# Health metrics and the spread of infectious diseases

**with machine learning applications and spatial model analysis**

Federica Gazzelloni

12/16/22

# Table of contents

# Preface

Health metrics and the spread of infectious diseases, with machine learning applications and spatial model analysis are the topics of this book.

Here you'll find everything you need to analyze the state of health of a country and compare it with that of other countries. You will also be able to evaluate the best model for

The author of this book is Federica Gazzelloni who is an actuary and a statistician graduated from the Sapienza of Rome who is also a collaborator of the Institute of Health and Metric evaluations (IHME).

# 1 Introduction

Health metrics and the spread of infectious diseases, with machine learning applications and spatial model analysis, is a manual and a textbook for an introductory health data analysis course. It can also turn out to be a useful source code for both practitioners and data scientists.

Public health metrics such as **DALYs**, **YLL**, and **YLD** are expressed in numbers of years of life lost or lived with disabilities whose sum expresses a key value generally used for ranking the health status of a population.

A focus on the impact of recent infectious disease outbreaks, such as Covid19, on the state of health of the population, will be provided along with the most affected locations. The book is structured with an alternation of text and chunks of code in the R language to let the reader be a practitioner of real-world case studies on the topic.

To be more specific, the metrics used to summarize the state of health of a population will be compared across other locations and a prediction level tested on a few key models will be provided. The idea is to use {tidymodels}, and {INLA} as modeling tools. Finally complete the material some interesting spatial visualization, made with using {ggplot2}, {leaflet}, {sf}, {rgdal} R packages, plus other main packages for allowing the user for a wider understanding of the potentiality of the R language for both spatial and health metrics.

The book is foreseen for practitioners at early stages and graduated students in STEM.

# Part I

# Health metrics

Health metrics are key variables to understand more about the state of health of a population. In this book we'll talk about **numbers of years of life lost (YLLs)** and the **numbers of years lived with disabilities (YLDs)**, to finally identify the key metric of the **DALYs** which means **disability adjusted life years**.
The numbers of years of life lost by a population in comparison to other population in a different country or to the Global mean trend is what is expected it should be based on the latest study results relative to the well being of a country in terms of the definition of a healthy life. To give an example, let's think about a population whose individuals are living a good life, so defined *healthy life* measured on **life expectancy** established to be 80 years on average, as most of the World population meets this as a deadline.

The part of the population who do not meet this age, but dies earlier, contributes to building blocks of the numbers of years of life lost (YLLs), as well as for all that is related with a healthy living, the numbers of years spent dealing with a disability of different kind contribute to increasing the numbers of years lived with disabilities for a country's population.

To establish the well being, meant as the healthy life defying the state of health of a population, based on the latest findings of the most updated studies, the sum of these two values YLLs and YLDs releases the key metric of DALYs. This metric value is used to quickly identify the level of health of a population compared to a Global review.

In addition, this level is used to improve the proportion of countries who are in need of a better health status recognition. To be more specific, the focus on the numbers of years can help identify the areas where most of the years are lost and need for improvement, whether in facilities, research or investments.

An improvement of health at a Global level is reached when the definition of a **healthy file** is met in most of the countries where it wasn't before.

Furthermore, what we want to analyze are a series of data that are produced taking into account the tables of mortality and future life expectancy these are defined as means that these metrics have been considered important for assessing the state of health of one point population

What we refer to are the health metrics and which refer to the number of years lost due to an increase in mortality or in any case to a mortality trend that is above a certain general level which we can consider as the level optimal health globally.

# 2 YLLs, YLDs and DALYs

A closer look at the metrics, their usage and potentiality for improvements

In this chapter are shown the methods used for building three key metrics: YLLs, YLDs and DALYs. These will be used throughout the book for making comparisons among the state of health of different countries.

# 3 Metrics components

- Life tables and Life expectancy used in the book
- How to use them and where to find them

This section is dedicated to a closer look at what are the components of the health metrics, how to build them and finally how to use them for making countries comparison. YLLs,YLDs and DALYs can be used for different illnesses, and at different age levels.

## 3.1 Components

Two fundamental components are used for calculating the DALYs:

- life tables
- life expectancy

Both of these elements are key for achieving a high level value of the state of health of a population.

### 3.1.1 Life tables

The life tables are selected among the most frequently used, more information about how to build a like table can be found in the Appendix A of this book.

```
library(tidyverse)
xmart <- read_csv("data/xmart.csv",skip = 1)

xmart %>% head
```

```
# A tibble: 6 x 17
  Indicator      Age G~1 Both ~2 Male.~3 Femal~4 Both ~5 Male.~6 Femal~7 Both ~8
  <chr>          <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
1 nMx - age-spe~ &lt;1 ~ 4.78e-2 5.13e-2 4.41e-2 5.66e-2 6.02e-2 5.27e-2 6.75e-2
2 nMx - age-spe~ 1-4 ye~ 3.61e-3 3.67e-3 3.55e-3 4.63e-3 4.67e-3 4.58e-3 6.43e-3
3 nMx - age-spe~ 5-9 ye~ 4.61e-4 4.69e-4 4.53e-4 6.68e-4 6.75e-4 6.62e-4 1.15e-3
```

```
4 nMx - age-spe~ 10-14 ~ 3.70e-4 3.87e-4 3.52e-4 4.98e-4 5.16e-4 4.79e-4 8.27e-4
5 nMx - age-spe~ 15-19 ~ 1.32e-3 1.46e-3 1.18e-3 2.22e-3 2.74e-3 1.67e-3 1.88e-3
6 nMx - age-spe~ 20-24 ~ 2.01e-3 2.22e-3 1.79e-3 3.41e-3 4.47e-3 2.27e-3 2.87e-3
# ... with 8 more variables: Male...10 <dbl>, Female...11 <dbl>,
#   `Both sexes...12` <dbl>, Male...13 <dbl>, Female...14 <dbl>,
#   `Both sexes...15` <dbl>, Male...16 <dbl>, Female...17 <dbl>, and
#   abbreviated variable names 1: `Age Group`, 2: `Both sexes...3`,
#   3: Male...4, 4: Female...5, 5: `Both sexes...6`, 6: Male...7,
#   7: Female...8, 8: `Both sexes...9`
```

```
xmart_yrs <- read_csv("data/xmart.csv")
```

```
New names:
Rows: 134 Columns: 17
-- Column specification
-------------------------------------------------------- Delimiter: "," chr
(17): ...1, ...2, 2019...3, 2019...4, 2019...5, 2015...6, 2015...7, 2015...
i Use `spec()` to retrieve the full column specification for this data. i
Specify the column types or set `show_col_types = FALSE` to quiet this message.
* `` -> `...1`
* `` -> `...2`
* `2019` -> `2019...3`
* `2019` -> `2019...4`
* `2019` -> `2019...5`
* `2015` -> `2015...6`
* `2015` -> `2015...7`
* `2015` -> `2015...8`
* `2010` -> `2010...9`
* `2010` -> `2010...10`
* `2010` -> `2010...11`
* `2005` -> `2005...12`
* `2005` -> `2005...13`
* `2005` -> `2005...14`
* `2000` -> `2000...15`
* `2000` -> `2000...16`
* `2000` -> `2000...17`
```

```
xmart_yrs <- xmart_yrs[-1,]%>%
  janitor::clean_names()%>%
  pivot_longer(cols=3:17,names_to="years",values_to="values")%>%
  mutate(values=as.numeric(values))
```

```
xmart_yrs %>% names
```

```
[1] "x1"     "x2"     "years"  "values"
```

```
xmart_yrs
```

```
# A tibble: 1,995 x 4
   x1                                                    x2         years  values
   <chr>                                                 <chr>      <chr>   <dbl>
 1 nMx - age-specific death rate between ages x and x+n &lt;1 year x2019~ 0.0478
 2 nMx - age-specific death rate between ages x and x+n &lt;1 year x2019~ 0.0513
 3 nMx - age-specific death rate between ages x and x+n &lt;1 year x2019~ 0.0441
 4 nMx - age-specific death rate between ages x and x+n &lt;1 year x2015~ 0.0566
 5 nMx - age-specific death rate between ages x and x+n &lt;1 year x2015~ 0.0602
 6 nMx - age-specific death rate between ages x and x+n &lt;1 year x2015~ 0.0527
 7 nMx - age-specific death rate between ages x and x+n &lt;1 year x2010~ 0.0675
 8 nMx - age-specific death rate between ages x and x+n &lt;1 year x2010~ 0.0723
 9 nMx - age-specific death rate between ages x and x+n &lt;1 year x2010~ 0.0625
10 nMx - age-specific death rate between ages x and x+n &lt;1 year x2005~ 0.0822
# ... with 1,985 more rows
```

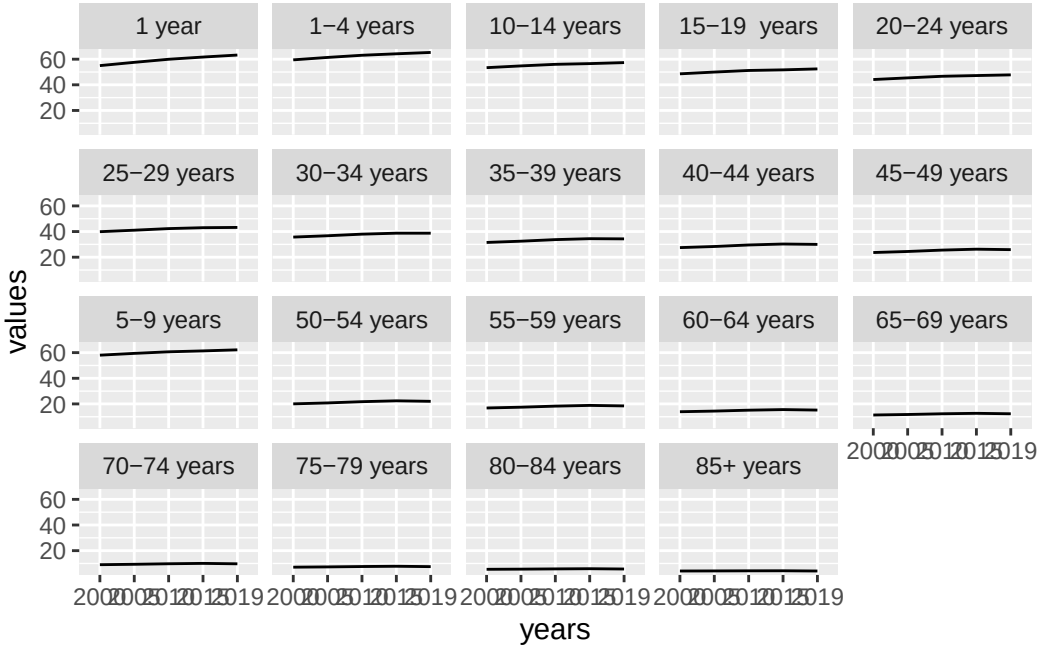
```
xmart_tidy <- xmart %>%
  janitor::clean_names()%>%
  pivot_longer(cols = 3:17,names_to="sex",values_to="values") %>%
  full_join(xmart_yrs,by=c("indicator"="x1","age_group"="x2","values")) %>%
  mutate(age_group=sub("&lt;","",age_group),
         sex=gsub("_\\d+","",sex),
         sex=ifelse(sex=="both_sexes","both",sex),
         years=sub("x","",years),
         years=gsub("_\\d+","",years))


xmart_tidy%>%count(indicator)
```

```
# A tibble: 7 x 2
  indicator                                             n
  <chr>                                             <int>
1 ex - expectation of life at age x                   285
```

```
2 lx - number of people left alive at age x              495
3 ndx - number of people dying between ages x and x+n    285
4 nLx - person-years lived between ages x and x+n        285
5 nMx - age-specific death rate between ages x and x+n   285
6 nqx - probability of dying between ages x and x+n      495
7 Tx - person-years lived above age x                    285
```

```
xmart_tidy %>%
  filter(sex=="both",
         indicator=="ex - expectation of life at age x")%>%
         #age_group=="1 year")%>%
  ggplot(aes(years,values,group=indicator))+
  geom_line()+
  facet_wrap(vars(age_group))
```




### 3.1.2 Life expectancy

The life expectancy rates are calculated with consideration of the probability of survival based on key parameter such as age, and deaths probabilities for that age. More info about how to calculate the life expectancy can be found #sec-tools of this book.

## 3.2 How to build the metrics

In this section a practical calculation of the health metrics is done for the practitioner to be able to replicate this calculation for further analysis based on these key elements.

### 3.2.1 YLLs

The number of years of life lost YLLs is the first of the three metrics that is calculated, and is important for releasing a first look at the status of a population. It is calculated for identifying the area where improvement is required for reducing the loss in health status and clearly reducing the probability of death.

Life expectancy for this book calculations is from worlddata

### 3.2.2 YLDs

### 3.2.3 DALYs

## 3.3 How to use the metrics

# 4 Causes and risks

- How to use the metrics
- overview of the causes and risks

# Part II

# Modeling

- Feature engineering
- Model selection
- Packages and functions to use for the analysis
- Attempt at predicting the future

# 5 Techniques

# 6 Packages and functions

# 7 Predicting the future

# Part III

# Data Visualizations

# 8 Application of the model results

# 9 Spatial data modeling

# 10 Examples of data visualizations

# Part IV

# Case Studies

# 11 Covid19

# 12 The state of health

# 13 Summary

# Conclusions

# References

https://cdn.who.int/media/docs/default-source/gho-documents/global-health-estimates/ghe2019_cod_metho

life tables:

- https://apps.who.int/gho/data/node.main.LIFECOUNTRY?lang=en
- https://ghdx.healthdata.org/record/ihme-data/gbd-2019-life-tables-1950-2019

# A Life tables and Life expectancy

# B Tools used to make this book

To set up the environment for replicating the code used in this book the R language is needed as well as R and Rstudio IDE environments. The following sections contain the directions for installing **R** and **RStudio**, how to set up a book with **quarto** and how to use **GitHub** as a version saver source.

## B.1 RStudio installation

Download and install R: Download and install RStudio IDE:

## B.2 Info on how to setup this project in quarto

quarto is the new version of **Rmarkdown**, it can be used for making notes, presentations, websites, books and more.

In this project the book has been made in quarto and version saved on github.

https://quarto.org/docs/publishing/github-pages.html In RStudio create a new project on a new directory and in terminal type: `add git quarto book project`

The automated process will create a `_quarto.yml` file, the top of the file will look like this one:

```
project: type: book
```

On terminal type: `quarto preview`

It creates a folder `_book`

# C github later:

# D source: https://happygitwithr.com/existing-github-last.html

# E connect with github

# F create a github repo with the same name from github website

# G then type

usethis::use_git() # this pushes all files in R to a remote folder designed to github repo

# H in terminal connect with github repo

git init # git remote add origin https://github.com/Fgazzelloni/infectious.git # git branch -M main # git push -u origin main #
#
# # publish your book on github pages # # change the _quarto.yml file into: # project: # type: book # output-dir: docs # # add a .nojekyll file …(terminal) # touch .nojekyll # # then type # quarto render # # some issues might arise if more than one # # calculation is made inside a single cunck # # split the cuncks! # # quarto render creates a folder # docs

## H.1 GitHub useful commands