

Covid-19 Shiny App

Rijin Baby

May 2020

Contents

1	Introduction	1
2	System modeling	2
2.1	Architecture, both logical and physical	2
2.2	Business processes and rules	2
2.3	Data structure and flow	3
2.4	Inter-process communication	4
2.5	System scope/scale	4
3	Dashboard Structure and Flow	4
3.1	Data Source	4
3.2	User Defined Functions - REMOVE	4
3.3	Tab Panels	5
4	Packages Used for Dashboard Development	5
5	Conclusion	6

1 Introduction

The outbreak of the Covid-19 epidemics in early 2020 has caused an unprecedented effort of the scientific community to produce models that could monitor and predict the evolution of the epidemics in a reliable way, also to advice governments to take actions which could mitigate the burden of hospitals to treat the affected patients, and thus reduce the mortality rate of the infection.

The COVID-19 outbreak in Italy has spread mainly in northern regions, particularly in Lombardy. However, even within the same region the virus has spread irregularly from province to province, producing real epicenters of infection in some provinces but also affecting other areas with relatively lower intensity.

In this dashboard we present some tools for analyzing and visualizing the COVID-19 outbreak in Italy at a provincial (NUTS-3) level by integrating official data from the Italian Ministry of Health with data extracted from official

press conferences of regional health authorities, especially regarding the number of deaths due to the Covid-19 which is not currently reported in official data releases. An adjusted time-dependent SIRD model is used to predict the epidemics behavior in the near future.

2 System modeling

2.1 Architecture, both logical and physical

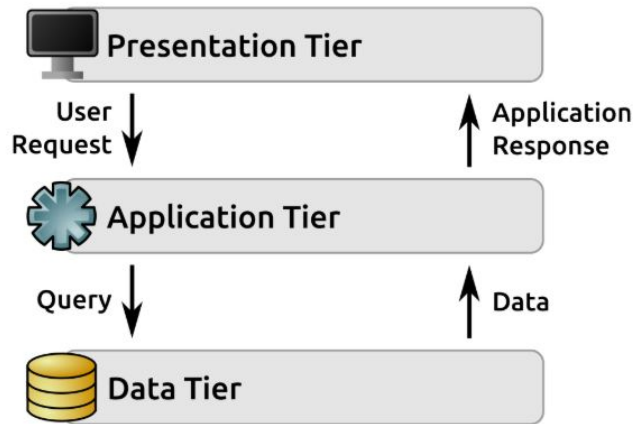


Figure 1: Logical Architecture

Logical Architecture This three-tier architecture is particularly common in web applications, where:

- The Presentation Tier is R Shiny dashboard
- The Application Tier is R shiny application that runs the necessary analysis on the data with the help of R program
- The Data Tier is the SQL database

2.2 Business processes and rules

As stated in the introduction the covid-19 death numbers on NUTS-3 level is the key highlight of this entire dashboard as this information is rare to find in the web. Primarily the focus is on two type of users.

- The Researcher:
 - Who worked on the SIRD model, and interested in viewing their work through a better visualized medium

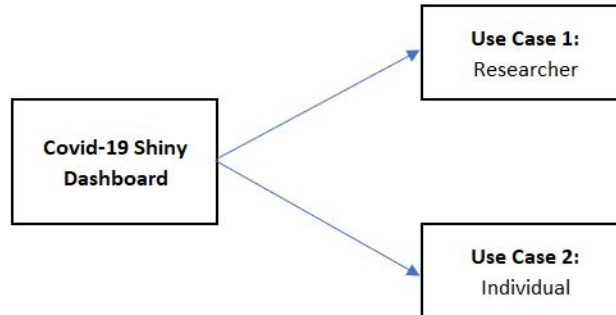


Figure 2: Use Cases

- Who are interested in exploring and validating the SIRD model working
- The Individual:
 - Who are interested in viewing and comparing the NUTS-3 level data among the provinces

2.3 Data structure and flow

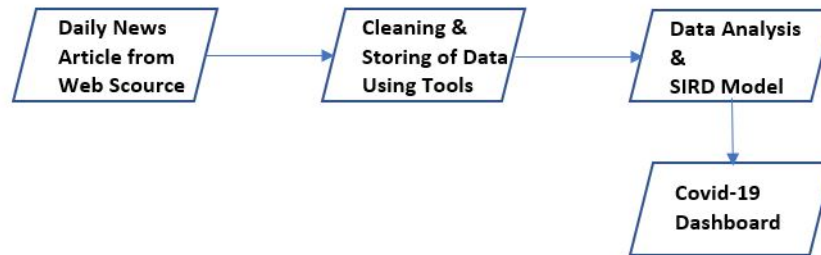


Figure 3: Data Flow

The daily news articles of all provinces are being automatically scrapped using python selenium to retrieve the NUTS-3 level death information, this data gets combined with the official GitHub data by the government that provides daily infection numbers. All these data are validated and are stored on SQL database on a regular basis. The further analysis, modeling and Rshiny applications are build using R programming.

For further reference on the SIRD model and analysis part, kindly refer to the paper available at <https://arxiv.org/abs/2005.12170>

2.4 Inter-process communication

2.5 System scope/scale

The current dashboard has a summary view, time-series view and SIRD model view panels. The aesthetics and the flow of data on each panel can be improved based on further research and on user feedback. The researchers are constantly working on adding additional dimensions to the application

Also, the application can be scaled for other counties based on the availability of relevant data.

3 Dashboard Structure and Flow

3.1 Data Source

1. Protezione Civile
 - Overall
 - Region
 - Province
2. COVID-Pro-Dataset - <https://github.com/CEEDS-DEMM/COVID-Pro-Dataset>
3. ISTAT

3.2 User Defined Functions - REMOVE

- **downAndSaveDataPMC()** - The function fetch data from the official GitHub repository of Protezione Civile
- **downDatiSQL()** - Download the daily province level death numbers from the database
- **downAndSaveDataISTAT()** - Download the clean ISTAT data from the local database
- **cumulDeathsProv()**
- **SIRDPParameterDataset()**
- **italy_data()** - The function fetch data from the official GitHub repository of Protezione Civile and returns the cumulative number along with daily changes for the confirmed, active, recovered and deceased COVID-19 cases.
- **dropdownMenuCustom()** - Function for the share and feedback options in the dashboard title panel.

3.3 Tab Panels

- **Province Map** - Brief overview on the COVID-19 numbers in Italy. Province level information on cumulative cases, cumulative rates and death rate.
- **Time Series** - Displays the time-series trend of multiple parameters that help in analysing the pandemic better. Also, comparison of covid deaths to ISTAT death numbers for provinces from the year 2015.
- **SIRD Model** - Model forecasting and parameter evaluation
- **About** - Description

4 Packages Used for Dashboard Development

- **tidyverse** - The 'tidyverse' is a set of packages that work in harmony because they share common data representations and 'API' design. This package is designed to make it easy to install and load multiple 'tidyverse' packages in a single step. Learn more about the 'tidyverse' at <https://tidyverse.org>
- **ggplot2** - A system for 'declaratively' creating graphics, based on "The Grammar of Graphics". You provide the data, tell 'ggplot2' how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.
- **plotly** - Create interactive web graphics from 'ggplot2' graphs and/or a custom interface to the (MIT-licensed) JavaScript library 'plotly.js' inspired by the grammar of graphics.
- **shiny** - Package from RStudio that makes it incredibly easy to build interactive web applications with R.
- **shinydashboard** - Create dashboards with 'Shiny'. This package provides a theme on top of 'Shiny', making it easy to create attractive dashboards.
- **shinydashboardPlus** - Extend 'shinydashboard' with 'AdminLTE2' components. 'AdminLTE2' is a free 'Bootstrap 3' dashboard template available at <https://adminlte.io>. Customize boxes, add timelines and a lot more.
- **shinycssloaders** - Automatically show loader animations while a Shiny output is (re)calculating. This is mostly a wrapper around the css-loaders.
- **leaflet** - Create and customize interactive maps using the 'Leaflet' JavaScript library and the 'htmlwidgets' package. These maps can be used directly from the R console, from 'RStudio', in Shiny applications and R Markdown documents.

- **geojsonio** - Convert data to 'GeoJSON' or 'TopoJSON' from various R classes, including vectors, lists, data frames, shape files, and spatial classes. 'geojsonio' does not aim to replace packages like 'sp', 'rgdal', 'rgeos', but rather aims to be a high level client to simplify conversions of data from and to 'GeoJSON' and 'TopoJSON'.
- **sp** - Classes and methods for spatial data; the classes document where the spatial location information resides, for 2D or 3D data. Utility functions are provided, e.g. for plotting data as maps, spatial selection, as well as methods for retrieving coordinates, for subsetting, print, summary, etc.
- **htmltools** - Tools for HTML generation and output.

5 Conclusion

Always keep an eye out for the version updates at <https://ceeds.unimi.it/covid-19-in-italy/>

References

<https://www.rdocumentation.org/> <https://www.istat.it/>