Introduction

I was able to blog during the year 2020 without mentioning the ongoing pandemic once. It's not that I made any conscious effort not to talk about it, but I did not really want to do something that had already been done a 1000 times. This changed this year, when I wanted to look at the spread of COVID-19, not only in the Grand-Duchy of Luxembourg, the country I live in, but also among our neighbours. You see, the Grand-Duchy of Luxembourg is like an island, but instead of being surrounded by water, it's surrounded by Belgians, Germans and Frenchmen. Many of them commute every day to Luxembourg to work, and even though they technically don't live inside the country, many aspects of their lives happen inside Luxembourguish borders. Their children might even come to school here, and sometimes they live so close by the border, that they can catch Luxembourguish public transportation in their towns. 200k commuters from Belgium, Germany and France work here every day. That's half our workforce! So that's why I thought that it would make sense to look at the spread of the disease at the level of the so-called *Greater Region*. This *Greater Region* is made up of the Grand-Duchy of Luxembourg, the Provinces of Liège and Luxembourg in Belgium (hence why I keep writing the *Grand-Duchy of* Luxembourg to refer to the country, and the *Province of Luxembourg* to refer to the Belgian province of the same name), and two German *Länders*, the Saarland and the Rhineland-Palatinate. Confused? Welcome to Europe, where supranational institutions literally have to have a page entitled Do not get confused so that citizens don't get lost (we still do).

So the Greater Region is not a state, but facilitates collaboration between the regions comprising it. To me, technically a citizen of the Greater Region, it feels like there was a want to **peacefully** correct for the randomness of history, where German-speaking regions ended up in both France and Belgium, and where Belgium and Luxembourg, well, somehow became independent countries.

Anyways, what I wanted to do was to first of all get the COVID-19 daily cases data for each of these regions. I did that, and even created a package called `{covidGrandeRegion}` hosted here that makes it very easy to download the latest data for the Greater Region. I will write another blog post about it, I have something in mind that I wanted to try for some time, and this was the first step. Then I thought that adding a function that would create a map could also be nice. And this is where the technical aspect of this blog post starts.

The problems to map the Greater Region

So how do you draw a map for an arbitrary landmass like the Greater Region? I wanted to draw the maps using `{echarts4r}`, and there's a very easy guide you can read. If you want to draw a map for one, or several, countries, this guide is all you need. But I wanted a map with only parts of France, Belgium and Germany. The only complete country was Luxembourg. So the first problem was how to get only parts of a country. The second problem, is that I had daily covid cases for the lowest administrative levels for France (which are *Départements*), Belgium (the *Provinces*) and Germany (*Land-* and *Stadtkreise*). But for the Grand-Duchy of Luxembourg, there's only data at the level of the country. So this would be another problem. How to draw a map with unequal levels of precision? One final problem: the names of the administrative divisions in my covid datasets are not the same than the ones that get downloaded if you follow the guide I linked before. So I had to rename them as well.

The solutions

Let's first start by following the guide, so loading the packages, and getting the maps I need:

```
library(echarts4r)
library(sp)
library(raster)
library(geojsonio)
france_dep <- getData("GADM", country = "FRANCE", level = 2)

ger_kreise <- getData("GADM", country = "GERMANY", level = 2)

be_province <- getData("GADM", country = "BELGIUM", level = 2)
```

The above lines of code load the required packages, and download the maps for France, Belgium and Germany with the required administrative level I need. I'll leave Luxembourg for last.

Let's take a look at what type of object we're dealing with:

```
class(france_dep)
## [1] "SpatialPolygonsDataFrame"
## attr(,"package")
## [1] "sp"
```

So it seems to be something like a data frame, but probably more complex. Looking for some help online, I saw that you can coerce it to a data frame:

```
as.data.frame(be_province)
##    GID_0  NAME_0   GID_1     NAME_1 NL_NAME_1     GID_2
NAME_2
## 1    BEL Belgium BEL.1_1  Bruxelles       BEL.1.1_1       Bruxelles
## 2    BEL Belgium BEL.2_1 Vlaanderen       BEL.2.1_1       Antwerpen
## 3    BEL Belgium BEL.2_1 Vlaanderen       BEL.2.2_1         Limburg
## 4    BEL Belgium BEL.2_1 Vlaanderen       BEL.2.3_1 Oost-Vlaanderen
## 5    BEL Belgium BEL.2_1 Vlaanderen       BEL.2.4_1  Vlaams Brabant
## 6    BEL Belgium BEL.2_1 Vlaanderen       BEL.2.5_1 West-Vlaanderen
## 7    BEL Belgium BEL.3_1   Wallonie       BEL.3.1_1  Brabant Wallon
## 8    BEL Belgium BEL.3_1   Wallonie       BEL.3.2_1         Hainaut
## 9    BEL Belgium BEL.3_1   Wallonie       BEL.3.3_1           Liège
## 10   BEL Belgium BEL.3_1   Wallonie       BEL.3.4_1      Luxembourg
## 11   BEL Belgium BEL.3_1   Wallonie       BEL.3.5_1           Namur
##
VARNAME_2
## 1  Brussel Hoofstadt|Brusselse Hoofdstedelijke
Gewest|Brüssel|Bruxelas|Région de Bruxelles-Capitale|Brussels|Bruselas
## 2
Amberes|Antuérpia|Antwerp|Anvers|Anversa
## 3
Limbourg|Limburgo
## 4                   Flandres Oriental|Fiandra Orientale|Flandes
Oriental|Flandre orientale|East Flanders|Ost Flandern
## 5                                                  Brabant
Flamand|Brabante Flamenco|Brabante Flamengo|Flemish Brabant
## 6           Fiandra Occidentale|Flandes Occidental|Flandre
occidentale|Flandres Ocidental|West Flandern|West Flanders
## 7
```

```
Waals Brabant|Walloon Brabant
## 8
Henegouwen|Hennegau
## 9
Luik|Liegi|Lieja|Lüttich
## 10
Lussemburgo|Luxemburg|Luxemburgo
## 11
Namen
##    NL_NAME_2                                TYPE_2      ENGTYPE_2
CC_2 HASC_2
## 1        Hoofdstedelijk Gewest|Région Capitale Capital Region
BE.BU
## 2                                      Provincie       Province
BE.AN
## 3                                      Provincie       Province
BE.LI
## 4                                      Provincie       Province
BE.OV
## 5                                      Provincie       Province
BE.VB
## 6                                      Provincie       Province
BE.WV
## 7                                       Province      Provincie
BE.BW
## 8                                       Province      Provincie
BE.HT
## 9                                       Province      Provincie
BE.LG
## 10                                      Province      Provincie
BE.LX
## 11                                      Province      Provincie
BE.NA
```

We're not going to convert them to data frames however; but this is an interesting clue; these `SpatialPolygonsDataFrame` objects share common methods with data frames. What this means is that we can use the usual, base R way of manipulating these objects.

So to get only the French *départements* I need, I can slice them like so:

```
lorraine <- france_dep[`%in%`(france_dep$NAME_2, c("Meurthe-et-
Moselle", "Meuse", "Moselle", "Vosges")),]
```

Same for the German *kreise*, here I select the *Länder* which are a higher administrative division than the Kreise, which makes it faster (so I don't need to type all the 40+ Kreise):

```
ger_kreise <- ger_kreise[`%in%`(ger_kreise$NAME_1, c("Rheinland-Pfalz",
"Saarland")),]
```

For Germany, many Kreise had a name which was different than on my covid data, so I had to rename them. So here again, the base R way of doing things works:

```
ger_kreise$NAME_2[ger_kreise$NAME_2 == "Eifelkreis Bitburg-Prüm"]  <-
```

```
"Bitburg-Prüm"
ger_kreise$NAME_2[ger_kreise$NAME_2 == "St. Wendel"]  <- "Sankt Wendel"
ger_kreise$NAME_2[ger_kreise$NAME_2 == "Altenkirchen (Westerwald)"]  <-
"Altenkirchen"
ger_kreise$NAME_2[ger_kreise$NAME_2 == "Neustadt an der Weinstraße"]
<- "Neustadt a.d.Weinstraße"
ger_kreise$NAME_2[ger_kreise$NAME_2 == "Landau in der Pfalz"]  <-
"Landau i.d.Pfalz"
ger_kreise$NAME_2[ger_kreise$NAME_2 == "Ludwigshafen am Rhein"]  <-
"Ludwigshafen"
ger_kreise$NAME_2[ger_kreise$NAME_2 == "Frankenthal (Pfalz)"]  <-
"Frankenthal"
```

Finally, I do the same for Belgium, and rename their province of Luxembourg, which was simply called “Luxembourg”, to “Province de Luxembourg”:

```
be_wallonia <- be_province[be_province$NAME_1 == "Wallonie", ]
be_wallonia$NAME_2[be_wallonia$NAME_2 == "Luxembourg"]  <- "Province de
Luxembourg"
```

I rename the province because the Grand-Duchy of Luxembourg is also only called “Luxembourg” in the data, and this would cause issues when mapping.

Now, comes Luxembourg. As I’ve written above, I only have data at the level of the country, so I download the country map:

```
lu_map_0 <- getData("GADM", country = "LUXEMBOURG", level = 0)
```

Let’s also see how it looks like as a data frame:

```
as.data.frame(lu_map_0)
##   GID_0     NAME_0
## 1   LUX Luxembourg
```

Unlike the previous `SpatialPolygonsDataFrame`s, there are much less columns and this will cause an issue. Indeed, in order to have a single `SpatialPolygonsDataFrame` object to draw my map, I will need to combine them. This will be very easy, by simple using the `rbind()` function. Again, simply using base R functions. However, this only works if the data frames have the same columns. Another issue, is that I will be using the names of the regions which are in the `SpatialPolygonsDataFrame`s’ column called `NAME_2`, but for Luxembourg, the name of the region (in this case the whole country) is in the column called `NAME_0`. So I need to add this columns to the `SpatialPolygonsDataFrame` object for Luxembourg:

```
lu_map_0$GID_1 <- NA
lu_map_0$NAME_1 <- NA
lu_map_0$NL_NAME_1 <- NA
lu_map_0$GID_2 <- NA
lu_map_0$NAME_2 <- "Luxembourg"
lu_map_0$VARNAME_2 <- NA
lu_map_0$NL_NAME_2 <- NA
lu_map_0$TYPE_2 <- NA
lu_map_0$ENGTYPE_2 <- NA
lu_map_0$CC_2 <- NA
lu_map_0$HASC_2 <- NA
```

Aaaand… that's it! Wasn't that hard, but a bit convoluted nonetheless. Now I can bind all the `SpatialPolygonsDataFrame` objects in one and use that for mapping:

```
grande_region <- do.call(rbind, list(lorraine, ger_kreise, be_wallonia,
lu_map_0))

as.data.frame(grande_region)
##     GID_0     NAME_0    GID_1          NAME_1 NL_NAME_1       GID_2
## 76    FRA     France  FRA.6_1       Grand Est         FRA.6.7_1
## 77    FRA     France  FRA.6_1       Grand Est         FRA.6.8_1
## 78    FRA     France  FRA.6_1       Grand Est         FRA.6.9_1
## 70    FRA     France  FRA.6_1       Grand Est        FRA.6.10_1
## 99    DEU    Germany DEU.11_1 Rheinland-Pfalz        DEU.11.1_1
## 110   DEU    Germany DEU.11_1 Rheinland-Pfalz        DEU.11.2_1
## 121   DEU    Germany DEU.11_1 Rheinland-Pfalz        DEU.11.3_1
## 129   DEU    Germany DEU.11_1 Rheinland-Pfalz        DEU.11.4_1
## 130   DEU    Germany DEU.11_1 Rheinland-Pfalz        DEU.11.5_1
## 131   DEU    Germany DEU.11_1 Rheinland-Pfalz        DEU.11.6_1
## 132   DEU    Germany DEU.11_1 Rheinland-Pfalz        DEU.11.7_1
## 133   DEU    Germany DEU.11_1 Rheinland-Pfalz        DEU.11.8_1
## 134   DEU    Germany DEU.11_1 Rheinland-Pfalz        DEU.11.9_1
## 100   DEU    Germany DEU.11_1 Rheinland-Pfalz       DEU.11.10_1
## 101   DEU    Germany DEU.11_1 Rheinland-Pfalz       DEU.11.11_1
## 102   DEU    Germany DEU.11_1 Rheinland-Pfalz       DEU.11.12_1
## 104   DEU    Germany DEU.11_1 Rheinland-Pfalz       DEU.11.14_1
## 103   DEU    Germany DEU.11_1 Rheinland-Pfalz       DEU.11.13_1
## 105   DEU    Germany DEU.11_1 Rheinland-Pfalz       DEU.11.15_1
## 106   DEU    Germany DEU.11_1 Rheinland-Pfalz       DEU.11.16_1
## 107   DEU    Germany DEU.11_1 Rheinland-Pfalz       DEU.11.17_1
## 108   DEU    Germany DEU.11_1 Rheinland-Pfalz       DEU.11.18_1
## 111   DEU    Germany DEU.11_1 Rheinland-Pfalz       DEU.11.20_1
## 109   DEU    Germany DEU.11_1 Rheinland-Pfalz       DEU.11.19_1
## 112   DEU    Germany DEU.11_1 Rheinland-Pfalz       DEU.11.21_1
## 113   DEU    Germany DEU.11_1 Rheinland-Pfalz       DEU.11.22_1
## 114   DEU    Germany DEU.11_1 Rheinland-Pfalz       DEU.11.23_1
## 115   DEU    Germany DEU.11_1 Rheinland-Pfalz       DEU.11.24_1
## 116   DEU    Germany DEU.11_1 Rheinland-Pfalz       DEU.11.25_1
## 117   DEU    Germany DEU.11_1 Rheinland-Pfalz       DEU.11.26_1
## 118   DEU    Germany DEU.11_1 Rheinland-Pfalz       DEU.11.27_1
## 119   DEU    Germany DEU.11_1 Rheinland-Pfalz       DEU.11.28_1
## 120   DEU    Germany DEU.11_1 Rheinland-Pfalz       DEU.11.29_1
## 122   DEU    Germany DEU.11_1 Rheinland-Pfalz       DEU.11.30_1
## 124   DEU    Germany DEU.11_1 Rheinland-Pfalz       DEU.11.32_1
## 123   DEU    Germany DEU.11_1 Rheinland-Pfalz       DEU.11.31_1
## 125   DEU    Germany DEU.11_1 Rheinland-Pfalz       DEU.11.33_1
## 126   DEU    Germany DEU.11_1 Rheinland-Pfalz       DEU.11.34_1
## 127   DEU    Germany DEU.11_1 Rheinland-Pfalz       DEU.11.35_1
## 128   DEU    Germany DEU.11_1 Rheinland-Pfalz       DEU.11.36_1
## 135   DEU    Germany DEU.12_1        Saarland        DEU.12.1_1
## 136   DEU    Germany DEU.12_1        Saarland        DEU.12.2_1
## 137   DEU    Germany DEU.12_1        Saarland        DEU.12.3_1
## 138   DEU    Germany DEU.12_1        Saarland        DEU.12.4_1
```

```
## 139   DEU    Germany DEU.12_1        Saarland        DEU.12.5_1
## 140   DEU    Germany DEU.12_1        Saarland        DEU.12.6_1
## 7     BEL    Belgium  BEL.3_1        Wallonie         BEL.3.1_1
## 8     BEL    Belgium  BEL.3_1        Wallonie         BEL.3.2_1
## 9     BEL    Belgium  BEL.3_1        Wallonie         BEL.3.3_1
## 10    BEL    Belgium  BEL.3_1        Wallonie         BEL.3.4_1
## 11    BEL    Belgium  BEL.3_1        Wallonie         BEL.3.5_1
## 1     LUX Luxembourg
##                                 NAME_2
VARNAME_2
## 76                 Meurthe-et-Moselle
## 77                              Meuse
## 78                            Moselle
Lothringen
## 70                              Vosges
## 99                            Ahrweiler
## 110                       Altenkirchen
## 121                        Alzey-Worms
## 129                       Bad Dürkheim
## 130                      Bad Kreuznach
## 131                Bernkastel-Wittlich
## 132                         Birkenfeld
## 133                        Cochem-Zell
## 134                    Donnersbergkreis
## 100                        Bitburg-Prüm
## 101                         Frankenthal
## 102                         Germersheim
## 104                       Kaiserslautern
## 103 Kaiserslautern (Kreisfreie Stadt)
## 105                              Koblenz
## 106                                Kusel
## 107                    Landau i.d.Pfalz
## 108                        Ludwigshafen
## 111                                Mainz
## 109                       Mainz-Bingen
## 112                      Mayen-Koblenz
## 113            Neustadt a.d.Weinstraße
## 114                              Neuwied
## 115                            Pirmasens
## 116                Rhein-Hunsrück-Kreis
## 117                    Rhein-Lahn-Kreis
## 118                   Rhein-Pfalz-Kreis
## 119                              Speyer
## 120                 Südliche Weinstraße
## 122                        Südwestpfalz
## 124                              Trier
## 123                      Trier-Saarburg
## 125                         Vulkaneifel
## 126                     Westerwaldkreis
## 127                              Worms
## 128                         Zweibrücken
## 135                       Merzig-Wadern
```

```
## 136                       Neunkirchen
## 137       Regionalverband Saarbrücken
## 138                         Saarlouis
## 139                   Saarpfalz-Kreis
## 140                      Sankt Wendel
## 7                     Brabant Wallon    Waals Brabant|Walloon
Brabant
## 8                              Hainaut
Henegouwen|Hennegau
## 9                                Liège
Luik|Liegi|Lieja|Lüttich
## 10             Province de Luxembourg Lussemburgo|Luxemburg|
Luxemburgo
## 11                               Namur
Namen
## 1                           Luxembourg
##     NL_NAME_2           TYPE_2  ENGTYPE_2  CC_2   HASC_2
## 76             Département Department    54    FR.MM
## 77             Département Department    55    FR.MS
## 78             Département Department    57    FR.MO
## 70             Département Department    88    FR.VG
## 99               Landkreis   District 07131 DE.RP.AR
## 110              Landkreis   District 07132 DE.RP.AT
## 121              Landkreis   District 07331 DE.RP.AW
## 129              Landkreis   District 07332 DE.RP.BD
## 130              Landkreis   District 07133 DE.RP.BK
## 131              Landkreis   District 07231 DE.RP.BW
## 132              Landkreis   District 07134 DE.RP.BR
## 133              Landkreis   District 07135 DE.RP.CZ
## 134              Landkreis   District 07333 DE.RP.DN
## 100              Landkreis   District 07232 DE.RP.EB
## 101       Kreisfreie Stadt   District 07311 DE.RP.FA
## 102              Landkreis   District 07334 DE.RP.GR
## 104              Landkreis   District 07335 DE.RP.KL
## 103       Kreisfreie Stadt   District 07312 DE.RP.KL
## 105       Kreisfreie Stadt   District 07111 DE.RP.KO
## 106              Landkreis   District 07336 DE.RP.KU
## 107       Kreisfreie Stadt   District 07313 DE.RP.LP
## 108       Kreisfreie Stadt   District 07314 DE.RP.LR
## 111       Kreisfreie Stadt   District 07315 DE.RP.MI
## 109              Landkreis   District 07339 DE.RP.MB
## 112              Landkreis   District 07137 DE.RP.MK
## 113       Kreisfreie Stadt   District 07316 DE.RP.NW
## 114              Landkreis   District 07138 DE.RP.NU
## 115       Kreisfreie Stadt   District 07317 DE.RP.PR
## 116              Landkreis   District 07140 DE.RP.RH
## 117              Landkreis   District 07141 DE.RP.RN
## 118              Landkreis   District 07338 DE.RP.RZ
## 119       Kreisfreie Stadt   District 07318 DE.RP.SE
## 120              Landkreis   District 07337 DE.RP.SW
## 122              Landkreis   District 07340 DE.RP.SD
```

```
## 124       Kreisfreie Stadt   District 07211 DE.RP.TI
## 123              Landkreis   District 07235 DE.RP.TS
## 125              Landkreis   District 07233 DE.RP.VL
## 126              Landkreis   District 07143 DE.RP.WS
## 127       Kreisfreie Stadt   District 07319 DE.RP.WR
## 128       Kreisfreie Stadt   District 07320 DE.RP.ZE
## 135              Landkreis   District 10042 DE.SL.MW
## 136              Landkreis   District 10043 DE.SL.NU
## 137              Landkreis   District 10041 DE.SL.SB
## 138              Landkreis   District 10044 DE.SL.SA
## 139              Landkreis   District 10045 DE.SL.SP
## 140              Landkreis   District 10046 DE.SL.SW
## 7                 Province  Provincie      BE.BW
## 8                 Province  Provincie      BE.HT
## 9                 Province  Provincie      BE.LG
## 10                Province  Provincie      BE.LX
## 11                Province  Provincie      BE.NA
## 1
```

And now I can continue following the tutorial from the `{echarts4r}` website, by converting this `SpatialPolygonsDataFrame` object for the Greater Region into a geojson file which can now be used to draw maps! You can take a look at the final result here.

I don't post the code to draw the map here, because it would require some more tinkering by joining the COVID data. But you can find my raw script here (lines 51 to 61) or you could also take a look at the `draw_map()` function from the package I made, which you can find here.

I really like the end result, `{echarts4r}` is really a fantastic package! Stay tuned part 2 of the project, which will deal with machine learning.