LifeExpentancy

October 20, 2021

1 Searching For Correlation between Air Polutions and Mortality

1.0.1 Importing necessarily modules

[1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import sklearn as skl
```

1.0.2 Importing Data, previously downloaded from: https://ec.europa.eu/eurostat

[2]:
```
# DataFrame from AirPolution CSV
ap_df = pd.read_csv("/Users/patrykdabkowski/Desktop/LifeExpentancy/
 ↪env_ac_ainah_r2/env_ac_ainah_r2_1_Data.csv", delimiter=",")
ap_df = ap_df[['TIME', 'GEO', 'AIRPOL', 'Value']]
ap_df
```

[2]:
```
       TIME                                        GEO          AIRPOL  \
0      2010  European Union - 27 countries (from 2020)  Carbon dioxide
1      2010  European Union - 27 countries (from 2020)  Carbon dioxide
2      2010  European Union - 27 countries (from 2020)  Carbon dioxide
3      2010  European Union - 27 countries (from 2020)  Carbon dioxide
4      2010  European Union - 27 countries (from 2020)  Carbon dioxide
…       …                                          …             …
23095  2019                                     Turkey   Nitrous oxide
23096  2019                                     Turkey   Nitrous oxide
23097  2019                                     Turkey   Nitrous oxide
23098  2019                                     Turkey   Nitrous oxide
23099  2019                                     Turkey   Nitrous oxide

                Value
0      2 809 855 835
1        101 646 467
2         30 133 743
3        818 010 355
4      1 127 012 607
…                …
23095              :
23096              :
```

```
23097              :
23098              :
23099              :

[23100 rows x 4 columns]
```

1.0.3 Removing data for all EU

[3]:
```
ap_df.drop([x for x in range(133)], inplace = True)
dropindex = ap_df[ ap_df['GEO'] == 'European Union - 28 countries (2013-2020)'].
 ↪index
ap_df.drop(dropindex, inplace = True)
dropindex = ap_df[ ap_df['GEO'] == 'European Union - 27 countries (from 2020)'].
 ↪index
ap_df.drop(dropindex, inplace = True)
ap_df['GEO'].unique()
```

[3]:
```
array(['Belgium', 'Bulgaria', 'Czechia', 'Denmark',
       'Germany (until 1990 former territory of the FRG)', 'Estonia',
       'Ireland', 'Greece', 'Spain', 'France', 'Croatia', 'Italy',
       'Cyprus', 'Latvia', 'Lithuania', 'Luxembourg', 'Hungary', 'Malta',
       'Netherlands', 'Austria', 'Poland', 'Portugal', 'Romania',
       'Slovenia', 'Slovakia', 'Finland', 'Sweden', 'Iceland', 'Norway',
       'Switzerland', 'United Kingdom', 'Serbia', 'Turkey'], dtype=object)
```

1.0.4 Summing all values for specific data for country in time and polution kind

[4]:
```
df = ap_df[['GEO', 'TIME', 'AIRPOL', 'Value']]
df
```

[4]:
```
           GEO  TIME          AIRPOL       Value
133    Belgium  2010  Carbon dioxide   2 316 656
134    Belgium  2010  Carbon dioxide     662 770
135    Belgium  2010  Carbon dioxide  35 630 974
136    Belgium  2010  Carbon dioxide  21 413 145
137    Belgium  2010  Carbon dioxide   1 118 837
…          …    …              …          …
23095   Turkey  2019   Nitrous oxide           :
23096   Turkey  2019   Nitrous oxide           :
23097   Turkey  2019   Nitrous oxide           :
23098   Turkey  2019   Nitrous oxide           :
23099   Turkey  2019   Nitrous oxide           :

[21779 rows x 4 columns]
```

1.0.5 Removing duplicated data

[5]:
```
df = df.drop_duplicates(['GEO', 'TIME', 'AIRPOL', 'Value'])
df.reset_index(inplace = True, drop = True)
df
```

[5]:

	GEO	TIME	AIRPOL	Value
0	Belgium	2010	Carbon dioxide	2 316 656
1	Belgium	2010	Carbon dioxide	662 770
2	Belgium	2010	Carbon dioxide	35 630 974
3	Belgium	2010	Carbon dioxide	21 413 145
4	Belgium	2010	Carbon dioxide	1 118 837
…	…	…	…	…
20374	Serbia	2019	Methane	:
20375	Serbia	2019	Nitrous oxide	:
20376	Turkey	2019	Carbon dioxide	:
20377	Turkey	2019	Methane	:
20378	Turkey	2019	Nitrous oxide	:

[20379 rows x 4 columns]

[ ]:
```
df_new = pd.DataFrame()
for geo in df['GEO']:
    for time in df['TIME']:
        for airpol in df['AIRPOL']:
            value = df['Value'].loc[(df['GEO'] == geo) & (df['TIME'] == time) &␣
↪(df['AIRPOL'] == airpol)].sum()
            new_row = {'GEO': geo, 'TIME': time, 'AIRPOL': airpol, 'Value':␣
↪value}
            df_new = df_new.append(new_row, ignore_index=True)
```

[ ]:
```
df_new
```

1.0.6 Pre-analysis for one kind of Air Polution

[ ]:
```
df_car = df.loc[df['AIRPOL']== 'Carbon dioxide']
df_car
```