IU Internationale Hochschule

Written Assignment

Visually exploring a data set

Data Analyst - Python (B2G-UPS DPDAPE)

Task for Course: DLBDSEDAV01 – Exploratory Data Analysis and Visualization

Date: 29.09.2023

Author: Boieru Mykola

Matriculation number: UPS10618765

Tutor: Visieu Lac

Table of content

Introduction

Visualization is one of the most important part of data analysis, because collected data is worth nothing if it cannot be presented. From sensible presentation of data depends how easily it will be understandable. Wrong presented data can lead to making wrong decisions that can have bad sequences for company in the future.

Important part of visualization is preparing data for it. Explorative data analysis consist of describing location, variance, distributions, covariance and correlation. This analysis allows to understand better, how the dataset looks like, from which parts consists it, how big dataset is, which relationships variables have and to determine trends of data. Based on this information and dependent on what exactly is expected from data, can be chosen an appropriate way to visualise data.

There are many ways and tools to visualize data. But not every graph, map or plot would be representable enough for certain type of data. Some type of graphs were popular earlier, but now there are more modern and suitable ways. It is also important not to overloaded graphs with data. The main purpose of visualization is to simplify and to facilitate data so, that it contains only important information for required tasks. Because of that it is much better to create several graphs with different aspects. One of the most powerful tool for data visualization is Python, mostly because of its powerful libraries, such plotly, seaborn, statistics, matplotlib, numpy, pandas etc.

First of all, it is needed to be provided explorative data analysis, then, after understanding structure of dataset, data will be visualized in the most suitable way. The whole code will be written in Jupyter Notebook and can be found with the following link on github. There can be found also a dataset:

https://github.com/MykolaBoieru/Explorative-Data-Analysis-and-Visualization

For the task was chosen a dataset with data about global alcohol consumption by country. There are also columns with amounts of beer, spirit and wine servings. This dataset allows to demonstrate learned skills for providing explorative data analysis and visualization, creating different types of graphs, plots and maps as well as finding location, distribution and relationships between different variables.

1. Exploratory Data Analysis

Explorative data analysis is an important part of preparing to data visualization. It helps to understand data better and to make a decision how visualization will look like, because different types of data are needed to be presented in different ways, dependent on what exactly this data can tell us. The first step to working with data, using Python is to import required libraries. On the picture below can be seen which libraries should de imported and how they are named for further process. Some of them are needed for calculating statistics and other will be used for creating graphs or maps.

```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import plotly.express as pdx
import plotly.offline as pyo
import statistics
```

To start with explorative data analysis, it is useful to see the structure of database. The following picture shows the code and result of its execution. Firstly, dataset is read with pandas function and then printed.

```
df = pd.read_csv("drinks.csv")
print(df)
```

```
         country  beer_servings  spirit_servings  wine_servings  \
0    Afghanistan              0                0              0
1        Albania             89              132             54
2        Algeria             25                0             14
3        Andorra            245              138            312
4         Angola            217               57             45
..           ...            ...              ...            ...
188    Venezuela            333              100              3
189      Vietnam            111                2              1
190        Yemen              6                0              0
191       Zambia             32               19              4
192     Zimbabwe             64               18              4

     total_litres_of_pure_alcohol
0                             0.0
1                             4.9
2                             0.7
3                            12.4
4                             5.9
..                            ...
188                           7.7
189                           2.0
190                           0.1
191                           2.5
192                           4.7

[193 rows x 5 columns]
```

It can be seen that the dataset has 193 rows (countries) and 5 columns. Data that is contained in those columns will be analysed and visualized. For explorative analysis it is necessary to find basic statistic indicators such as median, standard deviation and variance. In the dataset there are different data about alcohol consumption, amount of beer, spirit and wine servings, but also an amount of litres of pure alcohol per person by country. It would have more sense to calculate statistics for total amount of alcohol per person. One of the way to do it is to use statistics library with the following code.

```
median = statistics.median(df['total_litres_of_pure_alcohol'])
var = statistics.variance(df['total_litres_of_pure_alcohol'])
std = statistics.stdev(df['total_litres_of_pure_alcohol'])

print('Median of total litres of pure alcohol is: ', median)
print('Variance of total litres of pure alcohol is: ', var)
print('Standard deviation is: ', std)
```

```
Median of total litres of pure alcohol is:  4.2
Standard deviation is:  3.7732981643560835
Variance of total litres of pure alcohol is:  14.237779037132988
```

It can be seen from picture above that, for example, median is 4.2 litres of pure alcohol per person.

An alternative way to get basic statistics is to use function describe() from pandas library. The output is in table format and can be seen on picture below.

```
df.describe()
```

	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol
count	193.000000	193.000000	193.000000	193.000000
mean	106.160622	80.994819	49.450777	4.717098
std	101.143103	88.284312	79.697598	3.773298
min	0.000000	0.000000	0.000000	0.000000
25%	20.000000	4.000000	1.000000	1.300000
50%	76.000000	56.000000	8.000000	4.200000
75%	188.000000	128.000000	59.000000	7.200000
max	376.000000	438.000000	370.000000	14.400000

The output of this picture demonstrates additional information about data, for example, mean, the minimum and maximum data and percentile. Forasmuch, the indicator “count” in the table shows that the dataset contains no empty rows and no missing data.

Using pandas function head() with a number in brackets allow us to see appropriative amount of first rows of dataset.

```
df.head(5)
```

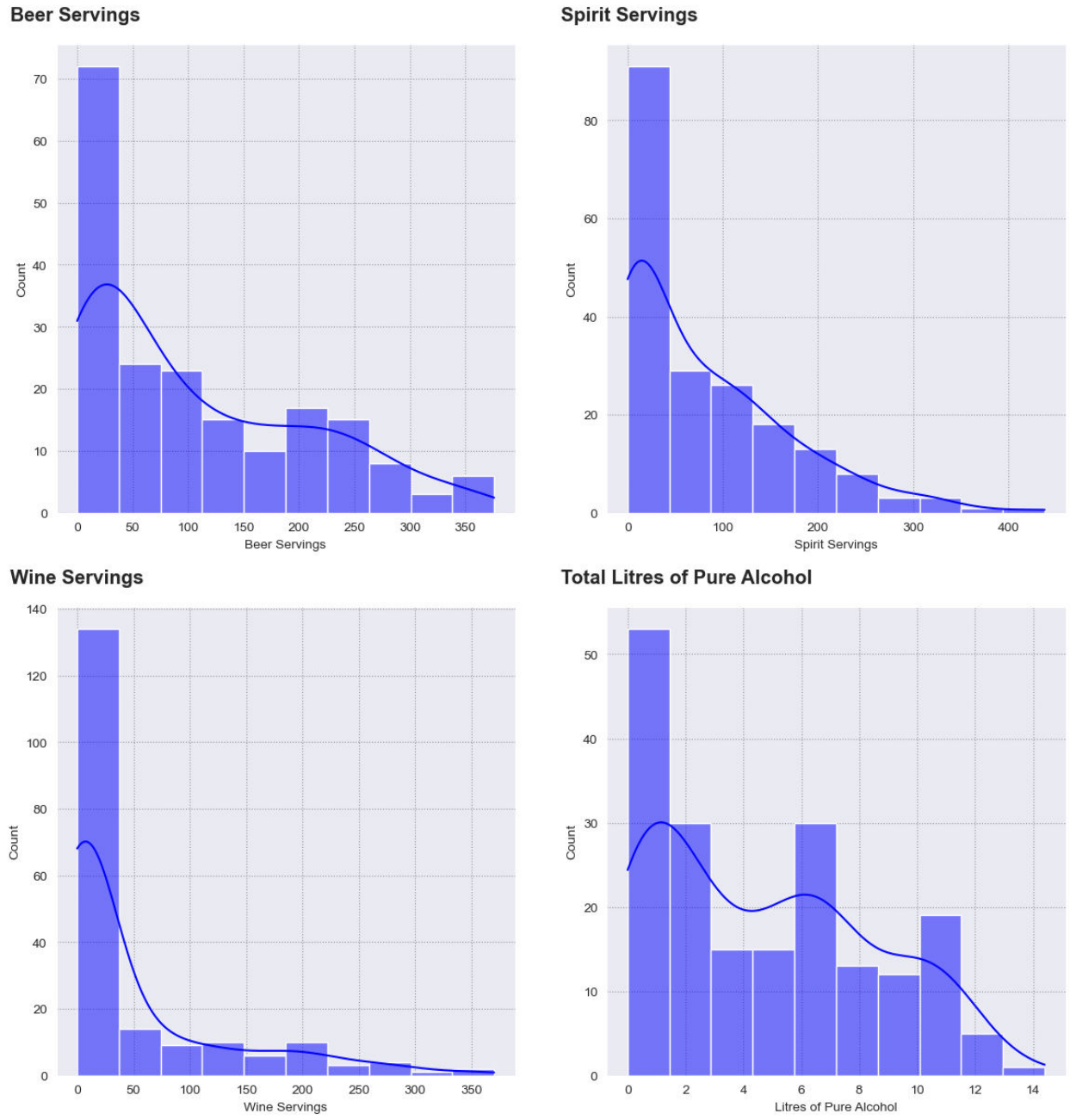
	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol
0	Afghanistan	0	0	0	0.0
1	Albania	89	132	54	4.9
2	Algeria	25	0	14	0.7
3	Andorra	245	138	312	12.4
4	Angola	217	57	45	5.9

This function will be especially useful if it is needed to create graphs with top countries in every section, so this function will be used mostly by visualization.

Providing exploratory data analysis it is important to find data distribution. It represents possible values and how often they occur. Data distribution will be calculated for every of variable and every one will be shown on the one figure. First of all, it will be set colour and style, as well as background, for graphs, created with seaborn. This line of code is needed only once for every graph that will be created in future. Then it is needed to create a figure, on which four graphs will be demonstrated. Creating every graph, it will be set an amount of bins, colour and name for x-label. In additional to that, in graph will be presented also kde-plot that represent density of data. For further convenience, every graphs should have name, its size, weight and intends.

```
sns.set_style("darkgrid", {"grid.color": ".6", "grid.linestyle": ":"}) # backround for seaborn graphs

fig = plt.figure(figsize=(14,14))

ax1 = fig.add_subplot(221)
sns.histplot(df['beer_servings'], bins=10, color='blue', kde=True)
ax1.set(xlabel='Beer Servings')

ax2 = fig.add_subplot(222)
sns.histplot(df['spirit_servings'], bins=10, color='blue', kde=True)
ax2.set(xlabel='Spirit Servings')

ax3 = fig.add_subplot(223)
sns.histplot(df['wine_servings'], bins=10, color='blue', kde=True)
ax3.set(xlabel='Wine Servings')

ax4 = fig.add_subplot(224)
sns.histplot(df['total_litres_of_pure_alcohol'], bins=10, color='blue', kde=True)
ax4.set(xlabel='Litres of Pure Alcohol')

ax1.text(-0.1, 1.05, 'Beer Servings', transform=ax1.transAxes, size=15, weight='bold')
ax2.text(-0.1, 1.05, 'Spirit Servings', transform=ax2.transAxes, size=15, weight='bold')
ax3.text(-0.1, 1.05, 'Wine Servings', transform=ax3.transAxes, size=15, weight='bold')
ax4.text(-0.1, 1.05, 'Total Litres of Pure Alcohol', transform=ax4.transAxes, size=15, weight='bold')
```

After execution the abovementioned code can be seen the following graphs.

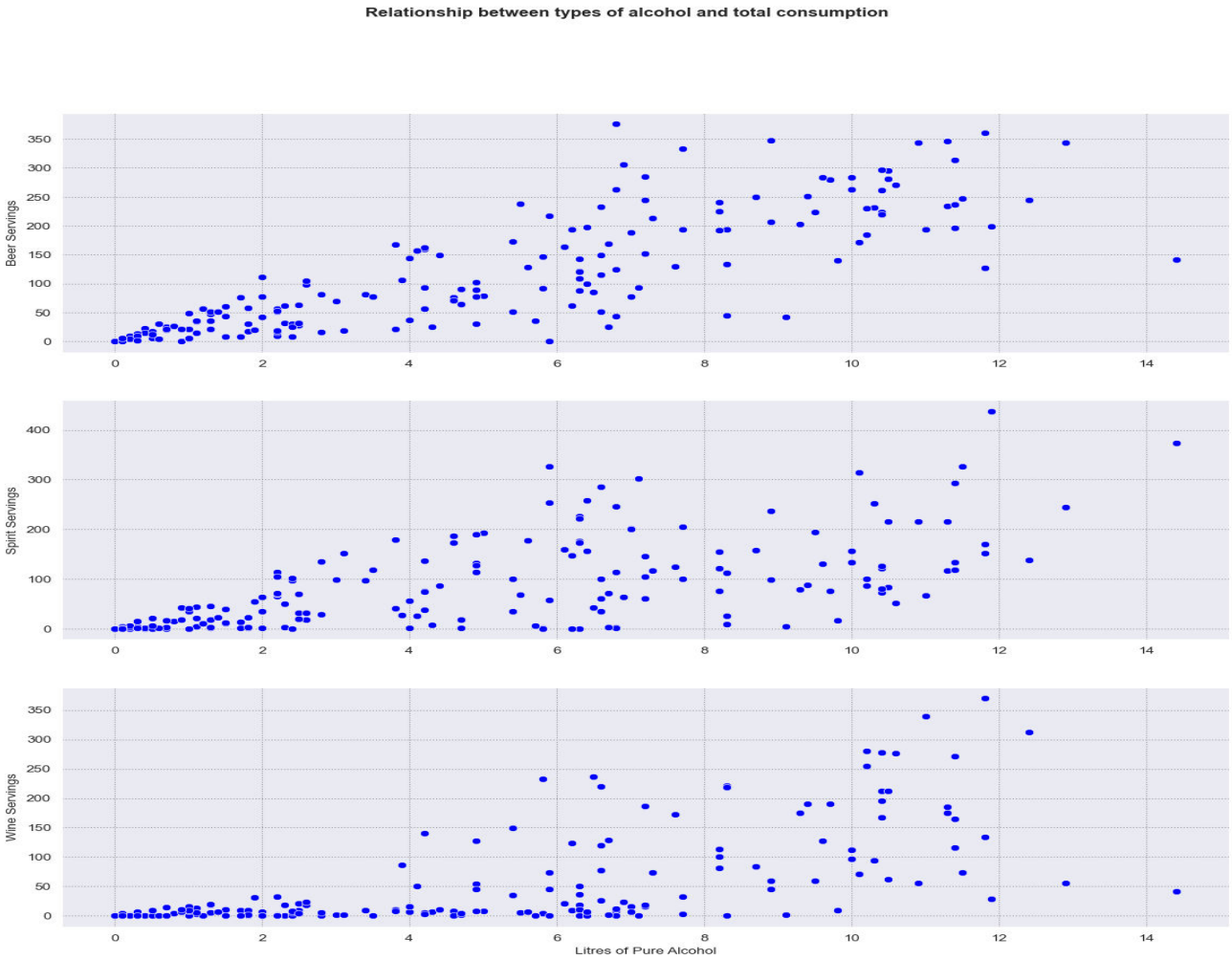


A useful indicator for data is covariance, that represents direction of relationships between two variables. If covariance is positive, it means that both variables are getting higher at the same time and if covariance is negative, it shows that if one variable is getting higher the second one is getting lower.

Firstly, creating a figure, we should resize it for further convenience. It has to be also named. This figure will contain three subplot, every subplot will have the own y-axis but x-axis will be the same for every subplot. Creating the first plot, we can set a colour and values for y-axis and x-axis. Every graph has the own name for y-label, but name for x-label will be written only for third x-label bottom because it is the same for all three graphs. The code can be seen below on the picture.

```
fig = plt.figure(figsize=(14,14))
fig.suptitle('Relationship between types of alcohol and total consumption', \
             fontsize='large', horizontalalignment ='center', fontweight='bold')


ax1 = fig.add_subplot(311)
sns.scatterplot(df, y=df['beer_servings'], x=df['total_litres_of_pure_alcohol'], color='blue')
ax1.set(xlabel=None, ylabel='Beer Servings')

ax2 = fig.add_subplot(312, sharex=ax1)
sns.scatterplot(df, y=df['spirit_servings'], x=df['total_litres_of_pure_alcohol'], color='blue')
ax2.set(xlabel=None, ylabel='Spirit Servings')

ax3 = fig.add_subplot(313, sharex=ax1)
sns.scatterplot(df, y=df['wine_servings'], x=df['total_litres_of_pure_alcohol'], color='blue')
ax3.set(xlabel='Litres of Pure Alcohol', ylabel='Wine Servings')
```

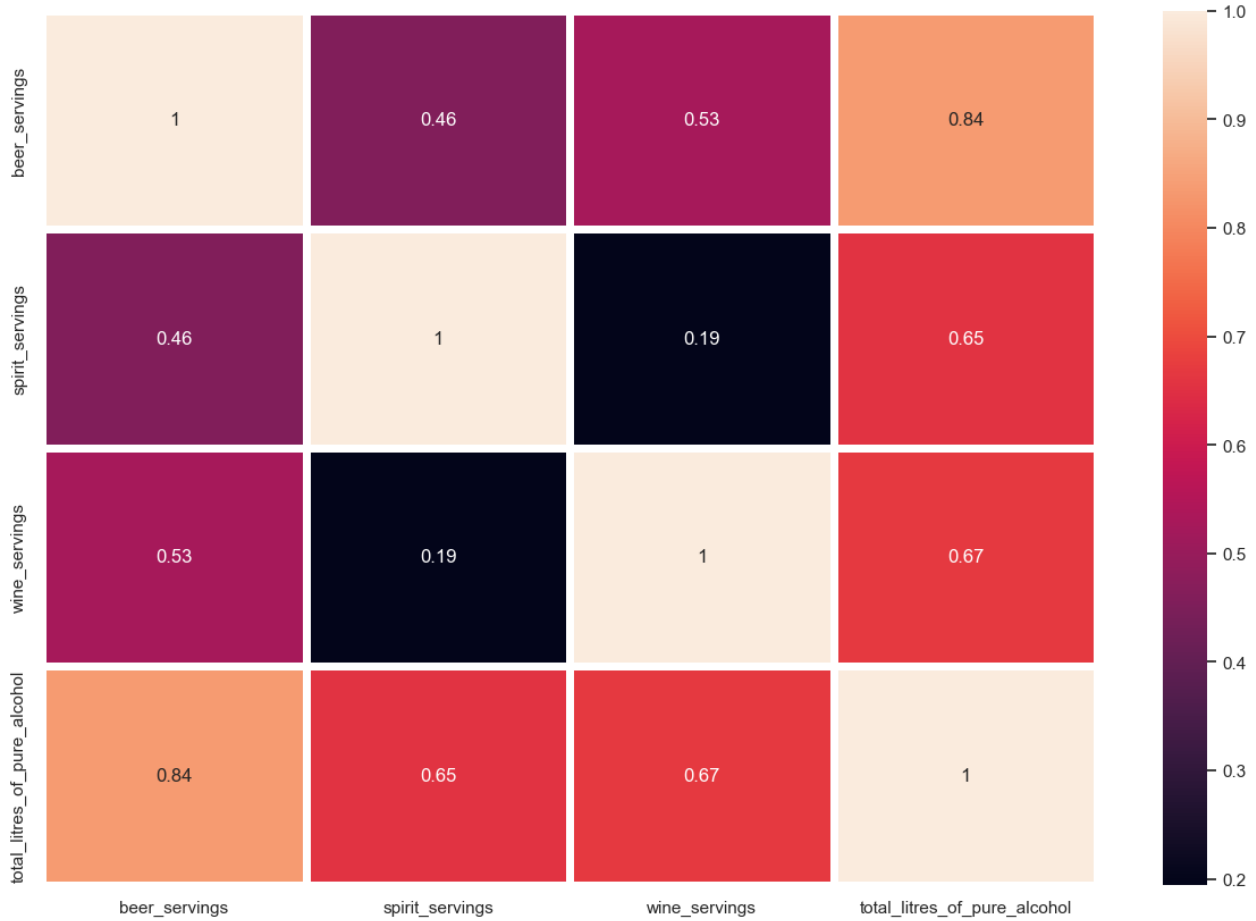
Execution of the code creates three different subplot with three different covariance on the same figure. It allows us to compare graphs easier. The trend of points demonstrates that in every case covariance is positive and it means that if every amount of beer, spirit or wine servings increases, it provoke a growth of total amount of pure litres of alcohol.



There is one more interesting indicator that can tell us much about data. Its name is correlation and it expresses how strong two variables influence on each other. A heatmap can be used to see correlations between different variables. For that, two libraries will be used: plotly and seaborn. But before creating a heatmap, we should use attribute “annot to get numerical expression of correlation.

```
plt.figure(figsize=(15,10))
cor = df.corr()
sns.heatmap(cor, annot=True, linewidths=5)
```

Now a heatmap can be created. It would be also useful to write a numerical correlation in each square for further convenience. Result of execution of abovementioned code can be seen below.



The heatmap demonstrates that the amount of total litres of pure alcohol has the highest correlation with beer servings at the same time spirit and wine servings have almost the same influence on that variable.

2. Visualization

To begin with visualization, it is needed to decide what the visualization is for and what we would like to see and to analyse. The main purpose of visualization is to simplify the data and to make it easier to understand and to analyse. The dataset contains data about alcohol consumption in different countries, there are also columns with amount of different type of alcohol that were consumed. Respectively, it could be created some graphs with, for example, top 10 countries with consumption of every of this types. There is also a column with an amount of total litres of pure alcohol, so it would be sensible to calculate and to demonstrate a percentage of drinks in total alcohol consumption. There are also list of countries with numerical data and it gives us the opportunity to create some maps to visualize higher and lower level of alcohol consumption. Creating graphs and maps, it is important to remember about brightness and intensiveness of colour, weight and size of text.

To begin with, graphs with top 10 countries in different variables will be created. The first one will be a graph that represent countries in which people consume beer the most. For that, it is needed to group and to sort these countries using pandas functions. Result should be assigned to a variable. The following picture demonstrates the code and grouped list of top 10 countries with number of beer servings per person.

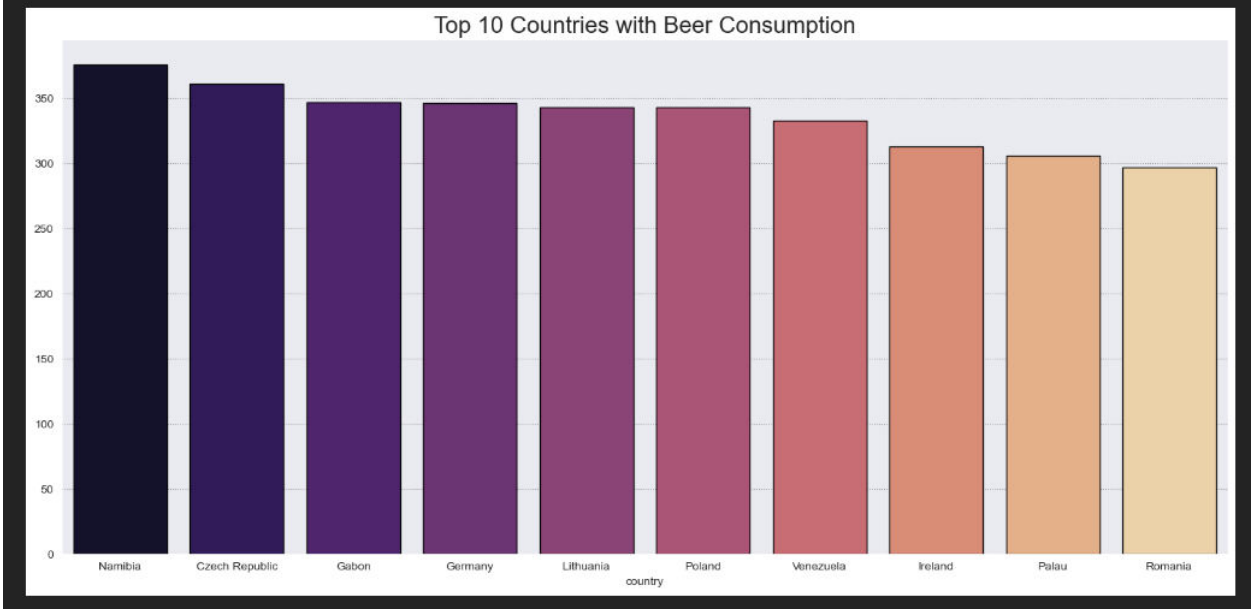
```
top_beer = df.groupby('country')['beer_servings'].sum().sort_values(ascending=False).head(10)
print(top_beer)
```

```
country
Namibia          376
Czech Republic   361
Gabon            347
Germany          346
Lithuania        343
Poland           343
Venezuela        333
Ireland          313
Palau            306
Romania          297
Name: beer_servings, dtype: int64
```

Based on this variable a graph can be built. First of all, a graph should be resized to be more comfortable for reading, then axes should be assigned, it is possible also to choose a colour for graph and a colour for frame line for bars and, finally, title and its size are also set.

After executing the code, the following graph is shown

```
plt.figure(figsize=(18,8))
a = sns.barplot(x=top_beer.index, y=top_beer.values, palette='magma',
                ec='black').set_title('Top 10 Countries with Beer Consumption', fontsize=20)
plt.show()
```



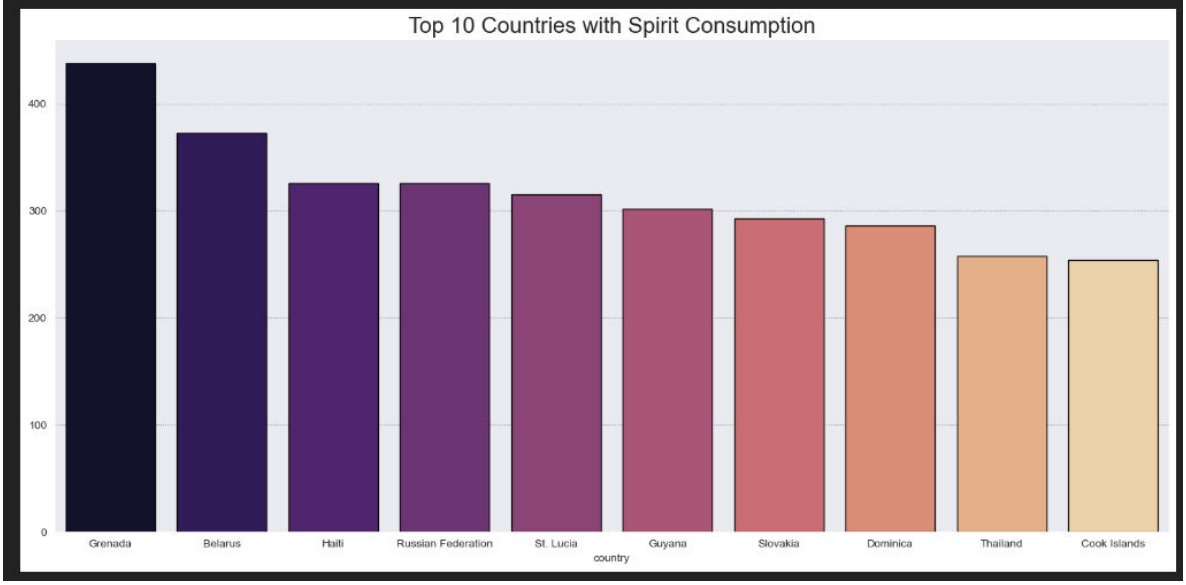
In the same way, top 10 countries with spirit consumption should be also grouped and sorted.

```
top_spirit = df.groupby('country')['spirit_servings'].sum().sort_values(ascending=False).head(10)
print(top_spirit)
```

```
country
Grenada              438
Belarus              373
Haiti                326
Russian Federation   326
St. Lucia            315
Guyana               302
Slovakia             293
Dominica             286
Thailand             258
Cook Islands         254
Name: spirit_servings, dtype: int64
```

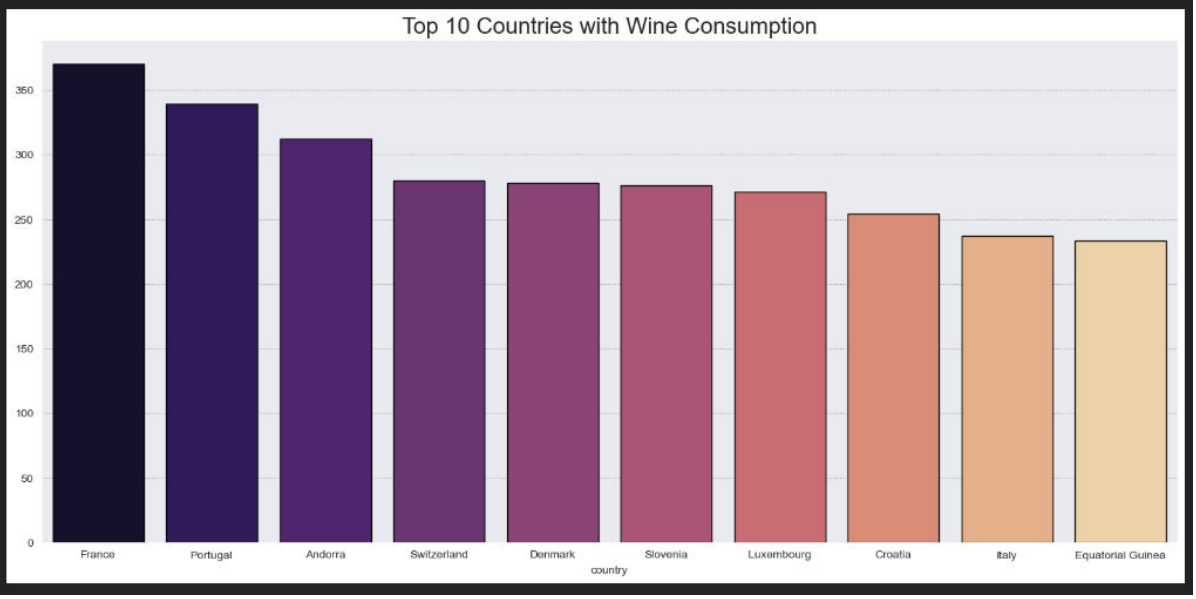
Next, the new variable can be used to create a similar graph but with beer consumption. Firstly, it is needed to resize a figure. Next, we should assign x-axis and y-axis, as well as, colour palette for the next graph and line for bars. Afterward, fuction set_title() creates name and its size for the graph.

```
plt.figure(figsize=(18,8))
sns.barplot(x=top_spirit.index, y=top_spirit.values, palette='magma',
            ec='black').set_title('Top 10 Countries with Spirit Consumption', fontsize=20)
plt.show()
```



Similarly, the code can be used for creation graph with wine consumption. As in previous examples, countries should be grouped and sorted. Next, a graph can be created based on the data.

```
plt.figure(figsize=(18,8))
sns.barplot(x=top_wine.index, y=top_wine.values, palette='magma',
            ec='black').set_title('Top 10 Countries with Wine Consumption', fontsize=20)

plt.show()
```



The dataset contains also information about total litres of pure alcohol per person in every country. That means that this data can be also used to create a graph with top 10 countries with alcohol consumption.

As it has been done for previous examples, at this time it should be also grouped and sorted.

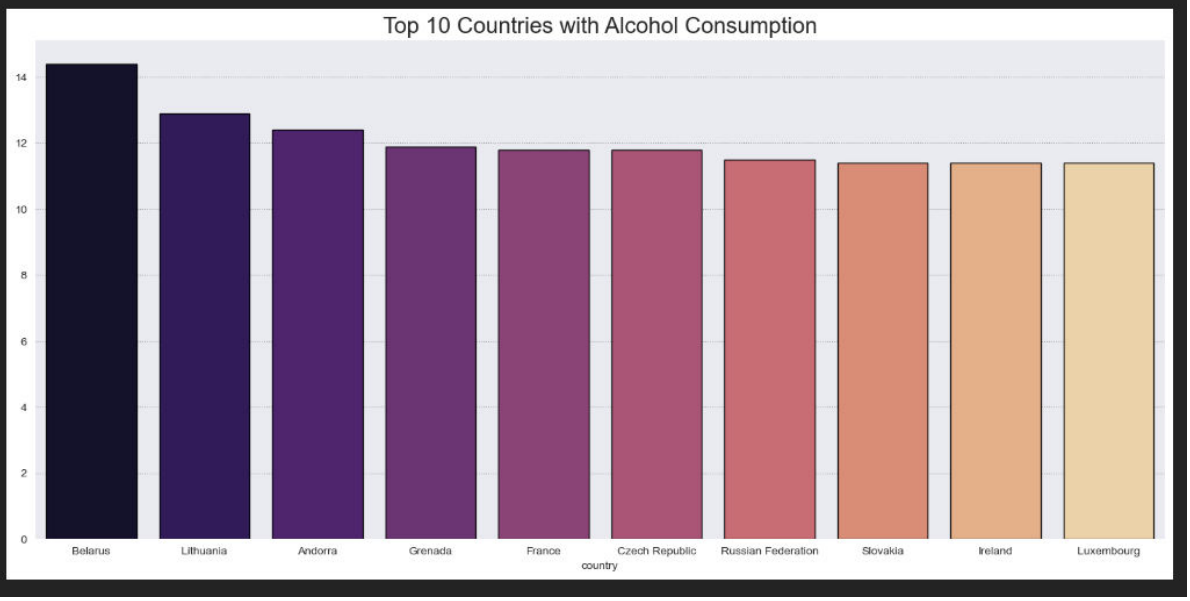
```
top_alco = df.groupby('country')[ \
    'total_litres_of_pure_alcohol'].sum().sort_values(ascending=False).head(10)
print(top_alco)
```

```
country
Belarus                14.4
Lithuania              12.9
Andorra                12.4
Grenada                11.9
France                 11.8
Czech Republic         11.8
Russian Federation     11.5
Slovakia               11.4
Ireland                11.4
Luxembourg             11.4
Name: total_litres_of_pure_alcohol, dtype: float64
```

A new variable can be now used for creation a graph.

```
plt.figure(figsize=(18,8))
sns.barplot(x=top_alco.index, y=top_alco.values, palette='magma',
            ec='black').set_title('Top 10 Countries with Alcohol Consumption', fontsize=20)
```

```
Text(0.5, 1.0, 'Top 10 Countries with Alcohol Consumption')
```



From this graph can be seen, for example, that top three countries with alcohol consumption per person are located in Europa, although beer is especially popular in Africa and spirit in South America. Of course, top 10 countries is now enough to generalize world consumption, but could

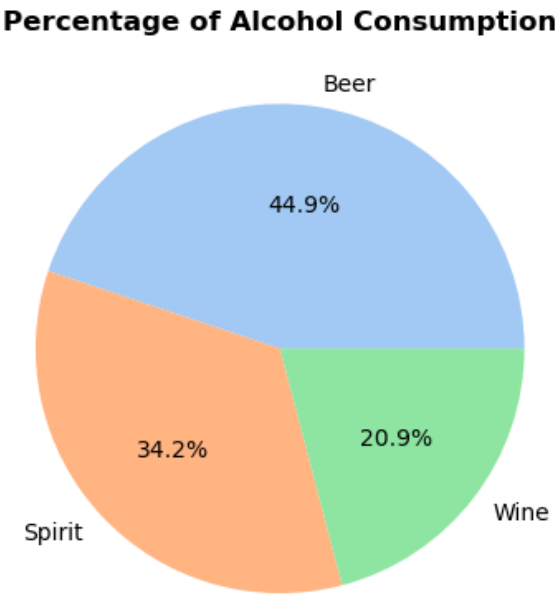
show some trend. This information can be useful if we would like to compare this data with health statistics or would like to research potential of market.

The dataset allows us also to analyse what type of alcohol is more popular in the world. Pie charts are considered as not the best way to demonstrate a percentage, because humans beings understand better length than angles, but for abovementioned purpose and with this type of data, pie chart could be more suitable for presentation percentage.

For that, a sum of all variable in appropriate columns should be calculated, using pandas and function sum(). Then a list of name and list of earlier calculated sums should be created as separate variables. After that a figure can be created and colours, data and labels are needed to be set. It is also sensible to write percentage on the parts of pie chart and a title of a chart for further convenience.

```
sum_of_beer = sum(df['beer_servings'])
sum_of_spirit = sum(df['spirit_servings'])
sum_of_wine = sum(df['wine_servings'])

list_names = ['Beer', 'Spirit', 'Wine']
list = [sum_of_beer, sum_of_spirit, sum_of_wine]
fig = plt.figure()

colors = sns.color_palette('pastel')
plt.pie(list, labels = list_names, autopct='%1.1f%%', colors=colors)
plt.title('Persentage of Alcohol Consumption', weight='bold')

plt.show()
```

After executing these lines of code the following pie chart will be demonstrated.

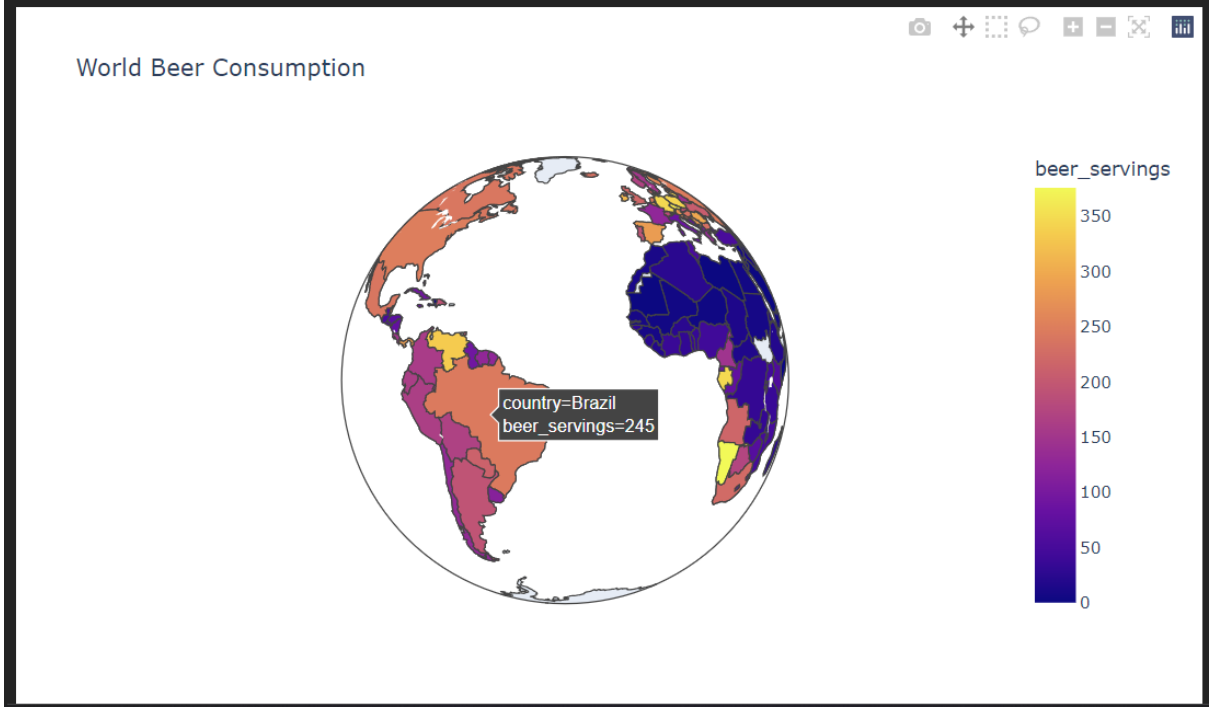


This pie chart is quiet well readable and understandable because it is not overloaded, colours are not too bright and percentage of every type of alcohol is written on the appropriate part of the pie chart.

The last but not least one way to visualize this type of data is creating maps. Maps are perfect for visualisation data which based on countries because it allows not only to compare different countries but also to find easily a country that is interesting for us.

For creation maps will be used the following libraries: plotly.express, plotly.offline and pandas. Plotly.express has a function choropleth() that creates beautiful map, we should only to write our data, locations, choose variable based on which countries will be differ. It is also recommend to change a projection. The code and its result after execution can be seen below on the picture.

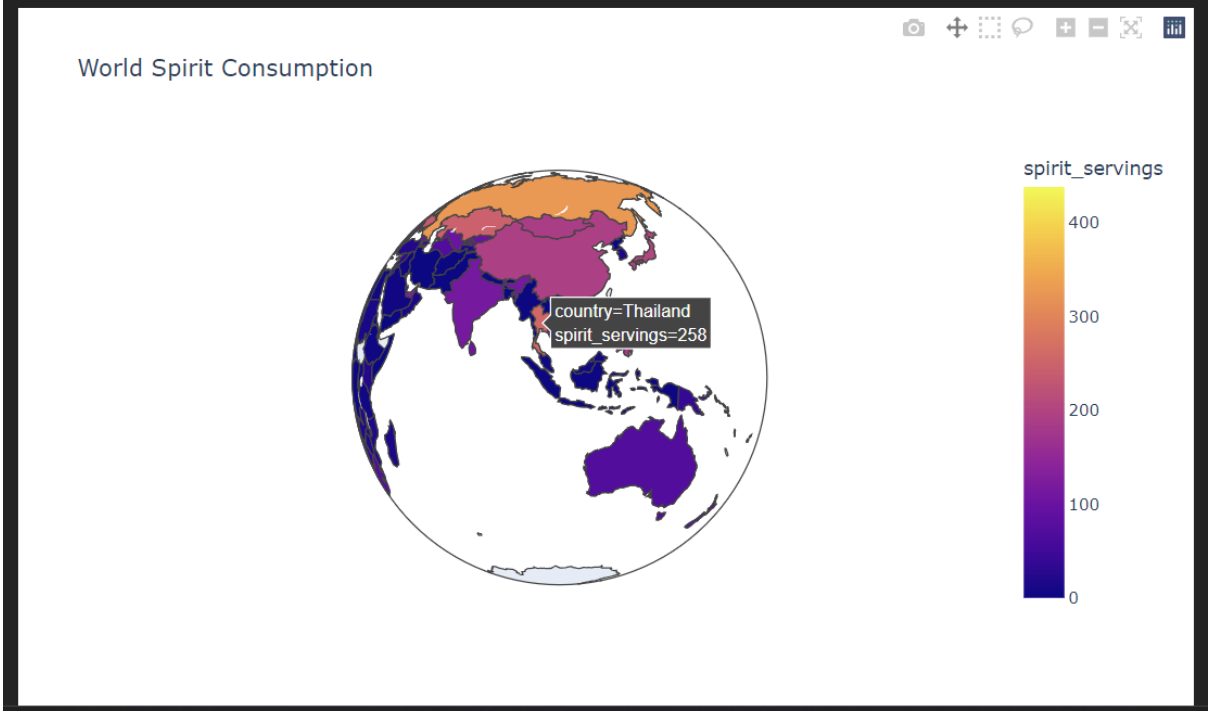
```
fig=pdx.choropleth(data_frame=df, locations='country',locationmode='country names',color='beer_servings',
                   projection='orthographic', title = 'World Beer Consumption')
fig.show()
```



On the picture above can be seen that every country has own colour and its intensiveness dependent on amount of beer servings. In addition to it, user can choose a country to see information exactly about certain country.

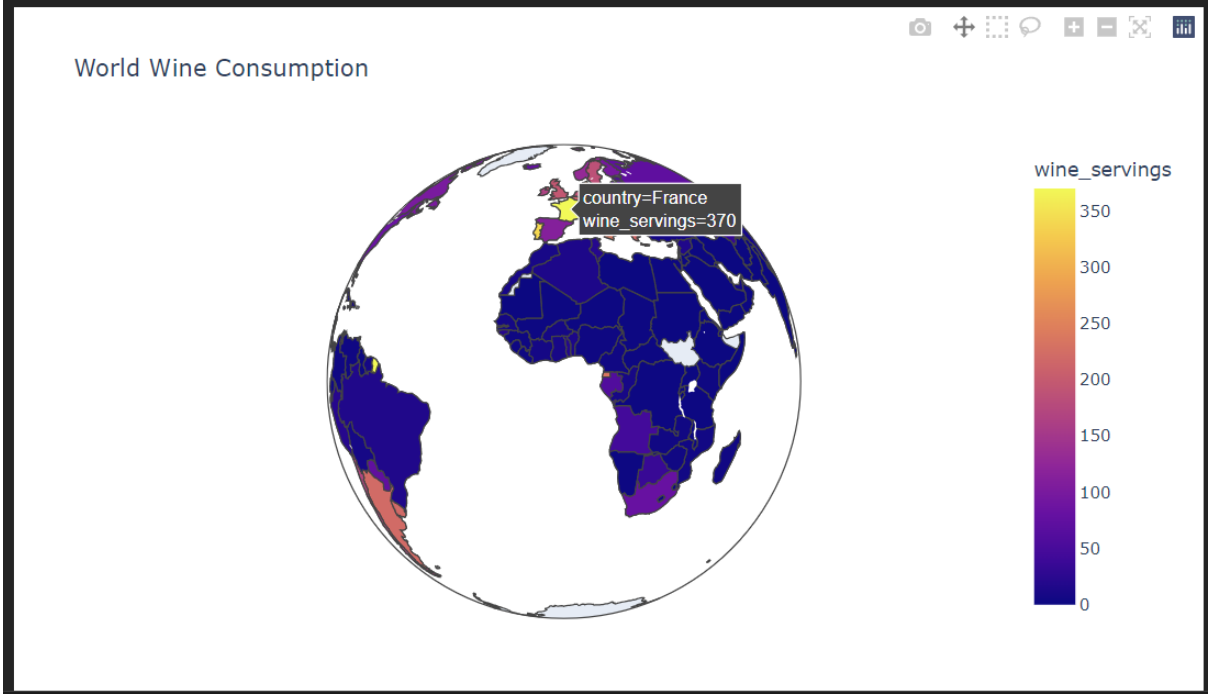
Similarly, it can be created a map for world spirit consumption. We should only change the variable in dataset that will define the colour on the map. On the picture below can be seen the code for creation the second map and its result.

```
fig=pdx.choropleth(data_frame=df, locations='country',locationmode='country names',color='spirit_servings',
                   projection='orthographic', title = 'World Spirit Consumption')
fig.show()
```



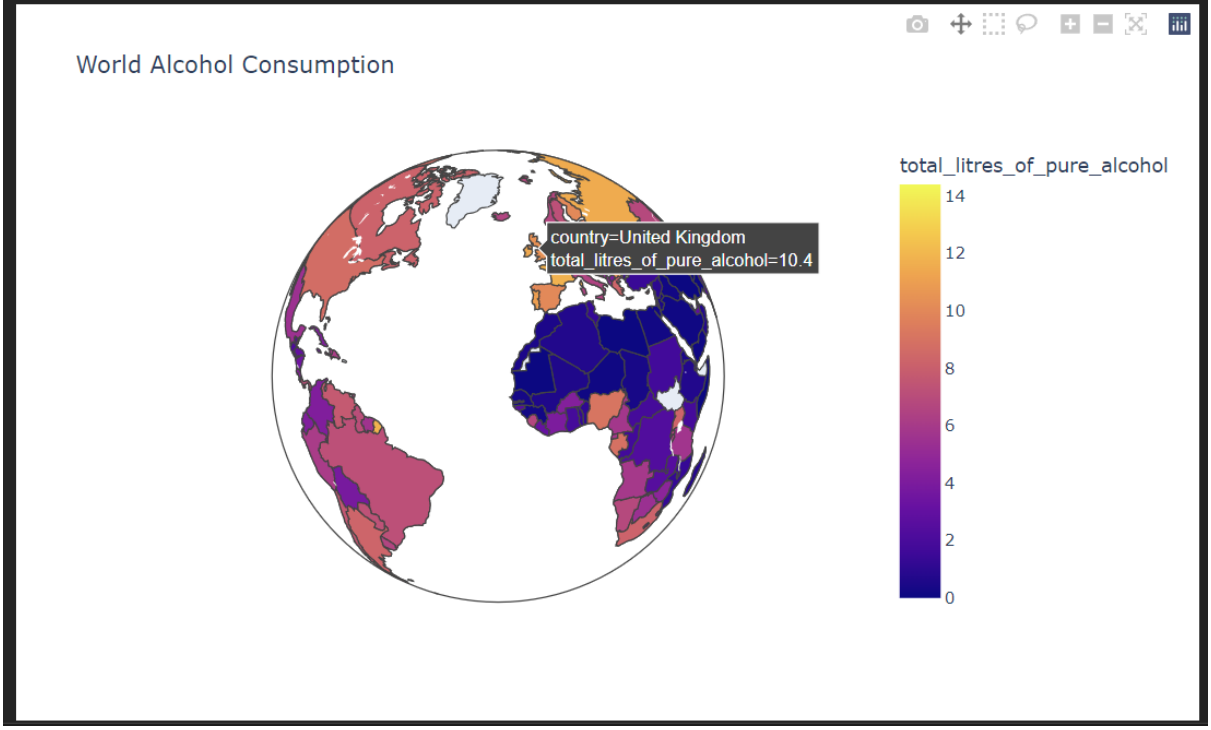
For world wine consumption we should change a data that define a colour of country. The code and its result are demonstrated below on the picture.

```
fig=pdx.choropleth(data_frame=df, locations='country',locationmode='country names',color='wine_servings',
                   projection='orthographic', title = 'World Wine Consumption')
fig.show()
```



The dataset allows us also to compare and research the total alcohol consumption in the world. For that, can be used already familiar code. We should change only data that will be used as colour for countries. Below can be seen the code and its result.

```
fig=pdx.choropleth(data_frame=df, locations='country',locationmode='country names',
                   color='total_litres_of_pure_alcohol',
                   projection='orthographic', title = 'World Alcohol Consumption')
fig.show()
```



Created visualization simplified understanding the data from csv file. It demonstrated a list of country with highest level of consumption every type of alcohol that was presented in researched dataset, it was also created a pie chart that showed the popularity of each type as percentage in the total amount of alcohol consumption. Finally, amount of servings of each type of alcohol as well as an amount of total litres of pure alcohol were presented in map format that facilitated to compare and to analyse consumption on the world map.

Conclusion

Exploratory data analysis was provided to find the most important indicators of data such as mean, median, standard deviation. Based on these indicators were created graphs and plots that showed density, covariance and correlation.

Afterwards, bar plots, pie chart and maps were created to demonstrate comparison between countries by different variables, percentage of alcohol consumption and location of countries which people consume more or less different type of alcohol.

Providing explorative data analysis and creating visualization, it was important to take into consideration pre-attentive attributes, form and function and design principles of creation visualization. It was chosen a dataset that contains data about alcohol consumption and amount of beer, spirit and wine servings. The data, which is contained in the dataset, determined what exactly should have been created and visualized.

There are much more types of graphs and plots that could be demonstrated, for example, violin plots, stacked histogram, overlapping density plot, but according to the dataset it would be not the best way to demonstrate the most important aspects of dataset and, it addition to this, would overload analysis. It could be also have created top 15 or top 20 countries for further analysis, but the number of counties dependent on a task and on a purpose of further analysis. The main purpose of visualization is to facilitate understanding of data, because of that, an approach for visualization should be chosen considerably.

Created visualization could be helpful, for example, by making decision what the market would have more potential for entering the market. Used approach and created graphs and plots could be also useful for researching correlation between people’s health and alcohol consumption not only in the whole world, but also in a certain country.

List of literature

1. Bruce, A., & Bruce, P. (2017*). Practical statistics for data scientists*. O'Reilly Media
2. Vo.T.H, P., Czygan, M., Kumar, A., & Raman, K. (2017). *Python: Data Analytics and Visualization* (1st ed.). Packt Publishing.
3. Wilke, C. O. (2019). *Fundamentals of data visualization*. O'Reilly Media