COS-R403. Special Research Methods. *Forecasting I: Introduction*

Hands-on excercises

Day 5 of intensive 5-day course

University of Helsinki, Finland

04.05.2020–08.05.2020

Lecturer: Christina Bohk-Ewald

Source: https://github.com/christina-bohk-ewald/2020-COS-R403-forecasting-I-introduction

Table of content:

1. Some preparations in R

1.1 Open a new script for day 5 in R and save it to a folder of your choice.

1.2 Create a filepath to a folder where you would like to save your outcome. For example,

```
the.plot.path <- c("C:/plots")
```

1.3 You can then set the working directory to this outcome path

```
setwd(the.plot.path)
```

2. Load required input data

The Github repository *demographic-scaling-model* provides an R implementation and information about required data to estimate COVID-19 infections. You will eventually need confirmed cases and reported deaths from COVID-19, population counts, and life tables, and scaled infection fatality rates.

2.1 Load COVID-19 data

Please start with loading confirmed cases and reported deaths attributable to COVID-19 (course day 4).

```
require(openxlsx)

confirmed <- read.csv("time_series_covid19_confirmed_global.csv",header=TRUE,
stringsAsFactors = FALSE)
## confirmed[1:2,]

deaths <- read.csv("time_series_covid19_deaths_global.csv",header=TRUE,
stringsAsFactors = FALSE)
## deaths[1:2,]
```

Brief data description. The data objects *confirmed* and *deaths* contain confirmed cases and reported deaths attributable to COVID-19 by country (or state or province) and day since January 22, 2020.

2.2 Load population counts

In a first step, please load UNWPP2019 total population counts (course day 1):

```
wom <- read.xlsx(file.path(the.data.path,
paste("WPP2019_INT_F03_3_POPULATION_BY_AGE_ANNUAL_FEMALE.xlsx",sep="")),
sheet = 1,startRow = 17)
wom_select <- wom[which(wom[,"Reference.date.(as.of.1.July)"]=="2019"),c(3,8:109)]

men <- read.xlsx(file.path(the.data.path,
paste("WPP2019_INT_F03_2_POPULATION_BY_AGE_ANNUAL_MALE.xlsx",sep="")),
sheet = 1,startRow = 17)
men_select <- men[which(men[,"Reference.date.(as.of.1.July)"]=="2019"),c(3,8:109)]
```

Brief data description. The data objects *wom_select* and *men_select* contain population counts by single years of age for all UN countries in 2019.

In a second step, please aggregate these 2019 population counts into 10-year age groups:

```r
wom_select_10y <- matrix(NA,nr=dim(wom_select)[1],ncol=length(seq(0,80,10)))
men_select_10y <- matrix(NA,nr=dim(men_select)[1],ncol=length(seq(0,80,10)))

rownames(wom_select_10y) <- wom_select[,"Region,.subregion,.country.or.area.*"]
rownames(men_select_10y) <- men_select[,"Region,.subregion,.country.or.area.*"]
colnames(wom_select_10y) <- seq(0,80,10)
colnames(men_select_10y) <- seq(0,80,10)

for(country in 1:dim(wom_select)[1]){

    current_wom_select <- wom_select[country,]
    current_men_select <- men_select[country,]

    for(age in 1:length(seq(0,80,10))){
        current_age <- seq(0,80,10)[age]

        wom_select_10y[country,age] <- sum( as.numeric( current_wom_select
        [as.character((current_age):(current_age+9))] ) )

        men_select_10y[country,age] <- sum( as.numeric( current_men_select
        [as.character((current_age):(current_age+9))] ) )

        if(current_age==80){
            wom_select_10y[country,age] <- sum( as.numeric
            ( current_wom_select[as.character((current_age):(current_age+20))] ) )

            men_select_10y[country,age] <- sum( as.numeric
            ( current_men_select[as.character((current_age):(current_age+20))] ) )
        }
    }
}
```

Brief data description. The matrices *wom_select_10y* and *men_select_10y* contain 2019 population counts by 10-year age groups for all UN countries.

2.3 Load abridged life tables

Please go to the UNWPP2019 website, download abridged life tables for both sexes together, save them into your project folder, and then load them into R.

```r
lt_1950_2020 <- read.xlsx("WPP2019_MORT_F17_1_ABRIDGED_LIFE_TABLE_BOTH_SEXES.xlsx",
sheet = 1,startRow = 17)
```

Brief data description. The data object *lt_1950_2020* contains abridged life tables for both sexes for all UN countries.

2.4 Load global pattern over age of COVID-19 deaths

Verity and colleagues (2020, page 5) report infection fatality rates for 10-year age groups for Hubei, China on page 5. Please create a data object *ifr* that contains these data or download it from the GitHub repository for this course.

```r
global_age_dist_deaths <- source("global_age_dist_deaths.R")
## global_age_dist_deaths
```

Brief data description. The data object *global_age_dist_deaths* contains the global pattern over 10-year age groups of COVID-19 deaths, based on data of Dudel and colleagues (2020).

2.5 Load infection fatality rates from Verity et al.

Verity and colleagues (2020, page 5) report infection fatality rates by 10-year age groups for Hubei, China, on page 5. Please create a data object *ifr_by_age_china_verity* that contains these data or download it from the GitHub repository for this course.

```
ifr_by_age_china_verity <- read.table("infection-fatality-rates-by-age-china-Verity.txt",
header=FALSE, stringsAsFactors = FALSE)

ifr_by_age_china_verity
```

```
##   V1      V2       V3       V4
## 1  0 1.6e-05 1.85e-06 0.000249
## 2 10 7.0e-05 1.50e-05 0.000500
## 3 20 3.1e-04 1.40e-04 0.000920
## 4 30 8.4e-04 4.10e-04 0.001850
## 5 40 1.6e-03 7.60e-04 0.003200
## 6 50 6.0e-03 3.40e-03 0.013000
## 7 60 1.9e-02 1.10e-02 0.039000
## 8 70 4.3e-02 2.50e-02 0.084000
## 9 80 7.8e-02 3.80e-02 0.133000
```

Brief data description. The data object *ifr_by_age_china_verity* contains the modal estimate as well as the lower and upper bound of the 95 percent credible interval for the infection fatality rates of Hubei, China, by 10-year age groups.

3. Estimate COVID-19 infections for Italy

3.1 Scale infection fatality rates of Hubei, China, to account for cross-country differences in the age structure, the health conditions, and the health care systems following the procedure introduced in Bohk-Ewald and colleagues (2020).

In a first step, please run and apply the basic function *to_ungroup* in order to disaggregate infection fatality rates into single years of age. This basic function makes use of the *smooth.spline.*

```
to_ungroup <- function(to_ungroup,nr_grouped_years){

    seq_ungrouped_years <- seq(0,length(to_ungroup)*nr_grouped_years)
    cumsum_to_ungroup <- cumsum(c(sum(to_ungroup),to_ungroup))
    grouped_time_points <- c(0,(1:length(to_ungroup))*nr_grouped_years)

    applied_smooth_spline <- smooth.spline(x=grouped_time_points,y=cumsum_to_ungroup)
    predict_cumsum_ungroup <- predict(applied_smooth_spline,x=seq_ungrouped_years)$y
    ungrouped <- diff(predict_cumsum_ungroup)
    return(ungrouped)
}
```

```
ungrouped_mode_ifr_by_single_age_china_sp <- to_ungroup(to_ungroup=
                ifr_by_age_china_verity[,2],nr_grouped_years=10)
```

Brief data description. The data object *ungrouped_mode_ifr_by_single_age_china_sp* contains infection fatality rates by single years of age.

In a second step, please run and apply the basic functions *get_ungrouped_ex_2015_2020* and *map_fr_betw_ref_and_coi_thanatAge* in order to scale Hubei's infection fatality rates and to account for differences in age structure, health conditions, and health care systems between China and Italy.

```
get_ungrouped_ex_2015_2020 <- function(country_name, lt_1950_2020){
    current_period_data <- lt_1950_2020[which(lt_1950_2020[,8]=="2015-2020"),]
    current_period_data <- current_period_data[which(current_period_data[,3]==country_name),]
    current_ex_data <- as.numeric(current_period_data[,19])
    smooth_current_ex_data <- smooth.spline(x=c(0,1,seq(5,100,5)),y=current_ex_data)
    new_x <- c(seq(0,0.99,0.01),seq(1,4.99,0.01),seq(5,100,0.01))
    predict_smooth_current_ex_data <- predict(smooth_current_ex_data,new_x,len=new_x)
    return(predict_smooth_current_ex_data)
}

map_fr_betw_ref_and_coi_thanatAge <- function(deaths,coi,lt_1950_2020,
    ungrouped_ifr_by_single_age_china_sp){

    cfr_coi_mapped_rc_china_based_on_thanat_x <-
    matrix(NA,nr=1,nc=length(ungrouped_ifr_by_single_age_china_sp))

    rownames(cfr_coi_mapped_rc_china_based_on_thanat_x) <- coi

    current_pop_insert <- coi

    for(chronAge in 1:90){
        current_ref_y <- get_ungrouped_ex_2015_2020(country_name="China",
        lt_1950_2020)$y

        current_ref_x <- get_ungrouped_ex_2015_2020(country_name="China",
        lt_1950_2020)$x

        current_coi_y <- get_ungrouped_ex_2015_2020(country_name=current_pop_insert,
        lt_1950_2020)$y

        current_coi_x <- get_ungrouped_ex_2015_2020(country_name=current_pop_insert,
        lt_1950_2020)$x

        current_y_ref_of_chronAge <- current_ref_y[which(current_ref_x==(chronAge-1))]
        equal_y <- which(round(current_coi_y,3)==round(current_y_ref_of_chronAge,3))[1]

        if(is.na(equal_y)){
            n <- 0.001
            while(is.na(equal_y)){
                equal_y <- which(round(current_coi_y,3)==
                (round(current_y_ref_of_chronAge,3)-n))[1]

                n <- n+0.001
            } ## while
        } ## if

        equivalent_x_coi <- current_coi_x[equal_y]

        if((round(equivalent_x_coi,0)+1)>length(ungrouped_ifr_by_single_age_china_sp)){
            equivalent_x_coi <- 89
        }

        cfr_coi_mapped_rc_china_based_on_thanat_x[1,equivalent_x_coi] <-
        ungrouped_ifr_by_single_age_china_sp[chronAge]

    } ## for chronAge


    return(cfr_coi_mapped_rc_china_based_on_thanat_x)

} ## function
```

In a third step, please fill in values for infection fatality rates that could not be mapped. This should be only very few values.

```
mapped_mode_ifr_thanatAge <-
map_fr_betw_ref_and_coi_thanatAge(deaths=
        deaths[which(deaths[,"Country.Region"]=="Italy"),(5:ncol(deaths))],
        coi="Italy",
        lt_1950_2020=lt_1950_2020,
        ungrouped_ifr_by_single_age_china_sp=ungrouped_mode_ifr_by_single_age_china_sp)

pos_na <- which(is.na(mapped_mode_ifr_thanatAge[1,]))
    if(length(pos_na)>0){
        for(pos in 1:length(pos_na)){
            if(pos_na[pos] < 6){
                mapped_mode_ifr_thanatAge[1,pos_na[pos]] <-
                    min(mapped_mode_ifr_thanatAge[1,],na.rm=TRUE)
            }
            if(pos_na[pos] >= 6){
                mapped_mode_ifr_thanatAge[1,pos_na[pos]] <-
                    mapped_mode_ifr_thanatAge[1,pos_na[pos]-1]
            }
        } ## for pos
    } ## if
```

In a fourth step, please aggregate scaled infection fatality rates into 10-year age groups.

```
mapped_mode_ifr_thanatAge_10y <- c(0)
for(group in 1:9){
    pos <- (1+10*(group-1)):(10+10*(group-1))
    mapped_mode_ifr_thanatAge_10y[group] <- sum(mapped_mode_ifr_thanatAge[pos])
}
```

Brief data description. The data object *mapped_mode_ifr_thanatAge_10y* contains scaled infection fatality rates by 10-year age groups for Italy.

To get an idea of how the scaling works, please visualize Hubei's infection fatality rates and and Italy's scaled infection fatality rates.

3.2 Disaggregate total deaths into 10-year age groups

```
deaths_by_age <- matrix(0,nr=length(seq(0,80,10)),nc=length(5:ncol(deaths)))
rownames(deaths_by_age) <- seq(0,80,10)
colnames(deaths_by_age) <- colnames(deaths[,5:ncol(deaths)])
for(day in 1:length(5:ncol(deaths))){
    deaths_by_age[,day] <- deaths[which(deaths[,"Country.Region"]=="Italy"),(day+4)] *
unlist(global_age_dist_deaths$value)
}
```

```r
deaths_by_age[,ncol(deaths_by_age)]
```

```
##            0           10           20           30           40           50
## 1.706457e-01 3.606504e+00 1.787970e+01 4.810981e+01 1.035155e+02 3.917200e+02
##           60           70           80
## 1.508741e+03 5.646769e+03 1.502449e+04
```

3.3 Estimate Italy's COVID-19 infections based on COVID-19 deaths and scaled infection fatality rates

```r
sum( deaths_by_age[,ncol(deaths_by_age)] / mapped_mode_ifr_thanatAge_10y )/1000
```

```
## [1] 1003.646
```

```r
confirmed[which(confirmed[,"Country.Region"]=="Italy"),ncol(confirmed)]/1000
```

```
## [1] 172.434
```

```r
(sum( deaths_by_age[,ncol(deaths_by_age)] / mapped_mode_ifr_thanatAge_10y )/1000) /
(confirmed[which(confirmed[,"Country.Region"]=="Italy"),ncol(confirmed)]/1000 )
```

```
## [1] 5.820464
```

Brief data description. As of April 17, 2020, Italy had 172k confirmed cases. According to the approach of Bohk-Ewald and colleagues, the number of infections is estimated to be almost six times higher, 1 million.

Something to think about. How can you adjust the R code to estimate the number of COVID-19 infections for a different point in time, as, for example, April 1, 2020? How have the numbers of infections developed over time?

4. Now is the time to do this again for Finland. As always, feel free to adapt the R code to your own needs and creativity :-)