investigate-a-dataset-project

August 31, 2018

1 Project: CO2 emissions, GDP and population growth in the last century

1.1 Table of Contents

1.2 Introduction

For this project I will analyze data from gapminder.org to explore worldwide CO2 emissions and their relationship to other variables. To start, I will explore CO2 emissions per country to discover trends. I will use per capita CO2 emissions as the dependent variable and time, GDP per capita and population growth as independent variables for my anaylsis.

The goals of this project (aside from learning and practicing data analytics) are to answer the following questions: 1. What trends can we see in CO2 emissions over the past century? 2. What is the relationship between CO2 emissions and GDP? 3. What is the relationship between CO2 emissions and population growth?

The world is dealing with unprecedented climate change. A major cause of this change is the rise in greenhouse gas emissions (of which CO2 is one). Understanding recent trends in CO2 emissions and their relationship with other variables is important if we want evidence-based measures to address climate change. My analysis will be limited in that it only looks at correlation, but it cannot determine causation. Also, the variables I will be analyzing are probably correlated, and my analysis will not try to disentangle them to determine their isolated effects on CO2 emissions.

In [1]:
```
# Import packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
% matplotlib inline
```

1.3 Data Wrangling

1.3.1 General Properties

In [2]:
```
# Load data
df_co2 = pd.read_csv('co2_emissions_tonnes_per_person.csv')
df_regions = pd.read_csv('regions.csv')
df_gdp = pd.read_csv('income_per_person_gdppercapita_ppp_inflation_adjusted.csv')
df_pop_growth = pd.read_csv('population_growth_annual_percent.csv')

# Rename first column to more descriptive 'country'
df_co2.rename(columns={"geo": "country"}, inplace=True)
df_gdp.rename(columns={"geo": "country"}, inplace=True)
df_pop_growth.rename(columns={"geo": "country"}, inplace=True)
```

In [3]:
```
# Print head of CO2 df
df_co2.head()
```

Out[3]:

	country	1800	1801	1802	1803	1804	1805	1806	1807	1808	...	\
0	Afghanistan	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	
1	Albania	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	
2	Algeria	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	
3	Andorra	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	
4	Angola	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	

	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014
0	0.0529	0.0637	0.0854	0.154	0.242	0.294	0.412	0.35	0.316	0.299
1	1.3800	1.2800	1.3000	1.460	1.480	1.560	1.790	1.68	1.730	1.960
2	3.2200	2.9900	3.1900	3.160	3.420	3.300	3.290	3.46	3.510	3.720
3	7.3000	6.7500	6.5200	6.430	6.120	6.120	5.870	5.92	5.900	5.830
4	0.9800	1.1000	1.2000	1.180	1.230	1.240	1.250	1.33	1.250	1.290

[5 rows x 216 columns]

In [4]:
```
# Print head of df_regions
df_regions.head()
```

Out[4]:

	geo	name	four_regions	eight_regions	\
0	afg	Afghanistan	asia	asia_west	
1	alb	Albania	europe	europe_east	
2	dza	Algeria	africa	africa_north	
3	and	Andorra	europe	europe_west	
4	ago	Angola	africa	africa_sub_saharan	

	six_regions	members_oecd_g77	Latitude	Longitude	\
0	south_asia	g77	33.00000	66.00000	
1	europe_central_asia	others	41.00000	20.00000	
2	middle_east_north_africa	g77	28.00000	3.00000	
3	europe_central_asia	others	42.50779	1.52109	

```
4        sub_saharan_africa              g77 -12.50000   18.50000

  UN member since           World bank region World bank income group 2017
0      19/11/1946                  South Asia                     Low income
1      14/12/1955       Europe & Central Asia           Upper middle income
2       8/10/1962  Middle East & North Africa           Upper middle income
3       28/7/1993       Europe & Central Asia                   High income
4       1/12/1976          Sub-Saharan Africa           Lower middle income
```

In [5]:
```
# Print head of GDP df
df_gdp.head()
```

Out[5]:
```
       country  1800  1801  1802  1803  1804  1805  1806  1807  1808  ...    \
0  Afghanistan   603   603   603   603   603   603   603   603   603  ...
1      Albania   667   667   667   667   667   668   668   668   668  ...
2      Algeria   715   716   717   718   719   720   721   722   723  ...
3      Andorra  1200  1200  1200  1200  1210  1210  1210  1210  1220  ...
4       Angola   618   620   623   626   628   631   634   637   640  ...

    2009   2010   2011   2012   2013   2014   2015   2016   2017   2018
0   1530   1610   1660   1840   1810   1780   1750   1740   1800   1870
1   9530   9930  10200  10400  10500  10700  11000  11400  11900  12400
2  12600  12900  13000  13200  13300  13500  13700  14000  13800  13700
3  41700  39000  42000  41900  43700  44900  46600  48200  49800  51500
4   5910   5900   5910   6000   6190   6260   6230   6030   5940   5850

[5 rows x 220 columns]
```

In [6]:
```
# Print head of pop growth df
df_pop_growth.head()
```

Out[6]:
```
       country  1960  1961  1962  1963  1964  1965  1966  1967  1968  ...    \
0  Afghanistan  1.82  1.88  1.94  1.99  2.05  2.11  2.13  2.15  2.21  ...
1      Albania  3.02  3.12  3.06  2.95  2.88  2.75  2.63  2.63  2.84  ...
2      Algeria  2.51  2.49  2.47  2.49  2.56  2.66  2.76  2.84  2.88  ...
3      Andorra  7.05  6.94  6.69  6.56  6.24  6.00  5.75  5.50  5.31  ...
4       Angola  1.90  1.93  1.95  1.93  1.87  1.79  1.70  1.65  1.68  ...

    2007   2008   2009    2010   2011   2012   2013   2014   2015   2016
0  2.760  2.510  2.570  2.8100  3.100  3.270  3.320  3.180  2.940  2.690
1 -0.756 -0.767 -0.674 -0.4960 -0.269 -0.165 -0.183 -0.207 -0.291 -0.160
2  1.530  1.620  1.720  1.8200  1.920  2.010  2.040  2.000  1.920  1.830
3  2.070  1.410  0.714 -0.0154 -0.830 -1.590 -2.010 -1.960 -1.540 -0.944
4  3.560  3.560  3.570  3.5700  3.570  3.560  3.530  3.490  3.430  3.370

[5 rows x 58 columns]
```

CO2 DataFrame

In [7]:
```
# Check for missing data
missing = df_co2.isnull().sum().sum()
total = df_co2.shape[0] * df_co2.shape[1]
print('Missing {} out of {} data points ({}%)'.format(missing, total, round(missing*10
```

```
Missing 24375 out of 41472 data points (59.0%)
```

In [8]:
```
# Get an idea for where missing data are located
df_co2.isnull().sum()
```

Out[8]:
```
country      0
1800       187
1801       187
1802       185
1803       187
1804       186
1805       187
1806       187
1807       186
1808       187
1809       187
1810       186
1811       186
1812       186
1813       186
1814       186
1815       186
1816       186
1817       186
1818       186
1819       185
1820       185
1821       185
1822       185
1823       185
1824       185
1825       185
1826       185
1827       185
1828       185
           ...
1985        20
1986        20
1987        20
1988        20
1989        20
1990        16
```

```
1991        15
1992         4
1993         4
1994         3
1995         3
1996         3
1997         3
1998         3
1999         3
2000         3
2001         3
2002         2
2003         2
2004         2
2005         2
2006         2
2007         1
2008         1
2009         1
2010         1
2011         1
2012         0
2013         0
2014         0
Length: 216, dtype: int64
```

As we can see above, this database has a lot of missing data (>50% of data points are missing). More data are available for that past 30 years or so than the first 30 years, for which only a few countries have data. This is to be expected since data collection has become easier and more widespread only in the last few decades. I suspect that this problem is worse for developing countries, which have fewer resources to focus on data collection, and could mean that my analysis could be biased if developed nations end up being over-represented in the data. For this reason, I will use several methods to fill in missing values as opposed to discarding countries.

GDP DataFrame

```
In [9]: # Check for missing data
        missing = df_gdp.isnull().sum().sum()
        total = df_gdp.shape[0] * df_gdp.shape[1]
        print('Missing {} out of {} data points ({}%)'.format(missing, total, round(missing*10
```

```
Missing 0 out of 42460 data points (0.0%)
```

Population Growth DataFrame

```
In [10]: # Check for missing data
         missing = df_pop_growth.isnull().sum().sum()
```

```
        total = df_pop_growth.shape[0] * df_pop_growth.shape[1]
        print('Missing {} out of {} data points ({}%)'.format(missing, total, round(missing*1
```

```
Missing 73 out of 11252 data points (1.0%)
```

In [11]:
```
# Get an idea for where missing data are located
df_pop_growth.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 194 entries, 0 to 193
Data columns (total 58 columns):
country    194 non-null object
1960       191 non-null float64
1961       192 non-null float64
1962       192 non-null float64
1963       192 non-null float64
1964       192 non-null float64
1965       192 non-null float64
1966       192 non-null float64
1967       192 non-null float64
1968       192 non-null float64
1969       192 non-null float64
1970       192 non-null float64
1971       192 non-null float64
1972       192 non-null float64
1973       192 non-null float64
1974       192 non-null float64
1975       192 non-null float64
1976       192 non-null float64
1977       192 non-null float64
1978       192 non-null float64
1979       192 non-null float64
1980       192 non-null float64
1981       192 non-null float64
1982       192 non-null float64
1983       192 non-null float64
1984       192 non-null float64
1985       192 non-null float64
1986       192 non-null float64
1987       192 non-null float64
1988       192 non-null float64
1989       192 non-null float64
1990       193 non-null float64
1991       193 non-null float64
1992       193 non-null float64
1993       193 non-null float64
1994       193 non-null float64
```

```
1995       193 non-null float64
1996       194 non-null float64
1997       194 non-null float64
1998       194 non-null float64
1999       194 non-null float64
2000       194 non-null float64
2001       194 non-null float64
2002       194 non-null float64
2003       194 non-null float64
2004       194 non-null float64
2005       194 non-null float64
2006       194 non-null float64
2007       194 non-null float64
2008       194 non-null float64
2009       194 non-null float64
2010       194 non-null float64
2011       193 non-null float64
2012       193 non-null float64
2013       193 non-null float64
2014       193 non-null float64
2015       193 non-null float64
2016       193 non-null float64
dtypes: float64(57), object(1)
memory usage: 88.0+ KB
```

The population growth DataFrame has few missing values. I will try to extrapolate some of these values, but ultimately getting rid of a couple of countries that still have missing values should not be very detrimental to my analysis.

1.3.2 Data Cleaning

CO2 DataFrame In order to deal with the missing data in the CO2 DataFrame, I will use several techniques. To start, I will use linear interpolation to fill the missing values which are in between two existing values.

```
In [12]: # Set 'country' column as index to allow for interpolation
         df_co2.set_index('country', inplace=True)

         # Use linear interpolation to fill missing values between two available values
         df_co2.interpolate(method='linear', axis=1, inplace=True)

         # Print number of remaining missing values
         df_co2.isnull().sum().sum()
```

```
Out[12]: 23722
```

While helpful, we still have 23,722 missing values after aplying interpolation . Next up, I will simply drop all years except for the last 100. While this number is somewhat arbitrary, analyzing the last 100 years should be enough to explore trends in CO2 emissions.

```
In [13]: df_co2 = df_co2.iloc[:,-100:]
         df_co2.head()
```

Out[13]:

country	1915	1916	1917	1918	1919	1920	1921	1922	1923	1924	...	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014
Afghanistan	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	0.0529	0.0637	0.0854	0.154	0.242	0.294	0.412	0.35	0.316	0.299
Albania	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	1.3800	1.2800	1.3000	1.460	1.480	1.560	1.790	1.68	1.730	1.960
Algeria	0.000582	0.00064	0.00126	0.0031	0.00306	0.00363	0.00418	0.00413	0.00233	0.0046	...	3.2200	2.9900	3.1900	3.160	3.420	3.300	3.290	3.46	3.510	3.720
Andorra	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	7.3000	6.7500	6.5200	6.430	6.120	6.120	5.870	5.92	5.900	5.830
Angola	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	0.9800	1.1000	1.2000	1.180	1.230	1.240	1.250	1.33	1.250	1.290

[5 rows x 100 columns]

```
In [14]: # Check for missing data
         missing = df_co2.isnull().sum().sum()
         total = df_co2.shape[0] * df_co2.shape[1]
         print('Missing {} out of {} data points ({}%)'.format(missing, total, round(missing*1
```

```
Missing 5201 out of 19200 data points (27.0%)
```

As seen above, we still have over 25% of missing values, even after dropping the years with a higher prevalence of missing values. I'd like to fill the remaining missing values with an average of CO2 emissions for each year. However, in order to avoid clumping regions with high emissions with regions with low emissions, I'd like to first group countries by region and use the region mean to populate missing values for other countries in that region. To do this, we'll have to use the DataFrame containing each country's region (df_regions).

```
In [15]: # Drop columns we won't be using for 'regions' DataFrame
         df_regions.drop(['geo', 'members_oecd_g77', 'Latitude', 'Longitude', 'UN member since

         # Rename first column to more descriptive 'country'
         df_regions.rename(columns={"name": "country"}, inplace=True)
         df_regions.head()
```

Out[15]:
```
       country four_regions       eight_regions               six_regions  \
0  Afghanistan         asia           asia_west                south_asia
1      Albania       europe         europe_east       europe_central_asia
2      Algeria       africa        africa_north  middle_east_north_africa
3      Andorra       europe         europe_west       europe_central_asia
4       Angola       africa  africa_sub_saharan        sub_saharan_africa

            World bank region World bank income group 2017
0                  South Asia                   Low income
1       Europe & Central Asia          Upper middle income
2  Middle East & North Africa          Upper middle income
3       Europe & Central Asia                  High income
4          Sub-Saharan Africa          Lower middle income
```

In [16]:
```
# Create new df with merged data
df_co2_w_region = df_regions.merge(df_co2, on='country')
df_co2_w_region.head()
```

Out[16]:
```
       country four_regions       eight_regions               six_regions  \
0  Afghanistan         asia           asia_west                south_asia
1      Albania       europe         europe_east       europe_central_asia
2      Algeria       africa        africa_north  middle_east_north_africa
3      Andorra       europe         europe_west       europe_central_asia
4       Angola       africa  africa_sub_saharan        sub_saharan_africa

            World bank region World bank income group 2017      1915     1916  \
0                  South Asia                   Low income       NaN      NaN
1       Europe & Central Asia          Upper middle income       NaN      NaN
2  Middle East & North Africa          Upper middle income  0.000582  0.00064
3       Europe & Central Asia                  High income       NaN      NaN
4          Sub-Saharan Africa          Lower middle income       NaN      NaN

      1917    1918  ...      2005    2006    2007   2008   2009   2010   2011  \
0      NaN     NaN  ...    0.0529  0.0637  0.0854  0.154  0.242  0.294  0.412
1      NaN     NaN  ...    1.3800  1.2800  1.3000  1.460  1.480  1.560  1.790
2  0.00126  0.0031  ...    3.2200  2.9900  3.1900  3.160  3.420  3.300  3.290
3      NaN     NaN  ...    7.3000  6.7500  6.5200  6.430  6.120  6.120  5.870
4      NaN     NaN  ...    0.9800  1.1000  1.2000  1.180  1.230  1.240  1.250

   2012   2013   2014
0  0.35  0.316  0.299
1  1.68  1.730  1.960
2  3.46  3.510  3.720
3  5.92  5.900  5.830
4  1.33  1.250  1.290

[5 rows x 106 columns]
```

In [17]:
```
# Show count of non-missing values per year
```

```
In [17]: df_co2_w_region.groupby('World bank region').count().iloc[:,5:]
```

Out[17]:

World bank region	1915	1916	1917	1918	1919	1920	1921	1922
East Asia & Pacific	7	7	7	7	7	7	7	7
Europe & Central Asia	36	36	36	36	36	36	36	36
Latin America & Caribbean	7	7	8	8	8	8	9	9
Middle East & North Africa	3	4	4	4	4	4	4	4
North America	2	2	2	2	2	2	2	2
South Asia	1	1	1	1	1	1	1	1
Sub-Saharan Africa	3	3	3	3	3	4	4	4

World bank region	1923	1924	...	2005	2006	2007	2008	2009
East Asia & Pacific	7	7	...	30	30	30	30	30
Europe & Central Asia	36	36	...	49	49	50	50	50
Latin America & Caribbean	9	9	...	33	33	33	33	33
Middle East & North Africa	4	5	...	21	21	21	21	21
North America	2	2	...	2	2	2	2	2
South Asia	1	1	...	8	8	8	8	8
Sub-Saharan Africa	4	4	...	47	47	47	47	47

World bank region	2010	2011	2012	2013	2014
East Asia & Pacific	30	30	30	30	30
Europe & Central Asia	50	50	50	50	50
Latin America & Caribbean	33	33	33	33	33
Middle East & North Africa	21	21	21	21	21
North America	2	2	2	2	2
South Asia	8	8	8	8	8
Sub-Saharan Africa	47	47	48	48	48

[7 rows x 100 columns]

When grouping by 'World Bank region' above, I can confirm my suspicion that developing countries will have more missing data than developed nations. As we can see, only 3/48 countries in Sub-Saharan Africa have data for the first few years, while 36/50 and 2/2 have data in Europe & Central Asia and North America, respectively. I will proceed with imputation based on mean by World Bank region, but we should keep in mind that earlier data will be less reliable than data for the past few years.

```
In [18]: # Show mean CO2 emissions by region
         df_co2_w_region.groupby('World bank region').mean()
```

Out[18]:

World bank region	1915	1916	1917	1918
East Asia & Pacific	1.422719	1.378370	1.387049	1.432520
Europe & Central Asia	1.989370	1.976942	1.706283	1.520525
Latin America & Caribbean	0.591614	0.637214	0.632725	0.748413

Middle East & North Africa	0.045067	0.050202	0.074140	0.098050	
North America	10.405000	11.965000	13.050000	13.550000	
South Asia	0.140000	0.140000	0.146000	0.163000	
Sub-Saharan Africa	1.122000	1.299200	1.375700	1.275200	

World bank region	1919	1920	1921	1922	\
East Asia & Pacific	1.340066	1.339243	1.219523	1.246697	
Europe & Central Asia	1.386217	1.630919	1.494358	1.708871	
Latin America & Caribbean	0.809500	1.169275	1.192974	1.167844	
Middle East & North Africa	0.111365	0.128807	0.166670	0.202457	
North America	11.305000	12.770000	10.890000	10.150000	
South Asia	0.177000	0.143000	0.153000	0.149000	
Sub-Saharan Africa	1.300633	1.064617	1.036808	0.877800	

World bank region	1923	1924	...	2005	\
East Asia & Pacific	1.284447	1.345371	...	3.988000	
Europe & Central Asia	1.794462	2.022525	...	7.008082	
Latin America & Caribbean	1.154022	1.304967	...	3.272970	
Middle East & North Africa	0.246458	0.226140	...	10.612381	
North America	13.300000	11.435000	...	18.450000	
South Asia	0.153000	0.163000	...	0.689863	
Sub-Saharan Africa	1.045500	1.080725	...	0.933315	

World bank region	2006	2007	2008	2009	\
East Asia & Pacific	3.934233	4.247133	4.425667	4.413100	
Europe & Central Asia	7.116735	6.987240	6.945940	6.359560	
Latin America & Caribbean	3.421030	3.517848	3.556333	3.578333	
Middle East & North Africa	10.780238	10.420714	10.372762	10.024762	
North America	17.900000	18.050000	17.650000	16.550000	
South Asia	0.753188	0.772887	0.819000	0.854000	
Sub-Saharan Africa	0.947328	0.931045	0.945347	0.891613	

World bank region	2010	2011	2012	2013	\
East Asia & Pacific	4.513300	4.646500	4.644767	4.715033	
Europe & Central Asia	6.669260	6.519420	6.325340	6.244720	
Latin America & Caribbean	3.665303	3.647152	3.673727	3.711455	
Middle East & North Africa	9.880952	9.580714	10.039429	9.522286	
North America	16.550000	16.300000	15.550000	15.550000	
South Asia	0.889750	0.975125	1.033625	1.024250	
Sub-Saharan Africa	0.892687	0.889591	0.877654	0.872792	

World bank region	2014
East Asia & Pacific	4.613100

```
Europe & Central Asia          5.987400
Latin America & Caribbean      3.631212
Middle East & North Africa     9.978714
North America                 15.800000
South Asia                     1.140000
Sub-Saharan Africa             0.929798

[7 rows x 100 columns]
```

In [19]:
```
# Copy df to be cleaned
df_co2_clean = df_co2_w_region

# Replace missing values with in-group mean by World Bank region
df_co2_clean.iloc[:,6:] = df_co2_clean.groupby("World bank region").transform(lambda x
# Show cleaned DF
df_co2_clean.head()
```

Out[19]:

	country	four_regions	eight_regions	six_regions	\
0	Afghanistan	asia	asia_west	south_asia	
1	Albania	europe	europe_east	europe_central_asia	
2	Algeria	africa	africa_north	middle_east_north_africa	
3	Andorra	europe	europe_west	europe_central_asia	
4	Angola	africa	africa_sub_saharan	sub_saharan_africa	

	World bank region	World bank income group 2017	1915	\
0	South Asia	Low income	0.140000	
1	Europe & Central Asia	Upper middle income	1.989370	
2	Middle East & North Africa	Upper middle income	0.000582	
3	Europe & Central Asia	High income	1.989370	
4	Sub-Saharan Africa	Lower middle income	1.122000	

	1916	1917	1918	...	2005	2006	2007	2008	2009	\
0	0.140000	0.146000	0.163000	...	0.0529	0.0637	0.0854	0.154	0.242	
1	1.976942	1.706283	1.520525	...	1.3800	1.2800	1.3000	1.460	1.480	
2	0.000640	0.001260	0.003100	...	3.2200	2.9900	3.1900	3.160	3.420	
3	1.976942	1.706283	1.520525	...	7.3000	6.7500	6.5200	6.430	6.120	
4	1.299200	1.375700	1.275200	...	0.9800	1.1000	1.2000	1.180	1.230	

	2010	2011	2012	2013	2014
0	0.294	0.412	0.35	0.316	0.299
1	1.560	1.790	1.68	1.730	1.960
2	3.300	3.290	3.46	3.510	3.720
3	6.120	5.870	5.92	5.900	5.830
4	1.240	1.250	1.33	1.250	1.290

[5 rows x 106 columns]

As we can see above, now I have a clean DataFrame with no missing values.

GDP DataFrame As we saw above, the GDP DataFrame has no missing values. The only data cleaning necessary will be to trim the DataFrame to include only the years we are interested in for this analysis.

```
In [20]: # Trim dataframe to include years 1915-2014
         df_gdp.set_index('country', inplace=True)
         df_gdp = df_gdp.iloc[:, -104:-4]
         df_gdp.head()
```

```
Out[20]:              1915  1916  1917  1918  1919  1920  1921  1922  1923  1924  \
         country
         Afghanistan   837   841   845   849   853   857   863   868   874   880
         Albania      1420  1440  1460  1480  1500  1520  1540  1560  1580  1600
         Algeria      2170  2210  2240  2280  2320  2360  2420  2480  2540  2600
         Andorra      3850  3920  3980  4050  4120  4190  4260  4330  4410  4480
         Angola       1080  1110  1140  1170  1210  1250  1290  1330  1370  1410

                       ...   2005   2006   2007   2008   2009   2010   2011   2012  \
         country       ...
         Afghanistan   ...   1140   1160   1290   1300   1530   1610   1660   1840
         Albania       ...   7460   7920   8450   9150   9530   9930  10200  10400
         Algeria       ...  12300  12300  12600  12700  12600  12900  13000  13200
         Andorra       ...  39800  42700  43400  41400  41700  39000  42000  41900
         Angola        ...   3950   4600   5440   5980   5910   5900   5910   6000

                       2013   2014
         country
         Afghanistan   1810   1780
         Albania      10500  10700
         Algeria      13300  13500
         Andorra      43700  44900
         Angola        6190   6260

         [5 rows x 100 columns]
```

Population Growth DataFrame As seen above, the population growth DataFrame has a few missing values (<1%). I will start by simply removing the last two years, since we don't have CO2 data for those years. After that, I will try to interpolate the missing values; if this is not possible, I will simply drop those countries that still have missing values.

```
In [21]: # Trim last 2 years of data
         df_pop_growth = df_pop_growth.iloc[:,:-2]
         df_pop_growth.head()
```

```
Out[21]:        country  1960  1961  1962  1963  1964  1965  1966  1967  1968  ...  \
         0  Afghanistan  1.82  1.88  1.94  1.99  2.05  2.11  2.13  2.15  2.21  ...
         1      Albania  3.02  3.12  3.06  2.95  2.88  2.75  2.63  2.63  2.84  ...
         2      Algeria  2.51  2.49  2.47  2.49  2.56  2.66  2.76  2.84  2.88  ...
```

```
3      Andorra  7.05  6.94  6.69  6.56  6.24  6.00  5.75  5.50  5.31  ...
4       Angola  1.90  1.93  1.95  1.93  1.87  1.79  1.70  1.65  1.68  ...

    2005   2006   2007   2008   2009    2010   2011   2012   2013   2014
0  3.870  3.230  2.760  2.510  2.570  2.8100  3.100  3.270  3.320  3.180
1 -0.512 -0.631 -0.756 -0.767 -0.674 -0.4960 -0.269 -0.165 -0.183 -0.207
2  1.380  1.460  1.530  1.620  1.720  1.8200  1.920  2.010  2.040  2.000
3  3.380  2.660  2.070  1.410  0.714 -0.0154 -0.830 -1.590 -2.010 -1.960
4  3.580  3.570  3.560  3.560  3.570  3.5700  3.570  3.560  3.530  3.490

[5 rows x 56 columns]
```

In [22]:
```
# Set 'country' column as index to allow for interpolation
df_pop_growth.set_index('country', inplace=True)

# Use linear interpolation to fill missing values between two available values
df_pop_growth.interpolate(method='linear', axis=1, inplace=True)

# Print number of remaining missing values
df_pop_growth.isnull().sum().sum()
```

Out[22]: 62

In [23]:
```
# Identify countries to be dropped
df_pop_growth[df_pop_growth.isnull().any(axis=1)]
```

Out[23]:
```
             1960  1961  1962  1963  1964  1965  1966  1967  1968  1969  ...    \
country                                                                  ...
Palestine     NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN  ...
Serbia        NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN  ...
Seychelles    NaN  2.81  2.65  2.54  2.51  2.51  2.49  2.46  2.42  2.38  ...

              2005   2006   2007   2008   2009   2010   2011   2012   2013  \
country
Palestine    2.560  2.560  2.560  2.880  2.890  2.900  3.000  3.010  2.980
Serbia      -0.300 -0.393 -0.405 -0.426 -0.401 -0.402 -0.789 -0.485 -0.487
Seychelles   0.463  2.080  0.511  2.240  0.393  2.790 -2.630  0.981  1.850

              2014
country
Palestine    2.960
Serbia      -0.469
Seychelles   1.560

[3 rows x 55 columns]
```

In [24]:
```
# Drop all rows with missing values
df_pop_growth.dropna(inplace=True)
df_pop_growth.shape
```

```
Out[24]: (191, 55)
```

I now have a DataFrame with no missing values and managed to keep 191 countries to be used for the analysis.

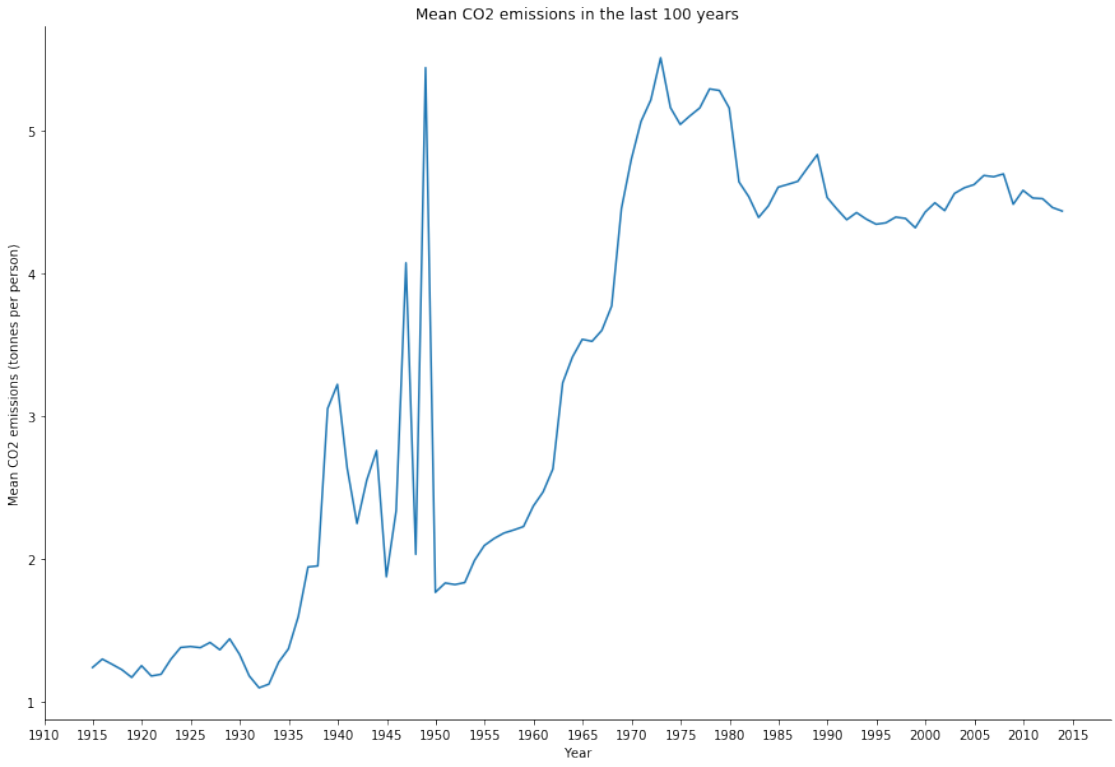
1.4 Exploratory Data Analysis

Now that I've trimmed and cleaned my data, I'm ready to move on to exploration.

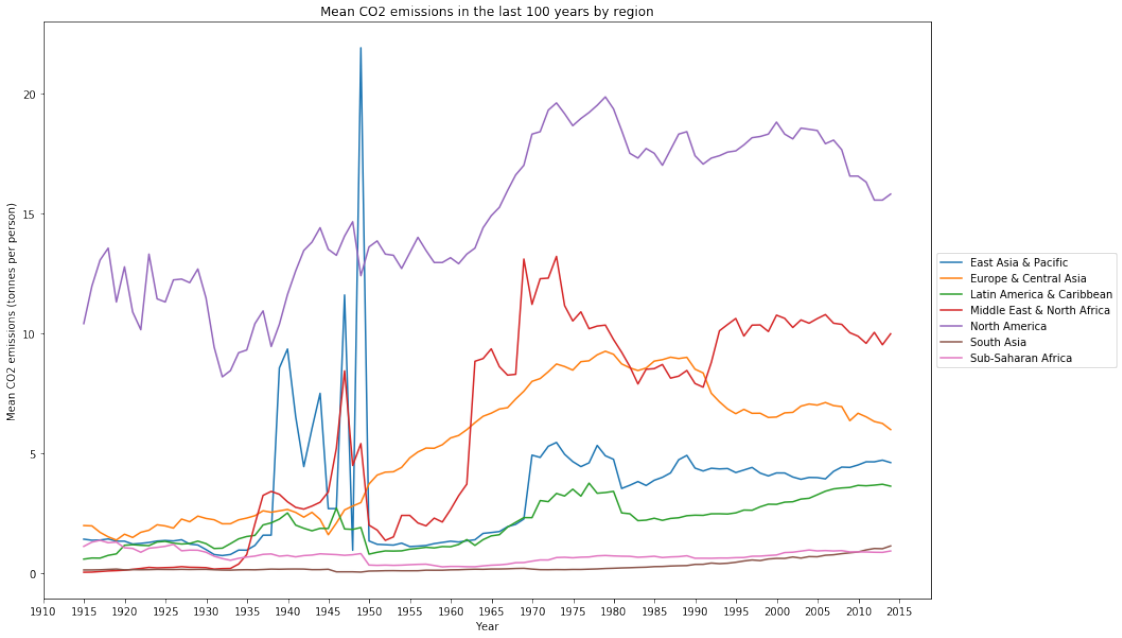
1.4.1 What trends can we see in CO2 emissions over the past century?

To help answer this question, I will start with a simple visualization of mean CO2 emissions over time, followed by a viasualization of this trend by region.

```
In [25]: # Plot mean CO2 emissions against time
         x = df_co2_clean.columns[6:].astype(int)
         y = df_co2_clean.mean()

         # Initialize figure
         fig, ax = plt.subplots(figsize=(15,10))

         # Format graph
         ax.set_title('Mean CO2 emissions in the last 100 years')
         ax.set_ylabel('Mean CO2 emissions (tonnes per person)')
         ax.set_xlabel('Year')
         ax.spines['top'].set_visible(False)
         ax.spines['right'].set_visible(False)

         # Plot
         ax.plot(x,y)

         # Change x ticks to every 5 years
         start, end = ax.get_xlim()
         ax.xaxis.set_ticks(np.arange(start, end, 5));
```



In [26]:
```
# Plot mean CO2 emissions against time by region
x = df_co2_clean.columns[6:].astype(int)
y = df_co2_clean.mean()

# Initialize figure
fig, ax = plt.subplots(figsize=(15,10))

# Format graph
ax.set_title('Mean CO2 emissions in the last 100 years by region')
ax.set_ylabel('Mean CO2 emissions (tonnes per person)')
ax.set_xlabel('Year')


# Plot
for i in range(df_co2_clean.groupby('World bank region').mean().shape[0]):
    ax.plot(x, df_co2_clean.groupby('World bank region').mean().iloc[i])

# Show legend
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))

# Change x ticks to every 5 years
start, end = ax.get_xlim()
ax.xaxis.set_ticks(np.arange(start, end, 5));
```



We can see some interesting trends in the above figures. To start, we can see a clear increase in CO2 emissions per capita from around 1.2 tonnes to around 4.5 tonnes in the past 100 years. But the increase hasn't always been smooth. There are two peaks in CO2 emissions, one around the 1940s and one in the 1970s. As for the one in the 1940s, I believe it is likely to be present due to spurious outliers, perhaps caused by errors in data gathering. I think this is the case because of the immense year-by-year variability which I would not expect. The peak in the 1970s is more likely to be real. The silver lining from this first figure is that, even though CO2 emissions are higher than they were 100 years ago, they are currently lower than they were 40 years ago.

The second figure gives us a more granular view of the trend. We can see that CO2 emissions have increased in every region except for Sub-Saharan Africa in the last 100 years. We can also see that CO2 emissions have been predominantly produced by North America (looking at you, USA). Importantly, the second figure seems to confirm that the spike in the 1940s is being produced by a handful of countries in East Asia & Pacific and Middle East & North Africa.

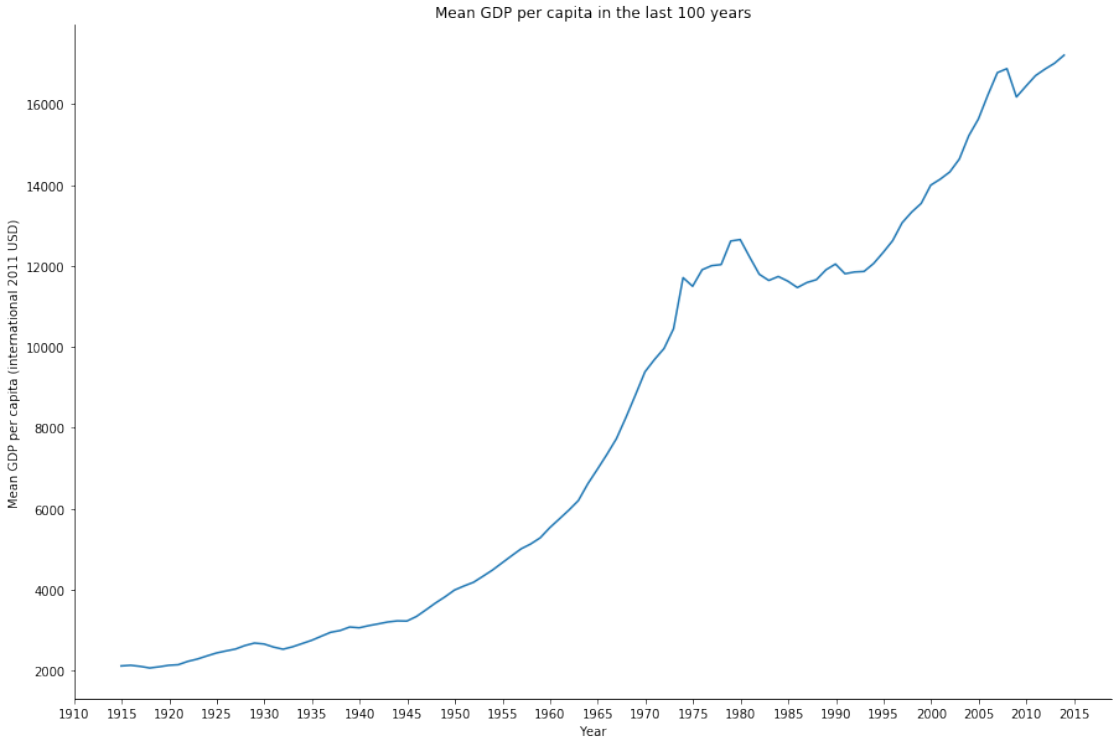
1.4.2 What is the relationship between CO2 emissions and GDP?

To explore this question we will first view the trend of per capita GDP in time, to see if it roughly follows the same upward path as CO2 emissions. I will also plot both GDP and CO2 emissions in the same graph so we can visualize both trends side by side.

Following this, I will plot a scattergraph to visualize the relationship better.

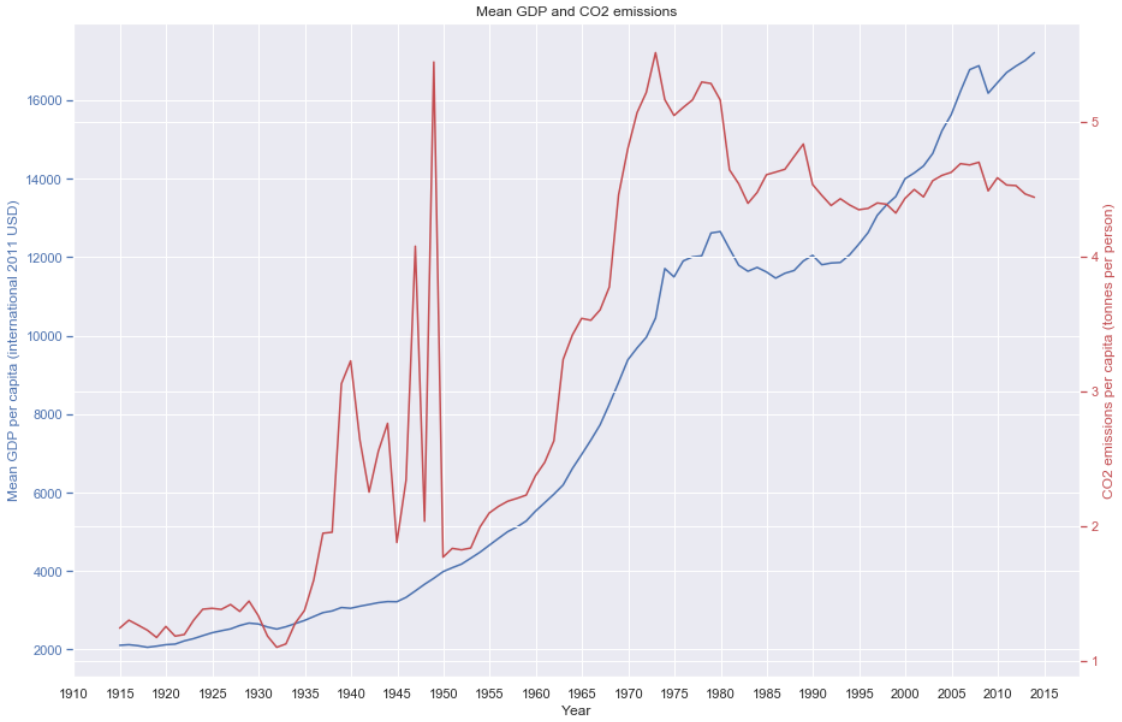
```
In [27]: # Plot mean GDP against time
         x = df_gdp.columns.astype(int)
         y = df_gdp.mean()

         # Initialize figure
         fig, ax = plt.subplots(figsize=(15,10))
```

```
        # Format graph
        ax.set_title('Mean GDP per capita in the last 100 years')
        ax.set_ylabel('Mean GDP per capita (international 2011 USD)')
        ax.set_xlabel('Year')
        ax.spines['top'].set_visible(False)
        ax.spines['right'].set_visible(False)

        # Plot
        ax.plot(x,y);

        # Change x ticks to every 5 years
        start, end = ax.get_xlim()
        ax.xaxis.set_ticks(np.arange(start, end, 5));
```



In [28]:
```
sns.set(style='darkgrid')

        # Plot mean GDP against time
        x = df_gdp.columns.astype(int)
        y1 = df_gdp.mean()
        y2 = df_co2_clean.mean()

        # Initialize figure
```

```
fig, ax1 = plt.subplots(figsize=(15,10))

# Format graph
ax1.set_title('Mean GDP and CO2 emissions')
ax1.set_ylabel('Mean GDP per capita (international 2011 USD)', color='b')
ax1.set_xlabel('Year')

# Make the y-axis label, ticks and tick labels match the line color.
ax1.tick_params('y', colors='b')
ax1.spines['top'].set_visible(False)

ax2 = ax1.twinx()
ax2.set_ylabel('CO2 emissions per capita (tonnes per person)', color='r')
ax2.tick_params('y', colors='r')

# Plot
ax1.plot(x, y1, 'b-')
ax2.plot(x, y2, 'r-')

# Change x ticks to every 5 years
start, end = ax1.get_xlim()
ax1.xaxis.set_ticks(np.arange(start, end, 5));
```



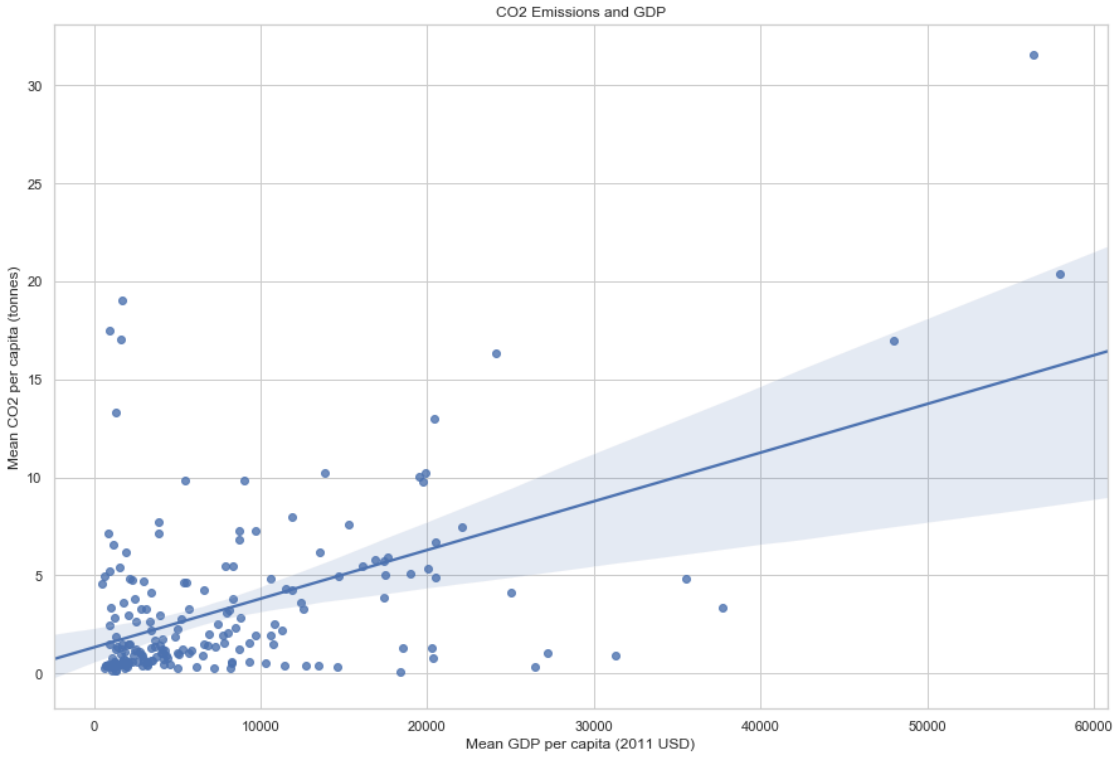
As we can see above, both CO2 emissions and GDP have increased over the past 100 years. They seem to be highly correlated (except for that outlying period in the 1940s), as they both seem

to have a peak around the 1970s. It is interesting to note that, after the financial crisis of 2007-2008, there seems to be some hope that an increase in GDP does not necessarily have to come with an increase in CO2 emissions, as emissions seem to be stable or decreasing while GDP rises.

In [29]:
```
# Assign first column as index
df_co2_clean.set_index('country', inplace=True)
```

In [30]:
```
# Set style
sns.set(style='whitegrid')

# Drop countries that are not in both dataframes
y = df_co2_clean.drop('Liechtenstein').iloc[:,5:].mean(axis=1)
x = df_gdp.drop(['Monaco', 'San Marino']).mean(axis=1)

fig, ax = plt.subplots(figsize=(15,10))

ax = sns.regplot(x, y)
ax.set(xlabel='Mean GDP per capita (2011 USD)', ylabel='Mean CO2 per capita (tonnes)'
```

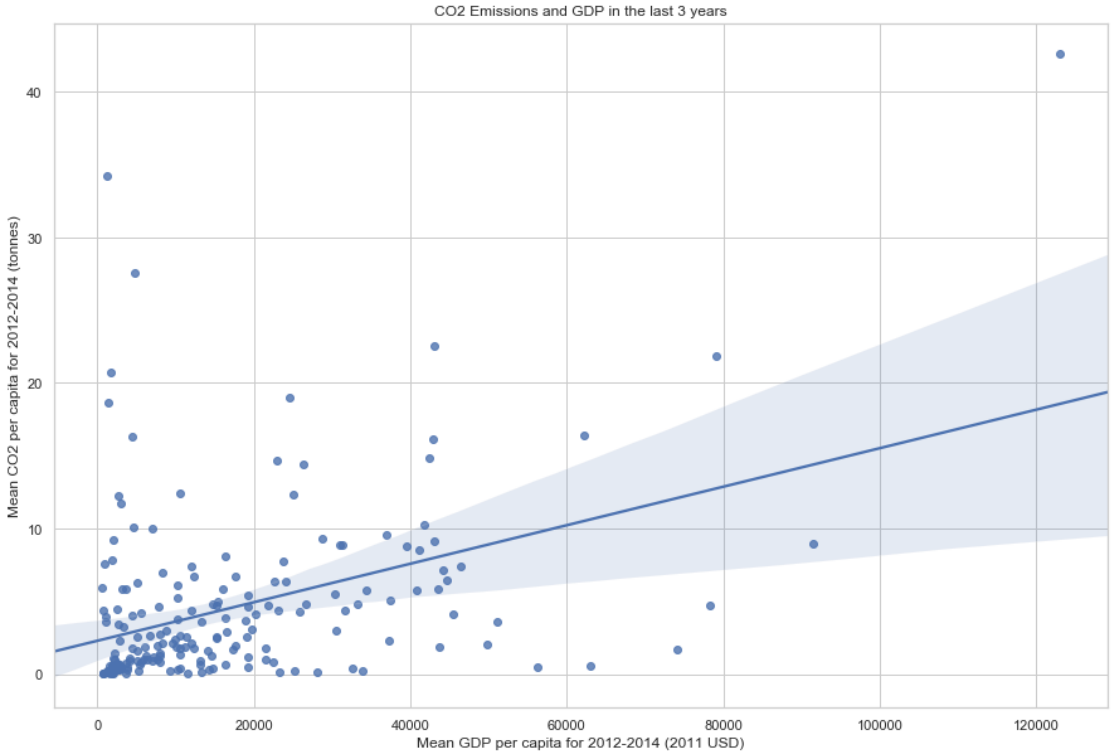
```
/anaconda/lib/python3.6/site-packages/scipy/stats/stats.py:1713: FutureWarning: Using a non-tu
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```



In [31]:
```
# Drop countries that are not in both dataframes and select only last 3 years
y = df_co2_clean.drop('Liechtenstein').iloc[:,-3:].mean(axis=1)
```

```
        x = df_gdp.drop(['Monaco', 'San Marino']).iloc[:,-3:].mean(axis=1)

        fig, ax = plt.subplots(figsize=(15,10))

        ax = sns.regplot(x, y)
        ax.set(xlabel='Mean GDP per capita for 2012-2014 (2011 USD)', ylabel='Mean CO2 per ca
```

```
/anaconda/lib/python3.6/site-packages/scipy/stats/stats.py:1713: FutureWarning: Using a non-tu
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```



As the above scatterplots show, there is in fact a positive correlation between GDP per capita and CO2 emissions per capita. Even when looking only at the last 3 years (in the second scatterplot), there still seems to be a positive correlation.

1.4.3 What is the relationship between CO2 emissions and population growth?

I will once again draw a line graph to see the evolution of population growth over the past 55 years. I will then combine all data into a single DataFrame to visualize relationships between all three variables (CO2, GDP and population growth).

```
In [32]: sns.set(style='white')

        # Plot pop growth against time
        x = df_pop_growth.columns.astype(int)
```

```
y = df_pop_growth.mean()

# Initialize figure
fig, ax = plt.subplots(figsize=(15,10))

# Format graph
ax.set_title('Population growth since 1960')
ax.set_ylabel('Annual population growth rate (%)')
ax.set_xlabel('Year')
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)

# Plot
ax.plot(x,y);
```

Interestingly, we can see that population growth, though still positive, has decreased greatly over the past 55 years.

```
In [33]: # Make DataFrames long
         df1 = df_pop_growth.reset_index().melt(id_vars='country', var_name='year', value_name
         df2 = df_co2_clean.iloc[:,5:].reset_index().melt(id_vars='country', var_name='year',

         #Merge DataFrames
         df_merged = df1.merge(df2, on=['country','year'])
         df_merged.head()
```

Out[33]:

country	year	pop_growth	co2
Afghanistan	1960	1.82	0.046100
Albania	1960	3.02	1.240000
Algeria	1960	2.51	0.554000
Andorra	1960	7.05	5.639227
Angola	1960	1.90	0.097500

In [34]:
```
# Make GDP DataFrame long
df3 = df_gdp.reset_index().melt(id_vars='country', var_name='year', value_name='gdp')

# Merge DataFrames
df_merged = df_merged.merge(df3, on=['country','year'])
df_merged.head()
```

Out[34]:

country	year	pop_growth	co2	gdp
Afghanistan	1960	1.82	0.046100	1210
Albania	1960	3.02	1.240000	2790
Algeria	1960	2.51	0.554000	6520
Andorra	1960	7.05	5.639227	15200
Angola	1960	1.90	0.097500	3860

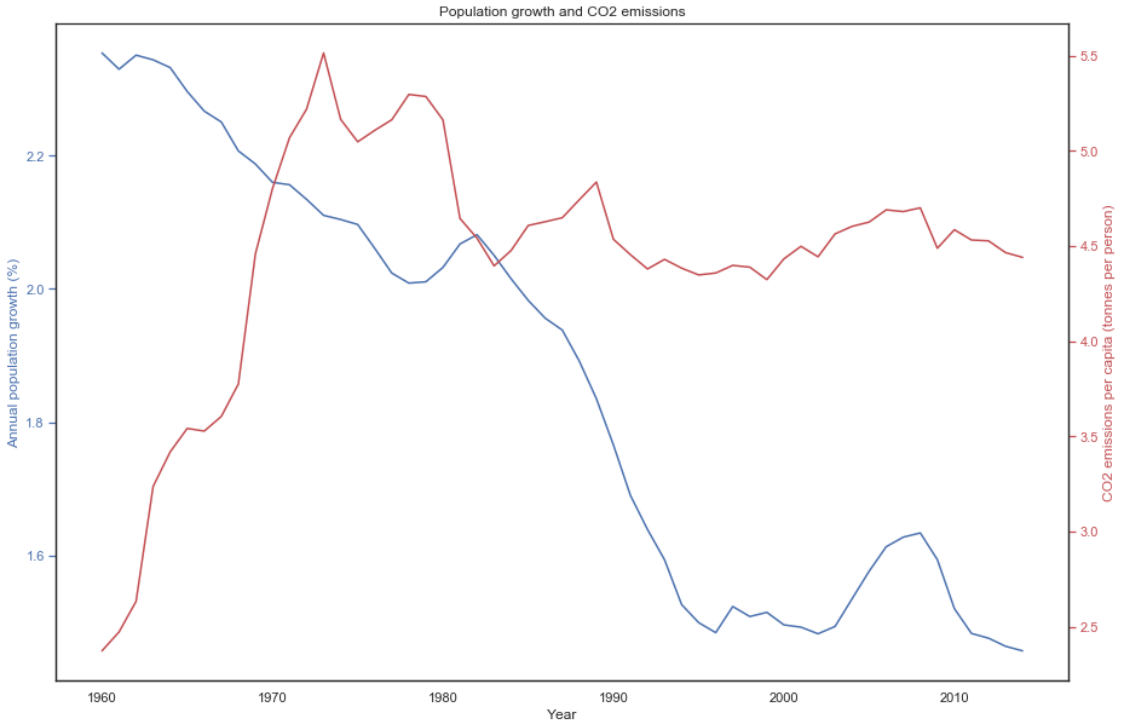
In [35]:
```
# Show all variables by country
df_merged.groupby('country').mean().head(10)
```

Out[35]:

country	pop_growth	co2	gdp
Afghanistan	2.383109	0.147507	1201.563636
Albania	1.119145	1.643909	5169.454545
Algeria	2.331636	2.449873	9576.909091
Andorra	3.357278	7.493673	31241.818182
Angola	2.875273	0.600264	4594.545455
Antigua and Barbuda	1.075582	4.824582	12586.181818
Argentina	1.364909	3.633455	13431.636364
Armenia	0.863087	1.581618	3978.909091
Australia	1.545327	14.540909	27892.727273
Austria	0.358811	7.158364	29085.454545

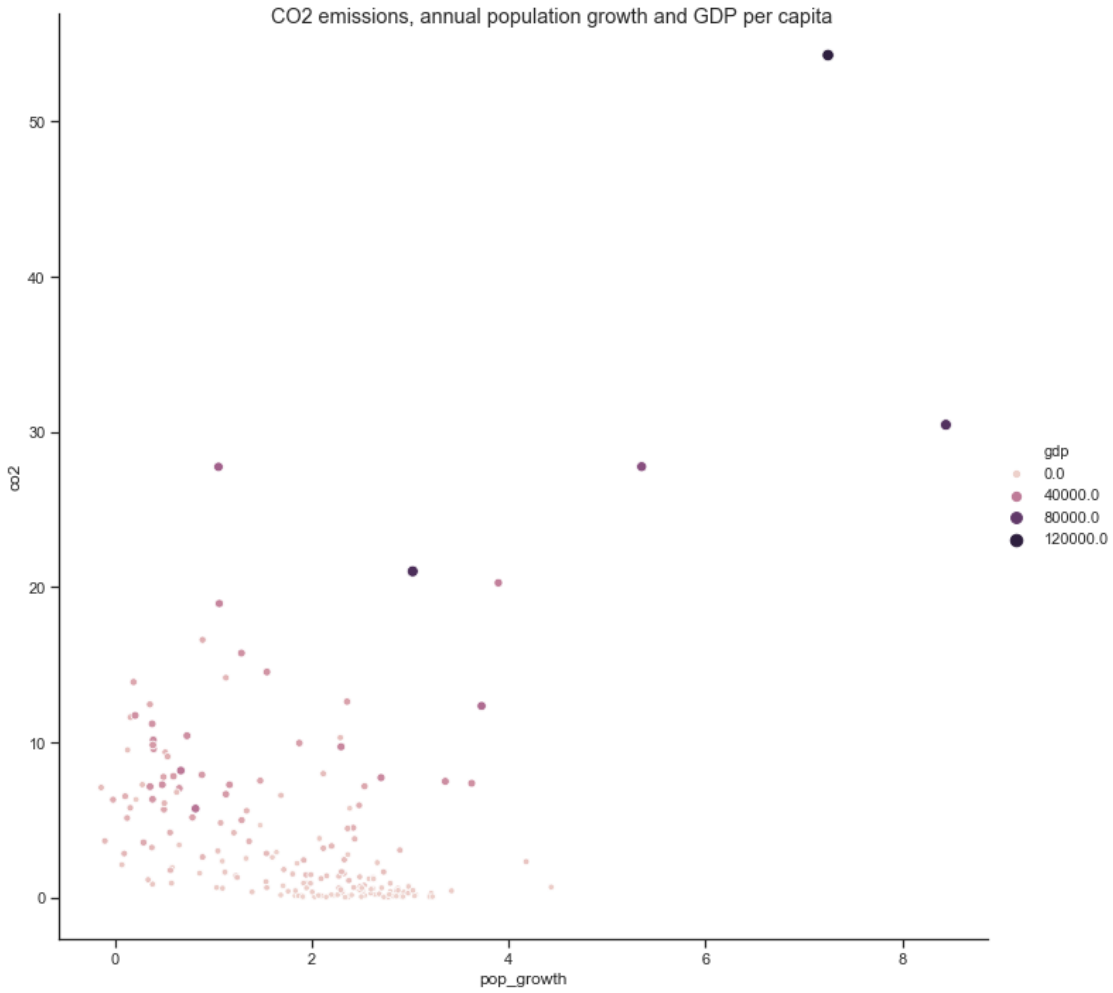
In [36]:
```
# Plot pop growth and CO2 emissions against time
x = np.arange(1960,2015)
y1 = df_pop_growth.mean()
y2 = df_co2_clean.mean()[-55:]

# Initialize figure
fig, ax1 = plt.subplots(figsize=(15,10))

# Format graph
ax1.set_title('Population growth and CO2 emissions')
```

```
ax1.set_ylabel('Annual population growth (%)', color='b')
ax1.set_xlabel('Year')

# Make the y-axis label, ticks and tick labels match the line color.
ax1.tick_params('y', colors='b')
ax1.spines['top'].set_visible(False)

ax2 = ax1.twinx()
ax2.set_ylabel('CO2 emissions per capita (tonnes per person)', color='r')
ax2.tick_params('y', colors='r')

# Plot
ax1.plot(x, y1, 'b-')
ax2.plot(x, y2, 'r-');
```



As seen above, there seems to be a negative correlation between population growth and CO2 emissions (which could be related to the fact that high-income countries tend to have lower population growth rates). While this has been true in average over the past 55 years, I want to create a 3D scatter-plot that will allow us to investigate relationships at the country level (i.e. does a country with higher GDP and population growth rate tend to have higher CO2 emissions?)

In [37]:
```
#fig, ax = plt.subplots(figsize=(15,10))
sns.set(style="ticks")
g = sns.relplot(x='pop_growth', y='co2', data=df_merged.groupby('country').mean(), he

g.fig.suptitle('CO2 emissions, annual population growth and GDP per capita');
```



As we can see above, when looking at each individual country (each point represents a country), it is still the case that higher population growth seems to be correlated with higher CO2 emissions and higher GDP per capita.

1.5 Conclusions

To recap, these were the questions that I set out to investigate at the onset: 1. What trends can we see in CO2 emissions over the past century? 2. What is the relationship between CO2 emissions and GDP? 3. What is the relationship between CO2 emissions and population growth?

After investigating the datasets, I have come to the following conclusions regarding these questions: 1. CO2 emissions have increased greatly over the past 100 years. Perhaps somewhat reassuring, this trend seems to have slowed down or even reversed since the financial crisis of 2007-2008. It would be interesting to see similar trends for other greenhouse gases (e.g. methane). It is important to note that this anaylsis is done on CO2 per capita, meaning that total CO2 emissions might still be increasing as population continues to grow globally. 2. GDP and CO2 emissions seem to be positively correlated. This comes as no surprise, since higher GDP usually means higher production of goods and consumption of resources. Interestingly, since 2007-2008, there seems to be

an increase in GDP that has not been met with an increase in CO2 emissions. Again, it would be interesting to evaluate other greenhouse gases as well. Also on this point, it will be interesting to continue to see what happens with CO2 emissions in Sub-Saharan Africa and South Asia as they increase their GDP. 3. When analyzing the timelines of population growth and CO2 emissions, it seems that these two variables are inversely correlated. What we have seen in the past 55 years is an increase in average CO2 emissions per capita as average population growth has been decreasing. It would be interesting to see if population growth by itself has any relation with CO2 emissions, or if low population growth simply reflects higher average GDP per capita.