'Malaysia', 'Maldives', 'Mali', 'Malta', 'Mauritania', 'Mauritius', 'Mexico', 'Mongolia', 'Montenegro', 'Morocco'
, 'Mozambique', 'Namibia', 'Nepal', 'Netherlands', 'New Zealand', 'Nicaragua', 'Niger', 'Nigeria', 'North Macedonia', 'Norway', 'Om
an', 'Pakistan', 'Panama', 'Paraguay', 'Peru', 'Philippines', 'Poland', 'Portugal', 'Romania', 'Russian Federation', 'Rwanda', 'Sai
nt Vincent and the Grenadines', 'Sao Tome and Principe', 'Saudi Arabia', 'Senegal', 'Serbia', 'Sierra Leone', 'Slovakia', 'Slovenia
', 'South Africa', 'Spain', 'Sri Lanka', 'Sudan', 'Suriname', 'Sweden', 'Switzerland', 'Thailand', 'Timor-Leste', 'Togo', 'Trinidad
and Tobago', 'Tunisia', 'Turkey', 'Uganda', 'Ukraine', 'United Arab Emirates', 'United Kingdom', 'United Republic of Tanzania', 'Un
ited States of America', 'Uruguay', 'Uzbekistan', 'Venezuela (Bolivarian Republic of)', 'Vietnam', 'Yemen', 'Zambia', 'Zimbabwe']
```

In [17]:
```
country_codes = ['AFG', 'ALB', 'DZA', 'AGO', 'ARG', 'ARM', 'AUS', 'AUT', 'AZE',
                 'BGD', 'BRB', 'BLR', 'BEL', 'BLZ', 'BEN', 'BOL', 'BIH',
                 'BWA', 'BRA', 'BGR', 'BFA', 'CPV', 'KHM', 'CMR', 'CAN', 'CAF',
                 'TCD', 'CHL', 'CHN', 'COL', 'COG', 'CRI', 'CIV', 'HRV',
                 'CUB', 'CYP', 'CZE', 'DNK', 'DJI', 'DMA', 'DOM', 'ECU', 'EGY',
                 'SLV', 'EST', 'SWZ', 'ETH', 'FJI', 'FIN', 'FRA', 'GAB',
                 'GMB', 'GEO', 'DEU', 'GHA', 'GRC', 'GTM', 'GIN', 'GNB', 'GUY',
                 'HTI', 'HND', 'HUN', 'ISL', 'IND', 'IDN', 'IRN', 'IRQ',
                 'IRL', 'ISR', 'ITA', 'JAM', 'JPN', 'JOR', 'KAZ', 'KEN', 'KOR',
                 'KWT', 'KGZ', 'LAO', 'LVA', 'LBN', 'LBR', 'LTU', 'LUX', 'MDG',
                 'MWI', 'MYS', 'MDV', 'MLI', 'MLT', 'MRT', 'MUS', 'MEX', 'MNG',
                 'MNE', 'MAR', 'MOZ', 'NAM', 'NPL', 'NLD', 'NZL', 'NIC', 'NER',
                 'NGA', 'MKD', 'NOR', 'OMN', 'PAK', 'PAN', 'PRY', 'PER',
                 'PHL', 'POL', 'PRT', 'ROU', 'RUS', 'RWA', 'VCT', 'STP', 'SAU',
                 'SEN', 'SRB', 'SLE', 'SVK', 'SVN', 'ZAF', 'ESP', 'LKA', 'SDN',
                 'SUR', 'SWE', 'CHE', 'THA', 'TLS', 'TGO', 'TTO', 'TUN', 'TUR',
                 'UGA', 'UKR', 'ARE', 'GBR', 'TZA', 'USA',
                 'URY', 'UZB', 'VEN', 'VNM', 'YEM', 'ZMB', 'ZWE']
```

Add the ISO codes as a new column to the existing dataframe

In [18]:
```
df['iso_alpha'] = country_codes

df.head()
```

Out[18]:

	country	animal products	vegetal products	obesity	deaths	iso_alpha
0	Afghanistan	9.4341	40.5645	4.5	0.000179	AFG
1	Albania	18.7684	31.2304	22.3	0.001085	ALB
2	Algeria	9.6334	40.3651	26.6	0.001044	DZA
3	Angola	4.9278	45.0722	6.8	0.000006	AGO
5	Argentina	19.3454	30.6559	28.5	0.000501	ARG

Now plot the values I want on a world map.

In [19]:

```
fig = px.choropleth(df, locations="iso_alpha",
                    color="animal products", # lifeExp is a column of gapminder
                    hover_name="country", # column to add to hover information
                    color_continuous_scale=px.colors.sequential.Cividis)
fig.write_image('animal_products_worldmap.pdf')
fig.show()
```