Week 5: Project

This notebook presents the final project of the author in the **Basic Data Processing and Visualization** course offered by Coursera.

The dataset analysed in this workbook was obtained from another Coursera course that the author is enrolled in, namely **Data Management and Visualization** by Wesleyan University (https://www.coursera.org/learn/data-visualization/supplement/F0UbG/course-data-sets (https://www.coursera.org/learn/data-visualization/supplement/F0UbG/course-data-sets)).

If the reader is interested, more information is available at https://www.gapminder.org/ (https://www.gapminder.org/).

GapMinder Dataset Description

GapMinder is a non-profit venture founded in Stockholm by *Ola Rosling*, *Anna Rosling Rönnlund* and *Hans Rosling*, which aims to **promote sustainable global development and achievement** of the *United Nations Millennium Development Goals*. It seeks to increase the use and understanding of statistics about social, economic, and environmental development at local, national, and global levels.

Since its conception in 2005, Gapminder has grown to include over 200 indicators, including gross domestic product, total employment rate, and estimated HIV prevalence. Gapminder contains data for all 192 UN members, aggregating data for Serbia and Montenegro. Additionally, it includes data for 24 other areas, generating a total of 215 areas.

GapMinder collects data from a handful of sources, including the *Institute for Health Metrics and Evaulation*, *US Census Bureau's International Database*, *United Nations Statistics Division*, and the *World Bank*.

This portion of the GapMinder data includes one year of numerous country-level indicators of health, wealth and development. Each entry is uniquely identified by their country name. The aim of the project is to predict the residential electricity consumption per person from other measurable variables, specifically alcohol consumption and urban population as these are the only two variables collected on the same year.

Below is the dataset description from the GapMinder Codebook (https://d396qusza40orc.cloudfront.net/phoenixassets/data-management-visualization/GapMinder%20Codebook%20.pdf (https://d396qusza40orc.cloudfront.net/phoenixassets/data-management-visualization/GapMinder%20Codebook%20.pdf)):

Name	Data Type	Measure	Description
incomeperperson	continuous	US$	2010 gross domestic product per capita with inflation taken into account
alcconsumption	continuous	litres	2008 alcohol consumption per adult (age 15+)
armedforcesrate	continuous	%	percentage of armed forces personnel of total labor force
breastcancerper100TH	continuous	rate	2002 breast cancer new cases per 100,000 female
co2emissions	continuous	metric tons	2006 cumulative CO2 emission since 1751
femaleemployrate	continuous	% of population	2007 percentage of adult female employed
HIVrate	continuous	%	2009 estimated HIV prevalence (Ages 15-49)
Internetuserate	continuous	rate	2010 Internet users per 100 people

Name	Data Type	Measure	Description
lifeexpectancy	continuous	years	2011 life expectancy at birth
oilperperson	continuous	tonnes / (year person)	2010 oil consumption per capita
polityscore	integer	polity	2009 overall polity score from -10 to 10 (overall = autocracy - democracy)
relectricperperson	continuous	kWh	2008 residential electricity consumption per person
suicideper100TH	continuous	rate	2005 suicide per 100,000 people (age adjusted)
employrate	continuous	% of population	2007 percentage of adult employed
urbanrate	continuous	% of population	2008 urban population

Data Preparation and Cleaning

Some data are missing in the dataset. Since all of the variables are numerical, the missing data could be replaced by zeros to allow the calculation of statistics later on.

However, for the computation of correlation scores between the variables, the number of data points for each variable has to be the same. Therefore, the countries with missing data of interest, namely *alcconsumption*, *relectricperperson* and *urbanrate*, are eliminated from the dataset.

```
# Import useful libraries
import csv
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import style
from collections import defaultdict
from scipy.stats import pearsonr
```

```
# Load the GapMinder dataset
f = open("./gapminder.csv")
all_lines = list(csv.reader(f, delimiter = ','))

headers = all_lines[0]  # Extract the headers
dataset = []
for line in all_lines[1:]:
    d = dict(zip(headers, line))
    for header in headers[1:]:
        if ' ' not in d[header]: d[header] = float(d[header])  # Cast the numerical v
        else: d[header] = 0
    if d['relectricperperson'] != 0 and d['urbanrate'] != 0 and d['alcconsumption'] !
        dataset.append(d)

print(dataset[0])  # First line
```

```
{'country': 'Albania', 'incomeperperson': 1914.99655094922, 'alcconsumption': 7.2
9, 'armedforcesrate': 1.0247361, 'breastcancerper100th': 57.4, 'co2emissions': 223
747333.333333, 'femaleemployrate': 42.0999984741211, 'hivrate': 0, 'internetuserat
e': 44.9899469578783, 'lifeexpectancy': 76.918, 'oilperperson': 0, 'polityscore':
9.0, 'relectricperperson': 636.341383366604, 'suicideper100th': 7.69932985305786,
'employrate': 51.4000015258789, 'urbanrate': 46.72}
```

Simple Statistics

It is paramount to check out some simple statistics of the variables of interest prior to further analyses of the data. Now that the missing data have been handled and dataset is cleaned, some of the statistics are computed below.

The mean value and range of each variable has been explored. The countries with corresponding maximum and minimum values of each variable are also investigated. However, it is hard to deduce the relationships between these variables without visualizing them at this stage.

```
num_countries = len(dataset)
avg_elec = sum(d['relectricperperson'] for d in dataset) / num_countries
max_elec, min_elec = max(d['relectricperperson'] for d in dataset), min(d['relectricp
max_elec_country = [d['country'] for d in dataset if d['relectricperperson'] == max_e
min_elec_country = [d['country'] for d in dataset if d['relectricperperson'] == min_e
avg_urban_pop = sum(d['urbanrate'] for d in dataset) / num_countries
max_urban_pop, min_urban_pop = max(d['urbanrate'] for d in dataset), min(d['urbanrate
max_urban_pop_country = [d['country'] for d in dataset if d['urbanrate'] == max_urbar
min_urban_pop_country = [d['country'] for d in dataset if d['urbanrate'] == min_urbar
avg_alc_con = sum(d['alcconsumption'] for d in dataset) / num_countries
max_alc_con, min_alc_con = max(d['alcconsumption'] for d in dataset), min(d['alcconsu
max_alc_con_country = [d['country'] for d in dataset if d['alcconsumption'] == max_al
min_alc_con_country = [d['country'] for d in dataset if d['alcconsumption'] == min_al

print(f"Total number of countries in dataset: {num_countries}\n")
print(f"Average residential electricity consumption per person over all countries: {a
print(f"Maximum residential electricity consumption per person: {max_elec} kWh")
print(f"Country with the highest residential electricity consumption per person: {max
print(f"Minimum residential electricity consumption per person: {min_elec} kWh")
print(f"Country with the lowest residential electricity consumption per person: {min_
print(f"Average urban population over all countries: {avg_urban_pop} %")
print(f"Highest urban population: {max_urban_pop} %")
print(f"Country with the highest urban population: {max_urban_pop_country}")
print(f"Lowest urban population: {min_urban_pop} %")
print(f"Country with the lowest urban population: {min_urban_pop_country}\n")
print(f"Average alcohol consumption per person over all countries: {avg_alc_con} litr
print(f"Highest alcohol consumption per person: {max_alc_con} litres")
print(f"Country with the highest alcohol consumption per person: {max_alc_con_country
print(f"Lowest alcohol consumption per person: {min_alc_con} litres")
print(f"Country with the lowest alcohol consumption per person: {min_alc_con_country}
```

```
Total number of countries in dataset: 129

Average residential electricity consumption per person over all countries: 1211.2
555865840645 kWh
Maximum residential electricity consumption per person: 11154.7550328078 kWh
Country with the highest residential electricity consumption per person: United A
rab Emirates
Minimum residential electricity consumption per person: 9.19239470829337 kWh
Country with the lowest residential electricity consumption per person: Haiti

Average urban population over all countries: 61.22961240310076 %
Highest urban population: 100.0 %
Country with the highest urban population: Singapore
Lowest urban population: 13.22 %
Country with the lowest urban population: Trinidad and Tobago

Average alcohol consumption per person over all countries: 7.281007751937983 litr
es
Highest alcohol consumption per person: 23.01 litres
Country with the highest alcohol consumption per person: Moldova
```

```
Lowest alcohol consumption per person: 0.05 litres
Country with the lowest alcohol consumption per person: Pakistan
```

Data Visualizations

The distributions of the data for the three variables are visualized using histograms as they are all numeric continuous variables. The histograms also enable the central tendency and skewness of the data to be shown clearly. Box plots are then used to mark the outliers.

The relationships between the variables are then visualized using scatter plots. The correlations are calculated and displayed on the graphs for easier reference.

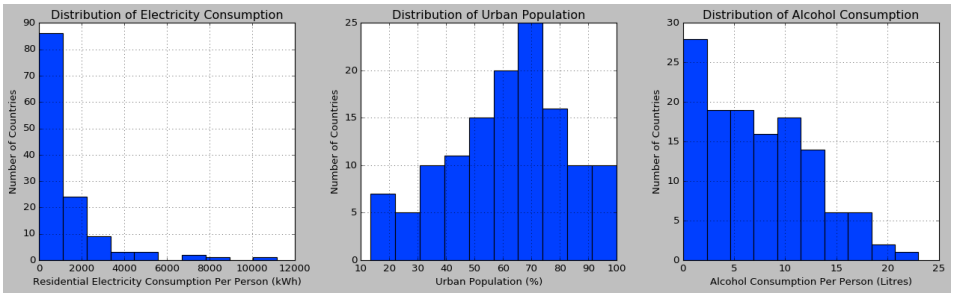
```
# Preparing data for visualization
elec = [d['relectricperperson'] for d in dataset]
elec.sort()
elec_con, elec_freq = defaultdict(int), []
for r in elec: elec_con[r] += 1
for key in elec_con.keys():
    for i in range(0, elec_con.get(key)): elec_freq.append(key)

urban = [d['urbanrate'] for d in dataset]
urban.sort()
urban_rate, urban_freq = defaultdict(int), []
for r in urban: urban_rate[r] += 1
for key in urban_rate.keys():
    for i in range(0, urban_rate.get(key)): urban_freq.append(key)

alc = [d['alcconsumption'] for d in dataset]
alc.sort()
alc_con, alc_freq = defaultdict(int), []
for r in alc: alc_con[r] += 1
for key in alc_con.keys():
    for i in range(0, alc_con.get(key)): alc_freq.append(key)

# print(elec, urban, alc)
```

Data Distribution

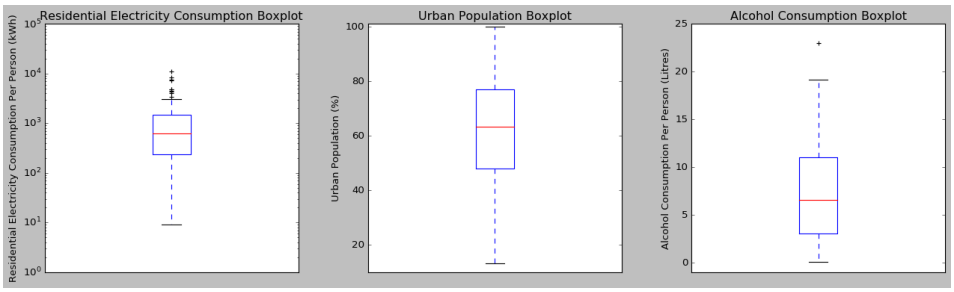
It is concluded from the histograms that the residential electricity consumption per person and alcohol consumption per person are strongly skewed to the right while the urban population roughly follows a normal distribution which peaks around urban population of 65%.

THe boxplots allows the identification of the outliers in the residential electricity consumption per person and alcohol consumption per person data. No data point is found to lie outside 1.5 * the interquartile range for the urban population variable. Note that a log scale was used for the plotting of residential electricity consumption per person due to the large order of magnitude in differences between the minimum and maximum of the variable. This allows clearer presentation of all points in the boxplot. However, readers should be careful when interpreting the plot. The median of the variable actually lies towards the lower side even though it appears to be right in the middle of the distribution.

```
# Plot the histograms
style.use('seaborn-bright')
plt.figure(figsize=(16, 5))
plt.subplot(1, 3, 1)
plt.gca().set(xlabel='Residential Electricity Consumption Per Person (kWh)', ylabel='
plt.grid()
plt.hist(elec_freq)
plt.subplot(1, 3, 2)
plt.gca().set(xlabel='Urban Population (%)', ylabel='Number of Countries', title='Dis
plt.grid()
plt.hist(urban_freq)
plt.subplot(1, 3, 3)
plt.gca().set(xlabel='Alcohol Consumption Per Person (Litres)', ylabel='Number of Cou
plt.grid()
plt.hist(alc_freq)
plt.tight_layout()
plt.show()
```



```
# Plot the box plots
plt.figure(figsize=(16, 5))
plt.subplot(1, 3, 1)
plt.boxplot(elec)
plt.xticks([])
plt.yscale('log')
plt.gca().set(ylabel='Residential Electricity Consumption Per Person (kWh)', title='F
plt.subplot(1, 3, 2)
plt.boxplot(urban)
plt.xticks([])
plt.ylim(None, 101)
plt.gca().set(ylabel='Urban Population (%)', title='Urban Population Boxplot')
plt.subplot(1, 3, 3)
plt.boxplot(alc)
plt.xticks([])
plt.ylim(-1, None)
plt.gca().set(ylabel='Alcohol Consumption Per Person (Litres)', title='Alcohol Consur
plt.tight_layout()
plt.show()
```



Data Correlation

The scatterplots clearly show that both urban population and alcohol consumption are positively correlated with the residential electricity consumption, with the correlation between alcohol consumption and residential electricity consumption being stronger. However, correlation does not imply causation. Further causality statistical studies would need to be conducted to determine the causality relationship between the variables.

It is interesting to note that the alcohol consumption and urban population are very strongly positively correlated. This could be explained by greater need for alcohol and greater buying power of the residents in the urban area.

```
# Compute the Pearson's correlation scores
elec_urban_corr = pearsonr(elec, urban)
elec_alc_corr = pearsonr(elec, alc)
urban_alc_corr = pearsonr(urban, alc)

# Graph the scatter plots
plt.figure(figsize=(20, 6))
plt.subplot(1, 3, 1)
plt.gca().set(xlabel='Residential Electricity Consumption Per Person (kWh)', ylabel='
plt.scatter(elec, urban, label='Pearson Correlation: {0:.2f}'.format(elec_urban_corr|
plt.legend()
plt.subplot(1, 3, 2)
plt.gca().set(xlabel='Residential Electricity Consumption Per Person (kWh)', ylabel='
plt.scatter(elec, alc, label='Pearson Correlation: {0:.2f}'.format(elec_alc_corr[0]))
plt.legend()
plt.subplot(1, 3, 3)
plt.gca().set(xlabel='Urban Population (%)', ylabel='Alcohol Consumption Per Person (
plt.scatter(urban, alc, label='Pearson Correlation: {0:.2f}'.format(urban_alc_corr[0
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7f4cd0283da0>
```