Visualization of Massive Data

03.03.2022

Ka Po Chau (Selena)
Epitech Project

Overview

This project is done by Ka Po Chau, for the Epitech module Visualization of Massive Data.

There are 4 tasks:

1. Artificial dataset generation
2. Dataset analysis
3. Dataset visualization
4. Quantitative analysis (I chose supervised learning.)

Artificial dataset generation

The file *artificial_dataset.py* contains the first task.

It generates 300 data points with 6 columns. This is the fake data of a normal person.

Name	Mean	STD	Type	Correlation
Age	25	5	integer	
Salary	4000	1000	integer	- with health
Love	0.8	0.5	integer	
Pets	2.5	1	integer	
Health	24	3	float	- with salary
Happiness	-	-	float	random generated from Love and Pets

In [2]:
```
import numpy as np
import pandas as pd

n = 300
data = {
        'Age': np.random.normal(25, 5, n).astype(int),
        'Salary': np.random.normal(4000, 1000, n).astype(int),
        'Love': np.random.normal(0.8, 0.5, n).astype(int),
        'Pets': np.random.normal(2.5, 1, n).astype(int),
        'Health': np.random.normal(24, 3, n),
       }

health = (10000 - (data['Salary']) - (10 * np.random.randn(n) + 50)) / 100    #health negative correlate with salary
happiness = (data['Love'] + data['Pets']) * (10 * np.random.randn(n) + 50)    #happiness correlate with Love and Pets

data['Health'] = health
data['Happiness'] = happiness

df = pd.DataFrame(data=data)
df
```

Out[2]:

	Age	Salary	Love	Pets	Health	Happiness
0	27	4370	0	3	55.922706	116.983234
1	23	2616	0	2	73.523259	106.525839
2	25	5488	1	2	44.580976	129.068220
3	19	3909	1	2	60.346671	173.193555
4	23	3439	1	1	65.139780	124.253195
...	...	...	...	...	...	...
295	24	3869	0	1	60.943686	39.492149
296	25	2617	0	2	73.525158	118.743647
297	23	4049	1	2	58.938685	143.085840
298	30	2423	0	1	75.162124	39.915500
299	20	5769	0	1	41.756433	50.703032

300 rows × 6 columns

Example output

Dataset

I have found a dataset via the links, this dataset is about countries.

This dataset has 44 columns of information about 263 data points of countries.

```
import matplotlib.pyplot as plt
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

df = pd.read_csv('data.csv', sep = ';')
df = df.drop(0)  # remove useless type data line

my_columns = df.columns.to_list()[1:45]
for i in my_columns:
   df[i] = df[i].astype('float')   # float type for all except country name

df.info()
print(df.head())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 263 entries, 1 to 263
Data columns (total 45 columns):
 #   Column                                      Non-Null Count  Dtype
---  ------                                      --------------  -----
 0   Country                                     263 non-null    object
 1   Area(sq km)                                 263 non-null    float64
 2   Birth rate(births/1000 population)          225 non-null    float64
 3   Current account balance                     149 non-null    float64
 4   Death rate(deaths/1000 population)          225 non-null    float64
 5   Debt - external                             201 non-null    float64
 6   Electricity - consumption(kWh)              215 non-null    float64
 7   Electricity - production(kWh)               213 non-null    float64
 8   Exports                                     224 non-null    float64
 9   GDP                                         230 non-null    float64
 10  GDP - per capita                            230 non-null    float64
 11  GDP - real growth rate(%)                   212 non-null    float64
 12  HIV/AIDS - adult prevalence rate(%)         168 non-null    float64
 13  HIV/AIDS - deaths                           148 non-null    float64
```

Analysis

In the *analysis.py*, we will see the quantitative variables “Country Area” and “GDP per capita”, and find out its important aspects, such as maximum, minimum, mean and standard deviation.

```
area = df["Area(sq km)"].dropna()
print("\n===============\nCountry Area\n===============\n")
print("Maximum:", area.max() )
print("Minimum:", area.min() )
def my_mean(x):
    return np.average(x, weights=np.ones_like(x) / x.size)

print("Mean:", area.mean() )
print("Std:", area.std() )

gdp_per_capita = df["GDP - per capita"].dropna()
print("\n===============\nGDP - per capita\n===============\n")
print("Maximum:", gdp_per_capita.max() )
print("Minimum:", gdp_per_capita.min() )
def my_mean(x):
    return np.average(x, weights=np.ones_like(x) / x.size)

print("Mean:", gdp_per_capita.mean() )
print("Std:", gdp_per_capita.std() )
```

```
===============
Country Area
===============

Maximum: 17075200.0
Minimum: 0.0
Mean: 584987.4866920152
Std: 1881415.5467778272

===============
GDP - per capita
===============

Maximum: 58900.0
Minimum: 400.0
Mean: 10552.760869565218
Std: 11104.610351385776
```

Visualization 1

In the first visualization *visualization_1.py*, the user can choose with the console input between 4 quantitative variables:

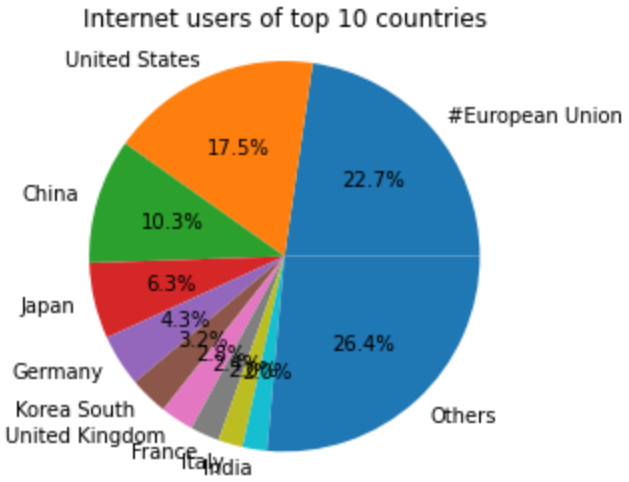
Country Size, GDP, Population and Number of Internet Users.

Default is Country Size.

After that, the optional parameter is how many countries to show. Default is 10.

The following is an example.

```
Please enter type of data to visualize: (enter a number 1-4)
1) country size (sq km) (default)
2) GDP
3) Population
4) Internet users
4
Please enter how many countries to show: (enter a number, default 10)
10
Internet users
```

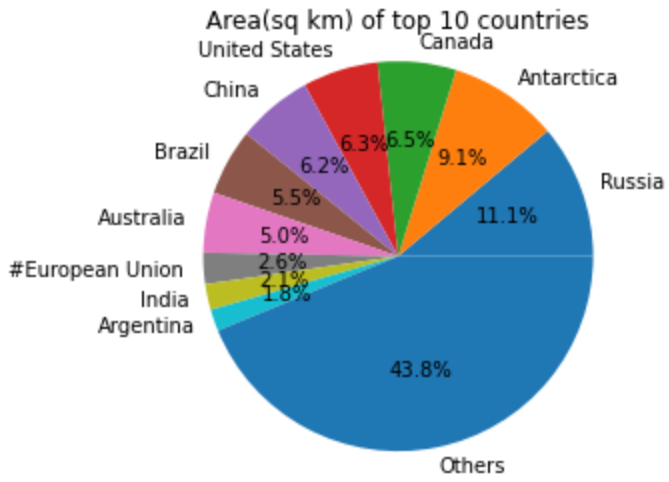


After the user enter 4 for Internet Users and 10 for 10 countries, a pie chart shows the percentage of Internet Users and where they are from.

We can see that the European Union has the most represented Internet Users, at 22.7% of all, followed by the US and then China.

The pie chart shows the 10 countries as the optional parameter. All other countries are grouped up as “Others”.

```
Please enter type of data to visualize: (enter a number 1-4)
1) country size (sq km) (default)
2) GDP
3) Population
4) Internet users
1
Please enter how many countries to show: (enter a number, default 10)
10
Area(sq km)
```



In this example, we can see the Country Size of all countries and how many percentage of the Earth's land that they occupy.

We can see that Russia is the largest, followed by Antarctica, Canada, US and China.

All other countries are grouped up into "Others". We can see that despite Russia being the largest, it is still only 11.1% of the Earth lands and small compared to the other countries combined.

Visualization 2

In the second visualization *visualization_2.py*, the user can choose with the console input between 4 quantitative variables:

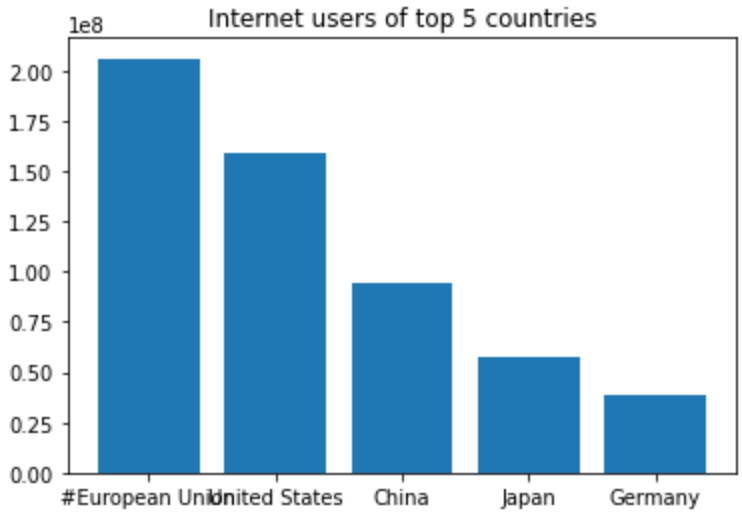
Country Size, GDP, Population and Number of Internet Users.

Default is Country Size.

After that, the optional parameter is how many countries to show. Default is 10.

The following is an example.

```
Please enter type of data to visualize: (enter a number 1-4)
1) country size (sq km) (default)
2) GDP
3) Population
4) Internet users
4
Please enter how many countries to show: (enter a number, default 10)
5
Internet users
```



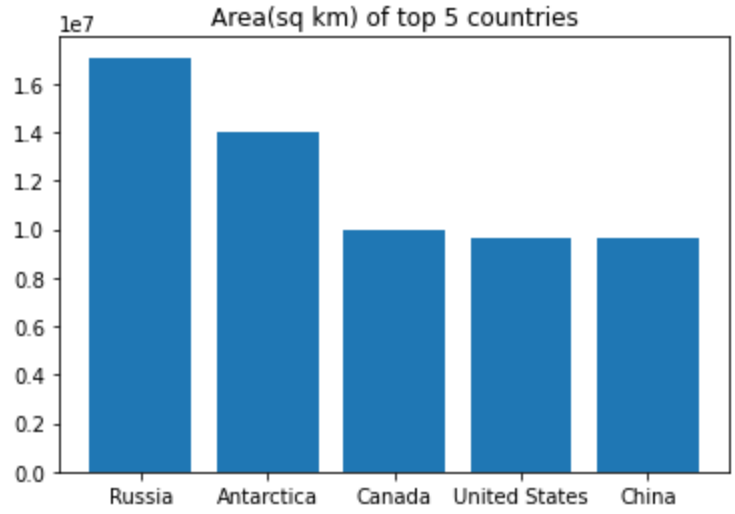
After the user enter 4 for Internet Users and 5 for 5 countries, a bar graph shows the percentage of Internet Users and where they are from.

We can see that the European Union has the most Internet Users, followed by the US, China, Japan and Germany.

The bar graph shows the 5 countries as the optional parameter. There is no “Others” option like for the pie chart, since the percentage of each country relative to overall is not

important in bar graphs like pie charts. The EU is about 4 times of Germany's number, which is interesting because the EU includes Germany, but the dataset contains both.

```
Please enter type of data to visualize: (enter a number 1-4)
1) country size (sq km) (default)
2) GDP
3) Population
4) Internet users
1
Please enter how many countries to show: (enter a number, default 10)
5
Area(sq km)
```



In this example, we can see the Country Size of the top 5 countries chosen.

We can see that Russia is the largest, followed by Antarctica, Canada, US and China.

Again, there is no "Others" category. We can see that Russia is the largest by a large margin, and then it is Antarctica. Canada, the US and China are almost the same size.

Quantitative analysis

For this task, I chose to do supervised learning, and used a simple linear regression model, to find out the correlation between Number of Internet Users and GDP per capita.

```
|: import numpy as np
   import pandas as pd
   from sklearn.linear_model import LinearRegression

   df = pd.read_csv('data.csv', sep = ';')
   df = df.drop(0)  # remove useless type data line

   my_columns = df.columns.to_list()[1:45]
   for i in my_columns:
      df[i] = df[i].astype('float')   # float type for all except country name

   # df.info()
   # print(df.head())


   df = df.dropna(subset=['GDP - per capita']).dropna(subset=['Internet users'])
   estimator = LinearRegression()

   X = df['GDP - per capita'].values.reshape(-1, 1)
   y = df['Internet users'].values.reshape(-1, 1)
   # Fit inputs to outputs
   estimator.fit(X, y)

   print(f"estimator score: {estimator.score(X, y)}")
   print(f"etimator coefficient: {estimator.coef_}")
   print(f"estimator b: {estimator.intercept_}")
```

```
estimator score: 0.07222072470042007
etimator coefficient: [[481.31867031]]
estimator b: [-774933.6031649]
```

We can see that there is a positive correlation between the number of Internet Users and the GDP per capita.

In this task, I used the third party library sklearn to do a simple linear regression and find out the coefficients with an estimator.