Ray_Yadav_DSI_E08_ST24

June 19, 2024

1 DSI Lecture - Exercise 08

Author: Philipp Wieder **Date**: 13/06/2024 **Content**: Exercise 08 of the Data Science Infrastructure lecture (ST 24) **Software needed**: Depends **Files**: *internet.csv*, *world-happiness.csv* (from Stud.IP)

1.1 Assignment I

Points: 3

In this assigment, we compare different representations of the same data. Use the *internet.csv* file from Stud.IP. It contains the development of internet usage in different continents from 1990 to 2019.

1.1.1 a) Evaluate data

Have a quick look at the table, the data records, and the different features. Is there anything in particular to consider with respect to data cleaning? If not, why?

[123]:

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv('internet.csv')
df
```

[123]:

```
          geo          name  time  Internet users (%)  Internet users  \
0      africa        Africa  1990                0.00             0.0
1      africa        Africa  1991                0.00          5010.0
2      africa        Africa  1992                0.00         15032.0
3      africa        Africa  1993                0.01         45667.0
4      africa        Africa  1994                0.02        105625.0
..        …             …    …                  …              …
115  americas  The Americas  2015               62.44     609314311.0
116  americas  The Americas  2016               67.81     667622700.0
117  americas  The Americas  2017               72.09     715968579.0
118  americas  The Americas  2018               72.76     728794052.0
119  americas  The Americas  2019               73.82     745506112.0

     Non-internet users  Non Internet users
```

```
0                  100.0         623927790.0
1                  100.0         640965311.0
2                  100.0         658212716.0
3                  100.0         675633003.0
4                  100.0         693196818.0
..                   …                 …
115                 38.0         366475365.0
116                 32.0         316960084.0
117                 28.0         277228904.0
118                 27.0         272852819.0
119                 26.0         264444018.0

[120 rows x 7 columns]
```

[114]:

```python
print(df.isnull().sum().sum())
print(df['geo'].unique())
print(df['name'].unique())
```

```
0
['africa' 'asia' 'europe' 'americas']
['Africa' 'Asia' 'Europe' 'The Americas']
```

In the table above, we can see that the rows and columns are properly aligned, and all elements are present. After checking for null elements, we can confirm that there are zero null elements in the table. However, the data contains several redundant columns. The 'Geo' and 'Name' columns are redundant, and the columns for 'Internet Users' and 'Non-Users' can be simplified by categorizing them into two separate categories 'Internet Users (%)' and 'Total Population'. Also, the calculation of the percentage of internet users seems to be incorrect. Therefore, data cleaning is required for these features.

1.1.2 b) Create data representations

Create three different plots representing the the data. You can use the programming language (and module/library)/tool of your choice. In case you are not familiar with any programming language, just use a program that can handle spreadsheets (like Excel, OpenOffice Calc, or alike).

Which representation do you think fits best?

Resources:s * From Data to Viz * Slides from Lecture 04 (Data Preprocessing) and Lecture 08 (Data Visualization)

Since the above data is untidy, I first cleaned the data as shown below and visualized it into three different ways. As I am familiar with Python, I will be using it throughout the exercise.

[115]:

```python
#Data Cleaning at first.
continents = df['geo']
total_population = df['Internet users'] + df['Non Internet users']
internet_users_percentage = df['Internet users'] / total_population * 100
new_df = pd.DataFrame({'continents': continents,
                       'time': df['time'],
                      'total_population': total_population,
                      'internet_users_percentage': internet_users_percentage})
new_df
```

[115]:

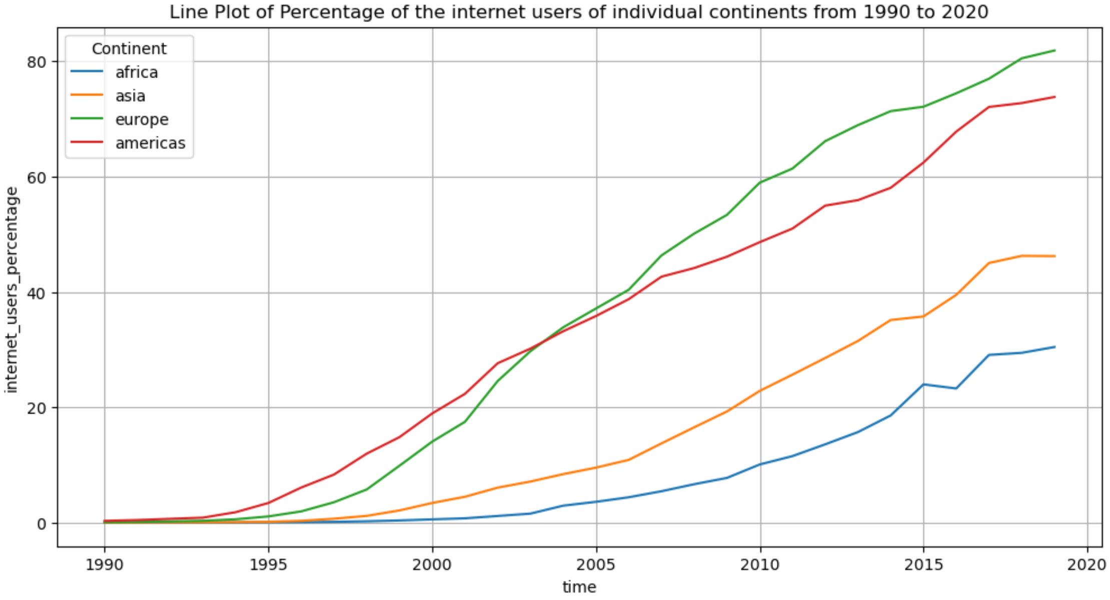
```
    continents  time  total_population  internet_users_percentage
0       africa  1990      6.239278e+08                   0.000000
1       africa  1991      6.409703e+08                   0.000782
2       africa  1992      6.582277e+08                   0.002284
3       africa  1993      6.756787e+08                   0.006759
4       africa  1994      6.933024e+08                   0.015235
..         …     …               …                          …
115   americas  2015      9.757897e+08                  62.443201
116   americas  2016      9.845828e+08                  67.807676
117   americas  2017      9.931975e+08                  72.087233
118   americas  2018      1.001647e+09                  72.759580
119   americas  2019      1.009950e+09                  73.816131

[120 rows x 4 columns]
```

[122]:

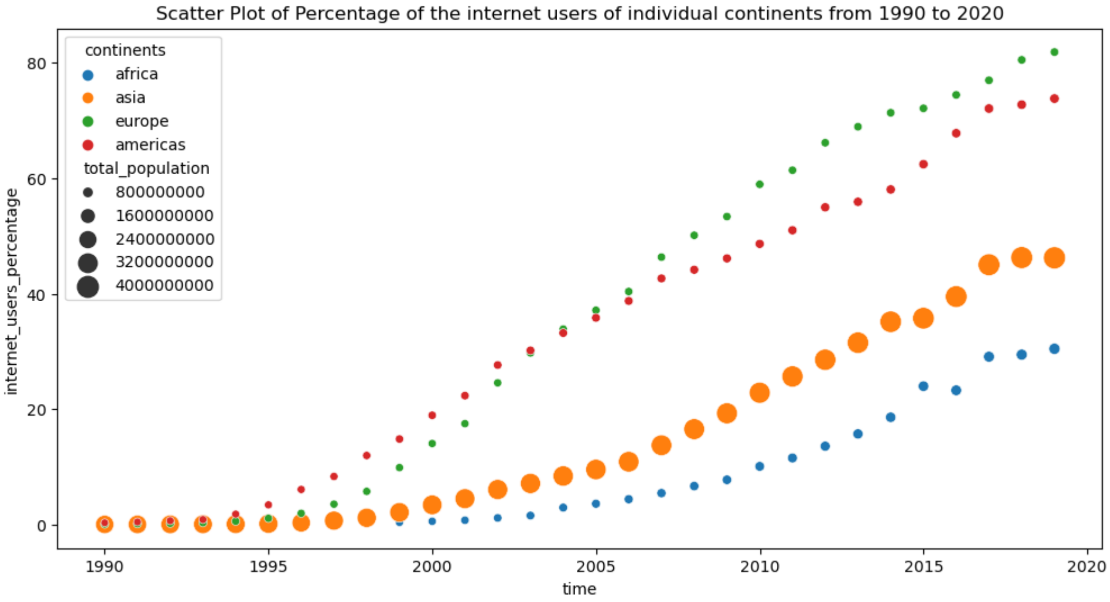
```
#Plot 1: Plotting the chart
plt.figure(figsize = (12,6))
sns.lineplot(data = new_df, x = new_df['time'], y =␣
 ↪new_df['internet_users_percentage'], hue = new_df['continents'])
plt.title("Line Plot of Percentage of the internet users of individual␣
 ↪continents from 1990 to 2020")
plt.legend(title = 'Continent')
plt.grid()
```

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



```
[117]: #Plot 2: Scatter Plot.
       plt.figure(figsize = (12, 6))
       sns.scatterplot(data = new_df, x = new_df['time'], y =␣
        ↪new_df['internet_users_percentage'], hue = 'continents', size =␣
        ↪'total_population', sizes =(20, 200))
       plt.title("Scatter Plot of Percentage of the internet users of individual␣
        ↪continents from 1990 to 2020")
```

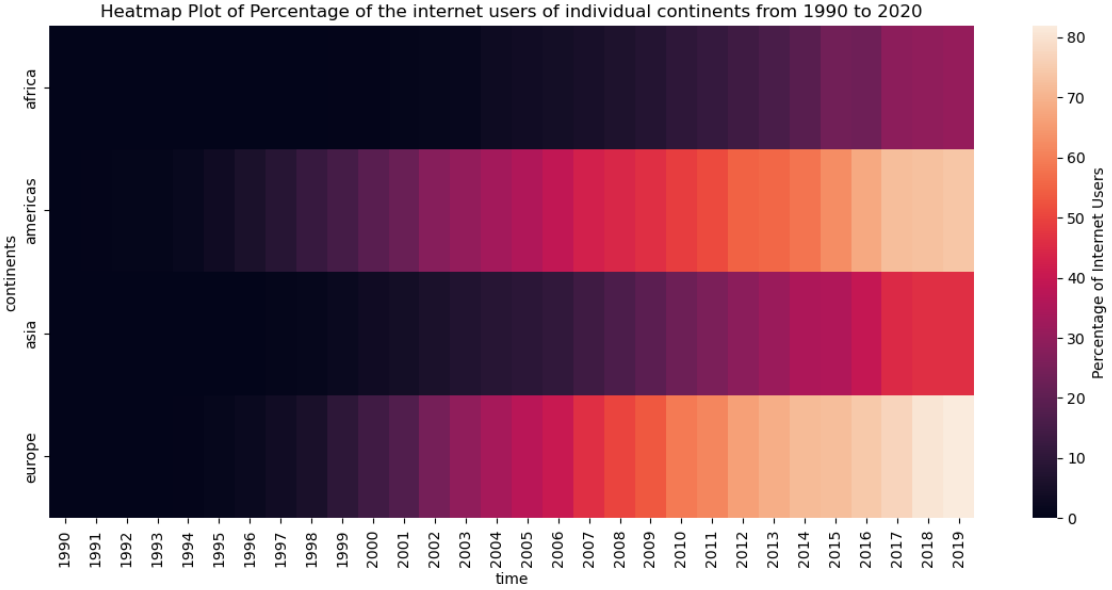
```
[117]: Text(0.5, 1.0, 'Scatter Plot of Percentage of the internet users of individual
       continents from 1990 to 2020')
```



[118]:
```
#Plot 3: Heatmap Plot
heatmap_data = new_df.pivot_table(index = 'continents', columns = 'time',␣
 ↪values = 'internet_users_percentage')

plt.figure(figsize = (14,6))
sns.heatmap(heatmap_data, cbar_kws = {'label': 'Percentage of Internet Users'})
plt.title("Heatmap Plot of Percentage of the internet users of individual␣
 ↪continents from 1990 to 2020")
```

[118]:
```
Text(0.5, 1.0, 'Heatmap Plot of Percentage of the internet users of individual
continents from 1990 to 2020')
```



1.1.3 Assignment II

Points: 2

In this assignment, you are free to choose the Python module for visualization and the representation. The data source is the file *world-happiness.csv* from Stud.IP (Please note: the delimiter is a semicolon, NOT a comma). The file contains data about the happiness of countries according to the so called *Happiness Score*.

Please represent the *Happiness Score* of all countries in the file for the year 2018 only. Choose the most suitable representation from your point of view and explain your choice.

Answer: After slicing the data for the year 2018, the only important columns that we are interested in are the country name and their corresponding happiness index. Therefore, a simple bar plot would be enough to represent the *Happiness Score* of all countries for the year 2018 only.

All of the process for the data representation is shown below.

```
[119]: df_happy = pd.read_csv('world-happiness.csv', delimiter = ';')
print(df_happy)
print(df_happy['geo'].unique())
print(df_happy['name'].unique())
print(df_happy['Happiness score (WHR)'].max())
print(df_happy['Happiness score (WHR)'].min())
```

```
      geo         name  time  Happiness score (WHR)
0     afg  Afghanistan  2008                     37
1     afg  Afghanistan  2009                     44
2     afg  Afghanistan  2010                     48
3     afg  Afghanistan  2011                     38
```

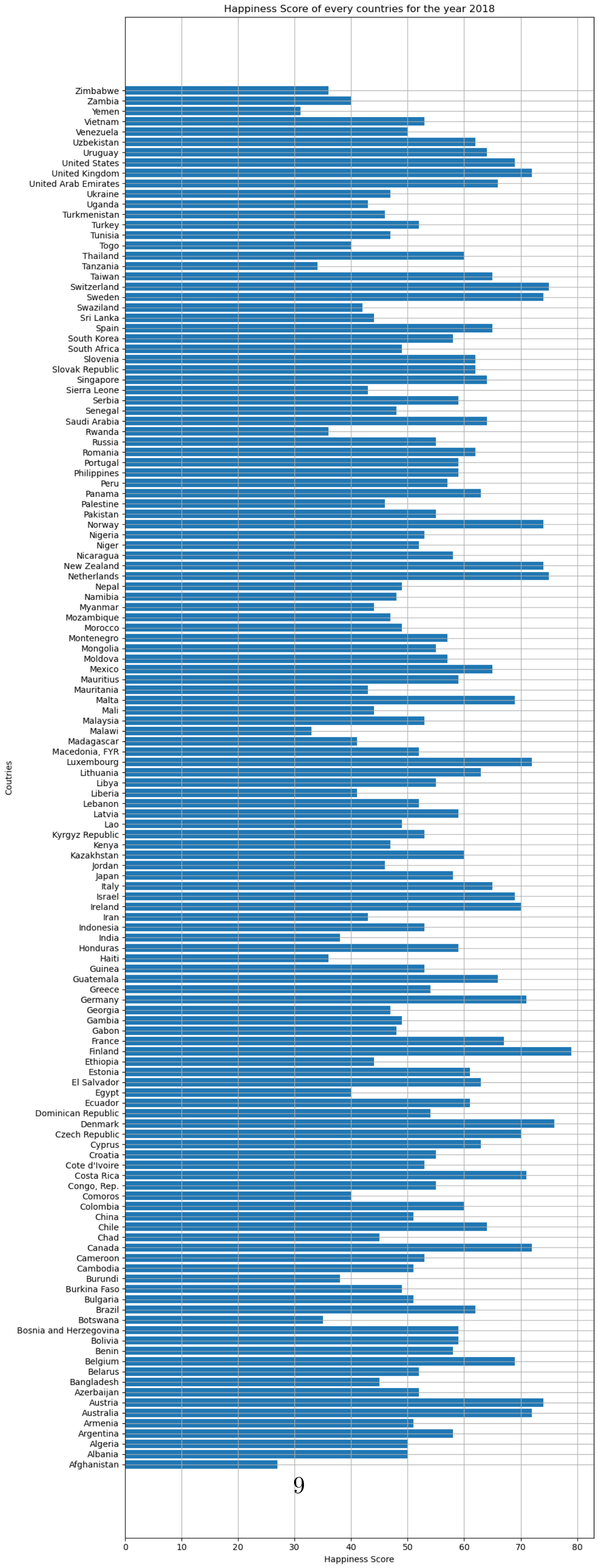
```
4     afg  Afghanistan  2012                     38
…   …          …   …                    …
1832  zwe     Zimbabwe  2015                     37
1833  zwe     Zimbabwe  2016                     37
1834  zwe     Zimbabwe  2017                     36
1835  zwe     Zimbabwe  2018                     36
1836  zwe     Zimbabwe  2019                     33

[1837 rows x 4 columns]
['afg' 'alb' 'dza' 'ago' 'arg' 'arm' 'aus' 'aut' 'aze' 'bhr' 'bgd' 'blr'
 'bel' 'blz' 'ben' 'btn' 'bol' 'bih' 'bwa' 'bra' 'bgr' 'bfa' 'bdi' 'khm'
 'cmr' 'can' 'caf' 'tcd' 'chl' 'chn' 'col' 'com' 'cod' 'cog' 'cri' 'civ'
 'hrv' 'cub' 'cyp' 'cze' 'dnk' 'dji' 'dom' 'ecu' 'egy' 'slv' 'est' 'eth'
 'fin' 'fra' 'gab' 'gmb' 'geo' 'deu' 'gha' 'grc' 'gtm' 'gin' 'guy' 'hti'
 'hnd' 'hkg' 'hun' 'isl' 'ind' 'idn' 'irn' 'irq' 'irl' 'isr' 'ita' 'jam'
 'jpn' 'jor' 'kaz' 'ken' 'kwt' 'kgz' 'lao' 'lva' 'lbn' 'lso' 'lbr' 'lby'
 'ltu' 'lux' 'mkd' 'mdg' 'mwi' 'mys' 'mdv' 'mli' 'mlt' 'mrt' 'mus' 'mex'
 'mda' 'mng' 'mne' 'mar' 'moz' 'mmr' 'nam' 'npl' 'nld' 'nzl' 'nic' 'ner'
 'nga' 'nor' 'omn' 'pak' 'pse' 'pan' 'pry' 'per' 'phl' 'pol' 'prt' 'qat'
 'rou' 'rus' 'rwa' 'sau' 'sen' 'srb' 'sle' 'sgp' 'svk' 'svn' 'som' 'zaf'
 'kor' 'ssd' 'esp' 'lka' 'sdn' 'sur' 'swz' 'swe' 'che' 'syr' 'twn' 'tjk'
 'tza' 'tha' 'tgo' 'tto' 'tun' 'tur' 'tkm' 'uga' 'ukr' 'are' 'gbr' 'usa'
 'ury' 'uzb' 'ven' 'vnm' 'yem' 'zmb' 'zwe']
['Afghanistan' 'Albania' 'Algeria' 'Angola' 'Argentina' 'Armenia'
 'Australia' 'Austria' 'Azerbaijan' 'Bahrain' 'Bangladesh' 'Belarus'
 'Belgium' 'Belize' 'Benin' 'Bhutan' 'Bolivia' 'Bosnia and Herzegovina'
 'Botswana' 'Brazil' 'Bulgaria' 'Burkina Faso' 'Burundi' 'Cambodia'
 'Cameroon' 'Canada' 'Central African Republic' 'Chad' 'Chile' 'China'
 'Colombia' 'Comoros' 'Congo, Dem. Rep.' 'Congo, Rep.' 'Costa Rica'
 "Cote d'Ivoire" 'Croatia' 'Cuba' 'Cyprus' 'Czech Republic' 'Denmark'
 'Djibouti' 'Dominican Republic' 'Ecuador' 'Egypt' 'El Salvador' 'Estonia'
 'Ethiopia' 'Finland' 'France' 'Gabon' 'Gambia' 'Georgia' 'Germany'
 'Ghana' 'Greece' 'Guatemala' 'Guinea' 'Guyana' 'Haiti' 'Honduras'
 'Hong Kong, China' 'Hungary' 'Iceland' 'India' 'Indonesia' 'Iran' 'Iraq'
 'Ireland' 'Israel' 'Italy' 'Jamaica' 'Japan' 'Jordan' 'Kazakhstan'
 'Kenya' 'Kuwait' 'Kyrgyz Republic' 'Lao' 'Latvia' 'Lebanon' 'Lesotho'
 'Liberia' 'Libya' 'Lithuania' 'Luxembourg' 'Macedonia, FYR' 'Madagascar'
 'Malawi' 'Malaysia' 'Maldives' 'Mali' 'Malta' 'Mauritania' 'Mauritius'
 'Mexico' 'Moldova' 'Mongolia' 'Montenegro' 'Morocco' 'Mozambique'
 'Myanmar' 'Namibia' 'Nepal' 'Netherlands' 'New Zealand' 'Nicaragua'
 'Niger' 'Nigeria' 'Norway' 'Oman' 'Pakistan' 'Palestine' 'Panama'
 'Paraguay' 'Peru' 'Philippines' 'Poland' 'Portugal' 'Qatar' 'Romania'
 'Russia' 'Rwanda' 'Saudi Arabia' 'Senegal' 'Serbia' 'Sierra Leone'
 'Singapore' 'Slovak Republic' 'Slovenia' 'Somalia' 'South Africa'
 'South Korea' 'South Sudan' 'Spain' 'Sri Lanka' 'Sudan' 'Suriname'
 'Swaziland' 'Sweden' 'Switzerland' 'Syria' 'Taiwan' 'Tajikistan'
 'Tanzania' 'Thailand' 'Togo' 'Trinidad and Tobago' 'Tunisia' 'Turkey'
 'Turkmenistan' 'Uganda' 'Ukraine' 'United Arab Emirates' 'United Kingdom'
 'United States' 'Uruguay' 'Uzbekistan' 'Venezuela' 'Vietnam' 'Yemen'
 'Zambia' 'Zimbabwe']
80
26
```

[126]:

```
df_2018 = df_happy[df_happy['time'] == 2018]

plt.figure(figsize = (10, 26))
plt.barh(df_2018['name'], df_2018['Happiness score (WHR)'])
plt.title("Happiness Score of every countries for the year 2018")
plt.xlabel("Happiness Score")
plt.ylabel("Coutries")
plt.grid()
plt.tight_layout()
```


Happiness Score of every countries for the year 2018
Zimbabwe
Zambia
Yemen
Vietnam
Venezuela
Uzbekistan
Uruguay
United States
United Kingdom
United Arab Emirates
Ukraine
Uganda
Turkmenistan
Turkey
Tunisia
Togo
Thailand
Tanzania
Taiwan
Switzerland
Sweden
Swaziland
Sri Lanka
Spain
South Korea
South Africa
Slovenia
Slovak Republic
Singapore
Sierra Leone
Serbia
Senegal
Saudi Arabia
Rwanda
Russia
Romania
Portugal
Philippines
Peru
Panama
Palestine
Pakistan
Norway
Nigeria
Niger
Nicaragua
New Zealand
Netherlands
Nepal
Namibia
Myanmar
Mozambique
Morocco
Montenegro
Mongolia
Moldova
Mexico
Mauritius
Mauritania
Malta
Mali
Malaysia
Malawi
Madagascar
Macedonia, FYR
Luxembourg
Lithuania
Libya
Liberia
Lebanon
Latvia
Lao
Kyrgyz Republic
Kenya
Kazakhstan
Jordan
Japan
Italy
Israel
Ireland
Iran
Indonesia
India
Honduras
Haiti
Guinea
Guatemala
Greece
Germany
Georgia
Gambia
Gabon
France
Finland
Ethiopia
Estonia
El Salvador
Egypt
Ecuador
Dominican Republic
Denmark
Czech Republic
Cyprus
Croatia
Cote d'Ivoire
Costa Rica
Congo, Rep.
Comoros
Colombia
China
Chile
Chad
Canada
Cameroon
Cambodia
Burundi
Burkina Faso
Bulgaria
Brazil
Botswana
Bosnia and Herzegovina
Bolivia
Benin
Belgium
Belarus
Bangladesh
Azerbaijan
Austria
Australia
Armenia
Argentina
Algeria
Albania
Afghanistan
Coutries
0
10
20
30
40
50
60
70
80
Happiness Score

[ ]:

```
```