Renewable energy sources and the effects of climate change

Report overview:

Global warming is one of the greatest existential threats to humans in modern times, it's a problem that will affect every country in the world and requires a combined effort from every continent to tackle before it's too late. One of the ways in which global warming can be tackled is by releasing less fossil fuels, and this can be achieved is by using more renewable energy sources. Over the past 10 years, our production of energy through fossil fuels has risen greatly. However, so has our energy consumption overall.
I have prepared data to outline the scale of this problem, which will highlight just how stark the problem is, as well as give insight into where we need to improve our renewable energy output. As well as providing a reminder about the importance of why we need to take action against climate change right now.

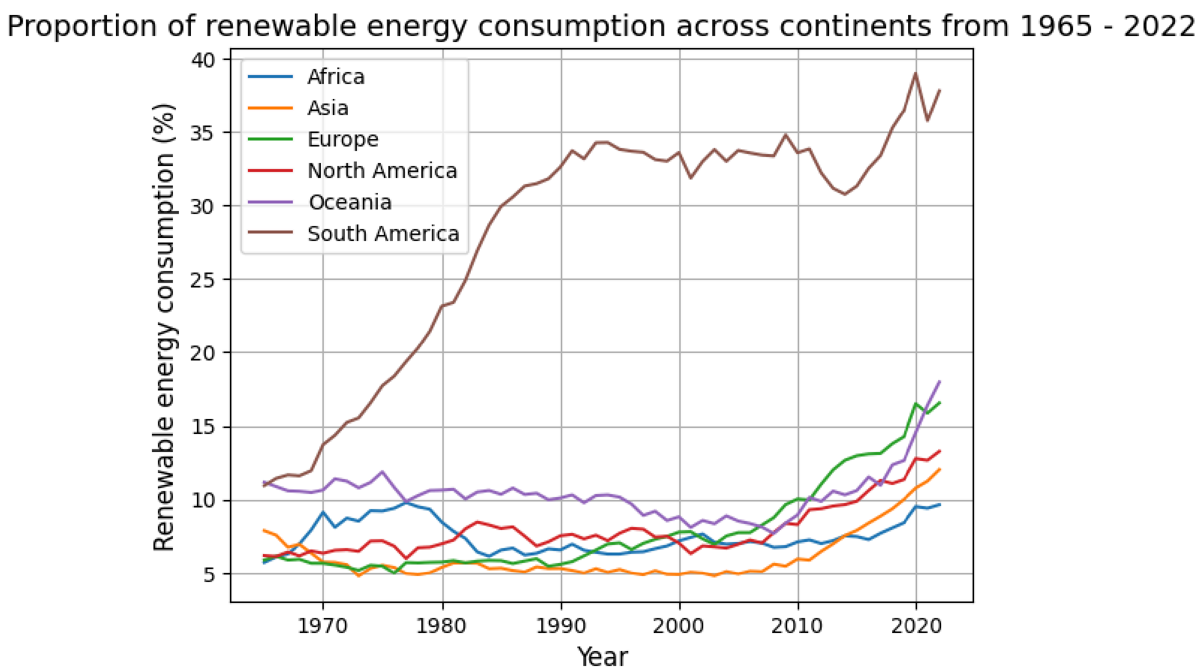
I decided to create my report on the subject matter of climate change because I have an interest in global warming and the effects of inaction, as it looks like it will likely have an overarching and disproportionate effect on my generation.
The data I used to create this report was found on https://ourworldindata.org/. I've used this data as it's collated by Oxford university which is a very credible world leading university.

The web location of the data used can be found at the following locations:

- https://ourworldindata.org/renewable-energy
- https://ourworldindata.org/energy
- https://ourworldindata.org/co2-and-greenhouse-gas-emissions
- https://ourworldindata.org/air-pollution

Proportion of renewable energy consumption

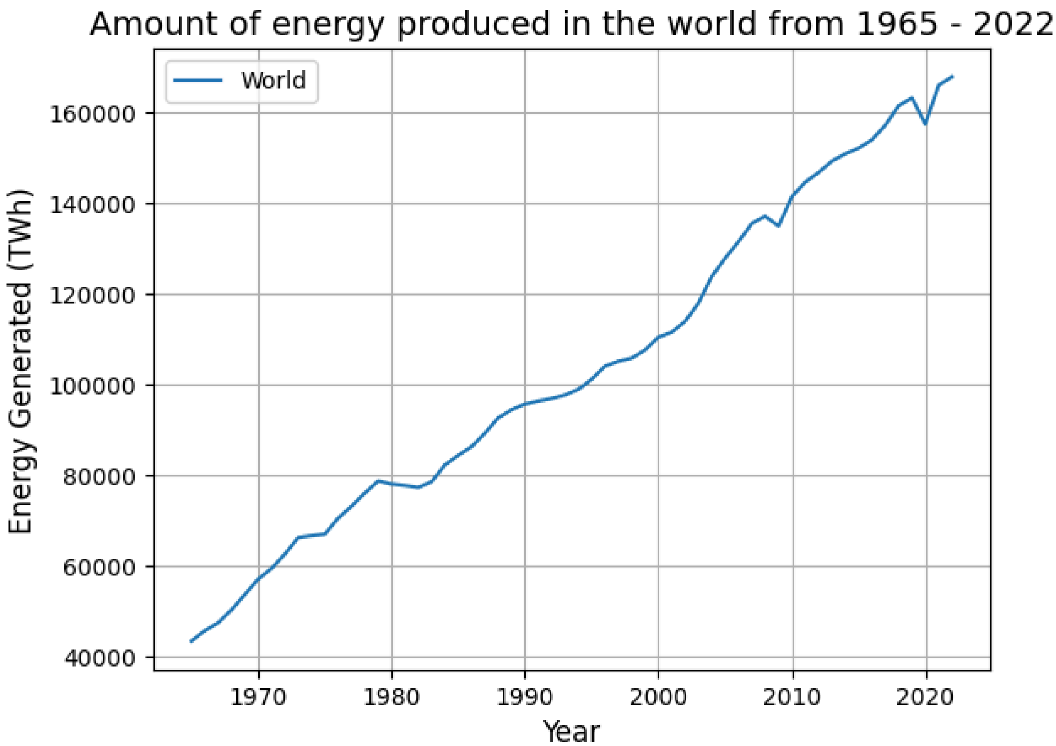
Renewable energy accounts for less than half of the energy consumed by continents across the globe. And until the last 10 years, it has been practically stagnant as a proportion of all energy output.



Graph 1.0

(Energy Institute - Statistical Review of World Energy (2023) – with major processing by Our World in Data)

This seems surprising, as new solar farms, wind turbines and hydroelectric power plants are built yearly. But we start to see the cause of why this happens in graph 1.1.

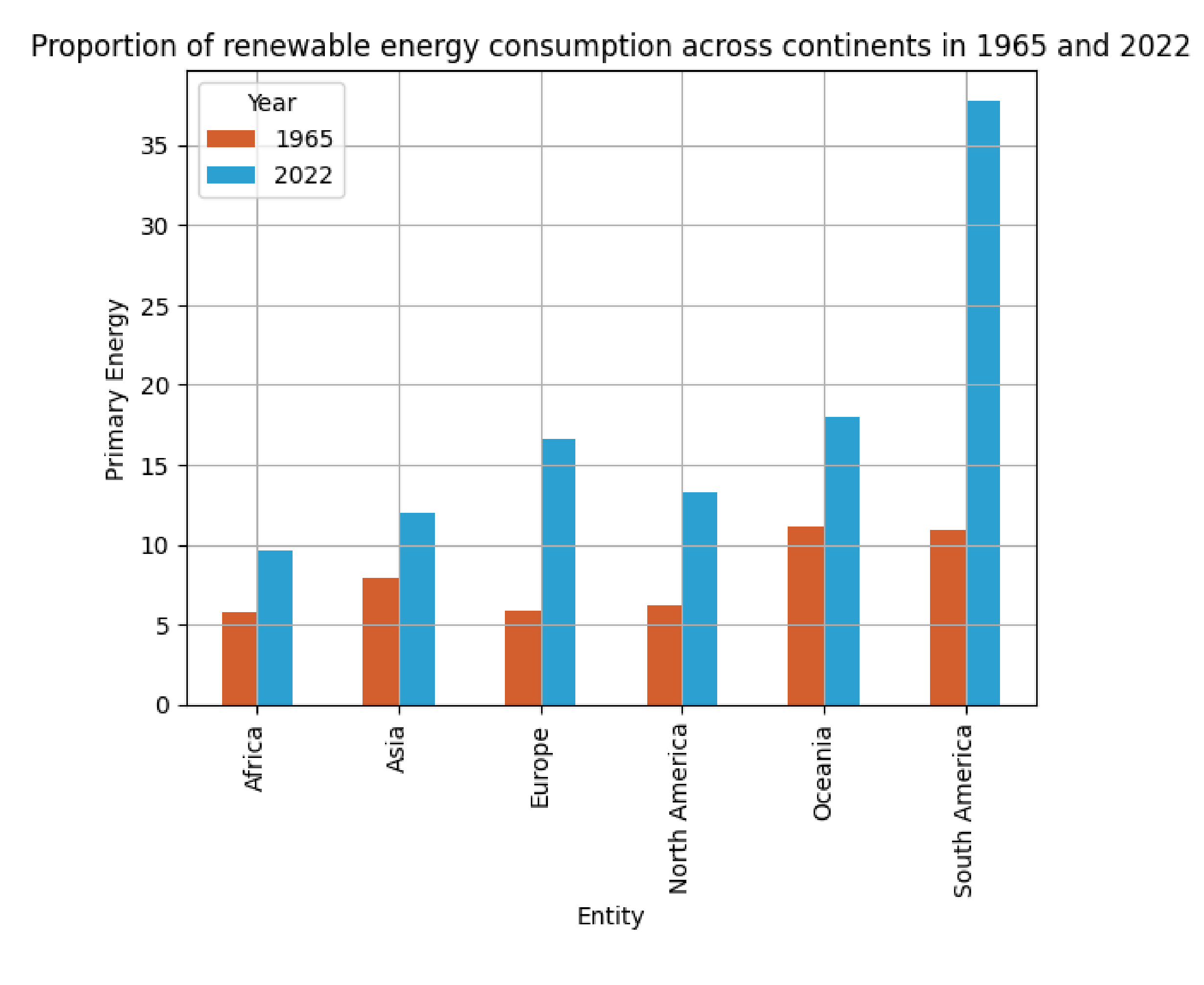


Graph 1.1

(U.S. Energy Information Administration (2023); Energy Institute - Statistical Review of World Energy (2023); Population based on various sources (2023) – with major processing by Our World in Data)

As we can see, the overall amount of energy being produced each year is going up, and although renewables are taking a greater share of this energy each year, it's not even close to being enough to catch up with our ever increasing energy needs in the next decade.

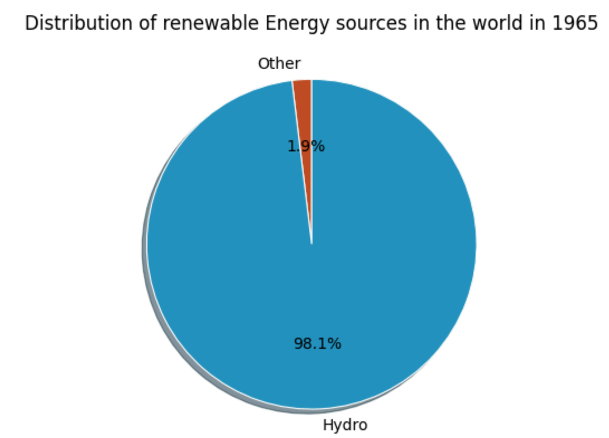
In graph 1.2, we can see that for most continents, the proportion of all energy used being produced by renewables has risen by only a few percentage points since 1965 excluding South America, and this has mostly been in the last 10 years. That is despite huge funding from governments and the private sector into renewable research and development across the world to try and tackle this problem.



Graph 1.2

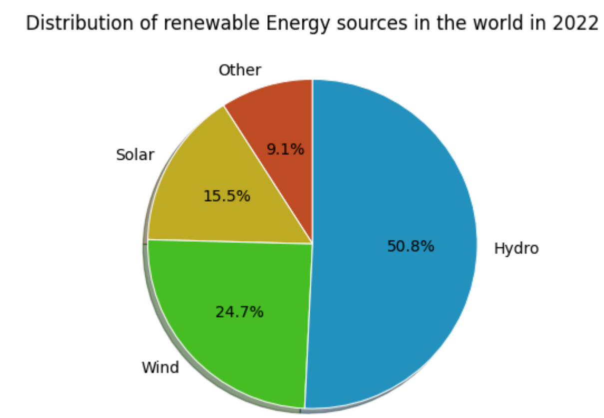
(Energy Institute - Statistical Review of World Energy (2023) – with major processing by Our World in Data)

One way in which the world is trying to solve this issue, is by bringing new ways of harnessing renewable energy. In the 60’s, solar and wind energy accounted for none of our renewable energy output, today they account for over 40%



<u>Graph 1.3</u>

(Energy Institute - Statistical Review of World Energy (2023) – with major processing by Our World in Data)



<u>Graph 1.4</u>

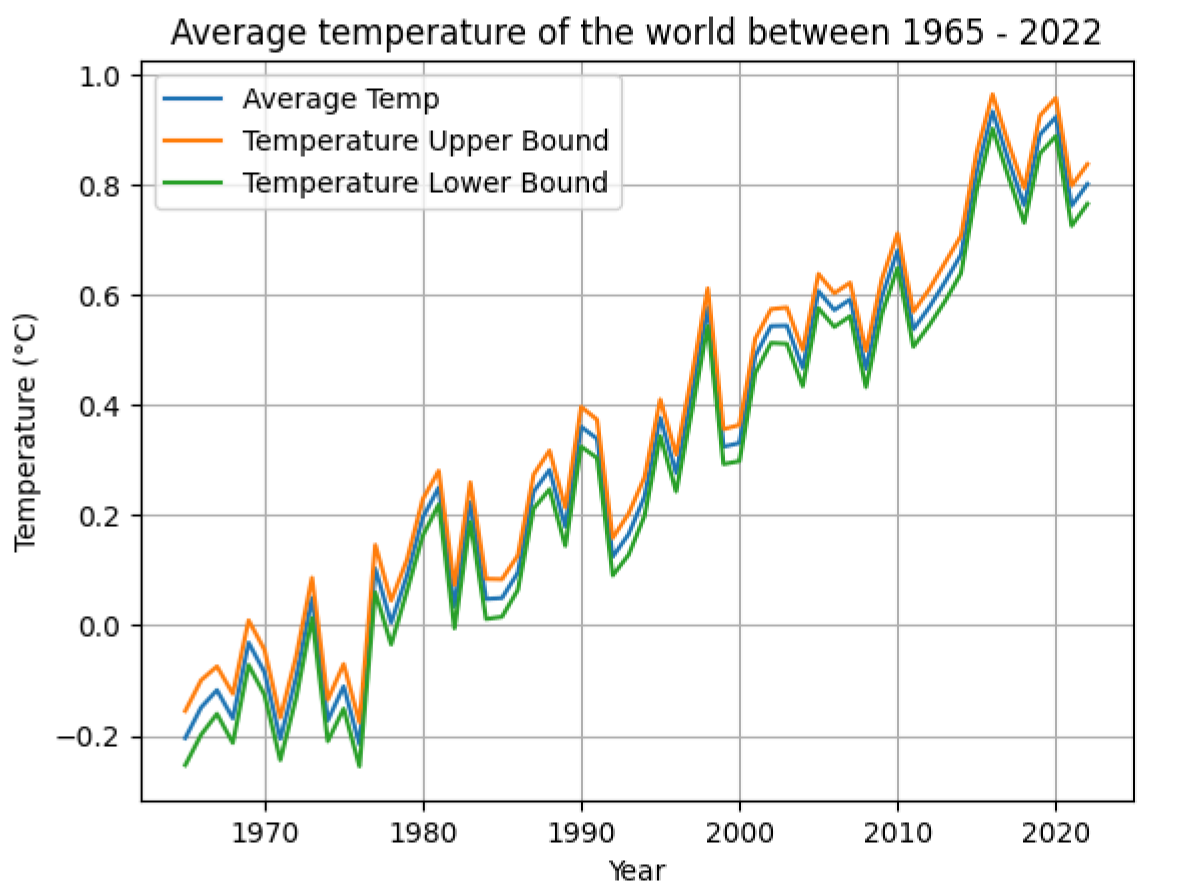
(Energy Institute - Statistical Review of World Energy (2023) – with major processing by Our World in Data)

Finding new ways to harness renewable energy is one of the key ways that will be required to solve global warming.

<u>Risks of Failure to act</u>

If we fail to solve this problem now, then what will happen to the world will be stark.

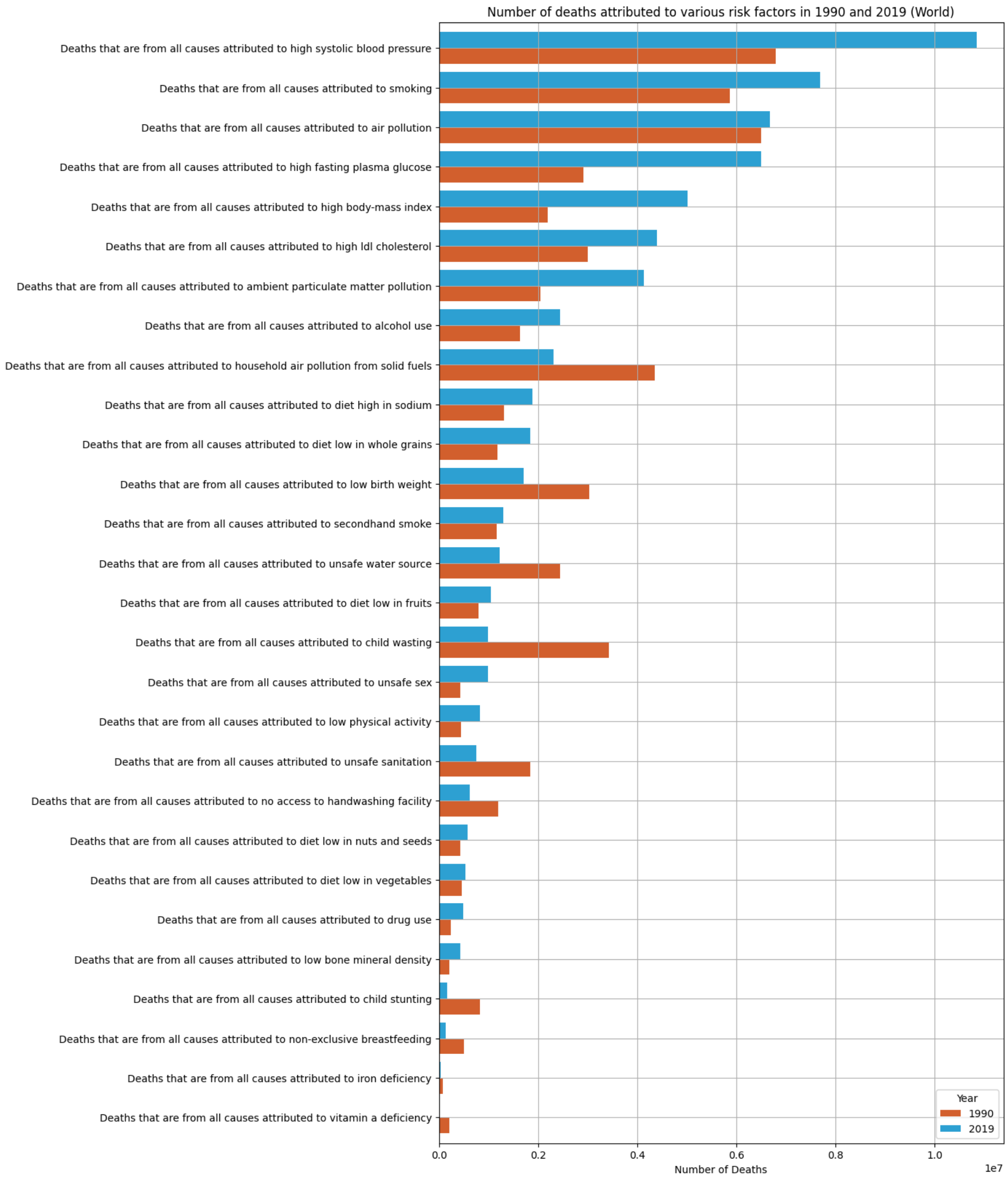
In graph 2.0, we can see that between the years 1965 and 2022, the global average temperature has increased dramatically, and it looks set to carry on increasing for years to come. If we do not find a way to control this, we risk global drought, food shortages, extreme weather and floods in the near future.



__Graph 2.0__

(Met Office Hadley Centre (2023) – processed by Our World in Data)

The problems with climate inaction also extend to the short term. Right now people are dying from the results of climate change and in particular the use of fossil fuels and air pollution. In graph 2.1 we can see that deaths caused by air pollution and deaths caused by ambient particulate matter pollution are in the millions of deaths per year. And this will only get worse unless we move away from fossil fuels.



Graph 2.1

(IHME, Global Burden of Disease Study (2019) – processed by Our World in Data)

Summary:

From the data gathered in this report, it is clear that the world has heavily invested in renewable energy, especially over the last 10 years. Additionally, the trend is that renewable energy sources will take up an increasingly large proportion of all the energy produced by the world over the next few decades. However, it is clear that the current level of energy produced by renewable sources is not close to being sufficient to cope with future demands of the world's energy consumption. With the introduction of AI being increasingly important in all aspects of business and with the increased energy consumption this will have, it is clear that if we are to avoid a climate disaster as well as deaths caused by air pollution, it is paramount that renewable energy production ramps up globally in the coming years.

A description of the steps included in your ETL script

Initially, I prototyped my code using Google Colab which is a hosted Jupyter Notebook service. Once I was happy with my data visualisation I took the code from Google Colab, and added functionality, readability and scalability to my code by building an ETL project using the PyCharm IDE.

My Code - Google Colab (Jupyter Notebook)

Below is the code used in my Google Colab Scripts. A copy of this can be found in the zip file containing the entire project under the name 'PythonDataProject_2.ipynb'

```
#My imports list
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
#Load in my CSV's
from google.colab import drive

drive.mount('/content/drive')
```

```
#Turn my CSV's into dataframes
df_renewable_energy_share =
pd.read_csv('/content/drive/MyDrive/Python/Datasets/renewable-share-energy.csv')
```

```
df_primary_energy_cons = pd.read_csv('/content/drive/MyDrive/Python/Datasets/primary-energy-cons.csv')

df_modern_renewable_energy_consumption =

pd.read_csv('/content/drive/MyDrive/Python/Datasets/modern-renewable-energy-consumption.csv')

df_temperature_anomaly = pd.read_csv('/content/drive/MyDrive/Python/Datasets/temperature-anomaly.csv')

df_number_of_deaths_by_risk_factor =

pd.read_csv('/content/drive/MyDrive/Python/Datasets/number-of-deaths-by-risk-factor.csv')
```

Graph 1.0 code:

```
#getting the data I want from my df_renewable_energy_share

#I require Entities: North America, South America, Oceania, Europe, Asia, Africa - Maybe include world as

well?

# Also require the uears and the Primary Energy consumption


#Remove the columns I do not want:

df = df_renewable_energy_share.drop(['Code'], axis=1)

df = df.rename(columns={'Renewables (% equivalent primary energy)' : 'Primary Energy'})


entities = ['North America', 'South America', 'Oceania', 'Europe', 'Asia', 'Africa']

#Get only the entities I require

df_selected = df[df.iloc[:,0].isin(entities)].iloc[:, [0, 1, 2]]

#df_selected is now the dataframe that I am to use for the current plot


grouped = df_selected.groupby('Entity')

for name, group in grouped:
```

```
    plt.plot(group['Year'], group['Primary Energy'], label=name)


plt.title('Proportion of renewable energy consumption across continents from 1965 - 2022', fontsize=14)
plt.xlabel('Year', fontsize=12)
plt.ylabel('Renewable energy consumption (%)', fontsize=12)
plt.grid(True)
plt.legend()
plt.show()
```

Graph 1.1 code:

```
# Now I need primary energy consumption of the world using df_primary_energy_cons dataframe


df = df_primary_energy_cons.drop(['Code'], axis=1)
df = df.rename(columns={'Primary energy consumption (TWh)' : 'Energy Consumption'})
df.head()
entities = ['World']
df_selected = df[df.iloc[:,0].isin(entities)].iloc[:, [0, 1, 2]]
df_selected.shape


grouped = df_selected.groupby('Entity')
for name, group in grouped:
    plt.plot(group['Year'], group['Energy Consumption'], label=name)


plt.title('Amount of energy produced in the world from 1965 - 2022', fontsize=14)
```

```
plt.xlabel('Year', fontsize=12)

plt.ylabel('Energy Generated (TWh)', fontsize=12)

plt.grid(True)

plt.legend(loc='upper left')

plt.show()
```

Graph 1.2 code:

```
# The next chart is using the df_renewable_energy_share dataframe again

df = df_renewable_energy_share.drop(['Code'], axis=1)

df = df.rename(columns={'Renewables (% equivalent primary energy)' : 'Primary Energy'})


entities = ['North America', 'South America', 'Oceania', 'Europe', 'Asia', 'Africa']

#Get only the entities I require

df_selected = df[df.iloc[:,0].isin(entities)].iloc[:, [0, 1, 2]]

#df_selected is now the dataframe that I am to use for the current plot


# grouped = df_selected.groupby('Entity')


df_selected_years = df_selected[df_selected['Year'].isin([1965, 2022])]

df_pivot = df_selected_years.pivot(index='Entity', columns='Year', values='Primary Energy')


df_pivot.plot(kind='bar', color=['#d25f2d', '#2da0d2'])

plt.xlabel('Entity')

plt.ylabel('Primary Energy')
```

```
plt.grid(True)

plt.title('Proportion of renewable energy consumption across continents in 1965 and 2022')

plt.show()
```

Graph 1.3 code:

```
# Now I want 2 pie charts one from renewable energy  sources in 1965 and another from 2022

# I need to use the following dataframe:

# df_modern_renewable_energy_consumption


# first, get rid of code

df = df_modern_renewable_energy_consumption.drop(['Code'], axis=1)


# now change the columns


df = df.rename(columns={'Other renewables (including geothermal and biomass) electricity generation -
TWh' : 'Other',

                    'Solar generation - TWh' : 'Solar',

                    'Wind generation - TWh' : 'Wind',

                    'Hydro generation - TWh' : 'Hydro'})




# now I need to get just the years I want:

entities = ['World']

df_selected = df[df.iloc[:,0].isin(entities)].iloc[:, [0, 1, 2, 3,4,5]]
```

```
df_year_1965 = df_selected[df_selected['Year'].isin([1965])]

df_year_2022 = df_selected[df_selected['Year'].isin([2022])]

labels = ['Other','Hydro']
colors = ['red','blue']

colors = ['#be4d23','#2394be']

selected_entity_row_1965 = df_year_1965[df_year_1965['Entity'] == 'World']
data_for_pie_1965 = selected_entity_row_1965.iloc[0].drop(['Entity','Year', 'Solar', 'Wind'])

data_for_pie_1965.plot(kind='pie', labels = labels, autopct='%1.1f%%',shadow=True, startangle=90, colors
= colors, wedgeprops = { 'linewidth' : 0.8, 'edgecolor' : 'white' })
plt.title('Distribution of renewable Energy sources in the world in 1965')
plt.ylabel('')  # this is to remove the default 'None' ylabel

plt.show()
```

Graph 1.4 code:

```
selected_entity_row_2022 = df_year_2022[df_year_2022['Entity'] == 'World']
data_for_pie_2022 = selected_entity_row_2022.iloc[0].drop(['Entity','Year'])
```

```
labels = ['Other', 'Solar', 'Wind', 'Hydro']
colors = ['red','yellow', 'green', 'blue']

colors = ['#be4d23', '#beaa23', '#46be23', '#2394be']


data_for_pie_2022.plot(kind='pie', labels = labels, autopct='%1.1f%%',shadow=True, startangle=90, colors
= colors, wedgeprops = { 'linewidth' : 0.8, 'edgecolor' : 'white' })
plt.title('Distribution of renewable Energy sources in the world in 2022')
plt.ylabel('')  # this is to remove the default 'None' ylabel
plt.show()
```

Graph 2.0:

```
# First I need to drop code column and then change the column headers.

df = df_temperature_anomaly.drop(['Code'], axis=1)

# now change the columns

df = df.rename(columns={'Global average temperature anomaly relative to 1961-1990' : 'Average Temp',
                  'Upper bound of the annual temperature anomaly (95% confidence interval)' : 'Upper Bound',
                  'Lower bound of the annual temperature anomaly (95% confidence interval)' : 'Lower
Bound'})
```

```
df = df[['Entity', 'Year', 'Average Temp', 'Upper Bound', 'Lower Bound']]
df= df[(df['Year'] >= 1965) & (df['Year'] <= 2022)]
df = df[df['Entity'] == 'Global']



# Now df_selected contains only necessary columns
grouped = df.groupby('Entity')

fig, ax = plt.subplots()

for name, group in grouped:
    group.plot(x='Year', y='Average Temp', ax=ax, label='Average Temp')
    group.plot(x='Year', y='Upper Bound', ax=ax, label='Temperature Upper Bound')
    group.plot(x='Year', y='Lower Bound', ax=ax, label='Temperature Lower Bound')




plt.ylabel('Temperature (°C)')
plt.grid(True)
plt.title('Average temperature of the world between 1965 - 2022')
plt.show()
```

Graph 2.1 code:

```python
# Deaths by risk factor using - df_number_of_deaths_by_risk_factor

# I want 2 line graphs on top of each other one from 1990 and another from 2019, by world


df = df_number_of_deaths_by_risk_factor



# First we get rid of Code and group by world

df = df.drop(['Code'], axis=1)

df = df[df['Entity'] == 'World']


# Now to get the dataframes for the correct years:

df= df[(df['Year'] == 1990) | (df['Year'] == 2019)]


df = df.drop(columns=['Entity'])


# Set 'Year' column as index

df = df.set_index('Year')


# Transpose the DataFrame to create a suitable format for a bar plot

df = df.T


df = df.sort_values(by=2019, ascending=True)

df.rename(index = lambda x: str(x)[:-29], inplace=True)
```

```
# Plot

df.plot(kind='barh', figsize=(10, 20), width=0.8, color=['#d25f2d', '#2da0d2'])

plt.title('Number of deaths attributed to various risk factors in 1990 and 2019 (World)')

plt.grid(True)

plt.xlabel('Number of Deaths')

plt.show()
```

My Code - ETL Project (PyCharm)

Once I knew the Graphs created in Google Colab displayed the data how I wanted it to look, I recreated the data as a PyCharm project. I did this for three main reasons.

Firstly, I wanted to use the skills I'd learned on the course but didn't cover in my Google Colab version of the graphs, such as OOP, inheritance and functions.

Secondly, although the outcome of the two methods of creating graphs look more or less the same, the PyCharm project allows me to create new graphs much faster as it has more code reuse and is structured in a better, more logical format.

Lastly, if I was to add more data visualisation and attach my graphs to a front end, it would be much easier to do so using the PyCharm version rather than the Google Colab version to do this.

Admittedly, the PyCharm version is probably overkill for creating this report, and much of the code is more experimenting with how Python works and learning new skills rather than being completely necessary. This said, I would like to experiment with creating an ETL pipeline using data from an API

I’ve created using the skills from the Java course to make some interesting graphs in the future. And the PyCharm project is a much better starting point for doing this than using Google Colab.

My Code is split into the following components:

CSVsExtracted:

- These are the output CSV files after they have been extracted and transformed by the program.

Data:

- This is the location of the raw CSV files downloaded from Our World in Data which is used to create the graphs.

Data_processors:

- This is the logic end for creating the dataframes for the graph scripts. Each graph has its own data processor which has its own logic to create the individual Graphs in the graph script. The advantage of doing this is that they all inherit a generic data_proccesor with methods to extract and load CSV’s. Ideally, I would prefer to have data_processors for each type of graph produced (for example, a pie chart data processor and a scatter graph processor). I didn’t do this for the current project because I wanted the flexibility for all of the graphs to be entirely different if need be. However, this may change by the time I submit the project.

Extractor:

- A class containing static methods to convert raw data to a pandas dataframe. At the moment I only have from CSV implemented as all the data I used was from CVS’s, but it may be useful to have the ability to extract from either CSV or JSON, and then have the parent DataProcessor class take the input data and decide for itself whether the data is in CSV or JSON format and use decide which Extractor method to use accordingly.

Loader:

- A class containing a static method which takes a pandas dataframe and converts it to a CSV file.

Transformer:

- A class containing static methods which transform the Pandas dataframe that was generated from the Extractor class. Doing my transformations this way means that I have a greater level of code reuse, as well as better exception handling. Also, it means that I can test individual transformation methods that are used across multiple graphs, rather than testing every individual graph transformation statement.

Graph scripts:

- The graph scripts bring all the other packages together and plot the graph created from the dataframes produced by the DataProcessor's.

Full code Location:

My Full code is too large for me to paste into this report. However, the full code can be found in the folder this report was found in. Additionally, a git repository of the project can be cloned from: https://github.com/Bazoopa/ETL

References

Energy Institute - Statistical Review of World Energy (2023) – with major processing by Our World in Data. "Share of primary energy consumption that comes from renewables – Using the substitution method" [dataset]. Energy Institute, "Statistical Review of World Energy" [original data].

U.S. Energy Information Administration (2023); Energy Institute - Statistical Review of World Energy (2023); Population based on various sources (2023) – with major processing by Our World in Data. "Primary energy consumption per capita" [dataset]. U.S. Energy Information Administration, "International Energy Data"; Energy Institute, "Statistical Review of World Energy"; Various sources, "Population" [original data].

Met Office Hadley Centre (2023) – processed by Our World in Data. "Mean" [dataset]. Met Office Hadley Centre, "HadCRUT5 HadCRUT.5.0.2.0" [original data].

IHME, Global Burden of Disease Study (2019) – processed by Our World in Data. "High blood pressure" [dataset]. IHME, Global Burden of Disease Study (2019) [original data].