Exploratory Data Analysis for Machine Learning – Course Project

IBM Machine Learning

Author: Isidro Brevers Gómez

E-mail: ibrevers@gmail.com

1. Brief description of the data set and a summary of its attributes

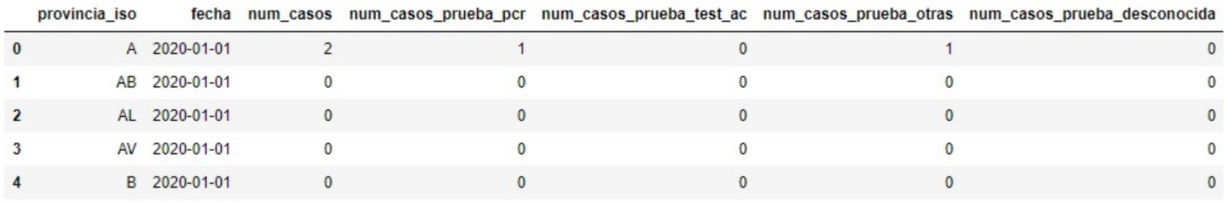
The data set used contains information about COVID-19 detected cases in Spain, by region and date, including a detail on the specific tests used for detection. This data is published and updated daily by the Spanish Ministry of Health, and the data set is available at https://cnecovid.isciii.es/ in CSV format file.

The data set includes de following information:

- *provincia_iso* [object]: ISO code for each region.
- *fecha* [object]: detection date.
- *num_casos* [int64]: number of cases detected.
- *num_casos_prueba_pcr* [int64]: number of cases detected through PCR tests.
- *num_casos_prueba_test_ac* [int64]: number of cases detected through AC tests.
- *num_casos_prueba_otras* [int64]: number of cases detected through other tests.
- *num_casos_prueba_desconocida* [int64]: number of cases detected through unknown tests.

2. Initial plan for data exploration

As a first step, the CSV file was downloaded and read into a dataframe for further exploration and analysis, resulting into the following (first 5 results):

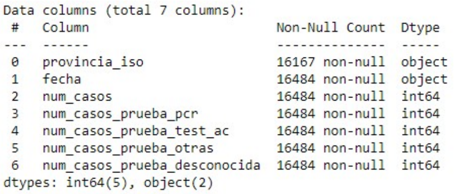


	provincia_iso	fecha	num_casos	num_casos_prueba_pcr	num_casos_prueba_test_ac	num_casos_prueba_otras	num_casos_prueba_desconocida
0	A	2020-01-01	2	1	0	1	0
1	AB	2020-01-01	0	0	0	0	0
2	AL	2020-01-01	0	0	0	0	0
3	AV	2020-01-01	0	0	0	0	0
4	B	2020-01-01	0	0	0	0	0

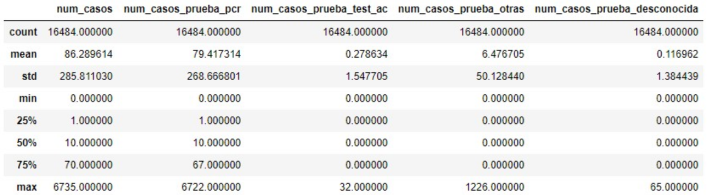
The above represents the number of COVID cases detected by region and date, including a detail on how many cases were detected through each kind of test.

Then, some initial checks were done in order to verify:

- The size of the data: 16,380 rows, 7 columns.
- Data does not include null or inconsistent values.
- Data types.
- Basic statistical attributes of the data.



```
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   provincia_iso                 16167 non-null  object
 1   fecha                         16484 non-null  object
 2   num_casos                     16484 non-null  int64
 3   num_casos_prueba_pcr          16484 non-null  int64
 4   num_casos_prueba_test_ac      16484 non-null  int64
 5   num_casos_prueba_otras        16484 non-null  int64
 6   num_casos_prueba_desconocida  16484 non-null  int64
dtypes: int64(5), object(2)
```

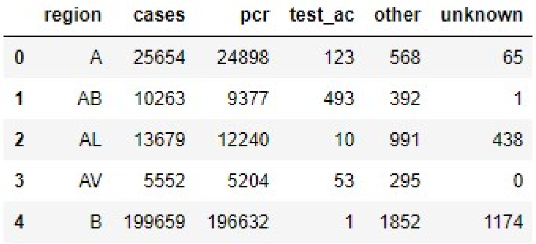


	num_casos	num_casos_prueba_pcr	num_casos_prueba_test_ac	num_casos_prueba_otras	num_casos_prueba_desconocida
count	16484.000000	16484.000000	16484.000000	16484.000000	16484.000000
mean	86.289614	79.417314	0.278634	6.476705	0.116962
std	285.811030	268.666801	1.547705	50.128440	1.384439
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	1.000000	1.000000	0.000000	0.000000	0.000000
50%	10.000000	10.000000	0.000000	0.000000	0.000000
75%	70.000000	67.000000	0.000000	0.000000	0.000000
max	6735.000000	6722.000000	32.000000	1226.000000	65.000000

3. Actions taken for data cleansing and feature engineering

Once data was read, some data cleaning was necessary in order to ensure a better understanding and further analysis of the information.

- Columns were renamed.
- Date column type was changed to datetime type.
- There was no need to analyze or substitute null values.
- Data was grouped and shorted (A-Z) by region.

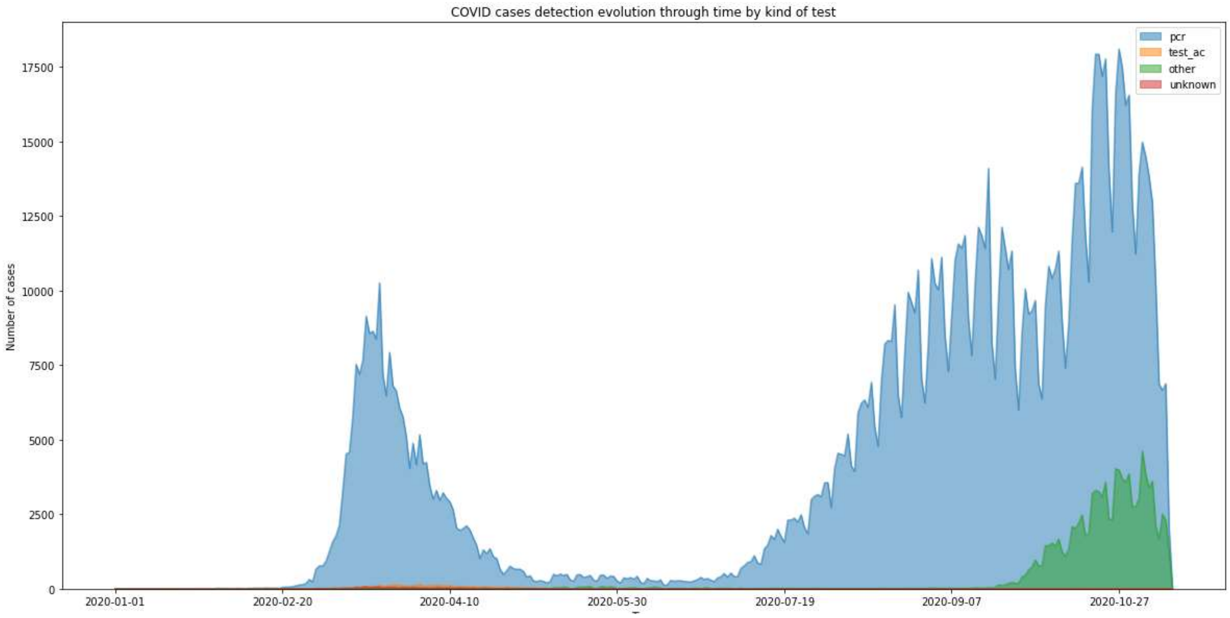


	region	cases	pcr	test_ac	other	unknown
0	A	25654	24898	123	568	65
1	AB	10263	9377	493	392	1
2	AL	13679	12240	10	991	438
3	AV	5552	5204	53	295	0
4	B	199659	196632	1	1852	1174

- For visualization purposes, specific dataframes were created grouping data, delete columns and setting specific indexes.

4. Key Findings and Insights, which synthesizes the results of Exploratory Data Analysis in an insightful and actionable manner

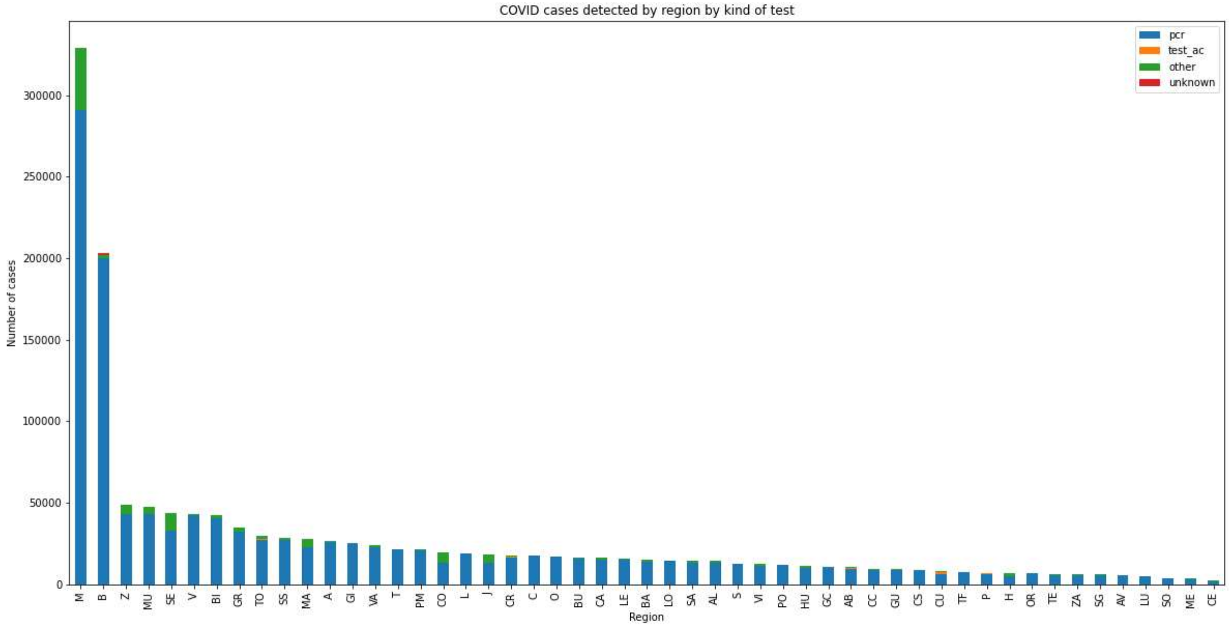
First, we visualized the evolution through time of COVID cases detected by each kind of test:



As per the above plot, we can extract the following insights:

- There are two waves: first between March and April 2020, second between August and November 2020.
- Maximum daily detection was over 10,000 cases in the first wave, and over 17,500 cases in the second wave.
- Second wave is deeper that the first one, almost double.
- Most of the cases have been detected through PCR test.
- AC tests have not significantly contributed to COVID detection.
- Since October 2020, other tests are increasingly contributing to COVID detection.
- Unknown tests were used between March and April 2020.

Then, we visualized the COVID detection by test cases by region:



As per the above stacked bar chart, we can extract the following insights:

- Madrid and Barcelona are the regions with higher detected cases.
- Most of the cases have been detected through PCR tests.
- Other tests have also been used in some regions, mainly in Madrid, Zaragoza, Murcia, Segovia, Córdoba and Jaen.
- Unknown tests have been used in Barcelona and Ciudad real.

5. Formulating 3 hypothesis about this data

Null hypothesis (H0) exists when the data scientist believes that there is no relationship between two variables, or that there is a lack of information to state a scientific hypothesis. In an attempt to disprove a null hypothesis, data scientist will seek to discover an alternative hypothesis (H1.

Hypothesis 1:

- H0: the kind of test done is irrelevant for COVID detection.
- H1: PCR tests improve COVID cases detection, opposite to other tests.

Hypothesis 2:

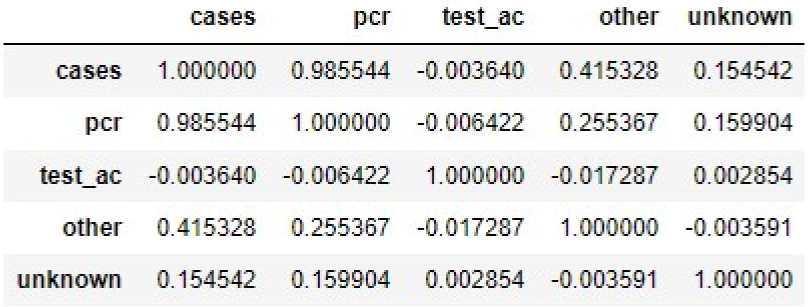
- H0: total population does not have influence on the relative number of cases.
- H1: there are other factors with influence rather than population (e.g. population concentration).

Hypothesis 3:

- H0: measures enforced by governments do not have influence on the number of cases detected.
- H1: some of the measures help to decrease the number of cases (e.g. lockdown), as opposed to others (e.g. curfews).

6. Significance test for one of the hypotheses and results' discussion

It was decided to test the first hypothesis. For these purposes, Pearson's correlation coefficient was selected to test whether variables had a linear relationship:



	cases	pcr	test_ac	other	unknown
cases	1.000000	0.985544	-0.003640	0.415328	0.154542
pcr	0.985544	1.000000	-0.006422	0.255367	0.159904
test_ac	-0.003640	-0.006422	1.000000	-0.017287	0.002854
other	0.415328	0.255367	-0.017287	1.000000	-0.003591
unknown	0.154542	0.159904	0.002854	-0.003591	1.000000

As per the results obtained:

- Pearson correlation coefficient between the number of cases detected and those detected through PCR tests is 0.9855, this meaning that there is a **strong positive correlation** between the number of cases detected and those detected through PCR tests; P-value is 0.00 (<0.05 -alpha-), this meaning that the correlation is **statistically significant**.
- Pearson correlation coefficient between the number of cases detected and those detected through AC tests is (0.0036), this meaning that there is a **no correlation** between the number of cases detected and those detected through AC tests; P-value is 0.64 (<0.05 -alpha-), this meaning that the correlation is **not statistically significant**.
- Pearson correlation coefficient between the number of cases detected and those detected through PCR tests is 0.4153, this meaning that there is a **weak positive correlation** between the number of cases detected and those detected through other tests; P-value is 0.00 (<0.05 -alpha-), this meaning that the correlation is **statistically significant**.

As per the above, it has been tested that **PCR tests are strongly correlated** with the number of cases detected, while **other tests are weakly correlated**, being these correlations statistically significant (p-value < alpha). On the other hand, the non-correlation of AC tests is not statistically significant.

Therefore, inasmuch as the kind of test done is relevant for COVID cases detection, **null hypothesis (H0) can be rejected**. In addition, based on the statistically significant strong positive correlation between the number of cases detected and those detected through PCR tests, the **alternative hypothesis (H1) can be accepted**.

7. Suggestions for next steps in analyzing the data

In order to improve and go deeper into the analysis done, the following is proposed:

- The current dataset to be updated and complemented with additional data, such us the full name of the regions, total population of each region or the number of tests done (including those where the result was negative).

- The other hypothesis formulated may be tested, by means of testing:

  - How total population influences the number of cases detected by region, in respect of other factors (e.g. population concentration).
  - Whether the different measures enforced by central and local governments (e.g. lockdowns, curfews…) have influence or not on the number of cases evolution.

- Due to the geographical distribution of the data, map visualization would be useful for EDA (e.g. folium).

- Model development (e.g. multi-linear regression) and evaluation (e.g. Residual plot, R-squared) could be introduced to develop and test different models, for the purposes of understanding how exactly variables impact.

8. Summary of the quality of the data set and request for additional data

The dataset quality was good, being that expected as it was coming from a government official source and used for public statistics purposes.

Additional data that could be useful for the analysis:

- Data about the total number of tests done, by kind of test, including those where the result was negative (so it could be measured the efficiency of each test).

- Data about the cost and time recurred for each kind of test (so efficiency could also consider the cost associated).

- Data about the regions, such as total population or population concentration (so it could be analyzed how different variables influence on the number of cases and its evolution).

In [1]:

```
# Import libraries
import pandas as pd
import numpy as np
from datetime import datetime
from scipy import stats
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:

```
# Download information (source: Spanish Ministry of Health https://cnecovid.is
ciii.es/)
!wget -q -O 'evolution.csv' https://cnecovid.isciii.es/covid19/resources/datos
_provincias.csv
```

In [3]:

```
# Read the CSV file downloaded into a dataframe
df_evol = pd.read_csv('evolution.csv', sep=',')
df_evol.head(10)
```

Out[3]:

	provincia_iso	fecha	num_casos	num_casos_prueba_pcr	num_casos_prueba_test_ac	num_ca
0	A	2020-01-01	2	1	0	
1	AB	2020-01-01	0	0	0	
2	AL	2020-01-01	0	0	0	
3	AV	2020-01-01	0	0	0	
4	B	2020-01-01	0	0	0	
5	BA	2020-01-01	0	0	0	
6	BI	2020-01-01	0	0	0	
7	BU	2020-01-01	0	0	0	
8	C	2020-01-01	0	0	0	
9	CA	2020-01-01	0	0	0	

In [4]:

```
# Check the size of the df
df_evol.shape
```

Out[4]:

```
(16484, 7)
```

In [5]:
```
df_evol.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16484 entries, 0 to 16483
Data columns (total 7 columns):
 #   Column                       Non-Null Count  Dtype 
---  ------                       --------------  ----- 
 0   provincia_iso                16167 non-null  object
 1   fecha                        16484 non-null  object
 2   num_casos                    16484 non-null  int64 
 3   num_casos_prueba_pcr         16484 non-null  int64 
 4   num_casos_prueba_test_ac     16484 non-null  int64 
 5   num_casos_prueba_otras       16484 non-null  int64 
 6   num_casos_prueba_desconocida 16484 non-null  int64 
dtypes: int64(5), object(2)
memory usage: 901.6+ KB
```

In [6]:
```
df_evol.describe()
```

Out[6]:

	num_casos	num_casos_prueba_pcr	num_casos_prueba_test_ac	num_casos_prueba_otra
count	16484.000000	16484.000000	16484.000000	16484.00000
mean	86.289614	79.417314	0.278634	6.47670
std	285.811030	268.666801	1.547705	50.12844
min	0.000000	0.000000	0.000000	0.00000
25%	1.000000	1.000000	0.000000	0.00000
50%	10.000000	10.000000	0.000000	0.00000
75%	70.000000	67.000000	0.000000	0.00000
max	6735.000000	6722.000000	32.000000	1226.00000

In [7]:

```
# Rename columns
df_evol.rename(columns={'provincia_iso':'region', 'fecha':'date', 'num_casos':
'cases', 'num_casos_prueba_pcr':'pcr', 'num_casos_prueba_test_ac':'test_ac',
'num_casos_prueba_otras':'other', 'num_casos_prueba_desconocida':'unknown'}, i
nplace=True)

# Date column to datetime type
df_evol['date'] = pd.to_datetime(df_evol['date'])
df_evol.dtypes

df_evol.head(10)
```

Out[7]:

	region	date	cases	pcr	test_ac	other	unknown
0	A	2020-01-01	2	1	0	1	0
1	AB	2020-01-01	0	0	0	0	0
2	AL	2020-01-01	0	0	0	0	0
3	AV	2020-01-01	0	0	0	0	0
4	B	2020-01-01	0	0	0	0	0
5	BA	2020-01-01	0	0	0	0	0
6	BI	2020-01-01	0	0	0	0	0
7	BU	2020-01-01	0	0	0	0	0
8	C	2020-01-01	0	0	0	0	0
9	CA	2020-01-01	0	0	0	0	0

In [8]:

```
# By region

df_region = df_evol.groupby('region').sum().reset_index()
df_region.sort_values(['cases'], ascending=False, axis=0, inplace=True)
del df_region['cases']
df_region.set_index('region', inplace=True)
df_region.head(10)
```

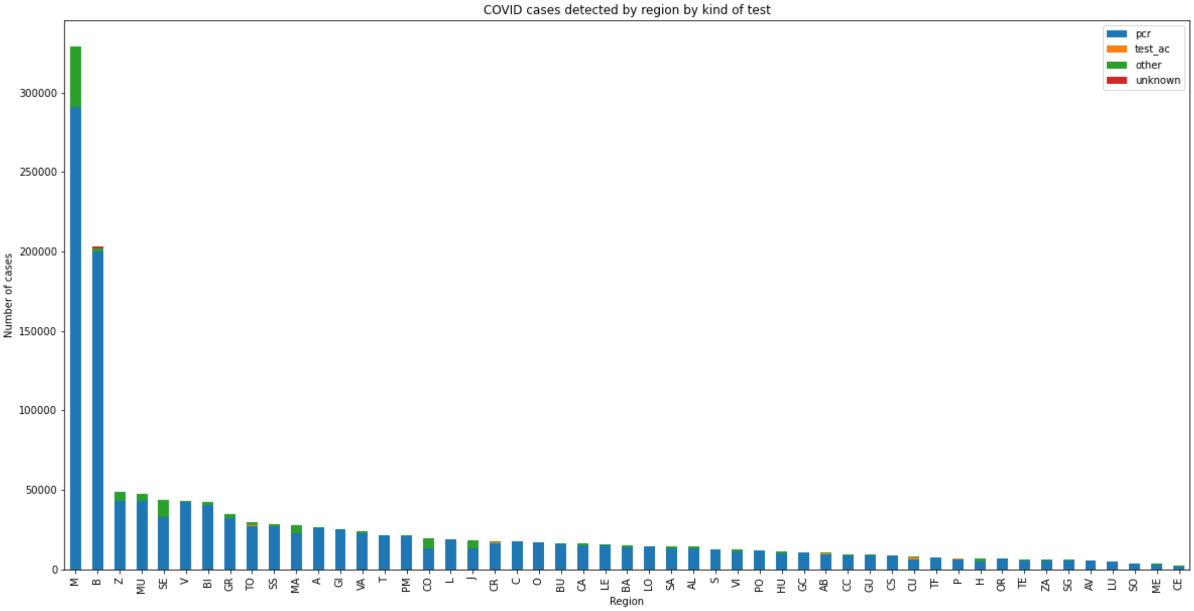
Out[8]:

	pcr	test_ac	other	unknown
region				
M	290866	22	37997	0
B	199826	1	1858	1175
Z	43241	0	5395	57
MU	43087	116	4279	17
SE	33068	58	10283	0
V	42144	202	503	85
BI	40563	0	1696	91
GR	31994	69	2477	0
TO	27296	638	1556	4
SS	27018	0	1160	54

```
In [9]: df_region.plot(kind='bar',
                       stacked=True,
                       figsize=(20, 10),
                       )

        plt.title('COVID cases detected by region by kind of test')
        plt.ylabel('Number of cases')
        plt.xlabel('Region')

        plt.show()
```



In [10]:

```
# By date

df_date = df_evol.groupby('date').sum().reset_index()
del df_date['cases']
df_date.sort_values(['date'], ascending=True, axis=0, inplace=True)
df_date.set_index('date', inplace=True)
df_date.head(10)
```

Out[10]:

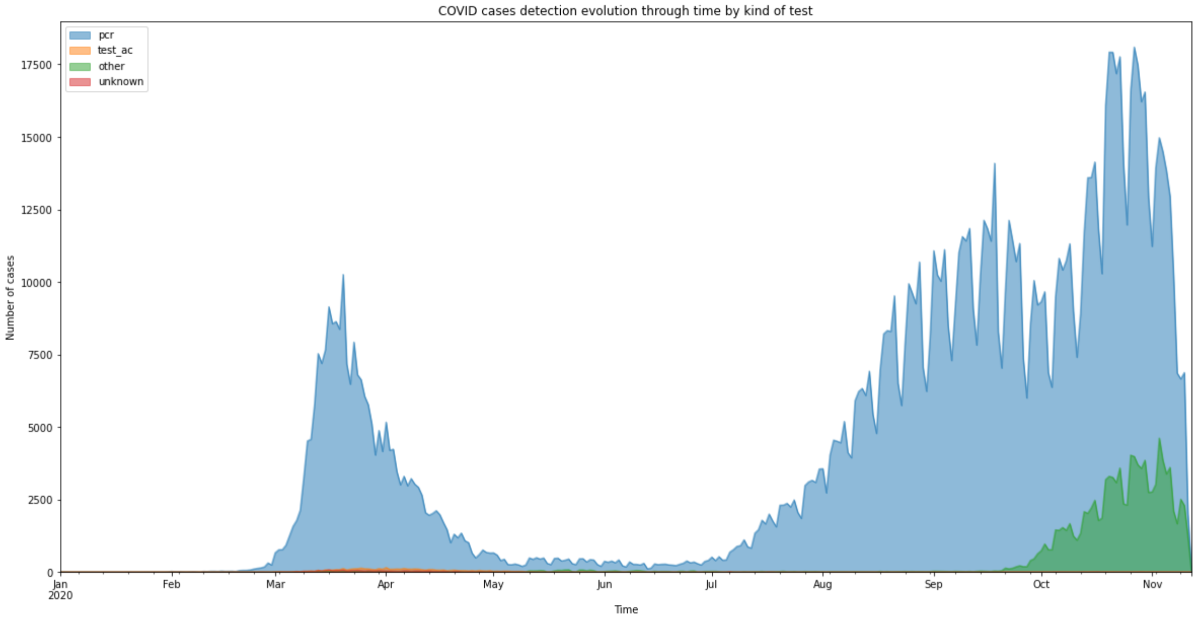
	pcr	test_ac	other	unknown
date				
2020-01-01	4	0	1	0
2020-01-02	6	0	0	0
2020-01-03	1	0	0	0
2020-01-04	1	0	0	0
2020-01-05	2	0	0	0
2020-01-06	0	0	1	0
2020-01-07	2	0	0	0
2020-01-08	1	0	1	0
2020-01-09	0	0	0	0
2020-01-10	2	0	0	0

In [11]:

```
df_date.plot(kind='area',
             stacked=False,
             alpha=0.5,
             figsize=(20, 10),
             )

plt.title('COVID cases detection evolution through time by kind of test')
plt.ylabel('Number of cases')
plt.xlabel('Time')

plt.show()
```



In [12]:

```
# Correlation

df_evol[['cases','pcr', 'test_ac', 'other', 'unknown']].corr()
```

Out[12]:

	cases	pcr	test_ac	other	unknown
cases	1.000000	0.985544	-0.003640	0.415328	0.154542
pcr	0.985544	1.000000	-0.006422	0.255367	0.159904
test_ac	-0.003640	-0.006422	1.000000	-0.017287	0.002854
other	0.415328	0.255367	-0.017287	1.000000	-0.003591
unknown	0.154542	0.159904	0.002854	-0.003591	1.000000

In [13]:

```
pearson_coef, p_value = stats.pearsonr(df_evol['cases'], df_evol['pcr'])
print("The Pearson Correlation Coefficient is", pearson_coef, " with a P-value
of P =", p_value)
pearson_coef, p_value = stats.pearsonr(df_evol['cases'], df_evol['test_ac'])
print("The Pearson Correlation Coefficient is", pearson_coef, " with a P-value
of P =", p_value)
pearson_coef, p_value = stats.pearsonr(df_evol['cases'], df_evol['other'])
print("The Pearson Correlation Coefficient is", pearson_coef, " with a P-value
of P =", p_value)
pearson_coef, p_value = stats.pearsonr(df_evol['cases'], df_evol['unknown'])
print("The Pearson Correlation Coefficient is", pearson_coef, " with a P-value
of P =", p_value)
```

```
The Pearson Correlation Coefficient is 0.9855440877276274  with a P-value of
P = 0.0
The Pearson Correlation Coefficient is -0.0036401312461791773  with a P-value
of P = 0.6402690399050847
The Pearson Correlation Coefficient is 0.41532776336637633  with a P-value of
P = 0.0
The Pearson Correlation Coefficient is 0.15454176062999933  with a P-value of
P = 1.2237858324025643e-88
```