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.

(appendix) Project repository tree structure

First of all, the project consists of two repositories and their respectives **subfolders**/*files*:

- **The main project repository** (*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 **tfm_uc3m_data** repository, then it commits and pushes the files to that repository.

- **data**: This folder is a subtree of the repository
    - **logs**
        - *update_database.log*
        - *update_history.log*
- **docker**
- **docs**
- **paper**
- *README.md*
- **The data repository** (*tfm_uc3m_data*)
- **ccaa_guide**
    - *guide_ccaa.json*
    - *guide_ccaa.txt*
    - *guide_ccaa_python_dict.p*