

Global Analysis

February 15, 2023

Air Quality in Catalonia challenge is asking participants to use the data from the Catalan Transparency Portal to analyze the evolution of air pollution in Catalonia over the past three decades and develop algorithms to predict air pollutant concentrations. We provide a global analysis of air quality, build algorithms to predict air pollutant concentrations, and write a final report.

0.1 Import Libraries

First, we import the libraries: matplotlib and seaborn for visualisation; pandas for data wrangling.

```
[1]: import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

pd.set_option('display.max_columns', 25)
plt.rcParams["figure.figsize"] = [10, 6]
plt.style.use('fivethirtyeight')
```

0.2 Import Data

Next, we read the csv data and store it as pandas dataframe. We also display the data.

```
[2]: df = pd.read_csv("9c820e0e5b3a4264aa5058f24a82386d.csv")
df
```

```
[2]:
```

	CODI EOI	NOM ESTACIO	DATA	MAGNITUD	\
0	43148003	Tarragona (Bonavista)	25/01/2023	10	
1	8137001	Montseny (La Castanya)	25/01/2023	12	
2	8124009	Mollet del Vallès	25/01/2023	7	
3	8114006	Martorell	25/01/2023	7	
4	8112003	Manlleu	25/01/2023	8	
...	
3106369	8125002	Montcada i Reixac	01/01/1991	14	
3106370	8019004	Barcelona (Poblenou)	01/01/1991	1	
3106371	8101001	L'Hospitalet de Llobregat	01/01/1991	6	
3106372	8125002	Montcada i Reixac	01/01/1991	7	
3106373	8101001	L'Hospitalet de Llobregat	01/01/1991	14	

	CONTAMINANT	UNITATS	TIPUS	ESTACIO	AREA URBANA	CODI INE	\
0	PM10	µg/m3	industrial	suburban	43148		

1	NOX	µg/m3	background	rural	8137
2	NO	µg/m3	traffic	suburban	8124
3	NO	µg/m3	background	suburban	8114
4	NO2	µg/m3	background	suburban	8112
...
3106369	O3	µg/m3	traffic	suburban	8125
3106370	SO2	µg/m3	background	urban	8019
3106371	CO	mg/m3	background	urban	8101
3106372	NO	µg/m3	traffic	suburban	8125
3106373	O3	µg/m3	background	urban	8101

	MUNICIPI	CODI	COMARCA	NOM COMARCA	...	\
0	Tarragona		36	Tarragonès	...	
1	Montseny		41	Vallès Oriental	...	
2	Mollet del Vallès		41	Vallès Oriental	...	
3	Martorell		11	Baix Llobregat	...	
4	Manlleu		24	Osona	...	
...
3106369	Montcada i Reixac		40	Vallès Occidental	...	
3106370	Barcelona		13	Barcelonès	...	
3106371	Hospitalet de Llobregat, l'		13	Barcelonès	...	
3106372	Montcada i Reixac		40	Vallès Occidental	...	
3106373	Hospitalet de Llobregat, l'		13	Barcelonès	...	

	17h	18h	19h	20h	21h	22h	23h	24h	ALTITUD	\
0	18.0	24.0	28.0	29.0	39.0	33.0	24.0	20.0	39	
1	14.0	9.0	4.0	3.0	3.0	3.0	2.0	2.0	693	
2	9.0	26.0	17.0	7.0	16.0	62.0	62.0	58.0	90	
3	5.0	2.0	2.0	4.0	7.0	7.0	3.0	1.0	78	
4	21.0	22.0	22.0	35.0	38.0	36.0	32.0	28.0	460	
...
3106369	22.0	10.0	0.0	0.0	0.0	0.0	0.0	0.0	34	
3106370	13.0	13.0	20.0	15.0	16.0	16.0	19.0	16.0	3	
3106371	0.6	1.0	1.9	1.5	1.6	1.6	1.1	1.0	29	
3106372	9.0	15.0	71.0	157.0	167.0	204.0	136.0	74.0	34	
3106373	47.0	18.0	6.0	13.0	11.0	2.0	12.0	12.0	29	

	LATITUD	LONGITUD	GEOREFERENCIA
0	41.115910	1.191999	POINT (1.1919986 41.11591)
1	41.779280	2.358002	POINT (2.358002 41.77928)
2	41.549183	2.212098	POINT (2.2120984 41.549183)
3	41.475384	1.921202	POINT (1.9212021 41.475384)
4	42.003307	2.287299	POINT (2.2872992 42.003307)
...
3106369	41.481972	2.188298	POINT (2.188298 41.481972)
3106370	41.403878	2.204501	POINT (2.204501 41.403878)
3106371	41.370475	2.114999	POINT (2.114999 41.370475)

```
3106372  41.481972  2.188298  POINT (2.188298 41.481972)
3106373  41.370475  2.114999  POINT (2.114999 41.370475)
```

[3106374 rows x 40 columns]

Here are the most frequently used Cabin names:

```
[70]: df['NOM ESTACIO'].value_counts()
```

```
[70]: Igualada                75800
      Constantí              73536
      Reus                   72855
      Tarragona (Bonavista)  72287
      Vila-seca              70140
      ...
      Veciana (estació agroalimentària)  2227
      Viladecans                1594
      Sant Just Desvern (CEIP Montseny)  1357
      Barcelona (Torre Girona)         1343
      el Prat de Llobregat (Sant Cosme)    26
      Name: NOM ESTACIO, Length: 116, dtype: int64
```

We display the most frequent 5-digit numeric code corresponding to the municipality (the first two digits correspond to the province, and the next three identify the Integer municipality)

```
[74]: df['CODI EOI'].value_counts()
```

```
[74]: 8102005    75800
      43047001   73536
      43123005   72855
      43148003   72287
      43171001   70140
      ...
      8297001    2227
      8301002    1594
      8221004    1357
      8019056    1343
      8169007     26
      Name: CODI EOI, Length: 124, dtype: int64
```

We display the most frequent pollutants:

```
[75]: df['MAGNITUD'].value_counts()
```

```
[75]: 8    547242
      7    546531
      14   486801
      1    452854
```

```

12      273300
6       266753
10      170283
65      147623
42       65304
3        57239
44       49797
30       16032
9        11424
11        6872
331       3577
53        3014
58        1728
Name: MAGNITUD, dtype: int64

```

We convert the data column to datetime and display the most frequent dates in the data:

```

[71]: df.DATA = pd.to_datetime(df.DATA, format="%d/%m/%Y")

df.DATA.value_counts()

```

```

[71]: 2021-10-25      368
      2021-11-01      368
      2021-10-11      368
      2021-10-12      368
      2021-10-13      368
      ...
      1991-02-23       25
      1991-03-25       23
      1991-01-01       22
      1991-01-04       21
      1991-01-08       19
Name: DATA, Length: 11713, dtype: int64

```

We create date features and one hot encode pollutants

```

[7]: df['year'] = df.DATA.dt.year
     df['month'] = df.DATA.dt.month
     df['day'] = df.DATA.dt.day

     cols = ['01h', '02h', '03h', '04h', '05h', '06h', '07h', '08h',
             '09h', '10h', '11h', '12h', '13h', '14h', '15h', '16h', '17h', '18h',
             '19h', '20h', '21h', '22h', '23h', '24h']

     df['sum_day'] = df[cols].sum(axis=1)

     pollutant_dummies = pd.get_dummies(df.MAGNITUD)
     df = pd.concat([df, pollutant_dummies], axis=1)

```

We further group the by pollutants and time:

```
[88]: df_monthly = df.groupby(['year', 'month', "MAGNITUD"])[ 'sum_day'].agg(['mean',  
↪ "median", "count"])
```

```
[89]: df_monthly
```

```
[89]:
```

			mean	median	count
year	month	MAGNITUD			
1991	1	1	416.633218	336.00	289
		3	2640.886364	2179.50	88
		6	41.639091	33.65	110
		7	1602.384146	1391.00	164
		8	955.169697	933.00	165
...		
2023	1	12	843.120000	608.00	1600
		14	967.541224	954.00	1225
		30	21.459200	16.20	125
		65	39.072000	33.50	300
		331	129.180000	96.30	25

[4016 rows x 3 columns]

From The best to worst months of the year in terms of pollution:

```
[94]: df.groupby(['month'])[ 'sum_day'].agg(['mean', "median", "count"]).reset_index().  
↪ sort_values('mean')
```

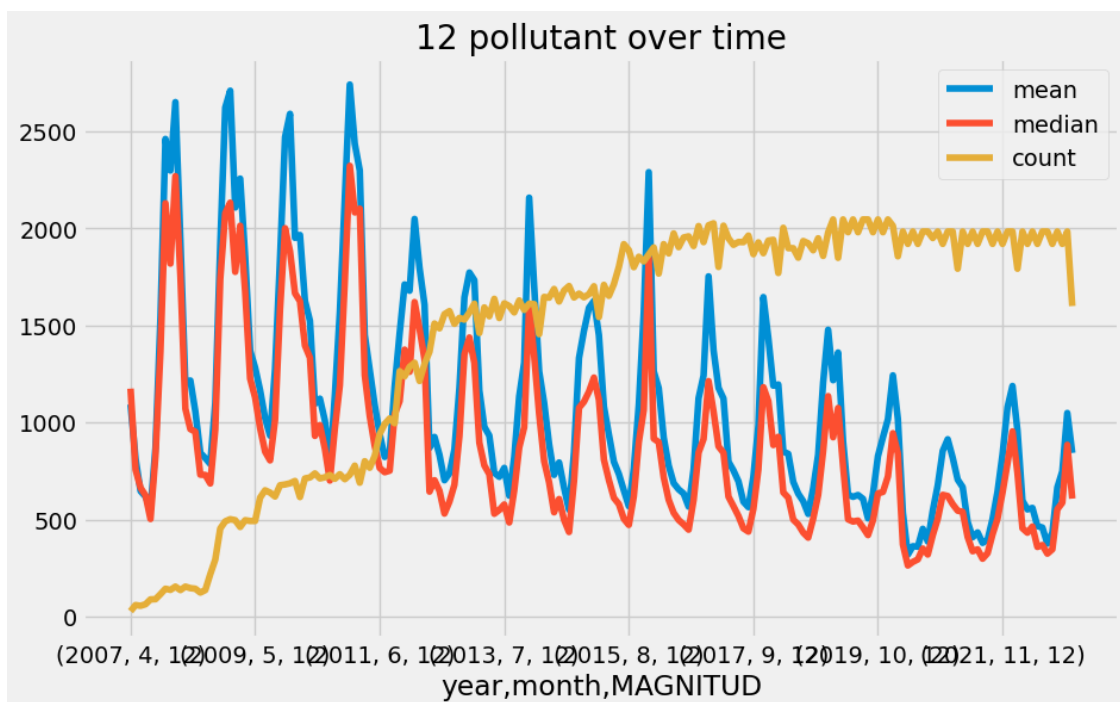
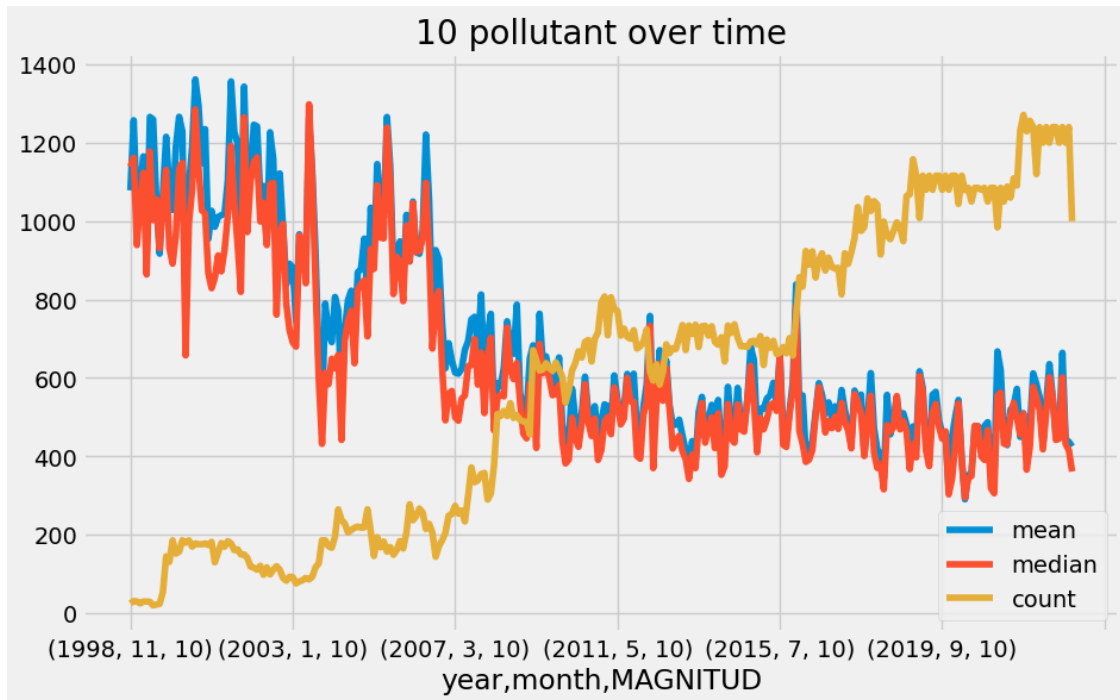
```
[94]:
```

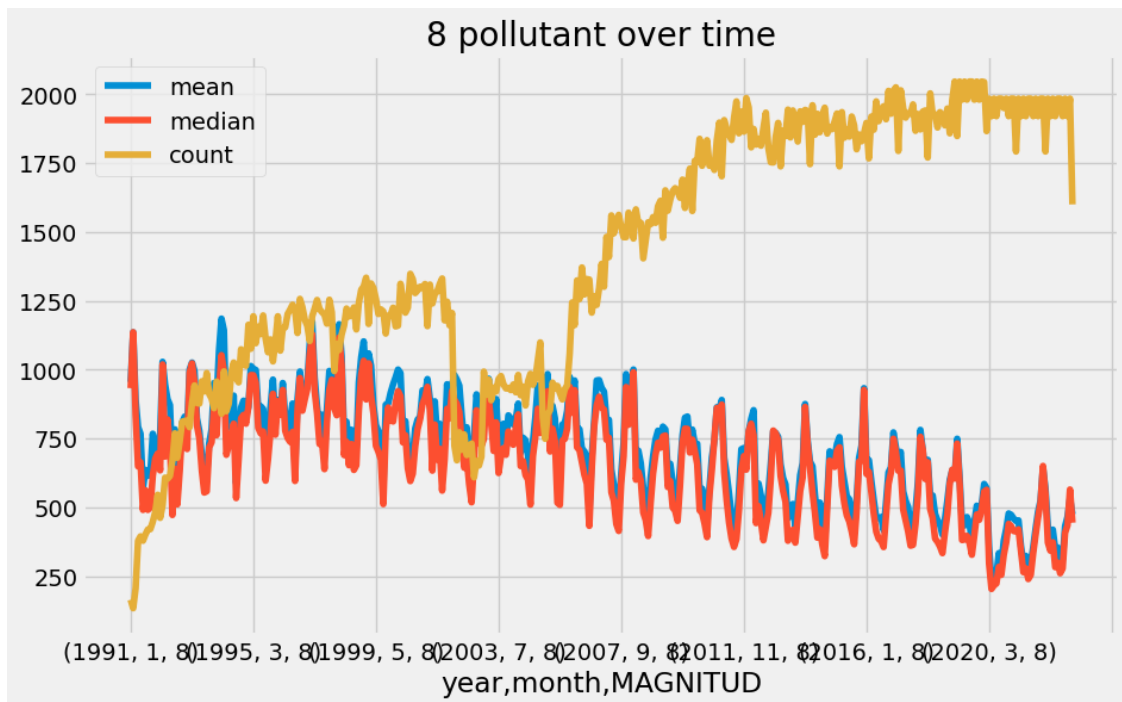
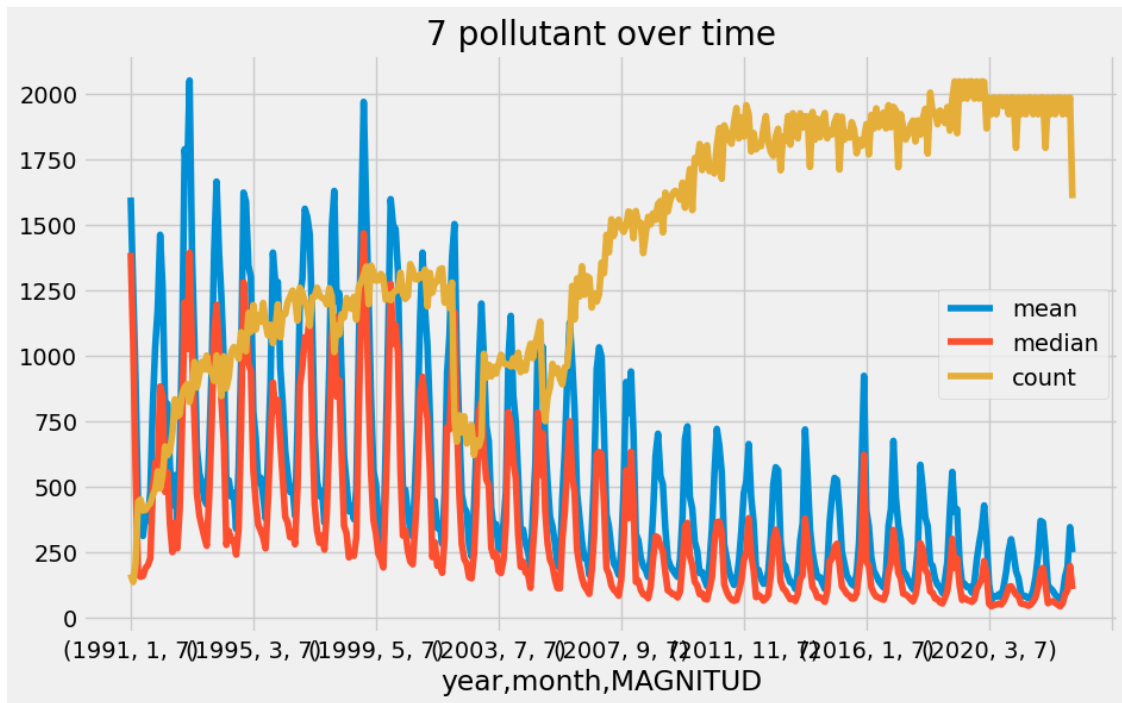
	month	mean	median	count
7	8	474.698796	232.0	264288
8	9	502.721422	274.0	255262
6	7	525.401401	244.0	263708
5	6	531.070319	253.0	255068
4	5	532.889356	262.0	261264
9	10	543.575806	336.0	264994
3	4	551.074431	279.0	253566
2	3	591.856147	341.0	260670
10	11	599.728589	340.0	258701
0	1	614.346660	348.0	266026
1	2	621.984413	383.0	236300
11	12	633.050464	343.0	266527

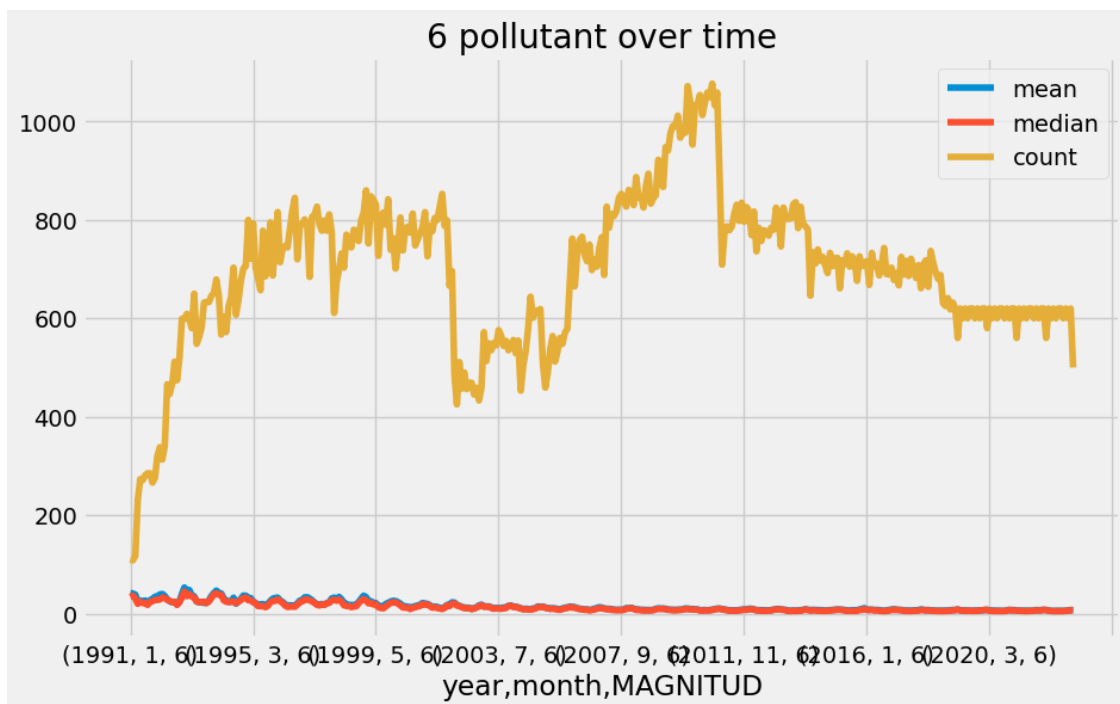
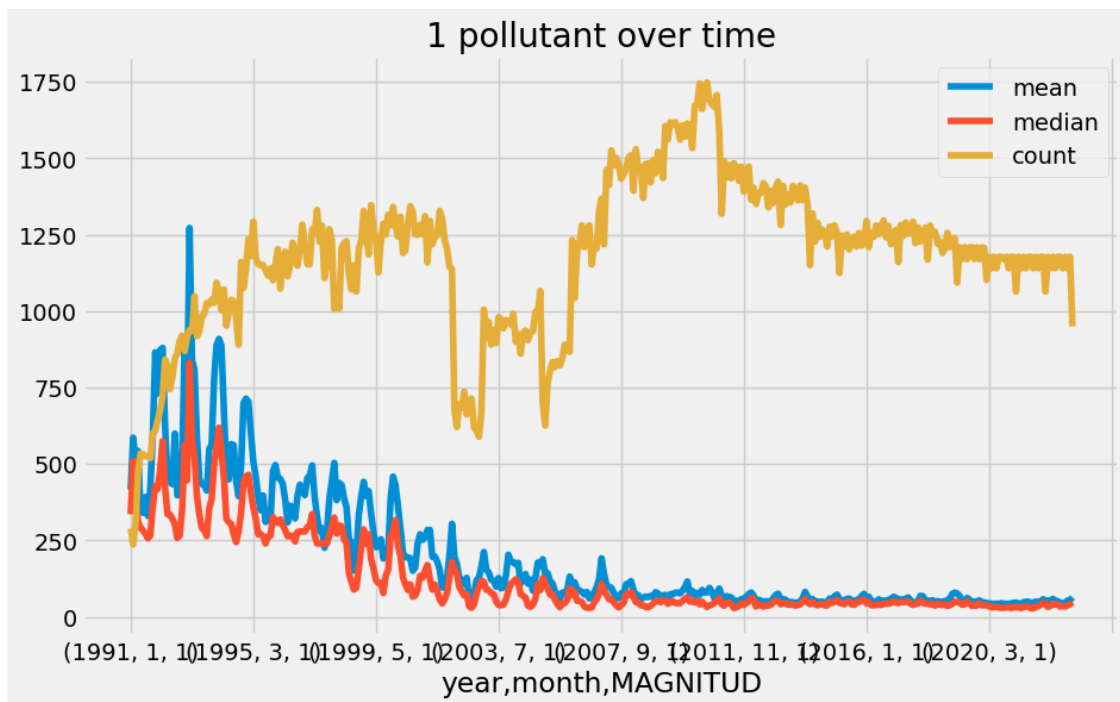
We display the aggregated data:

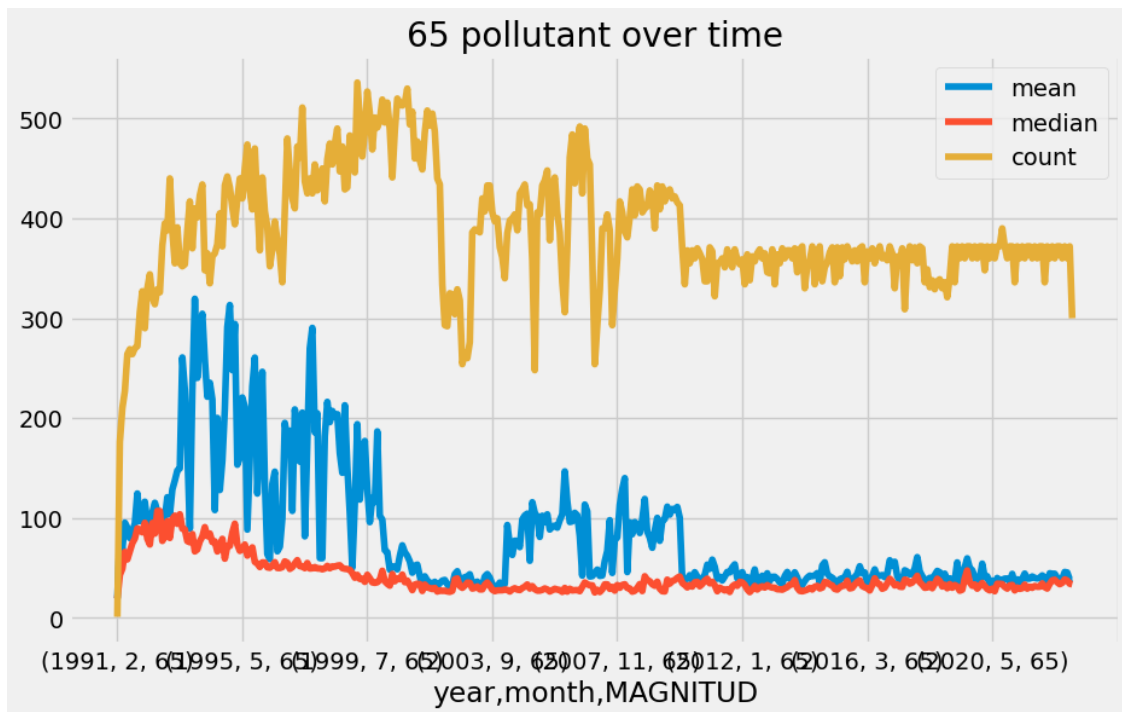
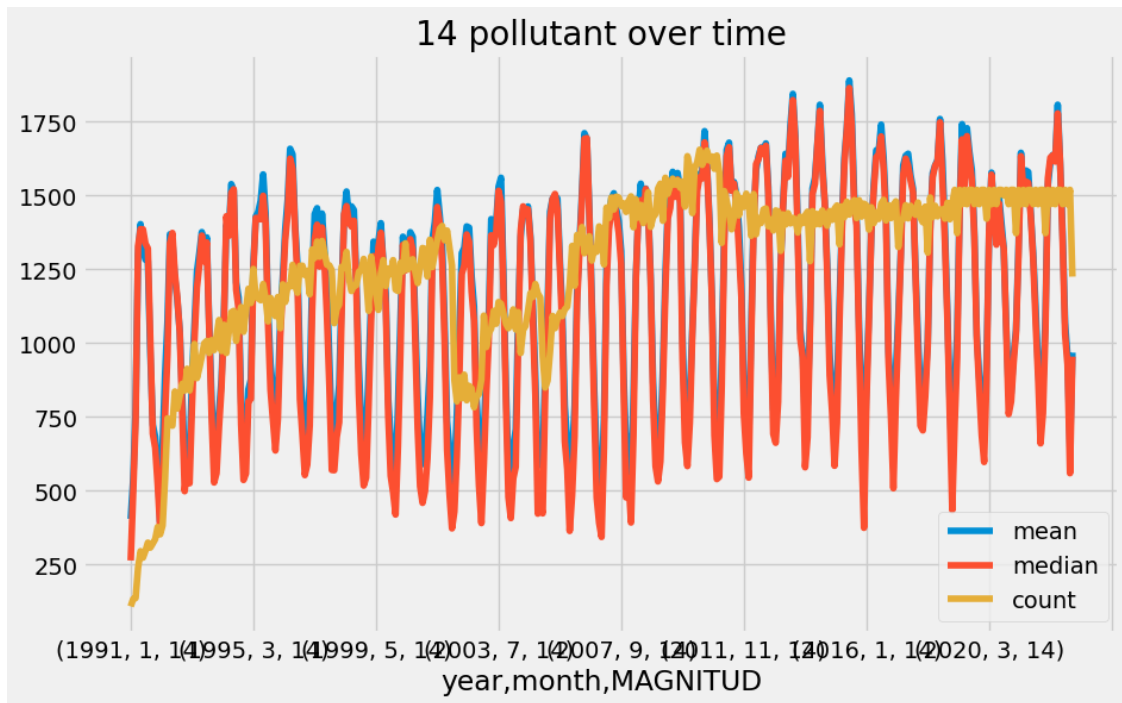
```
[78]: pollutantsIds = df["MAGNITUD"].unique()  
  
for x in pollutantsIds:
```

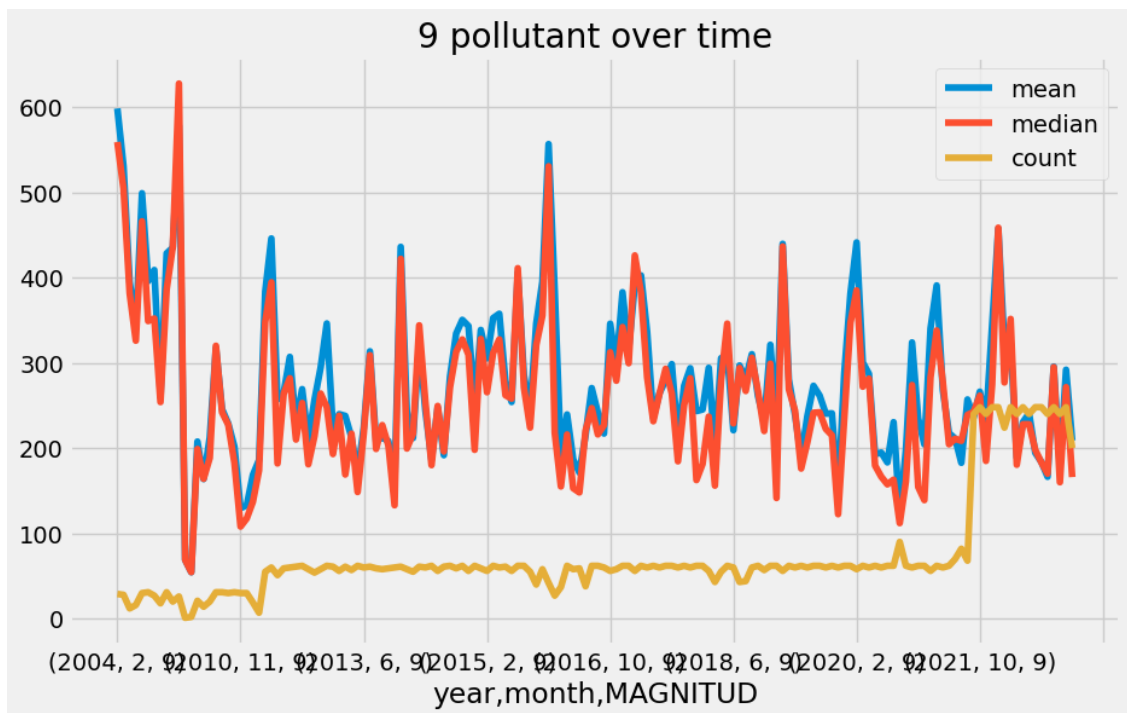
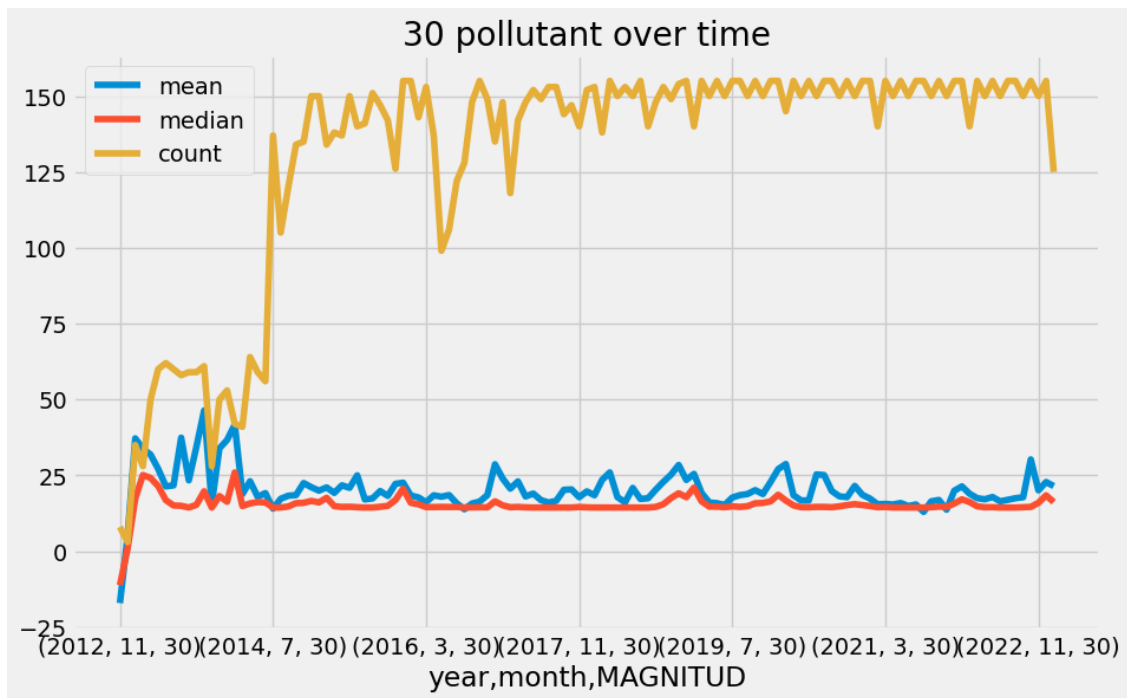
```
df_monthly.loc[(df_monthly.index.get_level_values('MAGNITUD') == x)].  
plot(title=f"{x} pollutant over time")
```

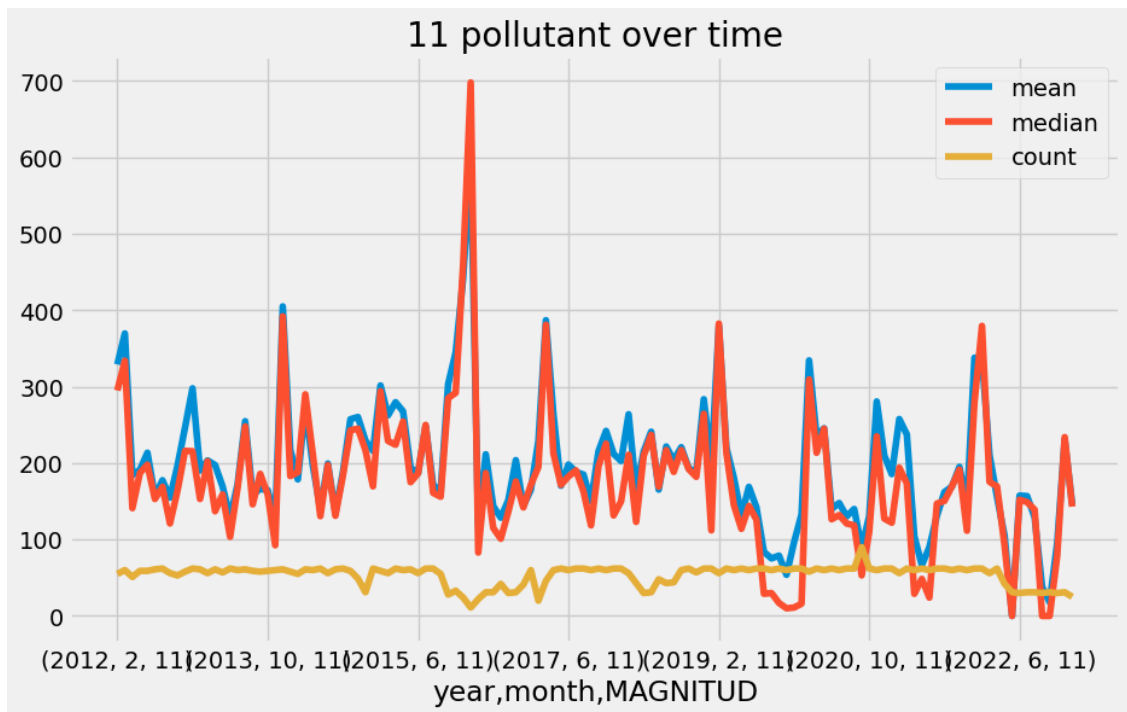
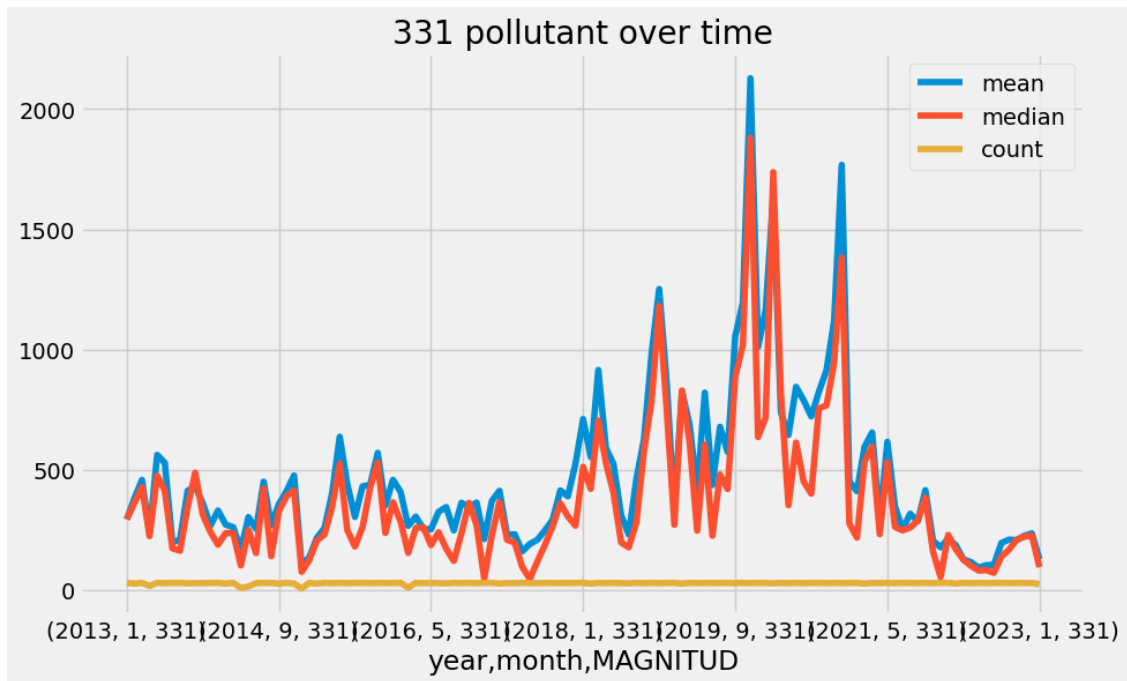


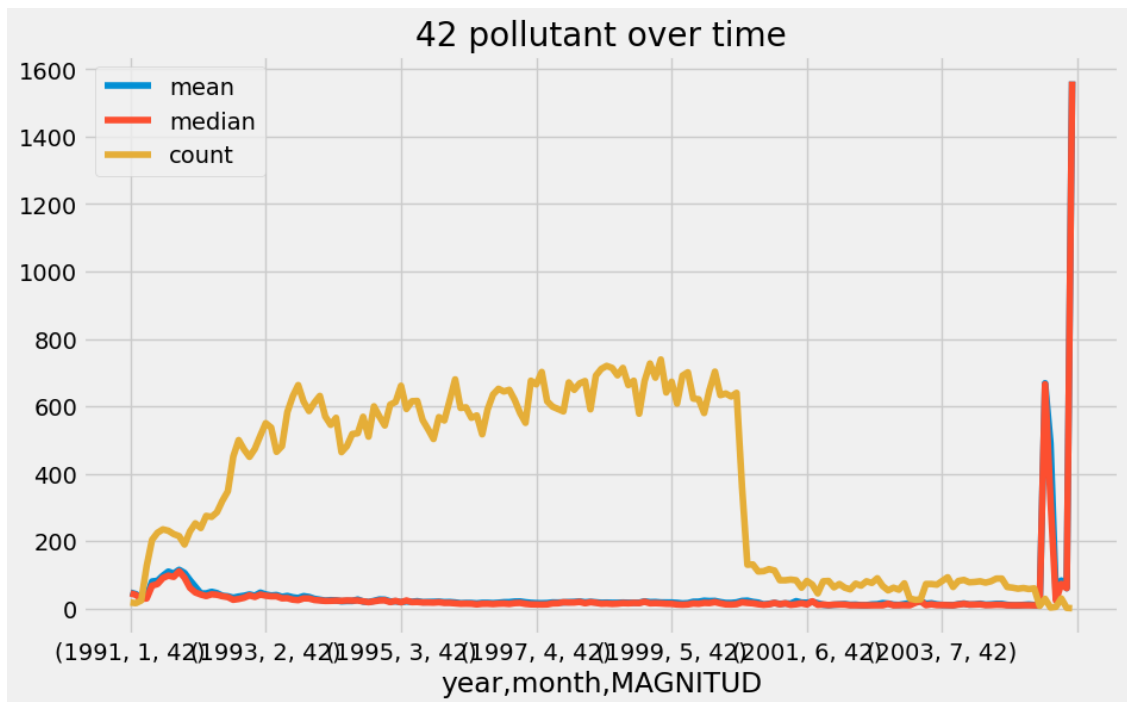
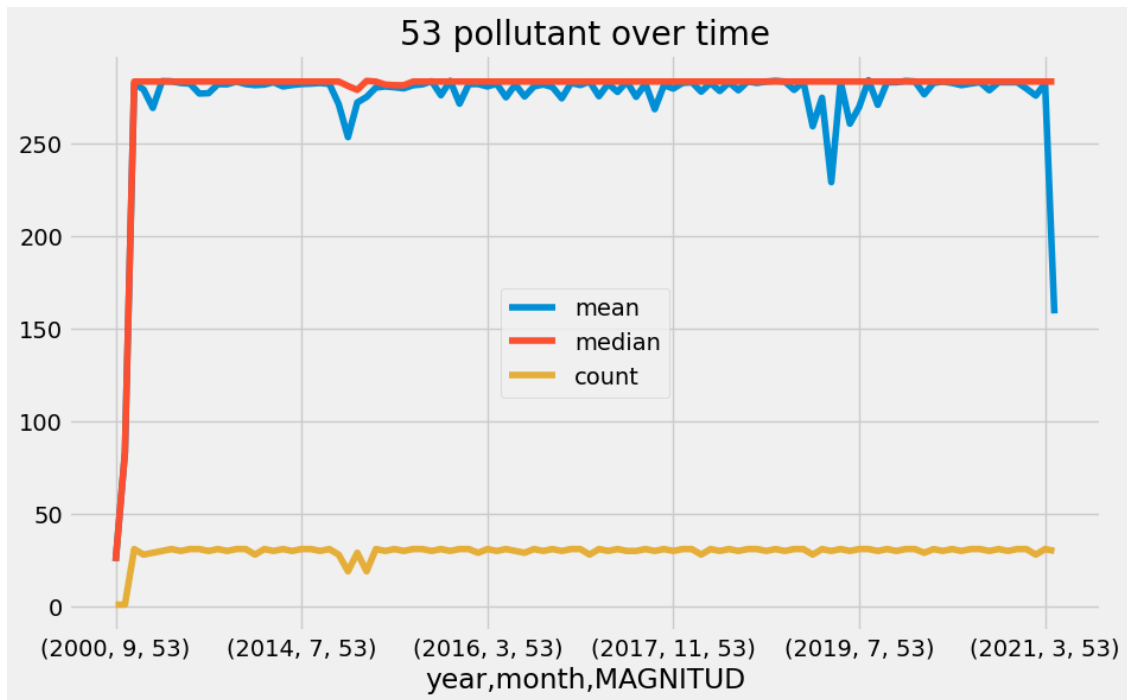


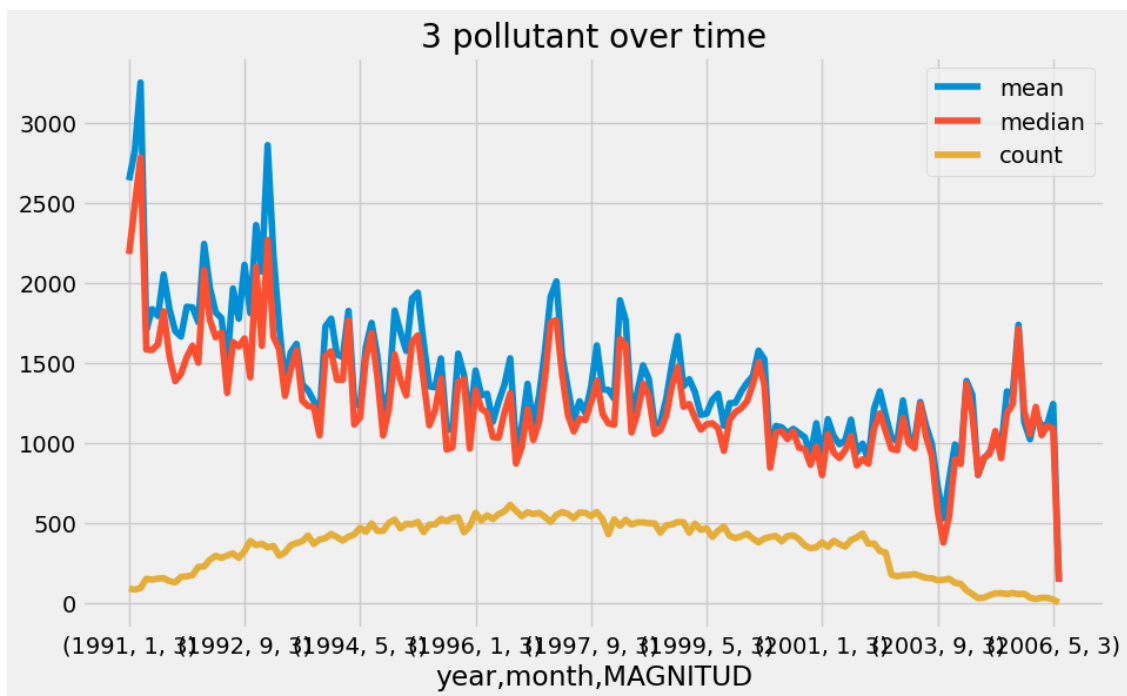
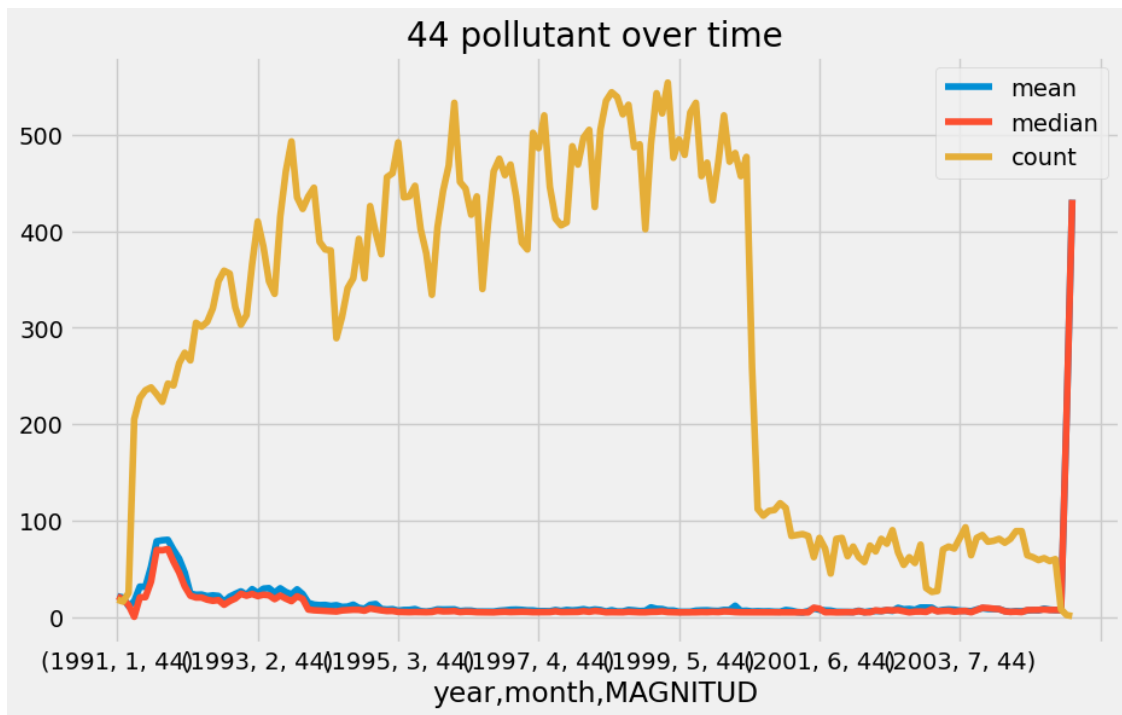


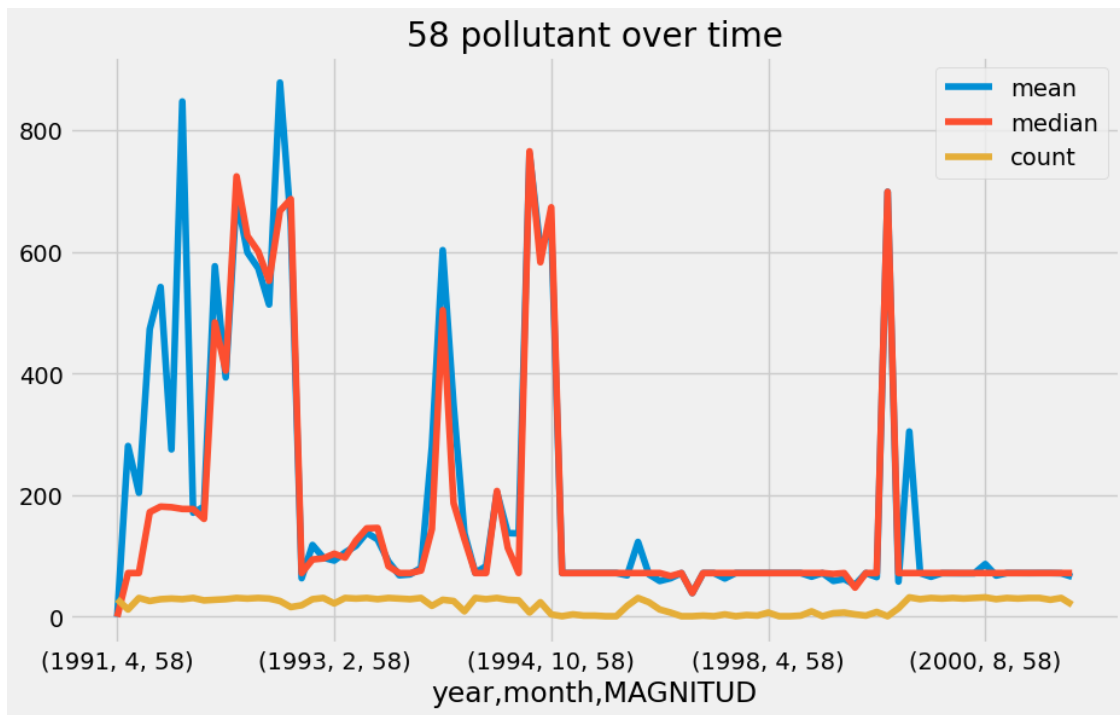












From the graphs above we can clearly see the dramatic rise in the number of observations.

We also note the periodic nature of pollution levels rises and falls. This is a very common pattern where Air pollution becomes actually worse during winter. Air pollution is often worse in winter due to a combination of meteorological and human-made factors. Cold, still air can cause atmospheric inversion layers to form, trapping pollutants close to the ground and leading to episodes of high levels of air pollution. Additionally, households often burn more fuel for heating during the winter months, leading to increased emissions of pollutants such as particulate matter, nitrogen oxides, and carbon monoxide. Lastly, areas with high population density and limited wind movement, such as cities and towns, tend to experience worse air quality in winter due to the buildup of pollutants.

Why air pollution is worse in winter?

Now we explore more granular hourly data:

```
[96]: df_hourly = df.groupby(["MAGNITUDE"])[cols].agg(['mean', "median", "count"])
df_hourly_mean = df.groupby(["MAGNITUDE"])[cols].agg(["mean"])
```

```
[106]: df_hourly_mean.T.style.highlight_max(color = 'lightgreen', axis = 0)
```

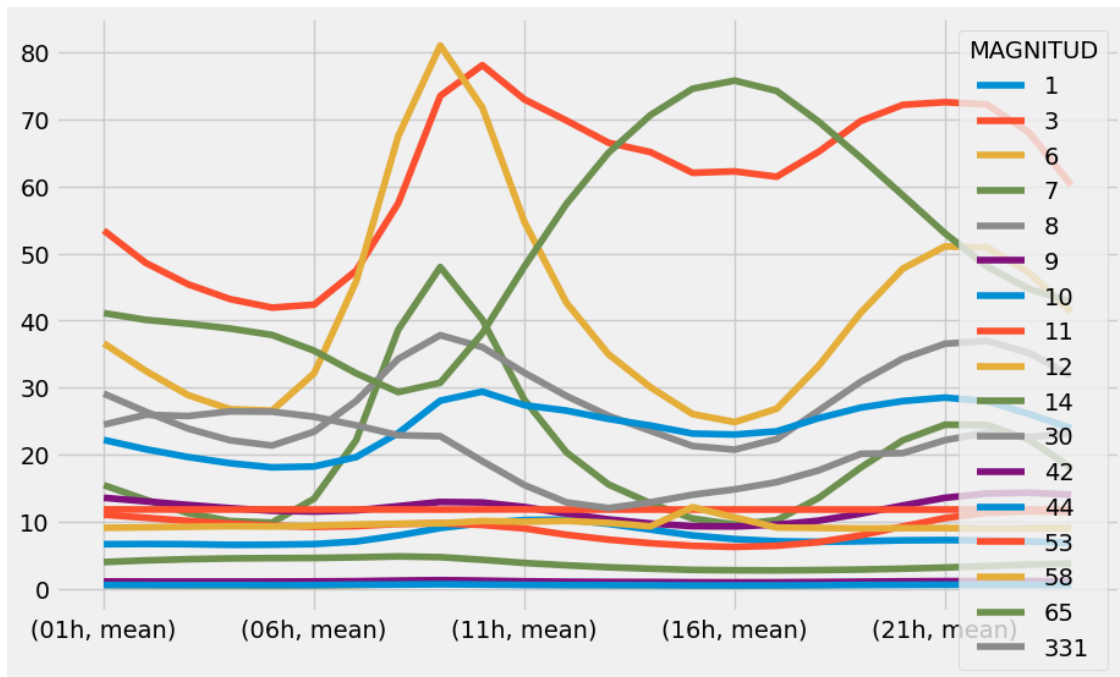
```
[106]: <pandas.io.formats.style.Styler at 0x7f0558cd2e50>
```

```
[107]: df_hourly_mean.T.style.highlight_min(color = 'lightgreen', axis = 0)
```

```
[107]: <pandas.io.formats.style.Styler at 0x7f0558cb48e0>
```

```
[99]: df_hourly_mean.T.plot()
```

```
[99]: <AxesSubplot: >
```



From the graph above we notice that the generally the worst hour is on average for largest amount of pollutants: 6, 7, 8, 12, 30, 42 and 44. We also noticed that the best hour for nearly similar mix of pollutants is 16h.

We find similar data patterns in the following article: [What Time of Day Is Air Pollution Lowest?](#)

This is due to the diurnal cycle, in which levels of pollutants can increase in the morning due to increased activity and decreased air circulation, and decrease in the afternoon due to increased air circulation and decreased activity. This cycle is affected by factors such as temperature, wind speed, sunlight, and mixing of air.

We analyze the relationship between altitude and concentration of particles in the air, and present your conclusions in graphical form.

```
[29]: corr = df.corr()
      corr["ALTITUD"].sort_values()
```

```
/tmp/ipykernel_679253/1270252458.py:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
```

```
corr = df.corr()
```

[29] :	LONGITUD	-0.104412
	CODI COMARCA	-0.070486
	09h	-0.067970
	65	-0.065791
	08h	-0.060106
	MAGNITUD	-0.052438
	10h	-0.049168
	CODI EOI	-0.043374
	CODI INE	-0.043374
	3	-0.035921
	30	-0.035770
	44	-0.033091
	42	-0.028931
	6	-0.028159
	07h	-0.024233
	11	-0.023837
	1	-0.021489
	9	-0.019179
	23h	-0.016602
	22h	-0.014888
	11h	-0.013039
	331	-0.010567
	58	-0.010439
	24h	-0.009751
	53	-0.009676
	21h	-0.004768
	7	-0.003579
	12	-0.003321
	8	-0.003199
	01h	-0.002495
	month	-0.000258
	day	0.000256
	sum_day	0.005850
	02h	0.006570
	20h	0.009954
	06h	0.010558
	03h	0.017351
	12h	0.021584
	04h	0.024476
	05h	0.027573
	19h	0.028619
	13h	0.042666
	18h	0.049340
	14h	0.054391
	10	0.054506
	15h	0.062837
	17h	0.063772


```

16h          0.066618
year         0.067959
14           0.108582
LATITUD      0.580733
ALTITUD      1.000000
Name: ALTITUD, dtype: float64

```

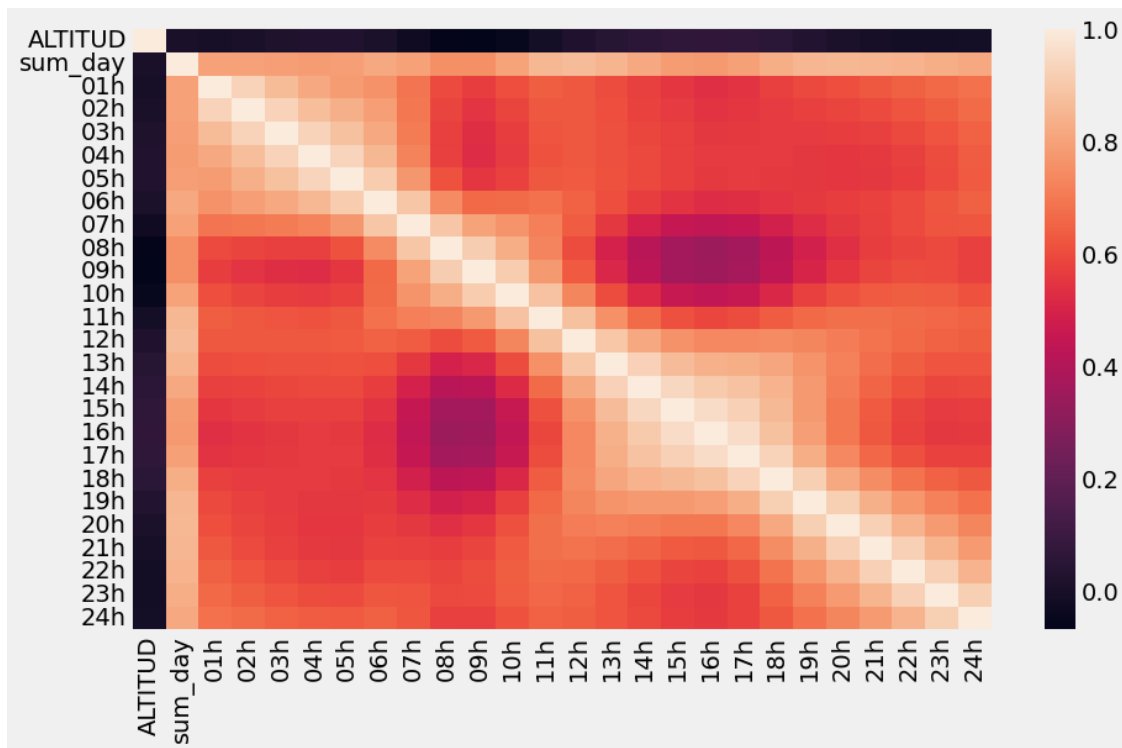
```

[30]: corr = df[["ALTITUD", "sum_day"] + cols].corr()

sns.heatmap(corr,
            xticklabels=corr.columns,
            yticklabels=corr.columns)

```

[30]: <AxesSubplot: >



We find no significant correlation between altitude and concentration of particles in the air.

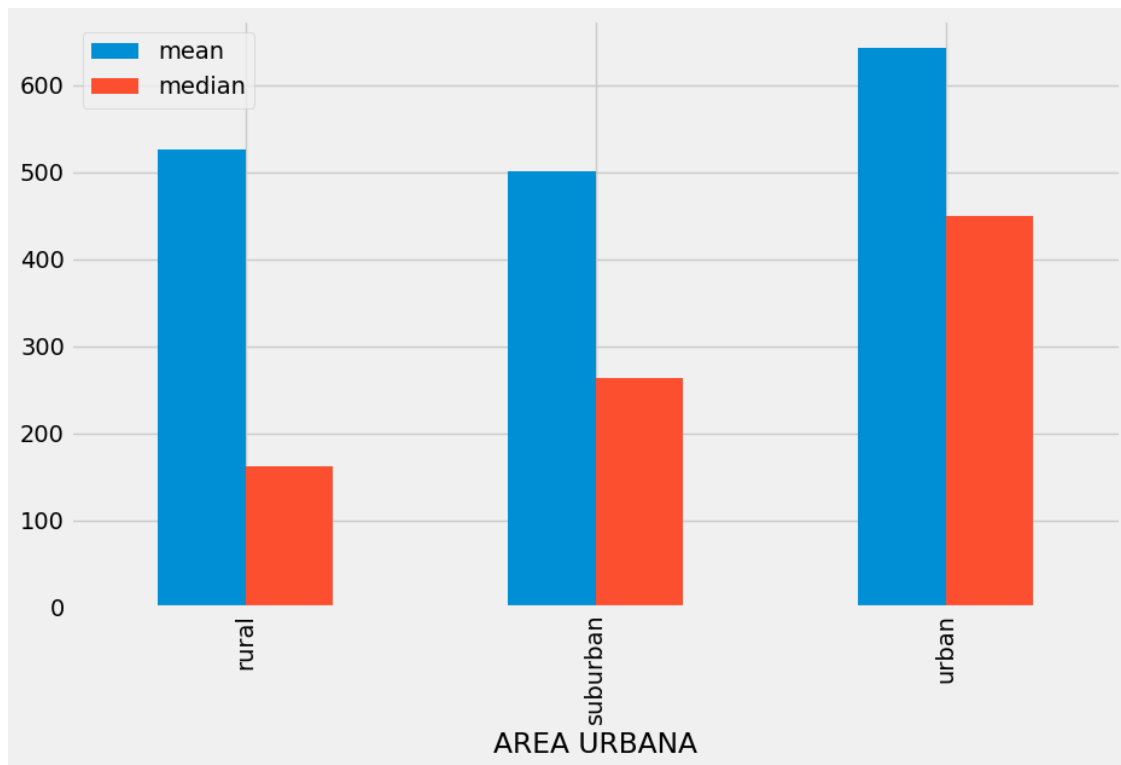
We Analyze the concentration of pollutants in urban, suburban and rural areas, and present your conclusion in graphical form.

```

[31]: df.groupby(['AREA URBANA'])["sum_day"].agg(['mean', "median"]).plot(kind="bar")

```

[31]: <AxesSubplot: xlabel='AREA URBANA'>



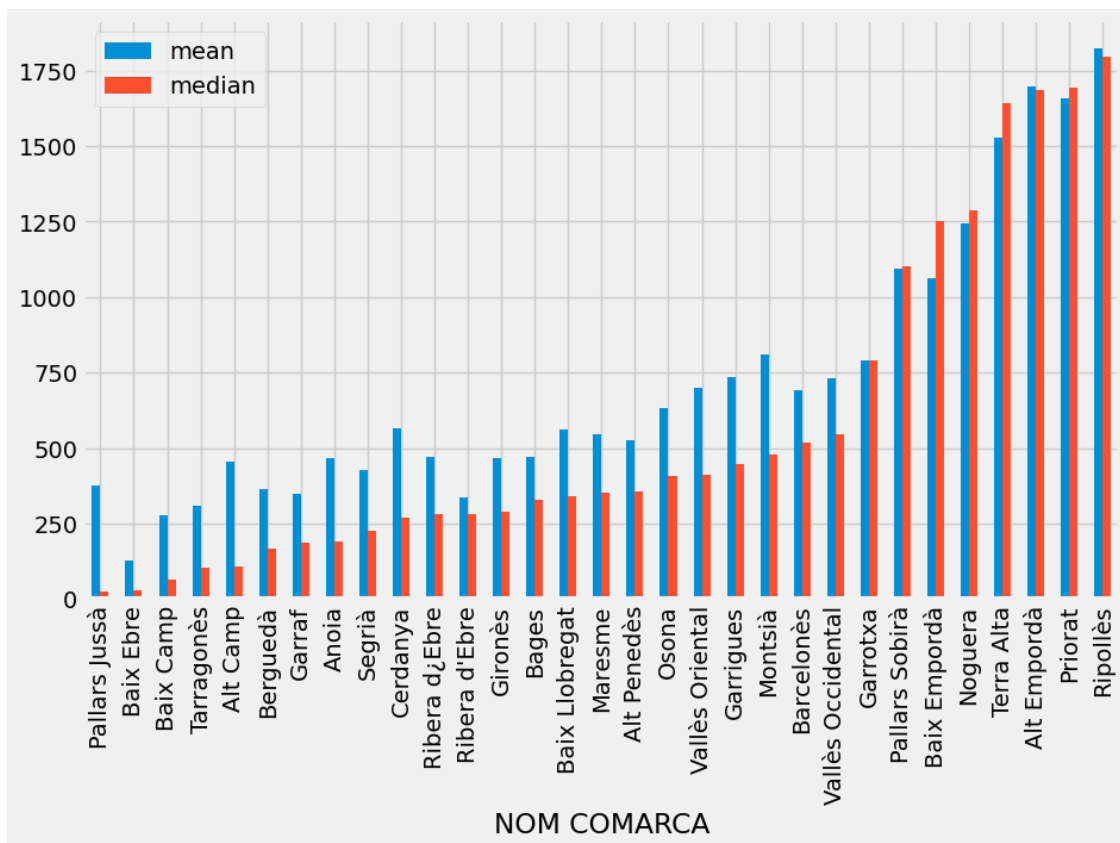
We find by far more pollution in urban areas on average and median. On median suburban are more polluted than rural areas.

Rank the cities in the dataset according to their level of pollution, and create best-5 and worst-5 lists.

Here we rank comarcas from best to worst:

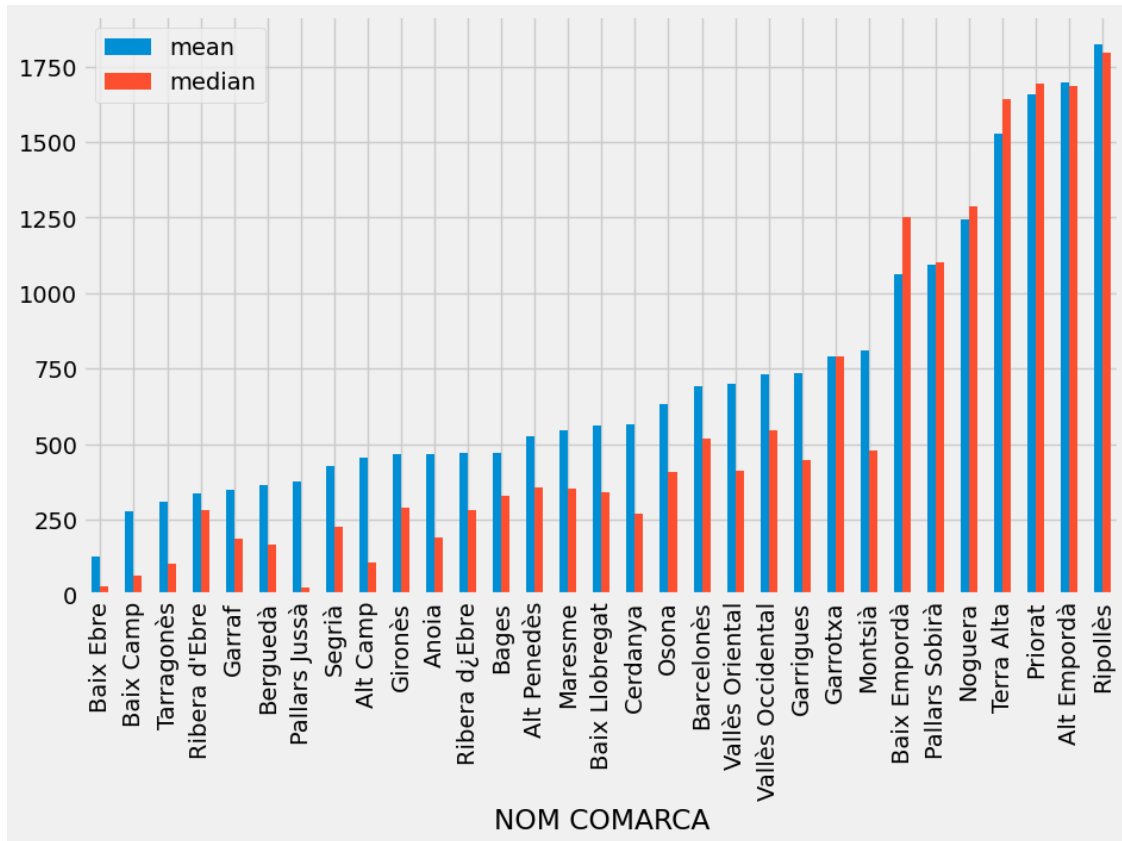
```
[32]: df.groupby(['NOM COMARCA'])["sum_day"].agg(['mean', "median"]).
      ↪sort_values(by='median').plot(kind="bar")
```

```
[32]: <AxesSubplot: xlabel='NOM COMARCA'>
```



```
[34]: df.groupby(['NOM COMARCA'])["sum_day"].agg(['mean', 'median']).
      ↪sort_values(by='mean').plot(kind="bar")
```

```
[34]: <AxesSubplot: xlabel='NOM COMARCA'>
```



```
[36]: df.groupby(['NOM COMARCA', "MAGNITUD"])[ "sum_day" ].agg(['mean', "median"]).
      ↪sort_values(by='mean')
```

```
[36]:
```

NOM COMARCA	MAGNITUD	mean	median
Pallars Jussà	6	3.013423	2.4
Baix Ebre	6	5.308080	4.8
Alt Camp	6	5.977174	5.4
Segrià	44	6.477778	5.2
Bages	44	6.626375	5.7
...	
Ripollès	14	1825.318131	1797.0
Baix Empordà	14	1895.151279	1880.0
Pallars Jussà	14	2136.343627	2123.0
Baix Llobregat	3	2161.889456	2179.0
	58	2744.966667	699.0

[218 rows x 2 columns]

0.2.1 More granular data on pollutant 8 for algo.

```
[38]: df[df['MAGNITUD'] == 8]['NOM ESTACIO'].value_counts()
```

```
[38]: Perafort (Puigdelfí)          11261
      Tarragona (Sant Salvador)     11123
      Tarragona (Bonavista)        11055
      Constantí                    11037
      Manresa                      10896
      ...
      Gavà (c/Girona - c/Progrés)  1232
      Sta. Coloma de Gr. (c/ Bruc)   942
      Vila-seca (IES Vila-seca)     877
      Barcelona (Torre Girona)      282
      el Prat de Llobregat (Sant Cosme)  5
      Name: NOM ESTACIO, Length: 96, dtype: int64
```

```
[39]: df[df['MAGNITUD'] == 8].groupby(['year', 'month', "NOM ESTACIO"])[['sum_day']].
      ↪agg(['mean', "median", "count"])
```

```
[39]:
```

			mean	median	count
year	month	NOM ESTACIO			
1991	1	Badalona	776.933333	834.0	30
		Barcelona (Poblenou)	1054.903226	1090.0	31
		Barcelona (St. Gervasi)	1760.058824	1663.0	17
		L'Hospitalet de Llobregat	1002.200000	1049.5	30
		Montcada i Reixac	1318.033333	1317.0	30
...		
2023	1	Vandellòs (Viver)	150.680000	116.0	25
		Vila-seca (IES Vila-seca)	394.520000	312.0	25
		Viladecans - Atrium	589.360000	611.0	25
		Vilafranca del Penedès	191.800000	147.0	25
		Vilanova i la Geltrú	436.000000	465.0	25

[18973 rows x 3 columns]

```
[67]: df[df['MAGNITUD'] == 8].groupby(['year', 'month', "NOM ESTACIO"])[['sum_day']].
      ↪agg(['mean']).reset_index()["NOM ESTACIO"].nunique()
```

```
[67]: 96
```

```
[68]: df[df['MAGNITUD'] == 8].groupby(['year', 'month', "NOM ESTACIO"])[['sum_day']].
      ↪agg(['mean']).reset_index()["NOM ESTACIO"].unique()
```

```
[68]: array(['Badalona', 'Barcelona (Poblenou)', 'Barcelona (St. Gervasi)',
      'L'Hospitalet de Llobregat', 'Montcada i Reixac',
      'Sant Adrià de Besòs', 'Vallcebre', 'Cercs (St. Corneli)'],
```

```
'la Nou de Berguedà (Malanyeu)', 'Constantí', 'Manresa',
'Perafort (Puigdelfí)', 'Tarragona (Bonavista)',
'Tarragona (pl. Generalitat)', 'Vila-seca',
'la Pobla de M./el Morell', 'Tarragona (Sant Salvador)',
'Igualada', 'Martorell', 'Terrassa', 'Vic', 'Sarrià de Ter',
'Granollers (av. Joan Prim)', 'Mollet del Vallès', 'Reus',
'Mataró', 'Barcelona (Sagrera)', 'Cercs (St. Jordi)', 'Lleida',
'Sabadell (pl. Creu de Barberà)', 'Sant Fost de Campsentelles',
'Sabadell', 'Sant Celoni', 'Rubí', 'Sta. Coloma de Gr. (c/ Bruc)',
'Sant Cugat del Vallès', 'Tarragona (Universitat Laboral)',
'Vilanova i la Geltrú', 'Fornells de la Selva (escola municipal)',
'Barcelona (Sants)', 'Granollers (c/ Joan Vinyoli)',
'Sta. Perpètua de Mogoda', 'Vilafranca del Penedès',
'Barcelona (Eixample)', 'Santa Coloma de Gramenet',
'Barcelona (Gràcia - Sant Gervasi)', 'Barberà del Vallès',
'Sant Andreu de la Barca', 'el Prat de Llobregat (església)',
'Sant Vicenç dels Horts (Ribot)', 'Gavà (c/Girona - c/Progrés)',
'Cornellà de Llobregat (Allende - Bonveí)',
'Tarragona (Parc de la Ciutat)', 'Cercs (Sant Jordi)',
'Bellver de Cerdanya', 'Barcelona (Ciutadella)',
'Girona (parc de la Devesa)', 'Gavà', 'Cubelles (Poliesportiu)',
'Tona', 'Alcover', 'Vallcebre (campanar)',
'Santa Perpètua de Mogoda', 'Castellet i la Gornal',
'Cercs (Sant Corneli)', 'Vandellòs (Els Dedalts)',
'Vandellòs (Viver)', 'Berga', 'Barcelona (Parc Vall Hebron)',
'Montseny (La Castanya)', 'Granollers', 'Viladecans - Atrium',
'el Prat de Llobregat (Sant Cosme)', 'Tona (Zona Esportiva)',
"L'Ametlla de Mar", 'Sta. Margarida i els Monjos (La Ràpita)',
'El Prat de Llobregat (Jardins de la Pau)', 'Amposta',
'Sitges (Vallcarca)', 'Vandellòs (Barranc del Terme)',
'Barcelona (Torre Girona)', 'Manlleu', 'Montsec',
'El Prat de Llobregat (Sagnier)', 'Barcelona (Palau Reial)',
'Girona (Escola de Música)', 'Pallejà (Roca de Vilana)', 'Alcanar',
'Sant Vicenç dels Horts', 'Sant Feliu de Ll. (CEIP Marti i Pol)',
'Sitges (Vallcarca - Oficines)', 'Juneda (Pla del Molí)', 'Begur',
'Santa Pau', 'Barcelona (Observatori Fabra)',
'Vila-seca (IES Vila-seca)'], dtype=object)
```

```
[42]: df[df['MAGNITUD'] == 8].groupby(['year', 'month', "NOM ESTACIO"])[['sum_day']].
      ↪agg(['mean']).reset_index()
```

```
[42]:
```

	year	month	NOM ESTACIO	mean
0	1991	1	Badalona	776.933333
1	1991	1	Barcelona (Poblenou)	1054.903226
2	1991	1	Barcelona (St. Gervasi)	1760.058824
3	1991	1	L'Hospitalet de Llobregat	1002.200000
4	1991	1	Montcada i Reixac	1318.033333

...
18968	2023	1	Vandellòs (Viver)	150.680000
18969	2023	1	Vila-seca (IES Vila-seca)	394.520000
18970	2023	1	Viladecans - Atrium	589.360000
18971	2023	1	Vilafranca del Penedès	191.800000
18972	2023	1	Vilanova i la Geltrú	436.000000

[18973 rows x 4 columns]

0.3 Prediction

To Build and publish an algorithm to predict the average concentration of one pollutant of your choice per month for the next 24 months - on average for all stations.

We have chosen the pollutant 8 as its the most frequent and We have aggregated the data by day and then month. We also label encoded the stations. We used Random Forest regressor to predict the target values for the next 24 months. The algo is available at [GitHub](#) and [Ocean Protocol](#).

To Build and publish an algorithm to predict the concentration of one pollutant of your choice for each hour of the day from February 15 to 28 - on average for all stations. We add time features and label encoded the stations. We used Random Forest regressor to predict the target values for the next 14 days. The algo is available at [GitHub](#) and [Ocean Protocol](#).

0.4 Summary

In this project, we have analyzed air quality in Catalonia by using the data from the Catalan Transparency Portal. We have used the data to explore the evolution of air pollution in Catalonia over the past three decades. We have conducted an exploratory data analysis and created visualisations to understand the data. We have also used machine learning algorithms to predict air pollutant concentrations. Finally, we have written a report to summarise our findings.

Overall, our analysis has shown that air pollution levels in Catalonia are significantly higher in urban areas compared to suburban and rural areas. We have also found that air pollution tends to be worse in winter due to meteorological and human-made factors, and that the diurnal cycle affects the concentration of pollutants in the air. We have also developed two machine learning algorithms to predict air pollutant concentrations for the next 24 months and 14 days.

0.4.1 Limitations

The main limitation of the project was the lack of data. Although we have used the data from the Catalan Transparency Portal, the data was only available from the year 1991 which limits our ability to explore the further trend of air pollution in Catalonia. Additionally, the data was only available for a limited amount of dates which limited our ability to model the data on a more granular level.

We recommend that future competitions should provide more data in order to be able to explore the trend of air pollution in Catalonia in more detail. Additionally, we recommend that the data should be available on a more granular level in order to be able to model the data more accurately. Finally, we recommend that the data should be provided from a wider range of sources in order to get a more comprehensive view of air pollution in Catalonia.