Development of an automatic tool for periodic surveillance of actuarial and demographic indicators

Daniel Alonso, María Luz Durbán Reguera, Bernardo D'Auria

July-August 2021

Contents

Abstract

As a result of the COVID-19 pandemic and its large impact across Spain, the monitoring of demographic measures as a direct result of deaths related to such pandemic and future similarly deadly events has become increasingly important. It is intended with this project to develop a tool in order to easily monitor a selection of demographic measures relating to collective deaths of individuals as a result of relevant worldwide events like the one mentioned previously. The tool consists of a shiny dashboard (developed in R) where such measures are displayed in different visualizations across time.

Introduction

The COVID-19 pandemic has led to a widespread and noticeable temporary increase in mortality and reduction in life expectancy throughout Spain. This arises a need to monitor these demographic measures more closely and in real time.

This project consists of a shiny dashboard with several features:

- Visualizing several mortality metrics:
  - Excess mortality
  - Cumulative mortality rate
  - Cumulative relative mortality rate
  - Mortality improvement factor
- Visualizing life expectancy and constructing life tables
- Visualizing a map of Spain with the previous metrics per autonomous community (CCAA)

All metrics are calculated weekly with data stretching back as far back as 2010.

Objectives

- Provide a simple-to-use, web-based, OS-agnostic tool to compute and visualize common mortality and life expectancy metrics in time series plots/maps
- Provide the user the ability to customize the plot parameters significantly
- Provide the user the ability to download the plots and the data (with or without filtering)

- Allow the user to update and push the data to the corresponding github repository hosting the data from within the application
- Have data updated in real-time from the official Spanish sources and Eurostat (also provided by INE)

Motivation

As we saw during the COVID-19 pandemic, the most widely publicized measures shown to the public in order to explain the status of the pandemic and the country as a whole were always related to incidence of the virus, death counts, recovery counts, amount of patients at ICU, hospitalized patients by COVID-19 vs total hospitalized patients, etc. However, **Death counts do not tell the whole story**, a death count merely tells us that an amount of people died, it does not tell us how much that amount of deaths affect the population. Also, **all these measures are static**, therefore, we cannot see the effect they cause on the population all by themselves, which is where measures like the *cumulative relative mortality rate* or *life expectancy* come in, measures that show the impact of deaths and their long lasting effect relative to the population over time.

As an example, 100 people dying over the course of a week within a commune of 10,000 inhabitants represents 1% of the population; that same amount of deaths would represent less than 0.01% of the population within a commune of 1 million inhabitants. Therefore, more robust measures to determine the impact of the deaths caused by a pandemic are needed to really gauge the effect of the pandemic as a whole.

The application allows monitoring some of these measures in real time. Whenever it is desired to fetch new data (if available), the data can be fetched with a click. The measures can be instantly computed with a click and visualized to observe the evolution of the population at any time and by region of Spain. Showing particular usefulness within events that cause large amounts of deaths or that reduce the general population life expectancy (as the application can also visualize evolution of life expectancy).

Inspiration

As somewhat of an inspiration, reports like the COVID-19 vaccine surveillance report: Week 27, published by the PHE (Public Health England), show visualizations and analyses where the evolution of mortality metrics is a section of the report. Along with this, the INE currently does not have a report or application calculating these metrics on a per-week basis.

Advantages an open source application over weekly reports in PDF format

There are several advantages an application could potentially have over static, weekly PDF reports like the one previously linked:

- The user has control over the visualizations they desire to see, as there are controls to manipulate the parameters of the visualizations and interact with them.
- The user can download the customized visualizations in their desired resolution or format.
- The user can choose what measures of those available to show.
- As the application is fully open source, developers wanting to expand the application and add extra features and contribute directly to the project's development.
- The measures will be as up-to-date as the data source is. As these reports take time to construct and analyze, they will take longer to be released, so the user can simply open the application and update the database whenever they desire to do so.

Why is monitoring mortality/life expectancy metrics important?

As it is with any demographic measure, knowing general metrics about population is important to many companies, institutions and individuals, to list a few:

- The government and the ministry of health, the main decisionmakers in terms of public health related issues. These two institutions can use these metrics to construct policies or regulations to protect the general health of the population, prioritize or purchase particular medication or medical utensils if needed, etc.
- Insurance companies, the companies which customers, companies, governments and other institutions transfer risk to in order to protect themselves financially from health related liabilities or death. Insurance companies directly use life tables to measure risk when providing life insurance to customers, and in order to remain profitable and offer a risk-assessed quality service to their customers these metrics modulate insurance premiums and aid in the process of decisionmaking when it comes to offering a policy to what the company could deem high or low risk customers.
- Ordinary people. Perhaps mere curiosity or desire to be informed, being able to know in a timely manner whether the population is at risk or healthy helps people make choices regarding how to take care of themselves or how to take care of their loved ones.

Metrics computed by the application

The application computes the following 5 different metrics:

- Excess mortality
- Cumulative mortality rate
- Cumulative relative mortality rate
- Mortality improvement factor
- Life expectancy

Excess mortality

The term *excess mortality* (EM) refers to a measurement which corresponds to the difference between the average deaths which have ocurred during the n years prior to the current t time. Typically, the window of years to compute the average deaths is often $n = 5$. For the application, a window of $n = 5$ years is used to compute the measurement.

If excess mortality is very high and above zero, then there have been more deaths than the previous n year average, if the number is smaller than zero, then there have been less deaths than the previous n year average.

Purpose

Excess mortality is used to assess how many more or less deaths than the previous years' average have ocurred. This way it's possible to determine if there is an increase in deaths that exceed what has been considered "normal".

During the COVID-19 pandemic the measurement was particularly useful as mortality spiked beyond the expectations of pretty much any country where the pandemic became widespread. Spain is no exception, and during the time where the deaths were at their peak, the excess mortality of that period of time was a highly publicized measure to illustrate how many deaths above what usually occurs happened.

Specification

As a result of the COVID-19 pandemic, future measures calculated after 2020 must exclude 2020 and 2021 from the calculation as the death counts are anomalous compared to previous and subsequent years prior the COVID-19 pandemic.

For the years between 2021 and 2027 inclusive, the application uses the same 5-year moving average used for 2020, meaning the average deaths corresponding to week w for $2015 \leq y \leq 2019$. After 2027 or prior to 2021 the average death counts used are computed from the 5 year window prior to the year(s) selected. For example, the average deaths for $w = 1$, 2019 will correspond to the average deaths during $w = 1$ for the years $2014 \leq y \leq 2018$, and the average deaths for $w = 1$, 2022 will correspond to the same average deaths observed on $w = 1$ for the years $2015 \leq y \leq 2019$.

Computation

The EM for week w of year y is computed as follows:

$$EM_{w,y} = \frac{D_{w,y}}{\overline{D_{w,y-5,y}}}$$

Where:

- $D_{w,y}$: death count for week w of year y
- $\overline{D_{w,y-5,y-1}}$: average death count for week w of years $y-5$, $y-4$, ..., $y-1$.

Exact definition:

$$EM_{w,y} = \frac{D_{w,y}}{\frac{1}{n}\sum_{k=1}^{n} D_{w,y-k}}$$

Where:

- $D_{w,y}$: death count for week w of year y
- n: years to look back to, default is $n = 5$
- $D_{w,y-k}$: death count for week w of year $y-k$, where k starts at $k = 1$ and ends at n with $k \in \mathbb{N}$.

Cumulative mortality rate

The *cumulative mortality rate* (CMR) represents the ratio between the sum of all deaths between the first week of a year or years up to a user-defined week, where weeks must be $1 \leq w \leq 52$.

Purpose

This ratio can be used standalone to gauge the amount of deaths in a time period divided by the population alive at that time period, but, it is most commonly used in this project as a component of the more informational *cumulative relative mortality rate* or *mortality improvement factor*.

Specification

The ratio is computed, depending on user selection, for years stretching as far back as 2010 and as updated as the current year. It can only be computed for any specified range between 1 and week w.

Computation

The CMR for year y and up to week s is computed as follows:

$$CMR_{w,y} = \frac{\sum_{w=1}^{s} D_{w,y}}{P_{w,y}}$$

Where:

- $D_{w,y}$: death count for week w of year y.
- $P_{w,y}$: population at week w of year y
- s: upper bound of week for which the death count is computed, with $1 \leq s \leq 52$ and $s \in \mathbb{N}$.

Cumulative relative mortality rate

The *cumulative relative mortality rate* ($CRMR$) corresponds to the ratio of the difference between the cumulative mortality rate (CMR) at a specified week and the average CMR between the years 2010 and 2019 inclusive, and the average CMR between the years 2010 and 2019 for the last week of those years (the 52nd week).

Purpose

The $CRMR$ allows us to show the medium to long-term effect of catastrophic or mortality-rising events on the population based on the CMR.

The effect of long-lasting events like the COVID-19 pandemic clearly show how mortality can part away from the mean significantly and how the cumulative effect of such mortality can rise to unprecedented levels for relatively long periods of time.

This particular metric can be affected significantly by events which we might perceive as not-so-dramatic when they occur (perhaps like a severe heat wave), but it lets us historically see how much worse or better (in terms of mortality) a given time period can be.

Specification

The metric is computed using the average CMR for years between 2010 and 2019. 2020 is explicitly excluded, as the effect of the COVID-19 pandemic would skew our perspective on 2021 mortality and further years.

The CMR for the last week of those years is averaged as it includes the entire year, given that the CMR is computed for a range of weeks starting at week 1 and ending at week s, for this metric we compute the denominator for week $s = 52$ and the numerator for a specified week range.

Computation

The CRMR for week w of year y is computed as follows:

$$CRMR_{w,y} = \frac{CMR_{w,y} - \frac{1}{10}\sum_{y=2010}^{2019} CMR_{w,y}}{\frac{1}{10}\sum_{y=2010}^{2019} CMR_{w=52,y}}$$

Where:

- $CMR_{w,y}$ is the *cumulative mortality rate* at week w of year y

Mortality improvement factor

Life expectancy

Appendix

Appendix A: Python

Appendix B: R

Appendix C: Project repository tree structure

The project consists of two repositories and their respectives **subfolders**/*files*:

The main project repository (*dreth/tfm_uc3m*)

- **api**: Contains all files related to the querying, acquisition, manipulation of new data from the two aggregated data sources (INE and Eurostat), along with various log files used to keep track of some metadata from the data sources (last updated data and provisional data).
  - **logs**
    - *earliest_eurostat_provisional.log*: Eurostat marks data as provisional when it can still be updated, as new deaths might occur. This file logs the earliest data entry marked as provisional.
    - *last_eurostat_update.log*: This file logs the last week for which there is available and updated data from Eurostat for Spain within the database *demo_r_mwk2_05*.
    - *last_ine_update.log*: This file logs the last week for which there is available and updated data from INE for table No. *9681*.
  - *dbs_check.py*: This script obtains a small sample of data from Eurostat and Ine and checks when the earliest provisional data point is, what is the latest obtainable week of data from both Eurostat and INE and writes it in the log files contained within **logs**.
  - *functions.py*: This script contains all the functions used for fetching, manipulating and outputting the data obtained from the aggregated data sources' respective APIs (Eurostat and INE).
  - *query.py*: This script will run a pre-constructed pipeline where the data is systematically queried per age group, manipulated and where new entries are added to the *.csv* files contained within the **data** folder.
- **dashboard**: Contains all files that allow the shiny app itself to run along with shapefiles for displaying

maps, shell scripts to check the databases' last update and last provisional status and the shell script to update the database from within the app.

- **www**
  - **maps**: Contains 2 folders which themselves contain the shapefiles for the map displayed using leaflet and the map displayed using ggplot2 in the maps section.
    - **map_shapefiles**: Contains the shapefiles used to display the map using the leaflet library.
    - **map_shapefiles_ggplot**: Contains the shapefiles used to display the map using the ggplot2 library.
  - **scripts**
    - *check_dbs.sh*: This script is ran every time the app is launched. This shell script runs *dbs_check.py* from the **api** folder.
    - *update_database.sh*: This is the update database script, basically checks the date, runs the *query.py* and *dbs_check.py* script from the **api** folder, appending new data (if available) to the data currently in the cloned repository tree. Then it copies the new files into the **dreth/tfm_uc3m_data** repository, then it commits and pushes the files to that repository.

- **data**: This folder is a subtree of the **dreth/tfm_uc3m_data** repository, therefore, it contains all the data used in the application along with logs showing diagnostic and historical information about database updates.
  - **logs**
    - *update_database.log*: This log file contains the date the database was last updated, it is written to every time the database is updated.
    - *update_history.log*: This log file contains the dates in which the database has been updated (historically)
- **docker**
- **docs**
- **paper**
- *README.md*

The data repository (*dreth/tfm_uc3m_data*)

- **ccaa_guide**
  - *guide_ccaa.json*

- *guide_ccaa.txt*
- *guide_ccaa_python_dict.p*