**UCDPA**

**Certificate in Data Analytics for Aviation**



Project Report by Daniel Frost.

Revision Date: 14-Dec-2022

Github URL: <https://github.com/FrostDan/UCDPA_Daniel_Frost>

## Abstract

Over the past few decades, air travel has become more affordable and as such has seen an increase in global flights.

This increase in flights has resulted in an increase in CO2 emissions globally.

This report applies data analytics to visualise this increase and its extent.

The Python coding language is used to facilitate this analysis and uses 2 main data sources for values.

## Introduction

Awareness of global warming and climate change are at an all time high. As stewards of our planet it is important that we look to preserve our environment.  
  
Whilst the increase in greenhouse gases in our planet’s atmosphere are the result of an increase in the availability of technology, we can also use technology to monitor and combat these effects.

This report will use such technology to analyse the relationship between the CO2 content of our atmosphere in relation to the growth of aviation.

## Datasets

This report utilises two datasets for statistical analysis.

### Dataset 1 - Passenger (PAX) Data

The first dataset lists values for the number of air travel passengers per year between 1980 and 2020.

**Additional**:

In aviation, Passengers are commonly referred to as ‘PAX.   
This term will be used throughout this report and the supporting code.

**Data Source:** <https://www.iea.org/data-and-statistics/charts/world-air-passenger-traffic-evolution-1980-2020>

Source chosen because the vendor is a recognisedAviation body.

### Dataset 2 - CO2 Data

The second dataset lists values for greenhouse gas emissions by country.

**Data Source:**  
<https://data.worldbank.org/indicator/EN.ATM.CO2E.LF.KT?end=2016&start=1980>

Source chosen because the vendor is a recognised global body.

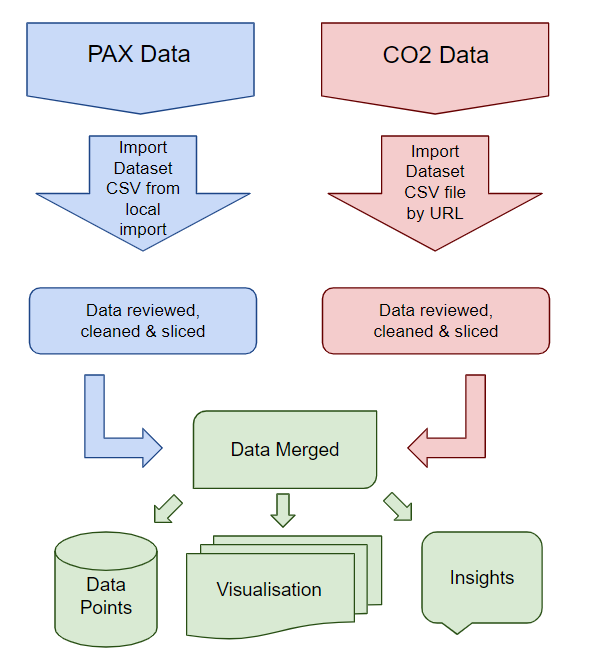
**Additional**:

The data represents CO2 quantity as a result of all fuel consumption, including cars etc.

## 

## Implementation Process

### Overview Flow Chart



### Implementation Process

This project was implemented using Python 3.11 and the Pycharm IDE.

The Python code is contained within a file called ‘main.py’.

In my code I have used the following Python packages:

1. **Pandas as pd**
   1. Used to modify and merge datasets.
2. **Numpy as NP**
   1. Used to import CSV file from local.
3. **Matplotlib.pyplot as plt**
   1. Used to create data visualisations.
4. **Warning**
   1. Used to suppress error messages related to future versions of Python.

The Print() function was used throughout development to ensure that results of functions worked as intended.

### Dataframe 1 Processing

This dataset was imported in CSV format using numpy loadtxt function and named csv1.

csv1 = np.loadtxt("Passengers.csv", delimiter=",", dtype=str)

Then converted to a Pandas dataframe named df1.

df1 = pd.DataFrame(csv1, columns =['Year','Total','Domestic','International'])

The dataset contains 4 columns which were named:

* Year
* Total
* International
* Domestic

Total column type was then changed to float to prevent visualisation error.

df1['Total'] = df1['Total'].astype(float)

As I intend to analyse data on a global scale, only the ‘Year’ & ‘Total’ columns are required.

The Pandas ‘drop’ method was used to remove these columns from the dataset.

df1 = df1.drop('Domestic', axis=1)

df1 = df1.drop('International', axis=1)

This dataset was very ‘clean’ and did not require any real manipulation to make it useful.

### Dataframe 2 Processing

This dataset was also imported in CSV format via url, using the Pandas read\_csv function and named ‘**df2**’

The dataset contains 8 columns:

* Location
* Indicator
* Subject
* Measure
* Time
* Value
* Flag Codes

After reviewing the data, it was clear that nations which are members of groups, were listed twice. First as their individual nation, and secondly as a part of the group.  
  
This creates value duplication. In order to prevent this, these following Locations have been removed:

1. EU27\_2020
2. EU28
3. G20
4. G7M
5. EA19

df2 = df2[~df2.LOCATION.str.contains('EU27\_2020')]

df2 = df2[~df2.LOCATION.str.contains('EU28')]

df2 = df2[~df2.LOCATION.str.contains('G20')]

df2 = df2[~df2.LOCATION.str.contains('G7M')]

df2 = df2[~df2.LOCATION.str.contains('EA19')]

The next step was to remove columns that were no longer required using the drop method.

df2 = df2.drop(['LOCATION','INDICATOR','SUBJECT','FREQUENCY','Flag Codes'], axis=1)

Columns removed:

* **Location**
  + Global values are required, not specific nations.
* **Indicator**
  + Data not useful or required.
* **Subject**
  + Data not useful or required.
* **Frequency**
  + Data not useful or required.
* **Flag Codes**
  + Data not useful or required.

Now that I am left with the columns that I do wish to analyse, I sliced the Measure column in order to only show values equal to MLN\_TONNES. This removed any entries that were not CO2 related, or were not measuring CO2 in the desired medium, while keeping the required Time and Value columns.

df2 = df2.loc[(df2['MEASURE'] == 'MLN\_TONNE'), ['TIME','Value']]

In order to create global value data by year, the values were grouped and summed by their corresponding year. The index was then reset using the reset\_index method to resolve index errors caused by the grouping.

df2 = df2.groupby('TIME')['Value'].sum().reset\_index()

### 

### Preparing Dataframes for merge

The dataframes had different year ranges. In order to keep these in line for both a clean merge, and to prevent missing data, the dataframes had years dropped in order to keep them equal.

Dataframe 1 year range: 1980-2020

Dataframe 2 year range: 1960-2021

Planned year range for both Dataframes: 1980-2021

In order to bring the Dataframes in-line by year, the following code was used to drop rows:

Drop the first 20 rows of Dataframe 2 (Years 1960-1979).

df2 = df2.drop(df2.index[range(20)])

Drop the last row of Dataframe 2 (Year 2021)

df2 = df2.drop(df2.index[-1]).reset\_index()

The reset\_index method was again used to prevent index related errors, however, upon printing the Dataframe for verification, the index had been turned into a new column.  
  
This was unexpected and not required and so was removed using the drop function.

df2 = df2.drop('index', axis=1)

### Merging the Dataframes

Now that I was content with the state of the 2 dataframes, I merged them by extracting the Value column from dataframe 2 and creating a new variable named ‘co2\_Value’.

co2\_Value = df2['Value']

co2\_Value was then inserted into Dataframe 1 as a new column, labelled ‘Value’.

df1.insert(1, 'Value', co2\_Value)

My updated Dataframe 1 now has 3 columns:

1. Time
2. Value
3. Total

I was not happy with these names as they were not very clear about the Data that they represented, and so they were renamed by updating the columns property of the dataframe:

1. Year
2. CO2
3. PAX

df1.columns = ['Year', 'CO2', 'PAX']

This now gives a clear indication that the columns represent the year the data was recorded, the CO2 values in millions of tonnes, and the Passengers (PAX) values in billions.

## 

## Results

#### Initial Graph - Evaluation

I created a simple line chart using matplotlib.subplots function showing 3 axis:

* X-Axis - Year
* Y-Axis - CO2
  + Line colour set to red, market set to ‘x’, Label: “CO2 (Mt)”

fig,ax = plt.subplots()

ax.plot(df1.Year, df1.CO2, color="red", marker="x")

ax.set\_xlabel("Year", fontsize = 14)

ax.set\_ylabel("CO2 (bn)", color="red", fontsize=14)

* Z-Axis - PAX
  + Line colour set to blue, marker to set ‘x’, Label: “PAX (bn)”

In order to create a 2nd Y-Axis (Z-Axis) I used the matplotlib.twinx function.

ax2=ax.twinx()

ax2.plot(df1.Year, df1.PAX, df1["PAX"],color="blue",marker="x")

ax2.set\_ylabel("PAX (bn)",color="blue",fontsize=14)

Finally I added a grid to the chart for easier viewing, and created it using the matplotlib.grid and matplotlib.show functions:  
  
plt.grid()

plt.show()

This code and dataframe resulted in the following line chart:

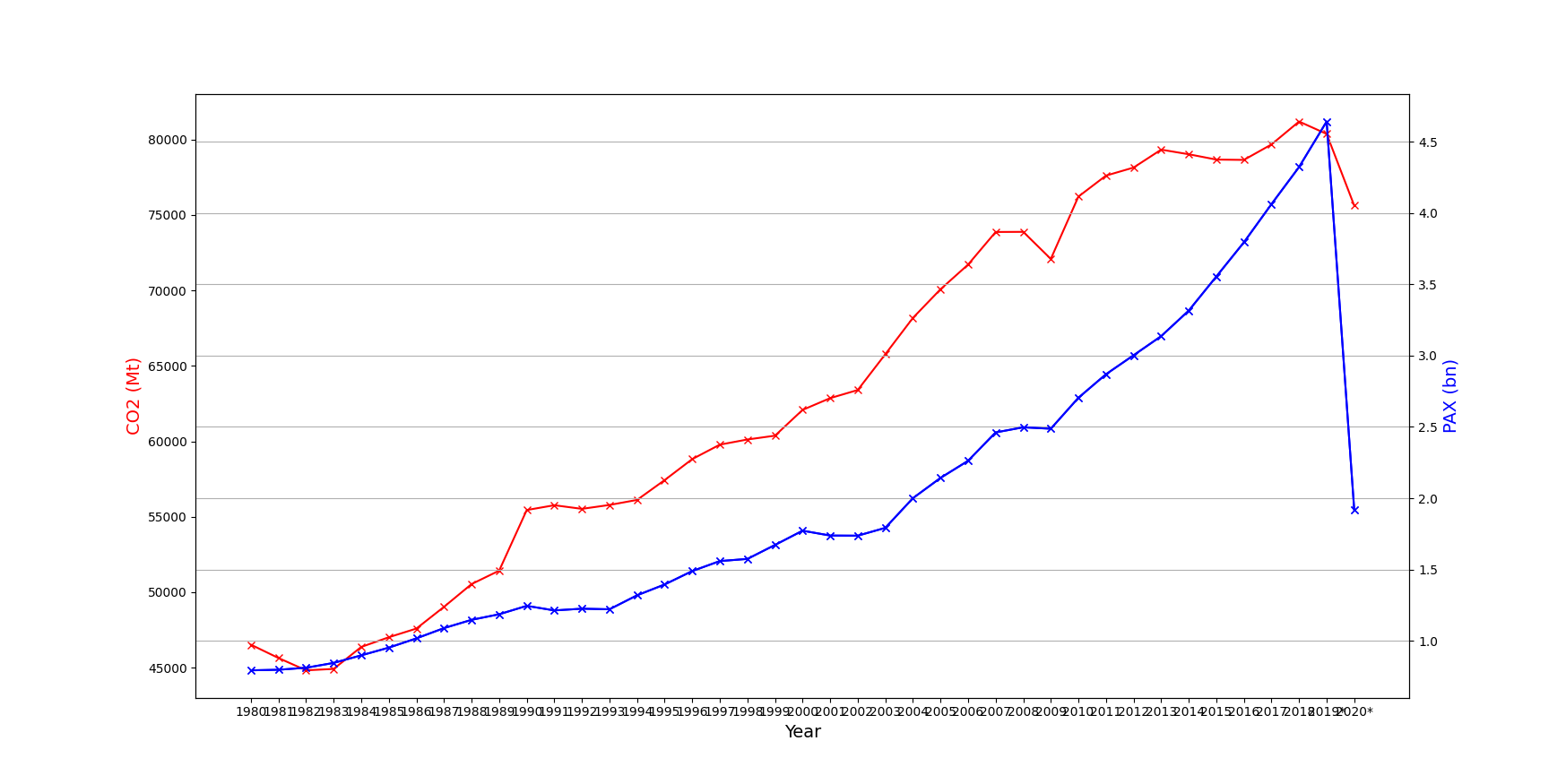


Chart 1

[Github link for improved viewing](https://github.com/FrostDan/UCD_Data_Course/blob/4a98dd1ef0ef0b162c552ee665e5b5ea5bbae69e/Graph%201.png)

With this first attempt I was not happy with the following aspects of this chart:

* X-Axis not easy to read
* Years 2019 & 2020 contain an asterisk.
* Changing CO2 values from millions to billions would decrease label size.
* Grid on X-Axis only.

#### Initial Graph - Fixes

##### Asterisks in Year Data

Asterisks in data for years 2019 & 2020 are unrequired so removed by converting the Year column’s Float type to a String type and extracting the asterisks to be replaced with nothing:

df1['Year'] = df1['Year'].astype(str).str.extract('(\d+)')

##### Year Values on X-Axis

In order to make the X-Axis easier to read, I used a For loop in order to rotate each label by 90°.

for tick in ax.get\_xticklabels():

tick.set\_rotation(90)

##### Convert CO2 Values from Millions to Billions

CO2 values divided by 1000 to convert from millions to billions.

Label updated to reflect change.

df1['CO2'] = df1['CO2']/1000

Added title to the graph for presentation.

ax.set\_title('Compare CO2 to Passenger over Time')

##### Grid on X-Axis only

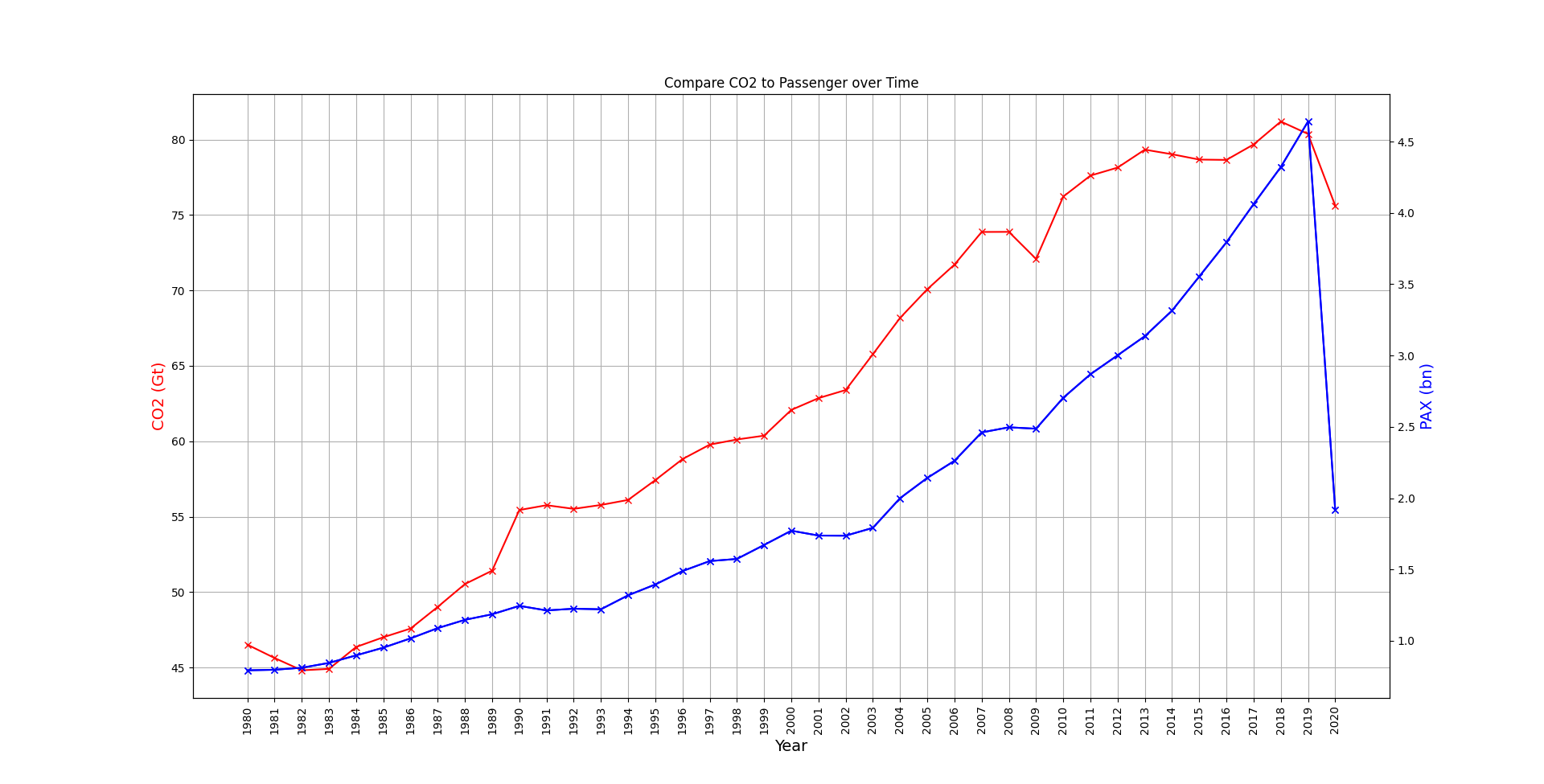
Create grid for X-Axis & Y-Axis:

plt.grid() updated to ax.grid(True)

#### 

#### Updated Graph

Graph recreated with new properties.



Graph 2

[Github link for improved viewing](https://github.com/FrostDan/UCD_Data_Course/blob/895000e7cd3154d0be091df82e515d7f17a65932/Graph%202.png)

## 

## Insights

### Insight 1 - High level analysis

As a high level analysis we can conclude that between the dates 1980 and 2019, the number of Passengers using air travel has had a steady increase.

This has had a mirror effect on CO2 quantity in the atmosphere caused by liquid fuel consumption.

### Insight 2 - World Event: Gulf War

From 1983 to 1990 we can see a sharp increase in Passengers, however in 1990 the Gulf War began and it would seem this event caused the Passenger rate to plateau until 1993.  
While the Gulf War ended in 1991, it is reasonable to suggest there was a lasting negative effect for the following 2 years.

This has a relative effect on CO2 consumption which also shows a decline based on the same timeframe. The Gulf war had a negative impact on global crude oil supply, and as such a sharp increase in crude oil prices which suggests its reduction in use.

### Insight 3 - World Events: Dot-Com bust & 9-11

The data suggests that around the year 2000, passenger numbers again began to decrease after a steady increase until 2003 (at which point it returned to passenger numbers similar to the year 2000).

.  
This is likely due to two world events:

1. Year 2000 - The ‘dot-com’ bust, a stock market crash.
2. Year 2021 - 9-11, a terror related event involving aircraft.

While these events had a clear negative impact on passenger rates, it would seem the consumption of fuel did not decrease and so CO2 levels in our atmosphere continued to increase.

### Insight 4 - World Event: 2007-2008 Financial Crisis.

The data suggests that between 2007 and 2009 the global passenger rate again temporarily slowed and then declined.

This is likely due to the 2007-2008 Financial Crisis, which created a global recession  
In 2009 various countries, including the USA promoted stimulus packages to help end the global recession.

The CO2 data correlates with the passenger data here and we can see it plateaus for roughly a year before a sharp decline within the same time frame.

### Insight 5 - World Event: COVID-19

In 2019 the world was hit by COVID-19 Pandemic.  
This created new restrictions to the movement of people on a global level and as such we can see an unprecedented decline in air passengers of roughly 41%, reducing passenger rates to that of 2003-2004.

Meanwhile, non-passenger air services such as freight would have continued, especially with new freight flights issued for global distribution of medicine such as vaccines and PPE for medical staff and the general public.

The CO2 data reflects this by showing that CO2 levels only decreased by roughly 10%.

## 

## References

**Dataset References**: Sources provided within Dataset sections, detailed earlier in this report.

**Insights References:**

Gulf War  
[Post Gulf War oil supply (OSTI.gov)](https://www.osti.gov/biblio/5366825#:~:text=With%20Iraq%20and%20Kuwait%20production,%2C%20worldwide%2C%20above%20normal%20levels.)

Dot-Com bust  
[Dot-Com Bubble (Wikipedia)](https://en.wikipedia.org/wiki/Dot-com_bubble)

9-11  
[How the Airline Industry Survived SARS, 9/11, the Global Recession and More (Apex.aero)](https://apex.aero/articles/aftershocks-coronavirus-impact/)

2007-2008 Financial Crisis  
[What Caused the Great Recession in 2008—and What Can We Learn From It? (Acorns.com)](https://www.acorns.com/learn/investing/what-caused-great-recession-of-2008/)

COVID-19

[Taking stock of the pandemic’s impact on global aviation (McKinsey & Company)](https://www.mckinsey.com/industries/travel-logistics-and-infrastructure/our-insights/taking-stock-of-the-pandemics-impact-on-global-aviation)