Analysis of COVID-19 Hospitalized Cases in the Canary Islands Data

By Ginevra Iorio

Abstract

Official datasets reporting COVID-19 cases registered in Canary Islands' hospitals are analyzed in a data quality point of view and manually labeled using MATLAB, obtaining ready-to-use training datasets.

Introduction

Canary Islands' COVID-19 hospitals statistics have been reported by the public authorities and made available for this study, with the goal of forecasting future cases through a time series analysis. The data points that are used for this goal are typically made up of consecutive measurements taken from a similar source over a period of time, which will be daily in this case [1]. Before being ready to use, the data needs to be studied and cleaned in case inconsistencies or missing entries are found, since it has to meet the process quality requirements to make them operate efficiently.

Data Cleaning

The reliability of a dataset is indicated by its data quality, which refers to the development and implementation of activities that use quality management techniques to ensure data is fit to serve the specific needs of this project. Duplicated data, incomplete data, inconsistent data, incorrect data, poorly defined data, poorly organized data, and poor data security are examples of data quality issues [2]. There are six common dimensions that ensure a high-quality dataset:

1. **Completeness:** the degree to which the necessary data is available for use;

2. **Uniqueness:** the degree to which the data is unique and cannot be mistaken for other entries;

3. **Consistency:** the degree to which the data is equal within and between datasets;

4. **Validity:** the degree to which the data is within defined requirements like format, type and range;

5. **Accuracy:** the degree to which the data represents the reality;

6. **Timeliness:** the degree to which the data is available at the time it is needed [3].

The data to be studied is recorded in different .xlsx files named *Acumulado.xlsx*, *Diario.xlsx, Ingresos.xlsx*, *Pacientes Covid 19.xlsx* and *Pacientes no Covid 19.xlsx.* One at a time, they are analyzed following the six data quality dimensions through MATLAB, a programming platform created specifically for engineers and scientists to analyze and design systems through the MATLAB language, a matrix-based language that allows for the most natural expressions of computational mathematics [4].

The following steps described are used to prepare the MATLAB environment for it to work with all the available data and are repeated for every existing sheet.
First of all, the sheets of the .xlsx files are loaded one at a time, in different MATLAB workspaces, using the *readtable* function:

```
inCOVpatients = readtable('Ingresos.xlsx', 'Sheet', 'Ingresos Covid por urgencias');
```

Subsequently, the first column is deleted, being useless for the sake of the study of the data, since it only contains the description of the entries, which can instead be represented only by the name of the table saved in the workspace. The other two columns, which respectively contain the dates of the measurements and the number of the resulting patients, are indeed renamed as “Date” and “NumCases”:

```
inCOVpatients = removevars(inCOVpatients, "Var1");
inCOVpatients.Properties.VariableNames(1) = "Date";
inCOVpatients.Properties.VariableNames(2) = "NumCases";
```

At this point, the Date column needs to be transformed to *datetime* format in order for the data to be represented in a time series chart efficiently. The initially available dates are in String format and the months’ names are in Spanish. This means that before the format transformation, the names of the months that differ from their English version need to be fixed:

```
inCOVpatients.Date = replace(inCOVpatients.Date(:,:), 'ene', 'jan')
inCOVpatients.Date = replace(inCOVpatients.Date(:,:), 'abr', 'apr')
inCOVpatients.Date = replace(inCOVpatients.Date(:,:), 'ago', 'aug')
inCOVpatients.Date = replace(inCOVpatients.Date(:,:), 'sept', 'sep')
inCOVpatients.Date = replace(inCOVpatients.Date(:,:), 'dic', 'dec')
```

The date can finally be transformed to *datetime* format using the following function:

```
inCOVpatients.Date = datetime(inCOVpatients.Date(:,:) ,'InputFormat','MMM dd yyyy','Format','dd/MMM/yyyy')
```

Data Labeling

The datasets are now ready to be used and are saved inside MATLAB workspaces that can easily be retrieved later in order to be studied and accurately labeled. Data is the process of identifying raw data (images, text files, videos, etc.) and adding one or more meaningful and informative labels to provide context for a machine learning model to learn from. The most common types of data labeling are:

- **Computer Vision.** Labeling images, pixels, or key points, or creating a border that completely encloses a digital image, known as a bounding box. This training data can then be used to create a computer vision model that can automatically categorize images, detect the location of objects, identify key points in an image, or segment an image.
- **Natural Language Processing.** Manually identifying important sections of text or tagging the text with specific labels. Sentiment analysis, entity name recognition, and optical character recognition are all applications of natural language processing models.
- **Audio Processing.** Audio processing is the process of converting unstructured sounds such as speech, wildlife noises (barks, whistles, or chirps), and building sounds (breaking glass, scans, or alarms) into a structured format that can be used in machine learning. Audio processing frequently necessitates manually transcribing into written text and then, by adding tags and categorizing the audio, it is easy to obtain more information about it [5].

For the purpose of this project, natural language processing will be used, manually labeling the dataset entries through the MATLAB matrix-oriented language.

COVID-19 Cases Data

In this section, the collected data on COVID-19 cases of the Canary Islands, such as new positivities, deaths and discharges will be studied.

The two available datasets, *Acumulado.xlsx* and *Diario.xlsx,* range from January 31st (or 20th in some daily cases), 2020, to March 29th, 2022, covering a total of 789 days (or 800) of pandemic.

The data reported in the two datasets have two different formats:

1. Accumulated Cases (*Acumulado.xlsx*)

    - COVID-19 Cases

      These are the data collected in the "*Casos*" sheet, containing the accumulated number of positive COVID-19 cases in the Canary Islands.

      The previously created workspace about this topic is loaded on MATLAB with the following command:

      ```
      load('accCasesWorkspace.mat')
      ```

      The number of entries of the dataset is 760, which is not equal to the number of days of the time period taken into consideration, meaning there are some missing data on this sheet, 29 days' data to be precise. In a data quality point of view, before working with the data it is necessary to deal with this lack. First of all, the available table needs to be transformed into a timetable:

      ```
      accumulatedCases = table2timetable(accumulatedCases)
      ```

      Subsequently, using the MATLAB "retime" function, the missing days are inserted using the value of the previous entry:

      ```
      accumulatedCases = retime(accumulatedCases,'daily', 'previous')
      ```

      A time series chart is created (Figure 1), representing the data in a filled area plot with the following lines of code:

      ```
      area(accumulatedCases.Date, accumulatedCases.NumCases)
      ylabel({'Daily Accumulated COVID-19 Cases'})
      xlabel({'Date'})
      ```

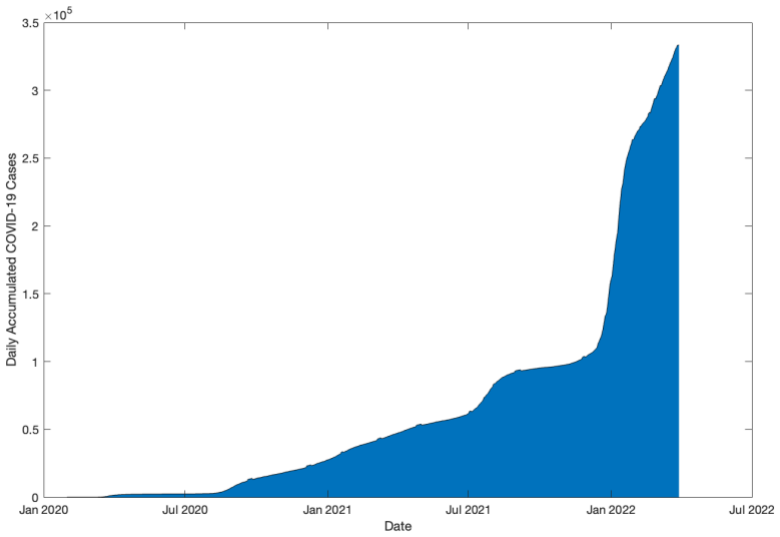

Figure 1

The result is a growth curve that, when more inclined, indicates a more rapid increase of positive COVID-19 cases.

- COVID-19 Deaths

These are the data collected in the "*Fallecidos*" sheet, containing the accumulated number of deaths caused by COVID-19 in the Canary Islands.

The previously created workspace about this topic is loaded on MATLAB with the following command:

```
load('accDeathsWorkspace.mat')
```

The number of entries of the dataset is 760, which is not equal to the number of days of the time period taken into consideration, meaning there are some missing data on this sheet, 29 days' data to be precise. In a data quality point of view, before working with the data it is necessary to deal with this lack. First of all, the available table needs to be transformed into a timetable:

```
accumulatedDeaths = table2timetable(accumulatedDeaths)
```

Subsequently, using the MATLAB "retime" function, the missing days are inserted using the value of the previous entry:

```
accumulatedDeaths = retime(accumulatedDeaths, 'daily', 'previous')
```

A time series chart is created (Figure 2), representing the data in a filled area plot with the following lines of code:

```
area(accumulatedDeaths.Date, accumulatedDeaths.NumCases)
ylabel({'Daily Accumulated COVID-19 Deaths'})
xlabel({'Date'})
```

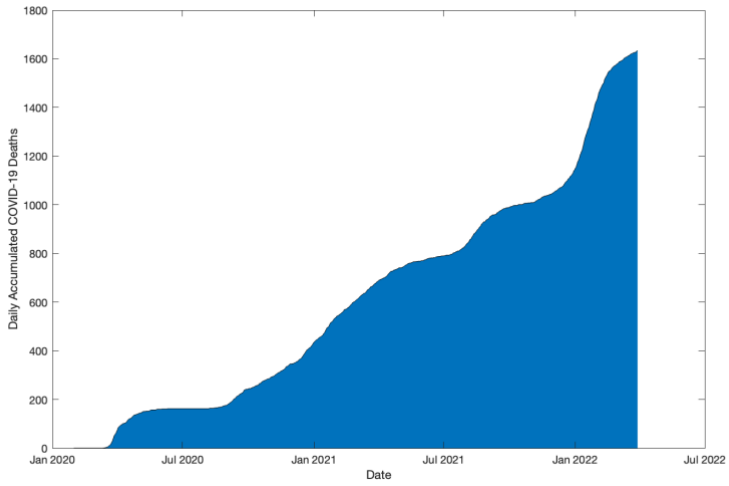

Figure 2

The result is a growth curve that, when more inclined, indicates a more rapid increase of COVID-19 death cases.

- <u>COVID-19 Discharged Patients</u>

These are the data collected in the "*Cerrados por alta médica*" sheet, containing the accumulated number of discharged COVID-19 patients in the Canary Islands.

The previously created workspace about this topic is loaded on MATLAB with the following command:

```
load('accDisWorkspace.mat')
```

The number of entries of the dataset is 760, which is not equal to the number of days of the time period taken into consideration, meaning there are some missing data on this sheet, 29 days' data to be precise. In a data quality point of view, before working with the data it is necessary to deal with this lack. First of all, the available table needs to be transformed into a timetable:

```
accumulatedDis = table2timetable(accumulatedDis)
```

Subsequently, using the MATLAB "retime" function, the missing days are inserted using the value of the previous entry:

```
accumulatedDis = retime(accumulatedDis, 'daily', 'previous')
```

A time series chart is created (Figure 3), representing the data in a filled area plot with the following lines of code:

```
area(accumulatedDis.Date, accumulatedDis.NumCases)
ylabel({'Daily Accumulated COVID-19 Discharged'})
xlabel({'Date'})
```

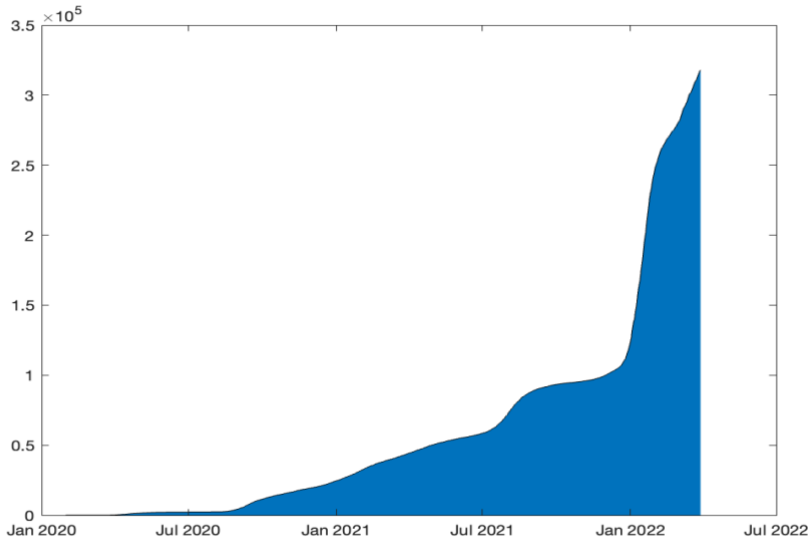

Figure 3

The result is a growth curve that, when more inclined, indicates a more rapid increase of COVID-19 patients' discharges.

Plotting the three variables with the accumulated data together (cases, deaths, and discharges) on the same chart (Figure 4), it is visible that cases and discharges grow with more or less the same pace, with a detachment of about two weeks, which is indeed the approximate duration of a COVID-19 infection.

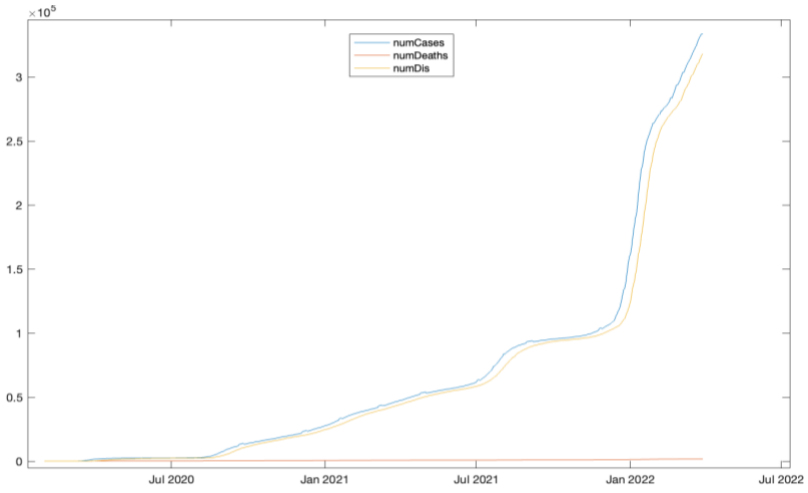

Figure 4

2. Daily Cases (*Diario.xlsx*)

- <u>COVID-19 Cases</u>

  These are the data collected in the "*Casos*" sheet, containing the number of daily COVID-19 new cases in the Canary Islands.

  The previously created workspace about this topic is loaded on MATLAB with the following command:

  ```
  load('casesWorkspace.mat')
  ```

  The number of entries of the dataset is 761, which is not equal to the number of days of the time period taken into consideration, meaning there are some missing data on this sheet, 28 days' data to be precise. In a data quality point of view, before working with the data it is necessary to deal with this lack. The first row of the table records several positive cases equal to zero (Figure 5), and since there are 11 missing entries afterwards, the best option would be deleting it (Figure 6):

	Date	1 NumCases
1	20/Jan/2020	0

  Figure 5

  ```
  cases(1,:) = [];
  ```

  Figure 6

  The available table now needs to be transformed into a timetable:

  ```
  cases = table2timetable(cases)
  ```

  Subsequently, using the MATLAB "retime" function, the missing days are inserted using the value of the previous entry:

  ```
  cases = retime(cases, 'daily', 'previous')
  ```

  A time series chart is created (Figure 7), representing the data in a filled area plot with the following lines of code:

  ```
  area(cases.Date, cases.NumCases)
  ylabel({'Daily COVID-19 Cases'})
  xlabel({'Date'})
  ```

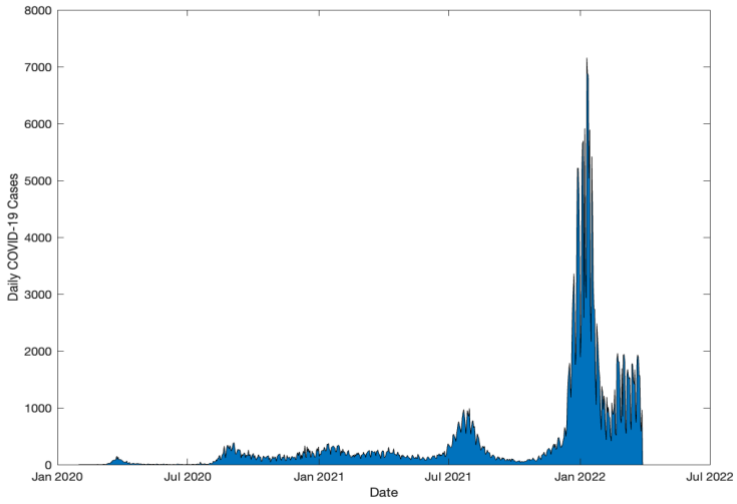

Figure 7

Looking at the obtained graph, it is easy to conclude that there is no seasonality, and no noise can be spotted. However, daily new COVID-19 cases are not constant, and peaks of higher quantities of positivities can be detected looking at the time series curve. Highlighting the peaks with the MATLAB's brushing tool, the following result is obtained:

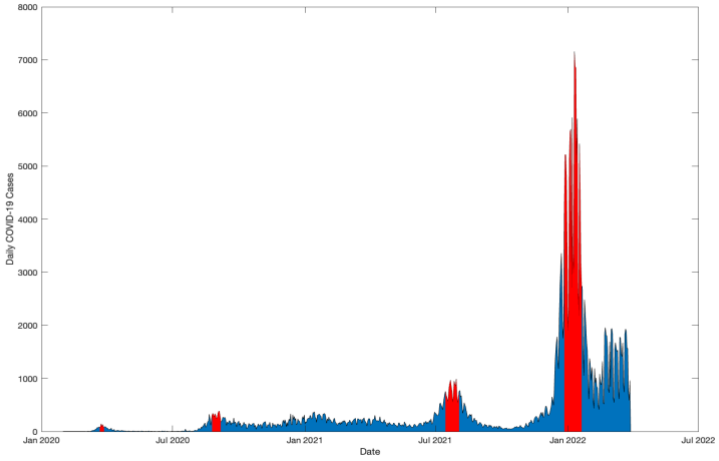

Figure 8

After the use of the brush, the time periods during which there were more positive COVID-19 cases in comparison to the usual result to be the following:

	From	To
Peak 1	22 March 2020	28 March 2020
Peak 2	25 August 2020	6 September 2020
Peak 3	15 July 2021	3 August 2021
Peak 4	27 December 2021	20 January 2022

Table 1

The dataset is further labeled saving in the workspace the time periods during which the cases are growing and those during which they are decreasing using the following lines of code:

```
for i=1:size
        if cases.NumCases(i)<cases.NumCases(i+1)
       growing = [growing; cases.Date(i+1)];
   elseif cases.NumCases(i)>cases.NumCases(i+1)
       decreasing = [decreasing; cases.Date(i+1)];
end
end
```

- COVID-19 Deaths

These are the data collected in the "*Fallecidos*" sheet, containing the number of daily COVID-19 deaths in the Canary Islands.

The previously created workspace about this topic is loaded on MATLAB with the following command:

```
load('deathsWorkspace.mat')
```

The number of entries of the dataset is 760, which is not equal to the number of days of the time period taken into consideration, meaning there are some missing data on this sheet, 29 days' data to be precise. In a data quality point of view, before working with the data it is necessary to deal with this lack. First of all, the available table needs to be transformed into a timetable:

```
deaths = table2timetable(deaths)
```

Subsequently, using the MATLAB "retime" function, the missing days are inserted using the value of the previous entry:

```
deaths = retime(deaths, 'daily', 'previous')
```

A time series chart is created (Figure 9), representing the data in a filled area plot with the following lines of code:

```
area(deaths.Date, deaths.NumCases)
ylabel({'Daily COVID-19 Deaths'})
xlabel({'Date'})
```

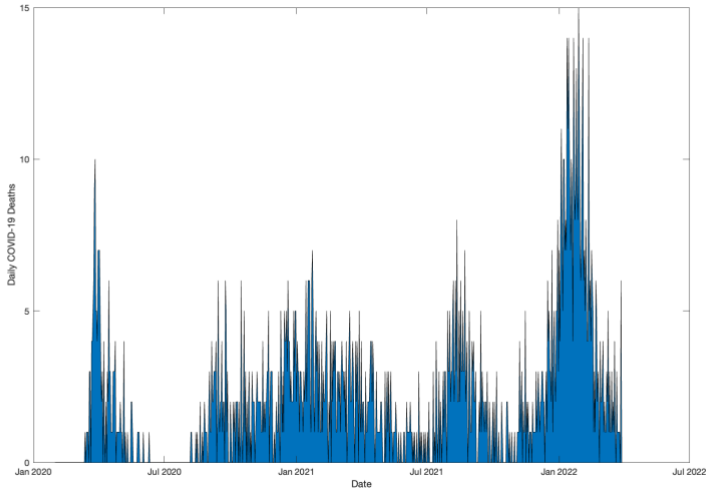

Figure 9

Looking at the obtained graph, it is easy to conclude that there is no seasonality, and no noise can be spotted. However, daily new COVID-19 deaths are not constant, and peaks of higher quantities of deaths can be detected looking at the time series curve. Highlighting the peaks with the MATLAB's brushing tool (Figure 10), the following result is obtained:

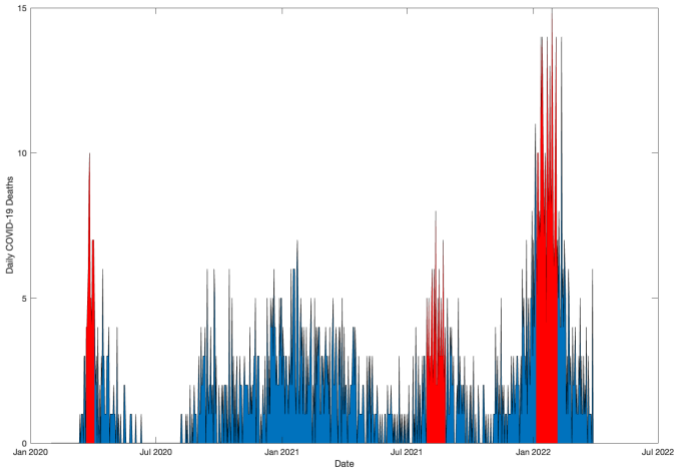

Figure 10

After the use of the brush, the time periods during which there were receiving more COVID-19 deaths in comparison to the usual result to be the following:

	From	To
Peak 1	22 March 2020	3 April 2020
Peak 2	30 July 2021	28 August 2021
Peak 3	5 January 2022	5 February 2022

Table 2

The dataset is further labeled saving in the workspace the time periods during which the cases are growing and those during which they are decreasing using the following lines of code:

```
>> for i=1:size
if deaths.NumCases(i)<deaths.NumCases(i+1)
growing = [growing; deaths.Date(i+1)];
elseif deaths.NumCases(i)>deaths.NumCases(i+1)
decreasing = [decreasing; deaths.Date(i+1)];
end
end
```

- <u>COVID-19 Discharged Patients</u>

These are the data collected in the "*Fallecidos*" sheet, containing the number of daily COVID-19 discharged patients in the Canary Islands.

The previously created workspace about this topic is loaded on MATLAB with the following command:

```
load('dischargedWorkspace.mat')
```

In this case the Date column needs to be fixed before being able to start working with the data. In fact, some of the month names are abbreviated with a dot, which doesn't allow to transform them to datetime format correctly. Indeed, the dot must be removed with the following command:

```
discharged.Date = replace(discharged.Date(:,:), '.', '')
```

The number of entries of the dataset is 761, which is not equal to the number of days of the time period taken into consideration, meaning there are some missing data on this sheet, 28 days' data to be precise. In a data quality point of view, before working with the data it is necessary to deal with this lack. The first row of the table records several positive cases equal to zero (Figure 11), and since there are 11 missing entries afterwards, the best option would be deleting it (Figure 12):

	Date	1 NumCases
1	20/Jan/2020	1

Figure 11

```
discharged(1,:) = [];
```

Figure 12

The available table now needs to be transformed in a timetable:

```
discharged = table2timetable(discharged)
```

Subsequently, using the MATLAB “retime” function, the missing days are inserted using the value of the previous entry:

```
discharged = retime(discharged, 'daily', 'previous')
```

A time series chart is created (Figure 13), representing the data in a filled area plot with the following lines of code:

```
area(discharged.Date, discharged.NumCases)
ylabel({'Daily Discharged COVID-19 Patients'})
xlabel({'Date'})
```

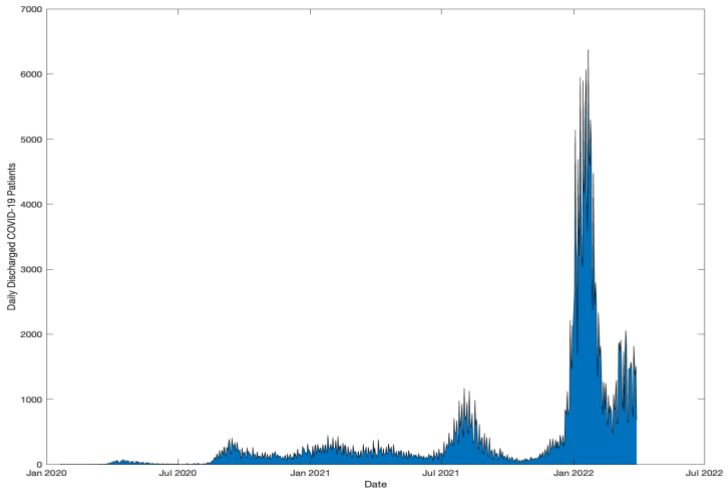

Figure 13

Looking at the obtained graph, it is easy to conclude that there is no seasonality, and no noise can be spotted. However, daily COVID-19 discharged patients are not constant, and peaks of higher quantities of deaths can be detected looking at the time series curve. Highlighting the peaks with the MATLAB’s brushing tool (Figure 14), the following result is obtained:

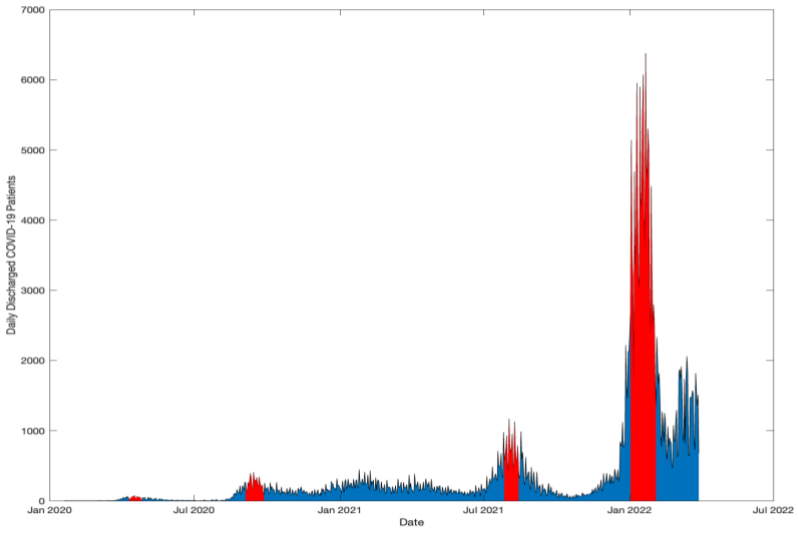

Figure 14

After the use of the brush, the time periods during which there were more discharges from COVID-19 in comparison to the usual result to be the following:

	From	To
Peak 1	10 April 2020	27 April 2020
Peak 2	4 September 2020	27 September 2020
Peak 3	27 July 2021	14 August 2021
Peak 4	2 January 2022	3 February 2022

Table 3

The dataset is further labeled saving in the workspace the time periods during which the cases are growing and those during which they are decreasing using the following lines of code:

```
>> for i=1:size
if discharged.NumCases(i)<discharged.NumCases(i+1)
growing = [growing; discharged.Date(i+1)];
elseif discharged.NumCases(i)>discharged.NumCases(i+1)
decreasing = [decreasing; discharged.Date(i+1)];
end
end
```

Plotting the three variables with the daily data together (cases, deaths, and discharges) on the same chart (Figure 15), it is visible that cases and discharges grow with more or less the same pace, with a detachment of about two weeks, which is indeed the approximate duration of a COVID-19 infection.

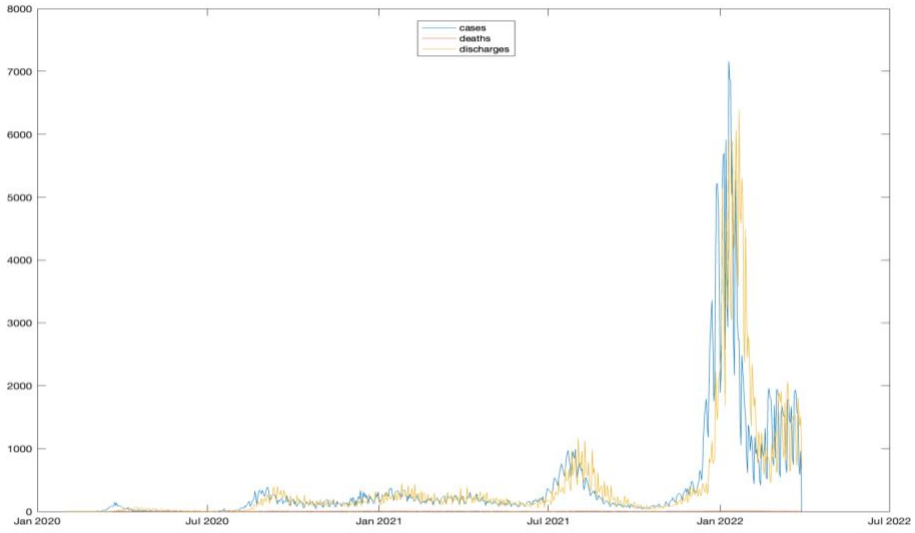

Figure 15

Care Capacity Data

In this section, the collected data on hospitals' capacity, such as incoming patients (both COVID-19 and not) or occupied beds, will be studied.

The three available datasets, *Ingresos.xlsx*, *Pacientes Covid 19.xlsx* and *Pacientes no Covid 19.xlsx,* range from March 23rd, 2020, to October 6th, 2022, covering a total of 928 days of pandemic. The number of entries of each dataset is also 928, meaning there is no missing data to deal with.

Ingresos.xlsx:

- Incoming COVID-19 Patients

  These are the data collected in the "*Ingresos Covid por urgencias*" sheet, containing the number of urgent daily incoming COVID-19 patients to the Cananary Islands' hospitals.

  The previously created workspace about this topic is loaded on MATLAB with the following command:

```
load('inCOVworkspace.mat')
```

  A time series chart (Figure 16) is created, representing the data in a filled area plot with the following lines of code:

```
area(inCOVpatients.Date, inCOVpatients.NumCases)
ylabel({'Urgent COVID-19 Incoming Patients'})
xlabel({'Date'})
```

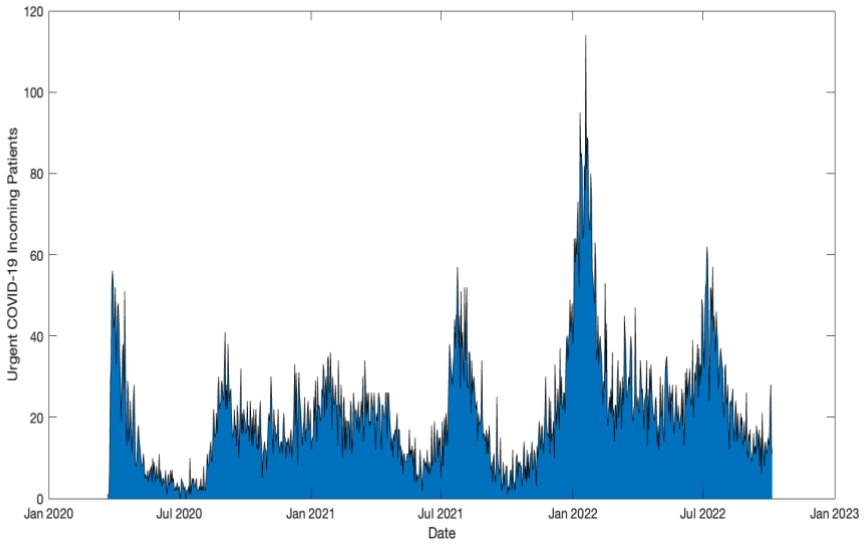

Figure 16

Looking at the obtained graph, it is easy to conclude that there is no seasonality, and no noise can be spotted. However, incoming hospital patients are not constant, and peaks of higher affluency can be detected looking at the time series curve. Highlighting the peaks with the MATLAB's brushing tool (figure 17), the following result is obtained:

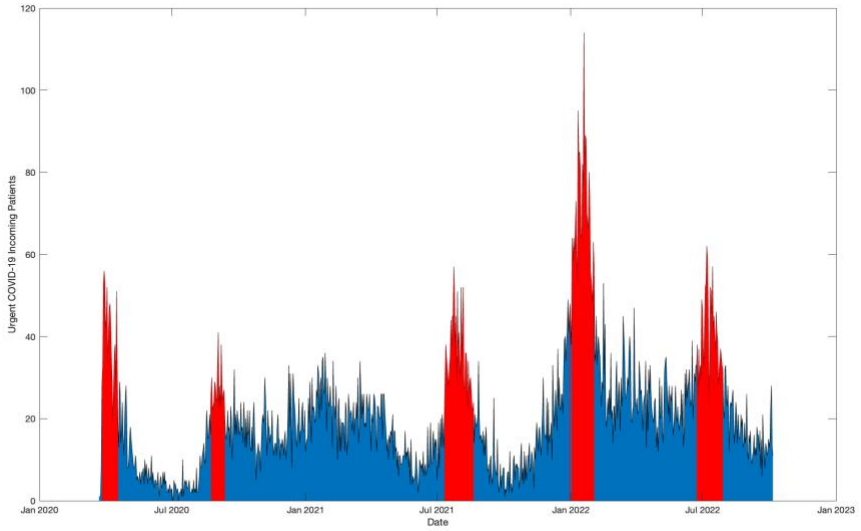

Figure 17

After the use of the brush, the time periods during which the hospitals were receiving high quantities of patients in comparison to the usual result to be the following:

	From	To
Peak 1	27 March 2020	18 April 2020
Peak 2	22 August 2020	15 September 2020
Peak 3	11 July 2021	20 August 2021
Peak 4	28 December 2021	5 February 2022
Peak 5	23 June 2022	29 July 2022

Table 4

The dataset is further labeled saving in the workspace the time periods during which the cases are growing and those during which they are decreasing using the following lines of code:

```
>> for i=1:size
if inCOVpatients.NumCases(i)<inCOVpatients.NumCases(i+1)
growing = [growing; inCOVpatients.Date(i+1)];
elseif inCOVpatients.NumCases(i)>inCOVpatients.NumCases(i+1)
decreasing = [decreasing; inCOVpatients.Date(i+1)];
end
end
```

- Total Hospital Incoming Patients

These are the data collected in the "*Ingresos totales hospitalarios*" sheet, where data about the total number of daily incoming patients to Canary Islands' hospitals is stored.

The previously created workspace about this topic is loaded on MATLAB with the following command:

```
load('inTOTworkspace.mat')
```

A time series chart (Figure 18) is created, representing the data in a filled area plot with the following lines of code:

```
area(inTOTpatients.Date, inTOTpatients.NumCases)
ylabel({'Total Incoming Patients'})
xlabel({'Date'})
```

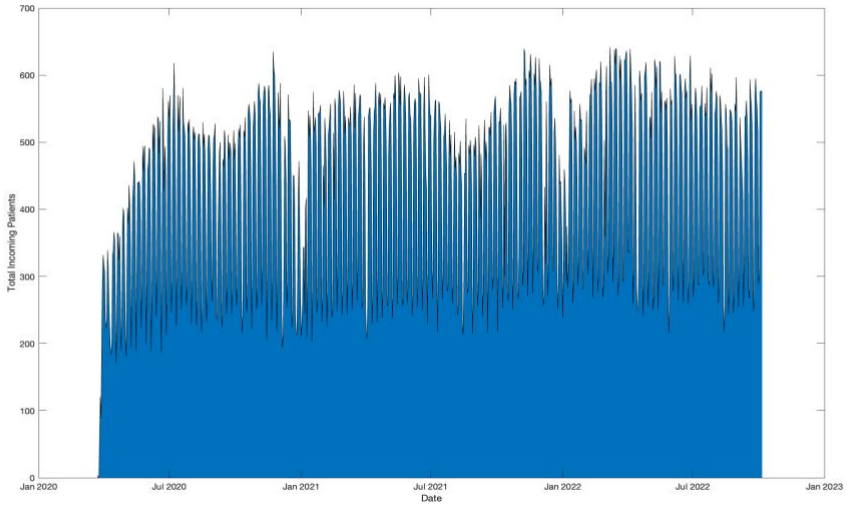

Figure 18

Outliers can be detected during the first days of the data acquisition. In fact, looking at the first entries of the '*inTOTpatients*' table, the total hospitalized cases result to be either zero or a very low number (such as 4) compared to all other results (Figure 19). Moreover, looking at the '*inCOVpatients*' table, which holds the data about the incoming COVID-19 patients, it is easy to notice an inconsistency in the data. The COVID-19 patients of March 24th and 25th result to be equal to 1, which means that the incoming total patients on the same days should at least be 1 (Figure 20). Dealing with the inconsistency from a data quality point of view, the best option would be deleting the first three entries of the '*inTOTpatients*' table.

	1 Date	2 NumCases
1	23/Mar/2020	4
2	24/Mar/2020	0
3	25/Mar/2020	0

Figure 19

	1 Date	2 NumCases
1	23/Mar/2020	1
2	24/Mar/2020	1
3	25/Mar/2020	1

Figure 20

In order to delete the three inconsistent rows, the following command is executed:

```
inTOTpatients(1:3,:) = []
```

Looking at the filled area chart, no visible peaks are easy to be observed. In addition, a seasonality on the incoming patients is deductible: every 6 or 7 days, there is a peak on the 3rd day and there are lows on the 1st and last days. This seasonality that recurs weekly can be attributed to the fact that some hospitals might be closed on the weekend or might not register the patients until the next Monday.

The dataset is further labeled saving in the workspace the time periods during which the cases are growing and those during which they are decreasing using the following lines of code:

```
>> for i=1:size
if inTOTpatients.NumCases(i)<inTOTpatients.NumCases(i+1)
growing = [growing; inTOTpatients.Date(i+1)];
elseif inTOTpatients.NumCases(i)>inTOTpatients.NumCases(i+1)
decreasing = [decreasing; inTOTpatients.Date(i+1)];
end
end
```

Pacientes Covid 19.xlsx:

- Critical COVID-19 Patients with Respirator

These are the data collected in the "*Críticas con respirador*" sheet, containing data about the number of critical COVID-19 patients that use a respirator.

The previously created workspace about this topic is loaded on MATLAB with the following command:

```
load('COVwithResWorkspace.mat')
```

In this case the Date column needs to be fixed before being able to start working with the data. In fact, some of the month names are abbreviated with a dot, which doesn't allow to transform them to datetime format correctly. Indeed, the dot must be removed with the following command:

```
COVwithRes.Date = replace(COVwithRes.Date(:,:), '.', '')
```

A time series chart is created (Figure 21), representing the data in a filled area plot with the following lines of code:

```
area(COVwithRes.Date, COVwithRes.NumCases)
ylabel({'Daily COVID-19 Patients Using a Respirator'})
xlabel({'Date'})
```

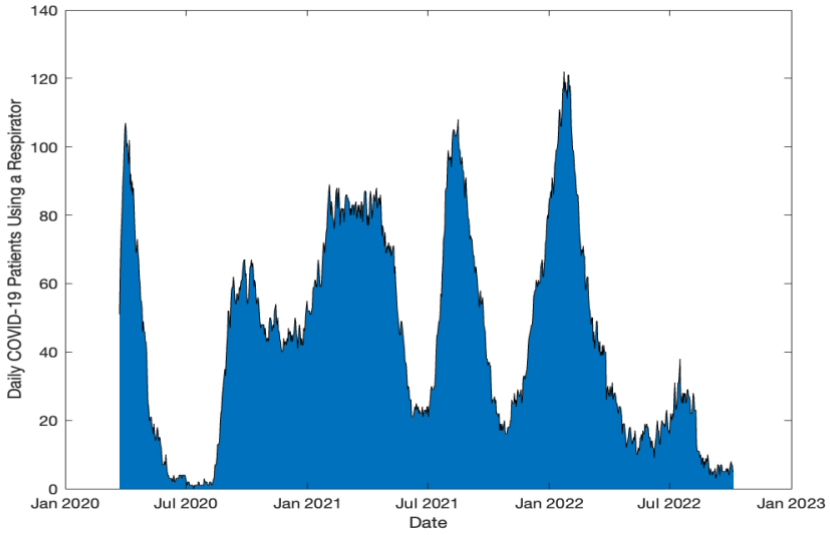

Figure 21

Looking at the obtained graph, it is easy to conclude that there is no seasonality, and no noise can be spotted. However, critical COVID-19 patients in the need of a respirator are not constant, and peaks of higher quantity can be detected looking at the time series curve. Highlighting these peaks with the MATLAB's brushing tool (Figure 22), the following result is obtained:

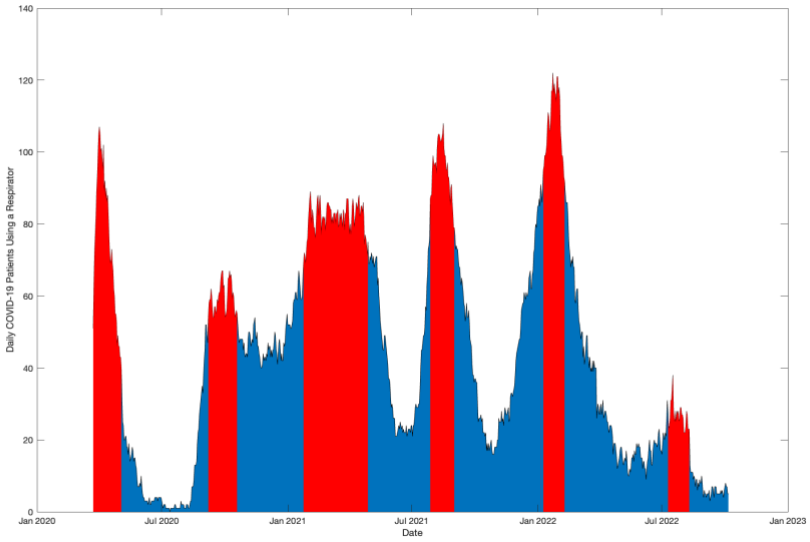

Figure 22

After the use of the brush, the time periods during which the hospitals were receiving high quantities of COVID-19 patients in comparison to the usual result to be the following:

	From	**To**
Peak 1	23 March 2020	3 May 2020
Peak 2	7 September 2020	19 October 2020

Peak 3	24 January 2021	28 April 2021
Peak 4	28 July 2021	1 September 2021
Peak 5	5 January 2022	9 February 2022
Peak 6	10 July 2022	10 August 2022

Table 5

The dataset is further labeled saving in the workspace the time periods during which the cases are growing and those during which they are decreasing using the following lines of code:

```
>> for i=1:size
if COVnoRes.NumCases(i)<COVnoRes.NumCases(i+1)
growing = [growing; COVnoRes.Date(i+1)];
elseif COVnoRes.NumCases(i)>COVnoRes.NumCases(i+1)
decreasing = [decreasing; COVnoRes.Date(i+1)];
end
end
```

- Critical COVID-19 Patients without Respirator

These are the data collected in the "*Críticas sin respirador*" sheet, containing data about the number of critical COVID-19 patients that don't have a respirator.

The previously created workspace about this topic is loaded on MATLAB with the following command:

```
load('COVnoResWorkspace.mat')
```

In this case the Date column needs to be fixed before being able to start working with the data. In fact, some of the month names are abbreviated with a dot, which doesn't allow to transform them to datetime format correctly. Indeed, the dot must be removed with the following command:

```
COVnoRes.Date = replace(COVnoRes.Date(:,:), '.', '')
```

A time series chart is created (Figure 23), representing the data in a filled area plot with the following lines of code:

```
area(COVnoRes.Date, COVnoRes.NumCases)
ylabel({'Daily COVID-19 Patients not Using a Respirator'})
xlabel({'Date'})
```

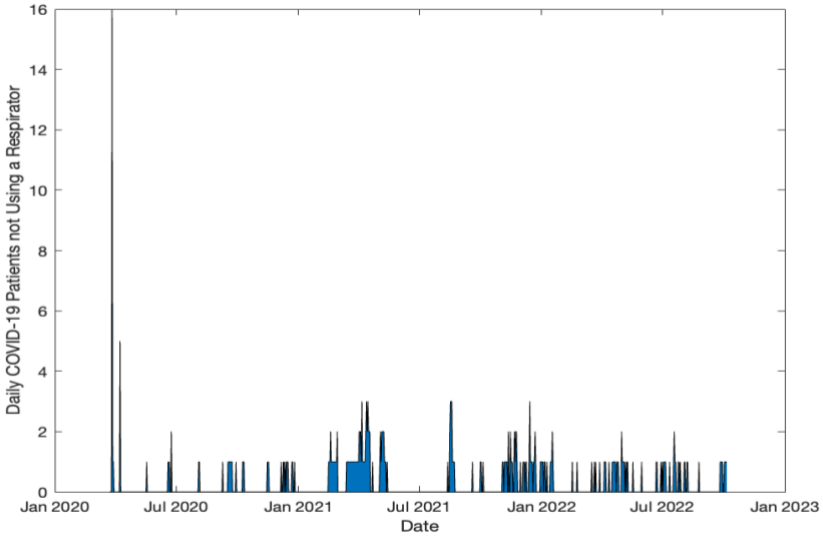

Figure 23

Looking at the obtained graph, it is easy to conclude that there is no seasonality. A noise can be spotted on March 27th, 2020, being the number of COVID-19 patients without respirator equal to 16, a number that is much higher compared to all the other days, where the number of these patients is 0 or not higher than 5.

The dataset is further labeled saving in the workspace the time periods during which the cases are growing and those during which they are decreasing using the following lines of code:

```
>> for i=1:size
if COVnoRes.NumCases(i)<COVnoRes.NumCases(i+1)
growing = [growing; COVnoRes.Date(i+1)];
elseif COVnoRes.NumCases(i)>COVnoRes.NumCases(i+1)
decreasing = [decreasing; COVnoRes.Date(i+1)];
end
end
```

- Rest of Daily COVID-19 Hospital Beds

These are the data collected in the "*Resto de camas hospitalarias*" sheet, indicating the beds occupied by COVID-19 patients that are not in a critical stadium of the disease.

The previously created workspace about this topic is loaded on MATLAB:

```
load('allCOVbedsWorkspace.mat')
```

In this case the Date column needs to be fixed before being able to start working with the data. In fact, some of the month names are abbreviated with a dot, which doesn't allow to transform them to datetime format correctly. Indeed, the dot must be removed with the following command:

```
allCOVbeds.Date = replace(allCOVbeds.Date(:,:), '.', '')
```

A time series chart is created (Figure 24), representing the data in a filled area plot with the following lines of code:

```
area(allCOVbeds.Date, allCOVbeds.NumCases)
ylabel({'Daily COVID-19 Occupied Beds'})
xlabel({'Date'})
```

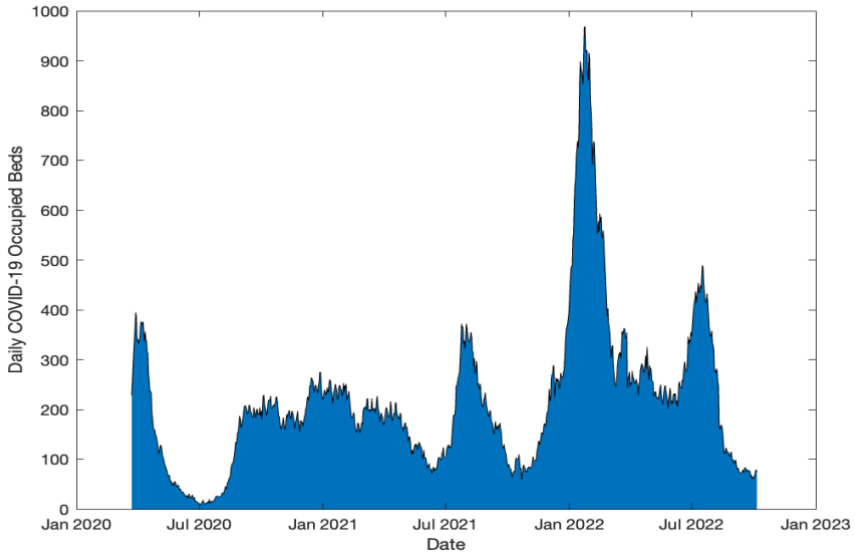

Figure 24

Looking at the obtained graph, it is easy to conclude that there is no seasonality, and no noise can be spotted. However, COVID-19 patients occupying a hospital bed are not constant, and peaks of higher quantity can be detected looking at the time series curve. Highlighting these peaks with the MATLAB's brushing tool (Figure 25), the following result is obtained:

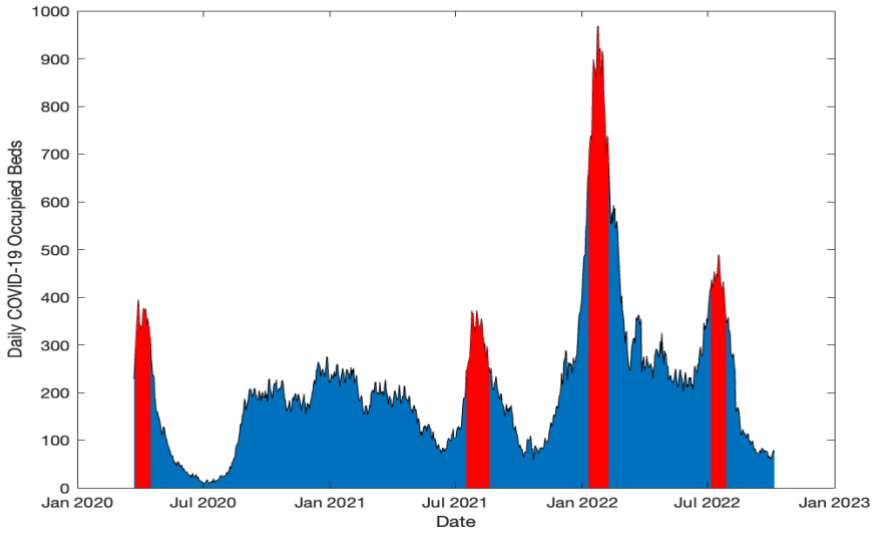

Figure 25

After the use of the brush, the time periods during which the hospitals had higher quantities of COVID-19 patients occupying beds in comparison to the usual result to be the following:

	From	To
Peak 1	25 March 2020	16 April 2020
Peak 2	17 July 2021	22 August 2021
Peak 3	10 January 2022	8 February 2022
Peak 4	6 July 2022	31 July 2022

Table 3

The dataset is further labeled saving in the workspace the time periods during which the cases are growing and those during which they are decreasing using the following lines of code:

```
>> for i=1:size
if allCOVbeds.NumCases(i)<allCOVbeds.NumCases(i+1)
growing = [growing; allCOVbeds.Date(i+1)];
elseif allCOVbeds.NumCases(i)>allCOVbeds.NumCases(i+1)
decreasing = [decreasing; allCOVbeds.Date(i+1)];
end
end
```

Pacientes no Covid 19.xlsx:

- Critical Patients with Respirator

These are the data collected in the "*Críticas con respirador*" sheet, containing data about the number of critical hospital patients (excluding COVID-19 patients) that use a respirator.

The previously created workspace about this topic is loaded on MATLAB with the following command:

```
load('withResWorkspace.mat')
```

In this case the Date column needs to be fixed before being able to start working with the data. In fact, some of the month names are abbreviated with a dot, which doesn't allow to transform them to datetime format correctly. Indeed, the dot must be removed with the following command:

```
resPatients.Date = replace(resPatients.Date(:,:), '.', '')
```

A time series chart is created (Figure 26), representing the data in a filled area plot with the following lines of code:

```
area(resPatients.Date, resPatients.NumCases)
ylabel({'Critical Patients with Respirator'})
xlabel({'Date'})
```

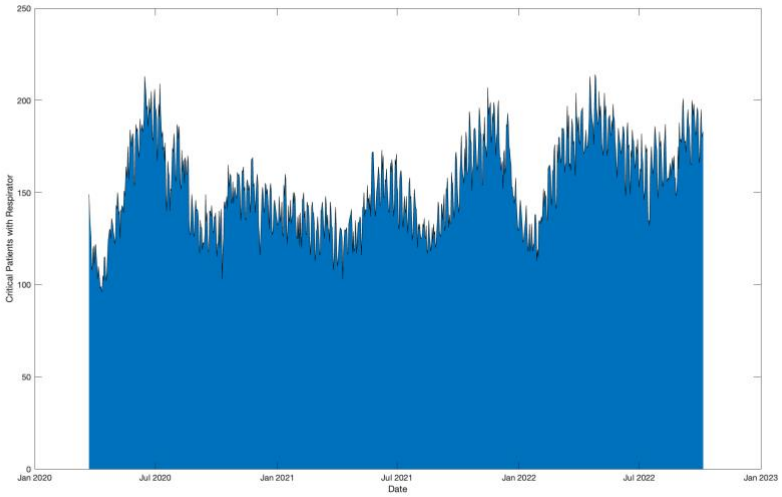

Figure 26

Looking at the obtained graph, it is easy to conclude that there is no seasonality, and no noise can be spotted. However, non-COVID-19 patients using a respirator are not constant, and peaks of higher quantity can be detected looking at the time series curve. Highlighting these peaks with the MATLAB's brushing tool (Figure 27), the following result is obtained:

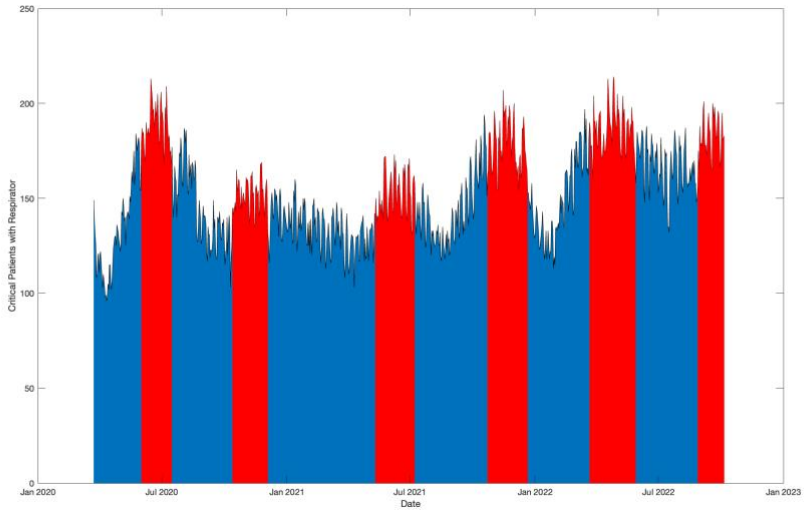

Figure 27

After the use of the brush, the time periods during which the hospitals had higher quantities of COVID-19 patients occupying beds in comparison to the usual results result to be the following:

	From	To
Peak 1	1 June 2020	16 July 2020
Peak 2	13 October 2020	4 December 2020
Peak 3	11 May 2021	8 July 2021
Peak 4	23 October 2021	21 December 2021
Peak 5	22 March 2022	29 May 2022
Peak 6	28 August 2022	6 October 2022

Table 6

The dataset is further labeled saving in the workspace the time periods during which the cases are growing and those during which they are decreasing using the following lines of code:

```
>> for i=1:size
if resPatients.NumCases(i)<resPatients.NumCases(i+1)
growing = [growing; resPatients.Date(i+1)];
elseif resPatients.NumCases(i)>resPatients.NumCases(i+1)
decreasing = [decreasing; resPatients.Date(i+1)];
end
end
```

- Critical Patients without Respirator

These are the data collected in the "*Críticas sin respirador*" sheet, containing data about the number of critical hospital patients (excluding COVID-19 patients) that don't have a respirator.

The previously created workspace about this topic is loaded on MATLAB with the following command:

```
load('noResWorkspace.mat')
```

In this case the Date column needs to be fixed before being able to start working with the data. In fact, some of the month names are abbreviated with a dot, which doesn't allow to transform them to datetime format correctly. Indeed, the dot must be removed with the following command:

```
noResPatients.Date = replace(noResPatients.Date(:,:), '.', '')
```

A time series chart is created (Figure 28), representing the data in a filled area plot with the following lines of code:

```
area(noResPatients.Date, noResPatients.NumCases)
ylabel({'Critical Patients without Respirator'})
xlabel({'Date'})
```

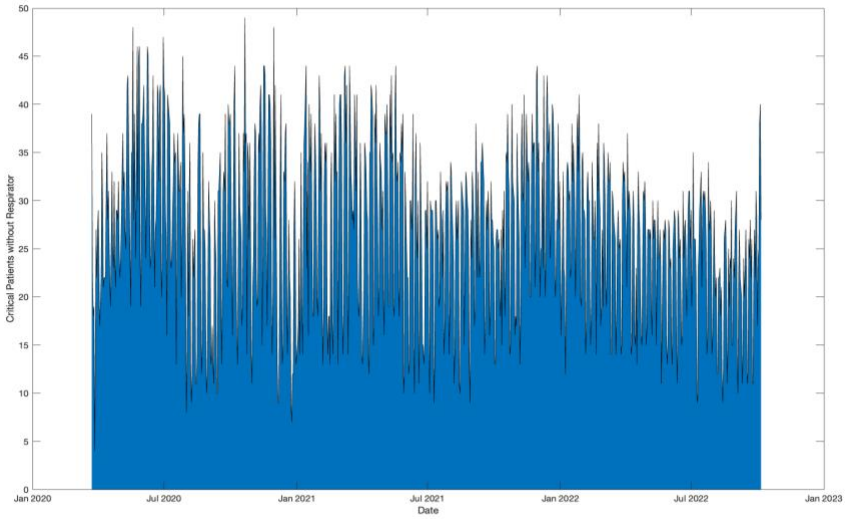

Figure 28

Looking at the filled area chart, no visible peaks are easy to be observed. In addition, a seasonality on the patients with no respirator is deductible: every 6 or 7 days, there is a peak on the 3rd day and there are lows on the 1st and last days. This seasonality that recurs weekly can be attributed to the fact that some hospitals might be closed on the weekend or might not register the patients until the next Monday.

The dataset is further labeled saving in the workspace the time periods during which the cases are growing and those during which they are decreasing using the following lines of code:

```
>> for i=1:size
if noResPatients.NumCases(i)<noResPatients.NumCases(i+1)
growing = [growing; noResPatients.Date(i+1)];
elseif noResPatients.NumCases(i)>noResPatients.NumCases(i+1)
decreasing = [decreasing; noResPatients.Date(i+1)];
end
end
```

- Rest of Daily Hospital Beds

These are the data collected in the "*Resto de camas hospitalarias*" sheet, reporting data about the number of hospital beds occupied by non-COVID-19 patients.

The previously created workspace about this topic is loaded on MATLAB with the following command:

```
load('allBedsWorkspace.mat')
```

In this case the Date column needs to be fixed before being able to start working with the data. In fact, some of the month names are abbreviated with a dot, which doesn't allow to transform them to datetime format correctly. Indeed, the dot must be removed with the following command:

```
allBeds.Date = replace(allBeds.Date(:,:), '.', '')
```

The number of occupied beds is in string format and needs to be convert it in double. In order to do so, the dot used to represent some of the 4-digits-numbers needs to first be removed:

```
allBeds.NumCases = replace(allBeds.NumCases(:,:), '.', '')
allBeds.NumCases = str2double(allBeds.NumCases)
```

A time series chart is created (Figure 29), representing the data in a filled area plot with the following lines of code:

```
area(allBeds.Date, allBeds.NumCases)
ylabel({'Daily Occupied Hospital Beds'})
xlabel({'Date'})
```

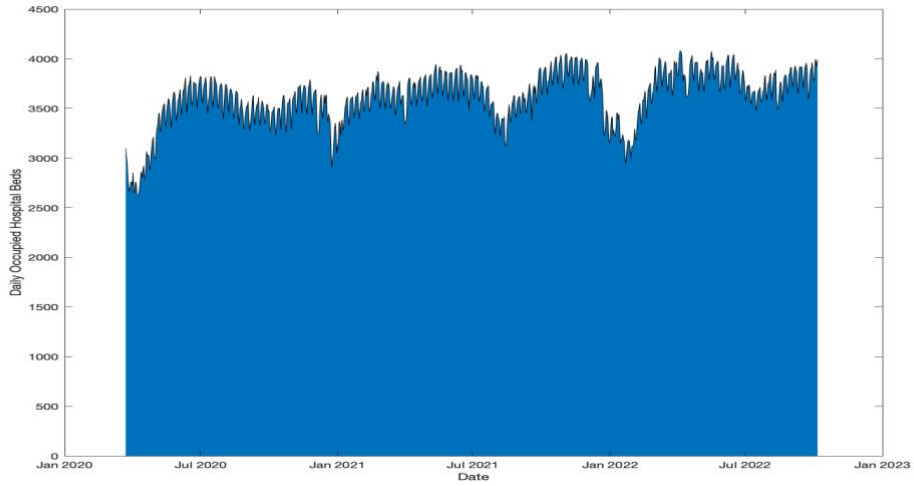

Figure 29

The dataset is further labeled saving in the workspace the time periods during which the cases are growing and those during which they are decreasing using the following lines of code:

```
>> for i=1:size
if allBeds.NumCases(i)<allBeds.NumCases(i+1)
growing = [growing; allBeds.Date(i+1)];
elseif allBeds.NumCases(i)>allBeds.NumCases(i+1)
decreasing = [decreasing; allBeds.Date(i+1)];
end
end
```

Conclusions

After this study, the data on COVID-19 cases are cleaned from inconsistencies, there are no missing values, and labels have been added in order to have a tidier and more understandable dataset. The cleaning of the data, carried out following the six data qualities dimensions, has been completed using the MATLAB matrix-oriented language and has additionally been visually represented on time series charts. The result is a ready-to-use dataset, that provides data inside proper MATLAB workspaces that will surely be useful for time series forecasting in the next steps of this project.

Bibliography

[1] LLC, Cogito Tech. “Time Series Data Labeling: A Complete Know-How for Efficient AI Implementation.” *Training Data for AI, ML With Human Empowered Automation | Cogito*. Accessed 3 Nov. 2022. www.cogitotech.com/blog/time-series-data-labeling-a-complete-know-how-for-efficient-ai-implementation

[2] “What Is Data Quality? Definition and FAQs | HEAVY.AI.” *What Is Data Quality? Definition and FAQs | HEAVY.AI*, www.heavy.ai/technical-glossary/data-quality. Accessed 3 Nov. 2022. https://www.heavy.ai/technical-glossary/data-quality

[3] “Data Quality – Lean-Data.” *Data Quality – Lean-Data*, www.lean-data.nl/data-quality. Accessed 3 Nov. 2022. https://www.lean-data.nl/data-quality/

[4] “What Is MATLAB?” *What Is MATLAB? - MATLAB & Simulink*, it.mathworks.com/discovery/what-is-matlab.html. Accessed 5 Nov. 2022. https://it.mathworks.com/discovery/what-is-matlab.html

[5] “What Is Data Labeling?” *Amazon Web Services, Inc.*, aws.amazon.com/sagemaker/data-labeling/what-is-data-labeling. Accessed 8 Nov. 2022. https://aws.amazon.com/sagemaker/data-labeling/what-is-data-labeling/