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

Hands-on excercises

Day 4 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 4 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. Download, load, and explore COVID-19 data

On day 4 we explore trends of the COVID-19 pandemic. We also think about how we could make use of demographic forecasting tools in order to tackle urgent questions during such a crisis. We first collect available data.

2.1 Download confirmed cases and reported deaths attributable to COVID-19

Please go to the website of the Johns Hopkins University CSSE. The files

- *time_series_covid19_confirmed_global.csv*
- *time_series_covid19_deaths_global.csv*

contain confirmed cases and reported deaths, respectively, for many countries on a daily basis since January 22, 2020. Please download these two files and save them in your project folder.

2.2 Load COVID-19 data

Please load the numbers of confirmed cases and reported deaths from COVID-19 in R using the function *read.csv* of the R-package *openxlsx*.

```
require(openxlsx)

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

```
##   Province.State Country.Region     Lat    Long X1.22.20 X1.23.20 X1.24.20
## 1                    Afghanistan 33.0000 65.0000        0        0        0
## 2                        Albania 41.1533 20.1683        0        0        0
##   X1.25.20 X1.26.20 X1.27.20 X1.28.20 X1.29.20 X1.30.20 X1.31.20 X2.1.20
## 1        0        0        0        0        0        0        0       0
## 2        0        0        0        0        0        0        0       0
##   X2.2.20 X2.3.20 X2.4.20 X2.5.20 X2.6.20 X2.7.20 X2.8.20 X2.9.20 X2.10.20
## 1       0       0       0       0       0       0       0       0        0
## 2       0       0       0       0       0       0       0       0        0
##   X2.11.20 X2.12.20 X2.13.20 X2.14.20 X2.15.20 X2.16.20 X2.17.20 X2.18.20
## 1        0        0        0        0        0        0        0        0
## 2        0        0        0        0        0        0        0        0
##   X2.19.20 X2.20.20 X2.21.20 X2.22.20 X2.23.20 X2.24.20 X2.25.20 X2.26.20
## 1        0        0        0        0        0        1        1        1
## 2        0        0        0        0        0        0        0        0
```

```
##   X2.27.20 X2.28.20 X2.29.20 X3.1.20 X3.2.20 X3.3.20 X3.4.20 X3.5.20 X3.6.20
## 1        1        1        1       1       1       1       1       1       1
## 2        0        0        0       0       0       0       0       0       0
##   X3.7.20 X3.8.20 X3.9.20 X3.10.20 X3.11.20 X3.12.20 X3.13.20 X3.14.20 X3.15.20
## 1       1       4       4        5        7        7        7       11       16
## 2       0       0       2       10       12       23       33       38       42
##   X3.16.20 X3.17.20 X3.18.20 X3.19.20 X3.20.20 X3.21.20 X3.22.20 X3.23.20
## 1       21       22       22       22       24       24       40       40
## 2       51       55       59       64       70       76       89      104
##   X3.24.20 X3.25.20 X3.26.20 X3.27.20 X3.28.20 X3.29.20 X3.30.20 X3.31.20
## 1       74       84       94      110      110      120      170      174
## 2      123      146      174      186      197      212      223      243
##   X4.1.20 X4.2.20 X4.3.20 X4.4.20 X4.5.20 X4.6.20 X4.7.20 X4.8.20 X4.9.20
## 1     237     273     281     299     349     367     423     444     484
## 2     259     277     304     333     361     377     383     400     409
##   X4.10.20 X4.11.20 X4.12.20 X4.13.20 X4.14.20 X4.15.20 X4.16.20 X4.17.20
## 1      521      555      607      665      714      784      840      906
## 2      416      433      446      467      475      494      518      539
##   X4.18.20 X4.19.20 X4.20.20 X4.21.20 X4.22.20 X4.23.20 X4.24.20 X4.25.20
## 1      933      996     1026     1092     1176     1279     1351     1463
## 2      548      562      584      609      634      663      678      712
##   X4.26.20 X4.27.20 X4.28.20 X4.29.20 X4.30.20 X5.1.20 X5.2.20
## 1     1531     1703     1828     1939     2171    2335    2469
## 2      726      736      750      766      773     782     789
```

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

```
##   Province.State Country.Region     Lat    Long X1.22.20 X1.23.20 X1.24.20
## 1                    Afghanistan 33.0000 65.0000        0        0        0
## 2                        Albania 41.1533 20.1683        0        0        0
##   X1.25.20 X1.26.20 X1.27.20 X1.28.20 X1.29.20 X1.30.20 X1.31.20 X2.1.20
## 1        0        0        0        0        0        0        0       0
## 2        0        0        0        0        0        0        0       0
##   X2.2.20 X2.3.20 X2.4.20 X2.5.20 X2.6.20 X2.7.20 X2.8.20 X2.9.20 X2.10.20
## 1       0       0       0       0       0       0       0       0        0
## 2       0       0       0       0       0       0       0       0        0
##   X2.11.20 X2.12.20 X2.13.20 X2.14.20 X2.15.20 X2.16.20 X2.17.20 X2.18.20
## 1        0        0        0        0        0        0        0        0
## 2        0        0        0        0        0        0        0        0
##   X2.19.20 X2.20.20 X2.21.20 X2.22.20 X2.23.20 X2.24.20 X2.25.20 X2.26.20
## 1        0        0        0        0        0        0        0        0
## 2        0        0        0        0        0        0        0        0
##   X2.27.20 X2.28.20 X2.29.20 X3.1.20 X3.2.20 X3.3.20 X3.4.20 X3.5.20 X3.6.20
## 1        0        0        0       0       0       0       0       0       0
## 2        0        0        0       0       0       0       0       0       0
##   X3.7.20 X3.8.20 X3.9.20 X3.10.20 X3.11.20 X3.12.20 X3.13.20 X3.14.20 X3.15.20
## 1       0       0       0        0        0        0        0        0        0
## 2       0       0       0        0        1        1        1        1        1
##   X3.16.20 X3.17.20 X3.18.20 X3.19.20 X3.20.20 X3.21.20 X3.22.20 X3.23.20
## 1        0        0        0        0        0        0        1        1
## 2        1        1        2        2        2        2        2        4
##   X3.24.20 X3.25.20 X3.26.20 X3.27.20 X3.28.20 X3.29.20 X3.30.20 X3.31.20
## 1        1        2        4        4        4        4        4        4
```

```
## 2        5        5        6        8       10       10       11       15
##   X4.1.20 X4.2.20 X4.3.20 X4.4.20 X4.5.20 X4.6.20 X4.7.20 X4.8.20 X4.9.20
## 1       4       6       6       7       7      11      14      14      15
## 2      15      16      17      20      20      21      22      22      23
##   X4.10.20 X4.11.20 X4.12.20 X4.13.20 X4.14.20 X4.15.20 X4.16.20 X4.17.20
## 1       15       18       18       21       23       25       30       30
## 2       23       23       23       23       24       25       26       26
##   X4.18.20 X4.19.20 X4.20.20 X4.21.20 X4.22.20 X4.23.20 X4.24.20 X4.25.20
## 1       30       33       36       36       40       42       43       47
## 2       26       26       26       26       27       27       27       27
##   X4.26.20 X4.27.20 X4.28.20 X4.29.20 X4.30.20 X5.1.20 X5.2.20
## 1       50       57       58       60       64      68      72
## 2       28       28       30       30       31      31      31
```

Describe these data. For which countries and states are they available, for which days are they available?

2.3 Explore data objects *confirmed* and *deaths*.

How many confirmed cases and reported deaths are there for Italy and for China most recently?

```
confirmed[which(deaths[,"Country.Region"]=="Italy"),c(1:4,ncol(confirmed))]
```

```
##     Province.State Country.Region Lat Long X5.2.20
## 138                         Italy  43   12  209328
```

```
deaths[which(deaths[,"Country.Region"]=="Italy"),c(1:4,ncol(deaths))]
```

```
##     Province.State Country.Region Lat Long X5.2.20
## 138                         Italy  43   12   28710
```

```
confirmed[which(deaths[,"Country.Region"]=="China"),c(1:4,ncol(confirmed))]
```

```
##    Province.State Country.Region     Lat     Long X5.2.20
## 50          Anhui          China 31.8257 117.2264     991
## 51        Beijing          China 40.1824 116.4142     593
## 52      Chongqing          China 30.0572 107.8740     579
## 53         Fujian          China 26.0789 117.9874     356
## 54          Gansu          China 37.8099 101.0583     139
## 55      Guangdong          China 23.3417 113.4244    1588
## 56        Guangxi          China 23.8298 108.7881     254
## 57        Guizhou          China 26.8154 106.8748     147
## 58         Hainan          China 19.1959 109.7453     168
## 59          Hebei          China 39.5490 116.1306     328
## 60   Heilongjiang          China 47.8620 127.7615     944
## 61          Henan          China 33.8820 113.6140    1276
## 62      Hong Kong          China 22.3000 114.2000    1039
## 63          Hubei          China 30.9756 112.2707   68128
## 64          Hunan          China 27.6104 111.7088    1019
## 65 Inner Mongolia          China 44.0935 113.9448     201
## 66        Jiangsu          China 32.9711 119.4550     653
## 67        Jiangxi          China 27.6140 115.7221     937
## 68          Jilin          China 43.6661 126.1923     112
## 69       Liaoning          China 41.2956 122.6085     146
## 70          Macau          China 22.1667 113.5500      45
## 71        Ningxia          China 37.2692 106.1655      75
## 72        Qinghai          China 35.7452  95.9956      18
## 73        Shaanxi          China 35.1917 108.8701     306
```

```
## 74       Shandong          China 36.3427 118.1498     787
## 75       Shanghai          China 31.2020 121.4491     652
## 76         Shanxi          China 37.5777 112.2922     197
## 77        Sichuan          China 30.6171 102.7103     561
## 78        Tianjin          China 39.3054 117.3230     190
## 79          Tibet          China 31.6927  88.0924       1
## 80       Xinjiang          China 41.1129  85.2401      76
## 81         Yunnan          China 24.9740 101.4870     185
## 82       Zhejiang          China 29.1832 120.0934    1268
```

```
deaths[which(deaths[,"Country.Region"]=="China"),c(1:4,ncol(deaths))]
```

```
##    Province.State Country.Region     Lat     Long X5.2.20
## 50          Anhui          China 31.8257 117.2264       6
## 51        Beijing          China 40.1824 116.4142       9
## 52      Chongqing          China 30.0572 107.8740       6
## 53         Fujian          China 26.0789 117.9874       1
## 54          Gansu          China 37.8099 101.0583       2
## 55      Guangdong          China 23.3417 113.4244       8
## 56        Guangxi          China 23.8298 108.7881       2
## 57        Guizhou          China 26.8154 106.8748       2
## 58         Hainan          China 19.1959 109.7453       6
## 59          Hebei          China 39.5490 116.1306       6
## 60   Heilongjiang          China 47.8620 127.7615      13
## 61          Henan          China 33.8820 113.6140      22
## 62      Hong Kong          China 22.3000 114.2000       4
## 63          Hubei          China 30.9756 112.2707    4512
## 64          Hunan          China 27.6104 111.7088       4
## 65 Inner Mongolia          China 44.0935 113.9448       1
## 66        Jiangsu          China 32.9711 119.4550       0
## 67        Jiangxi          China 27.6140 115.7221       1
## 68          Jilin          China 43.6661 126.1923       1
## 69       Liaoning          China 41.2956 122.6085       2
## 70          Macau          China 22.1667 113.5500       0
## 71        Ningxia          China 37.2692 106.1655       0
## 72        Qinghai          China 35.7452  95.9956       0
## 73        Shaanxi          China 35.1917 108.8701       3
## 74       Shandong          China 36.3427 118.1498       7
## 75       Shanghai          China 31.2020 121.4491       7
## 76         Shanxi          China 37.5777 112.2922       0
## 77        Sichuan          China 30.6171 102.7103       3
## 78        Tianjin          China 39.3054 117.3230       3
## 79          Tibet          China 31.6927  88.0924       0
## 80       Xinjiang          China 41.1129  85.2401       3
## 81         Yunnan          China 24.9740 101.4870       2
## 82       Zhejiang          China 29.1832 120.0934       1
```

```
sum(confirmed[which(confirmed[,"Country.Region"]=="China"),ncol(confirmed)])
```

```
## [1] 83959
```

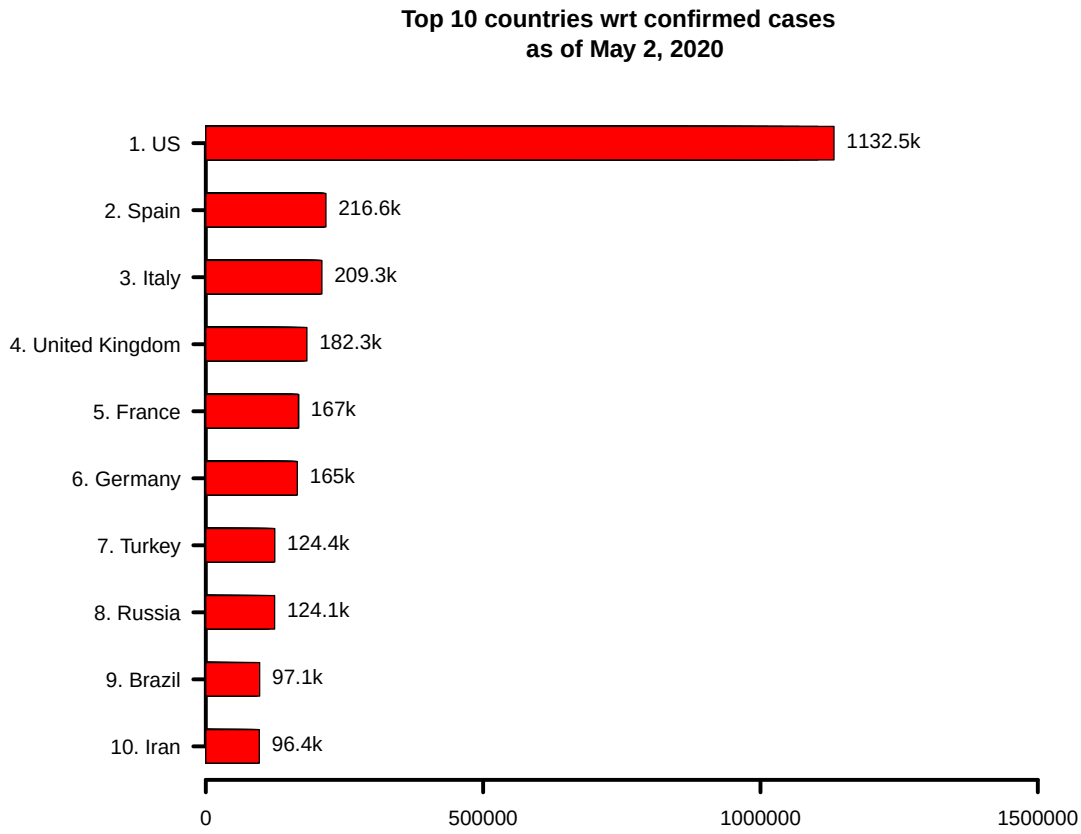
```
sum(deaths[which(deaths[,"Country.Region"]=="China"),ncol(deaths)])
```

```
## [1] 4637
```

3. Plot confirmed cases and reported deaths attributable to COVID-19

We now want to visualize the numbers of confirmed cases and reported deaths from COVID-19. We focus on the ten countries with the most cases or deaths so far.

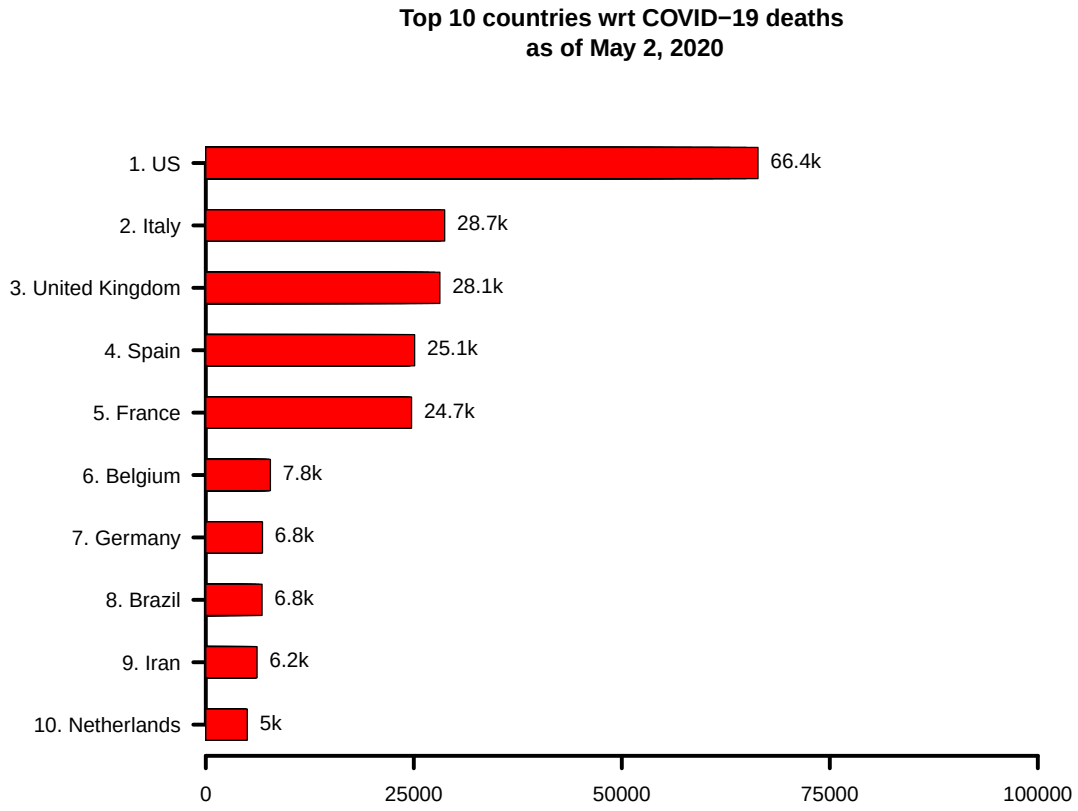
```
par(fig = c(0,1,0,1), las=1, mai=c(0.4,2.4,0.8,0.4))

plot(x=-100,y=-100,xlim=c(0,1500000),ylim=c(0,10),xlab="",ylab="",
    main="Top 10 countries wrt confirmed cases \n as of May 2, 2020",axes=FALSE)

country_labels <- c(0)
for(pop in 1:10){
    current_pop <- confirmed[order(confirmed[,ncol(confirmed)],decreasing=TRUE),][pop,1:2]
    if(!current_pop["Province.State"]==''){
        country_labels[pop] <- current_pop["Province.State"]
    }
    if(current_pop["Province.State"]==''){
        country_labels[pop] <- current_pop["Country.Region"]
    }
}

axis(side=1,at=seq(0,1500000,500000),labels=TRUE,lwd=3,pos=0)
axis(side=2,at=seq(0.5,9.5,1),
labels=paste(rev(seq(1,10,1)),". ",rev(country_labels),sep=""),lwd=3,pos=0)

for(pop in 1:10){
    rect(xleft=0,xright=confirmed[order(confirmed[,ncol(confirmed)],
    decreasing=TRUE),][pop,5:ncol(confirmed)],ybottom=9.25-1*(pop-1),
    ytop=9.25-1*(pop-1)+0.5,col="red")

    text(confirmed[order(confirmed[,ncol(confirmed)],decreasing=TRUE),][pop,ncol(confirmed)],
    9.25-1*(pop-1)+0.25,paste(round(confirmed[order(confirmed[,ncol(confirmed)],
    decreasing=TRUE),][pop,ncol(confirmed)]/1000,1),"k",sep=""),pos=4)
}
```



Please do the same for reported deaths.

```
par(fig = c(0,1,0,1), las=1, mai=c(0.4,2.4,1.2,0.4))

    plot(x=-100,y=-100,xlim=c(0,100000),ylim=c(0,10),xlab="",ylab="",
        main="Top 10 countries wrt COVID-19 deaths\n as of May 2, 2020",axes=FALSE)

    country_labels <- c(0)
    country_row_number <- c(NA)
    for(pop in 1:10){
        current_pop <- deaths[order(deaths[,ncol(deaths)],decreasing=TRUE),][pop,1:2]
        country_row_number[pop] <- rownames(current_pop)
        if(!current_pop["Province.State"]==''){
            country_labels[pop] <- current_pop["Province.State"]
        }
        if(current_pop["Province.State"]==''){
            country_labels[pop] <- current_pop["Country.Region"]
        }
    }

    axis(side=1,at=seq(0,100000,25000),labels=TRUE,lwd=3,pos=0)
    axis(side=2,at=seq(0.5,9.5,1),labels=paste(rev(seq(1,10,1)),". ",
    rev(country_labels),sep=""),lwd=3,pos=0)

    for(pop in 1:10){
        rect(xleft=0,xright=deaths[order(deaths[,ncol(deaths)],decreasing=TRUE),]
```

```
        [pop,5:ncol(deaths)],ybottom=9.25-1*(pop-1),ytop=9.25-1*(pop-1)+0.5,col="red")

        text(deaths[order(deaths[,ncol(deaths)],decreasing=TRUE),][pop,ncol(deaths)],
        9.25-1*(pop-1)+0.25,paste(round(deaths[order(deaths[,ncol(deaths)],
        decreasing=TRUE),][pop,ncol(deaths)]/1000,1),"k",sep=""),pos=4)
    }
```



Compare the ranking of the top ten countries with respect to most confirmed cases and reported deaths. What similarities and differences do you observe?

4. What would you like to investigate here?

There are so many things to explore. For example, how did the ranking of the top ten countries with respect to most confirmed cases and reported deaths change over time?

How reliable are confirmed cases and reported deaths from COVID-19?

How to get to know how many people are actually infected with COVID-19, and how many people are likely to die from COVID-19?

How helpful could be demographic forecasting tools in this matter?

Please read the paper on *A demographic scaling model for estimating the total number of COVID-19 infections* which is available on medrxiv at https://doi.org/10.1101/2020.04.23.20077719.