Project Report

UCD Professional Academy

Certificate in Data Analytics

[CIDA 2022-02-15](https://learn.ucdpa.ie/course/view.php?id=413)

Hendry Duvenage

2022-04-19

# GitHub URL

<https://github.com/hendryduvenage/UCDPA_hendryduvenage>

# Abstract

The project aims to illustrate skills gained in using Python for data analysis. For this purpose, two datasets were used with relatable data, necessary formatting, grouping and cleaning of the data was done after which the two resulting data frames were joined into one. Two graphs visualizing insights from the data is included to summarize the important and relevant points.

# Introduction

I discovered the dataset for the hours of sunshine per city on Kaggle, and since I have an interest in solar energy decided to compare it with the amount of solar power installed in each of these countries. It provided the opportunity to perform several data analysis actions to prepare the two datasets for joining them – from which visualizations can be retrieved to compare columns which may provide some insightful results.

# Dataset

Two data sources were used in this project. The first consists of a .csv file retrieved from Kaggle [1], which is a table showing hours of sunshine in several major cities across the world. Data is split into columns of monthly values and a cumulative value for the year. The data in this source was already clean with no need for further filtering or cleaning to make it usable. A function was run to group the mean values for all the countries the European Union as a new row in the table. Also grouping all the cities in the same country was done taking the maximum value from each country and finally the total for all rows were calculated to add a World Total row to the end of the data.

The second data source was retrieved using web scraping from Wikipedia [2]. The table contained a list of countries with columns detailing the amount of solar power in KW installed between 2016 and 2020 – split into two columns for each year showing the new capacity added and the total after the year in question. Further columns showing the W per capita in 2019 and the share of the country’s total consumption is also shown and the World total value is added to the last row. Since this table is coming from Wikipedia it required several steps of cleaning to prepare the data, such as removing all unnecessary columns and rows, removing all reference tests between ‘[]’ brackets.

With the data prepared and having the Country column similar between the two tables, the tables are joined with based on similar countries to create the final dataset.

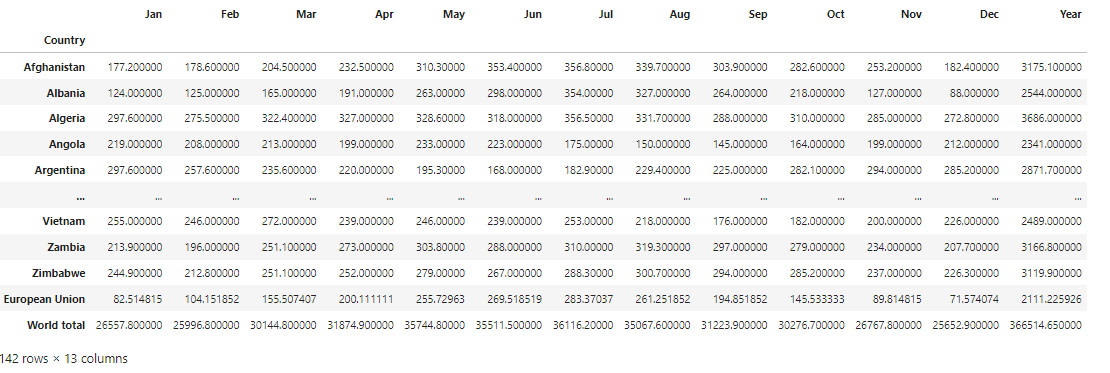
# Implementation Process

The structure of the Python software was done as follows:

The first section was used to import all the necessary Python libraries. Libraries used include pandas, re, requests, BeautifulSoup, seaborn and matplotlib.

A function named Is\_Europe is defined written in the following section, which is the function used to check if a country is in Europe. A list of countries in Europe obtained from [3] was entered as a list – with a country being fed to the function and then checked if it is contained in the list. The function then returns a true or false string depending on the result – via an if and else function.

The next sections are dealing with the import of the tables, the first table via Web scraping from [2] named solar and the second from a CSV file retrieved from [1] and stored in the local project folder named sunshine. These tables are both made to be pandas data frames. The sorting of the sunshine table follows, with the aim to get the table similar to the solar table in terms of the data fields available. Using the Is\_Europe function and a while loop, each row is checked and flagged if the relevant country returns a true into a new column that was added for the table. Following this, the rows which has similar countries is grouped and only the maximum value retained for each group. Two new rows of data are then to be added – firstly a world total series is generated taking the sum of all values for each month. After this a series for all countries in the European Union is created taking the mean value of all the countries. These two series are then appended to the data frame of the sunshine table. Finally, the temporary column for flagging the European countries is hidden.



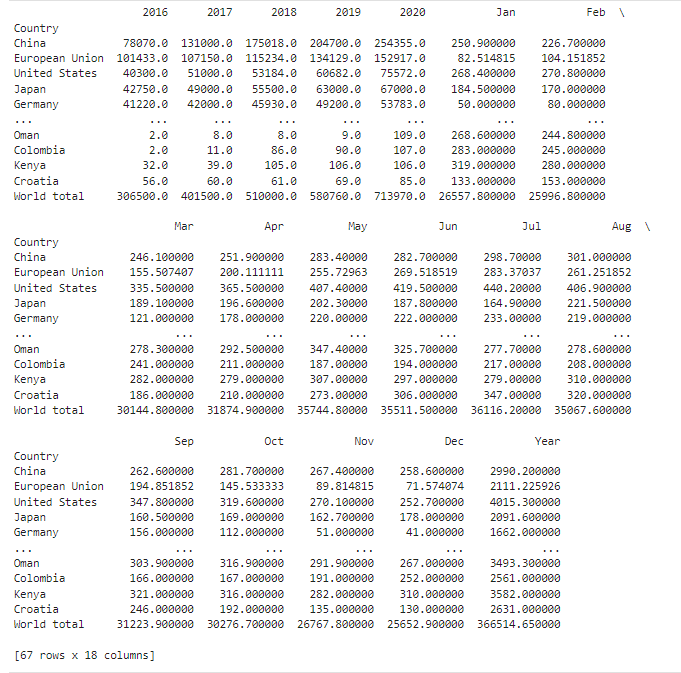
*Figure 1 – Sunshine data frame*

The solar data frame required some more cleaning functions to prepare the data. Firstly, a row was removed at the end of the table which only contained some referencing information from Wikipedia. After this the column names were renamed to remove the Wikipedia referencing brackets. The Watt per capita and share of total consumption columns were not necessary for this project and they were dropped from the data frame. To remove all the Wikipedia referencing brackets in the Country names, a while loop was set up to scan through each row and replace the bracketed strings with blanks. The index for the data frame is set to be the Country column, to match the sunshine data frame and enable joining. The columns are defined to be the year number, to ensure the column headers are single level. The data values in the data frame is converted to float, as in the grouping procedure they turned into objects which would cause issues when visualizing later.



*Figure 2 – Solar data frame*

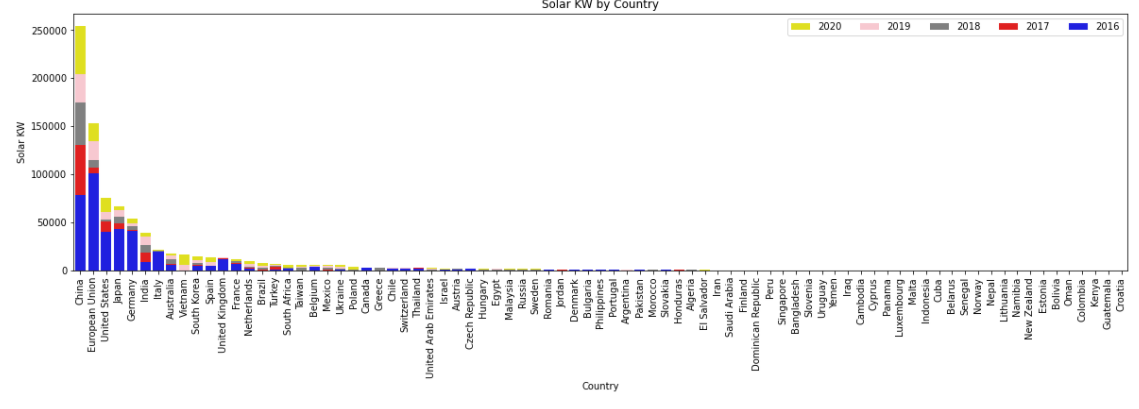
In the next section the joining of the two data frames is done. Since the sunshine data frame had a lot more countries in than the solar data frame is was decided to join left using the countries in the solar data frame as reference. This would decrease the number of rows returned with data missing in some of the columns. Following the join a function was run to drop the remaining rows with Na as data in any of the columns – to ensure all entries in the final table contains a usable value. The final data frame, named solar\_sunshine is returned as a result.



*Figure 3 – Solar\_Sunshine data frame*

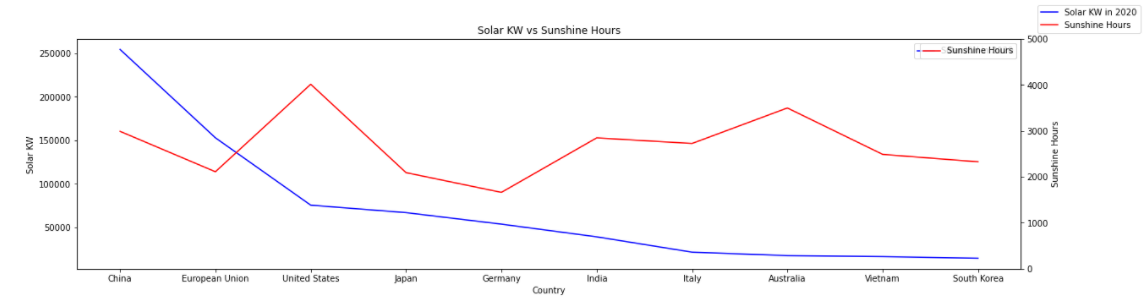
In the visualisation section, the first chart used the solar dataset to indicate in a stacked bar chart how the amount of solar power installed in each country has increased on a yearly base. This used 5 x plots with data from columns from 2016, 2017, 2018, 2019 and 2020 shown in different colors. The chart is formatted with a label ‘Solar KW by Country’, with a legend and axis labels included. The second chart makes use of the solar\_sunshine dataframe to show the relationship between the number of sunshine hours and the amount of solar power installed for the first 10 countries listed. It was necessary to create 2 y-axes in this instance since the scaling of the two values differed greatly. The chart consists of 2-line plots which are stacked. The chart is formatted with a label ‘Solar KW vs Sunshine Hours’ with a legend and axis labels included.

# Results



*Figure 4 – Solar KW by Country*

The chart above shows the amount of solar power generated by each of the countries, stacked to indicate how the value has grown between 2016 and 2020. The chart mainly shows the values of the first 10 countries, as the remaining countries value is too small to be relevant in this chart.



*Figure 5 – Solar KW vs Sunshine Hours*

The chart above shows the amount of solar power compared to the number of sunshine hours for the first 10 countries listed in the data frame. The reason for only using 10 rows is to negate the impact seen in the first chart from the remaining values being very low and not really giving data of any relevance.

# Insights

From the Solar KW by Country chart, the following can be derived:

* China has superior capacity of Solar power installations, and has constantly been adding new capacity between 2016 and 2020.
* The European Union did initially have the highest capacity in 2016, but lower addition in 2017 and 2018 resulted in them falling behind.

From the Solar KW vs Sunshine Hours chart the following can be derived:

* The United States and Australia has the most hours of sunshine available, with Australia especially being under utilised in terms of Solar Capacity installed by 2020.
* China and the European union have lower available sunshine hours but manages to have the highest capacity of solar power installed.

Machine learning can be used to approximate the following factors:

* Based on the rate of new solar capacity being added – machine learning can determine for instance what the installed solar capacity will be at a future date for example in 2030.
* Based on the amount of sunshine hours available in each country and using China as a benchmark – a calculation can be made to determine the optimal solar power capacity which can be installed for each country and what the current shortfall is in comparison with China.

# References

[1] Kaggle.com. 2022. *Sunshine duration by city*. [online] Available at: <https://www.kaggle.com/datasets/prasertk/sunshine-duration-by-city> [Accessed 8 April 2022].

[2] En.wikipedia.org. 2022. *Solar power by country - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Solar\_power\_by\_country> [Accessed 8 April 2022].

[3] En.wikipedia.org. 2022. *Solar power in the European Union - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Solar\_power\_in\_the\_European\_Union> [Accessed 8 April 2022].