Covid 19 - some Challenges - some Data

Introduction

This lab has been written in the first week of April 2020.

In this lab we will investigate some of the data which can help inform issues around the Covid 19 Pandemic.

Worldwide there is a huge effort being undertaken by specialists of all colours to understand and reduce the threat of this pandemic and of course there aare huge efforts undertaken by many specialists to help us all live under these new conditions.

Here we will illuminate some of the challenges and questions to which people with data skills can contribute.

There are a number of great place to start any such journey:

- https://coronavirus.jhu.edu/ at the John Hopkins University. There you can find a collation of data but also articles on the topic. have to achieve the following tasks/learning outcomes:
- https://ourworldindata.org/coronavirus has a dedicated Covid-19 page where they review some of the latest numbers. A particularly interesting element of this page is that they provide a discussion of why they use the daily updates provided by the https://www.ecdc.europa.eu/en/geographical-distribution-2019-ncov-cases, rather than other sources. This is a particularly good case of careful datawork.
- The data competition site Kaggle has a dedicated https://www.kaggle.com/covid19 section of challenges.

In fact we start with the latter site. At the time of writing the structure their data challenges into three categories. “Use Natural Language Processing to answer key questions from the scientific literature”, “Forecast COVID-19 cases and fatalities to help understand what drives transmission rates”, “Curate COVID-19 related datasets to further research” and “Use exploratory analysis to answer research questions that support frontline responders”.

Exploring the latter of these you will find that 12 particular tasks were posed:

- Which populations are at risk of contracting COVID-19?
- What is the incidence of infection with coronavirus among cancer patients?
- Which patient populations pass away from COVID-19?
- Are hospital resources being diverted from providing oncology care to support the COVID-19 response?
- How is the implementation of existing strategies affecting the rates of COVID-19 infection?
- Which populations have contracted COVID-19 and require ventilators?
- Which populations have contracted COVID-19 who require the ICU?
- What is the change in turnaround time for routine lab values for oncology patients?
- Which populations of clinicians are most likely to contract COVID-19?
- Which populations assessed should stay home and which should see an HCP?
- Which populations of clinicians and patients require protective equipment?
- How are patterns of care changing for current patients (i.e. cancer patients)?

Quoting from the Kaggle website: “The tasks associated with this dataset were developed and evaluated by global frontline healthcare providers, hospitals, suppliers, and policy makers. They represent key research questions where insights developed by the Kaggle community can be most impactful in the areas of at-risk population evaluation and capacity management.”

Some exploratory data analysis

Let's do some exploratory analysis using a dataset published by the ECDC. This dataset is used by the Our World in Data page and is also part of the dataset in the Kaggle challenge. ## Preparing your workfile

We add the basic libraries needed for this week's work:

```
library(tidyverse)    # for almost all data handling tasks
library(ggplot2)      # to produce nice graphiscs
library(stargazer)    # to produce nice results tables
library(AER)          # access to HS robust standard errors
library(readxl)       # enable the read_excel function
```

Data Upload

Very helpfully the ECDC webpage through which you can https://www.ecdc.europa.eu/en/publications-data/download-todays-data-geographic-distribution-covid-19-cases-worldwide provides an R script which allows you to download the most current dataset. This is the script replicated in the next code block. You could of course download the datset to your computer and then upload the excel or csv file, but here the ECDC has build a direct pipeline into their data.

```
#these libraries need to be loaded
library(utils)
library(httr)

#download the dataset from the ECDC website to a local temporary file
GET("https://opendata.ecdc.europa.eu/covid19/casedistribution/csv", authenticate(":", ":", type="ntlm")
```

```
## Response [https://opendata.ecdc.europa.eu/covid19/casedistribution/csv/]
##   Date: 2020-04-04 20:29
##   Status: 200
##   Content-Type: application/octet-stream
##   Size: 447 kB
## <ON DISK>  C:\Users\msassrb2\AppData\Local\Temp\Rtmp2HVWZt\file7c05d1c7c.csv
```

```
#read the Dataset sheet into "R". The dataset will be called "data".
data <- read.csv(tf)
```

Some data cleaning

Let's look at the structure of this dataset. We wnat to make sure we understand all the variables and give them sensible names we want to work with.

```
str(data)
```

```
## 'data.frame':    8704 obs. of  10 variables:
##  $ dateRep               : Factor w/ 96 levels "01/01/2020","01/02/2020",..: 16 12 8 4 95 93 91 88 8
##  $ day                   : int  4 3 2 1 31 30 29 28 27 26 ...
##  $ month                 : int  4 4 4 4 3 3 3 3 3 3 ...
##  $ year                  : int  2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ...
##  $ cases                 : int  0 43 26 25 27 8 15 16 0 33 ...
##  $ deaths                : int  0 0 0 0 0 1 1 1 0 0 ...
##  $ countriesAndTerritories: Factor w/ 204 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ geoId                 : Factor w/ 203 levels "AD","AE","AF",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ countryterritoryCode  : Factor w/ 201 levels "","ABW","AFG",..: 3 3 3 3 3 3 3 3 3 3 ...
```

```
##  $ popData2018             : int  37172386 37172386 37172386 37172386 37172386 37172386 37172386 3717
```

Some of the variables have obvious meaning, sich as `day`, `month`, `year`, `countriesAndTerritories` and `popData2018`, the latter giving the population of the respective country in 2018. `geoId` and `countryterritoryCode` are common appreviations for the respective country.

For starters we want to shorten the name of `countriesAndTerritories` to `country` and `countryterritoryCode` to `countryCode` and `dateRep` to `dates`.

```
names(data)[names(data) == "countriesAndTerritories"] <- "country"
names(data)[names(data) == "countryterritoryCode"] <- "countryCode"
names(data)[names(data) == "dateRep"] <- "dates"
```

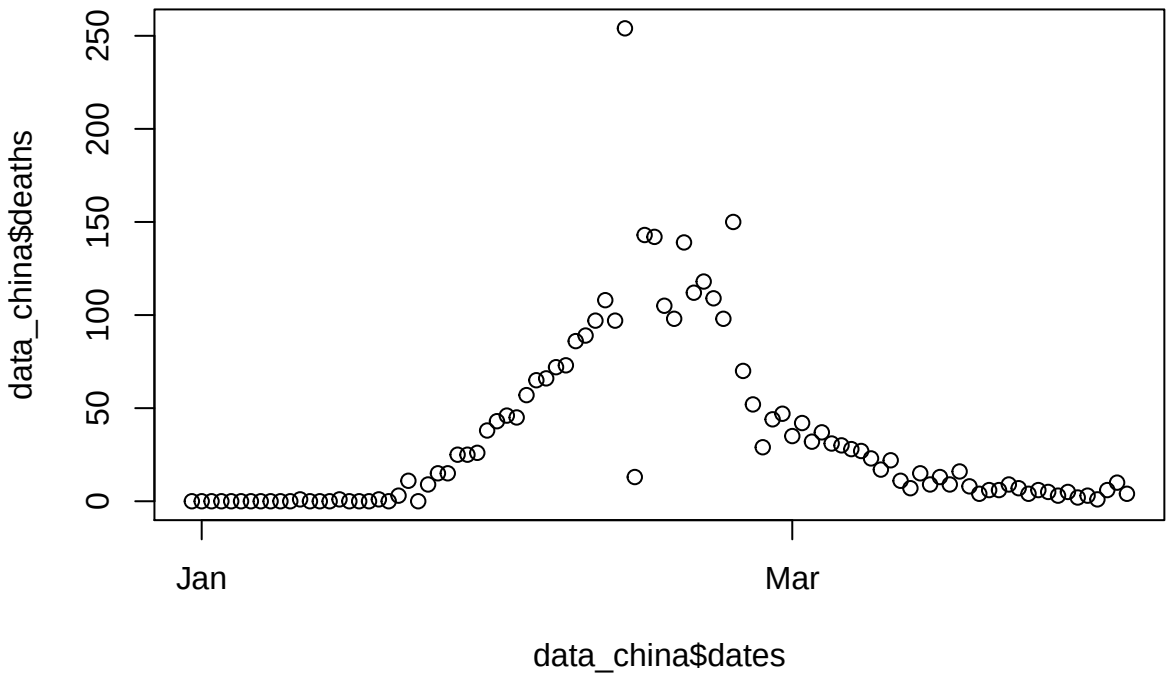
The first variable is a factor variable which includes the entire date string. At this stage R does not recognise this as a date. So let's change this as it will be useful for R to know that this variable represents a date. Dates are of the format 24/01/2020. In the `format` option we specify this date such that R knows how to translate to dates (see https://www.stat.berkeley.edu/~s133/dates.html to understand how to specify the format string).

```
data$dates <- as.Date(as.character(data$dates),format = "%d/%m/%Y")
```

Last, but most importantly, there are two variables called `cases` and `deaths`. These are daily data. So without further explanation it is not obvious whether these are the cumulative data (e.g. all the Covid-19 cases which have been identified in a country by a certain day, or whether these are the cases identified on that particular day). YOu could either go back to the data source to find an explanation orwe could use our understanding of the problem at hand. The two series should look very different.

To investigate this let's look at one country in particular, say China, where this particular virus was first identified. We use the plot function which provides the standard build-in R plotting facility. Later we will look at using `ggplot` to produce nicer plots.

```
data_china <- data %>%  filter(country == "China")
plot(data_china$dates,data_china$deaths)  # specifies variable for x and y axis
```



We can clearly see that after an increase in the numbers of fatalities in China in January and February we see a decrease in numbers in March. If these were cumulaive numbers we would not see a decrease. So these are the deaths which occured on a particular day. You can find the same for cases.

Before continuing we may also want to highlight a particularity in these data. YOu can see that in the middle of March there is one day (13 Feb 2020) on which almost 100 more deaths have been reported than at any other day. And in fact, the day before tehre were only 13 reported deaths. It turns out that these irregularities are the result of changes in data definitions as reported, for instance, in this https://www.cnbc.com/2020/02/26/confusion-breeds-distrust-china-keeps-changing-how-it-counts-coronavirus-cases.html.

Knowing that we are talking about daily statistics for new confirmed infections and daily deaths, we may also want to calculate the accumulated infections and deaths. This is achieved with the `cumsum` (cumulative sum) command. This takes a vector of data and keeps adding these up.

To illustrate what this command does, let's use an example.

```
test <- c(0,0,2,4,9,2)
cumsum(test)
```

```
## [1]  0  0  2  6 15 17
```

Before we apply this to our data in `data`, we need to make sure that we only accumulate by country (`group_by(country)`) and that the data are arranged by date (`arrange(dates)`) before we apply the `cumsum` function.

```
data <- data %>% group_by(country) %>%
          arrange(dates) %>%
          mutate(c_cases = cumsum(cases), c_deaths = cumsum(deaths))
```

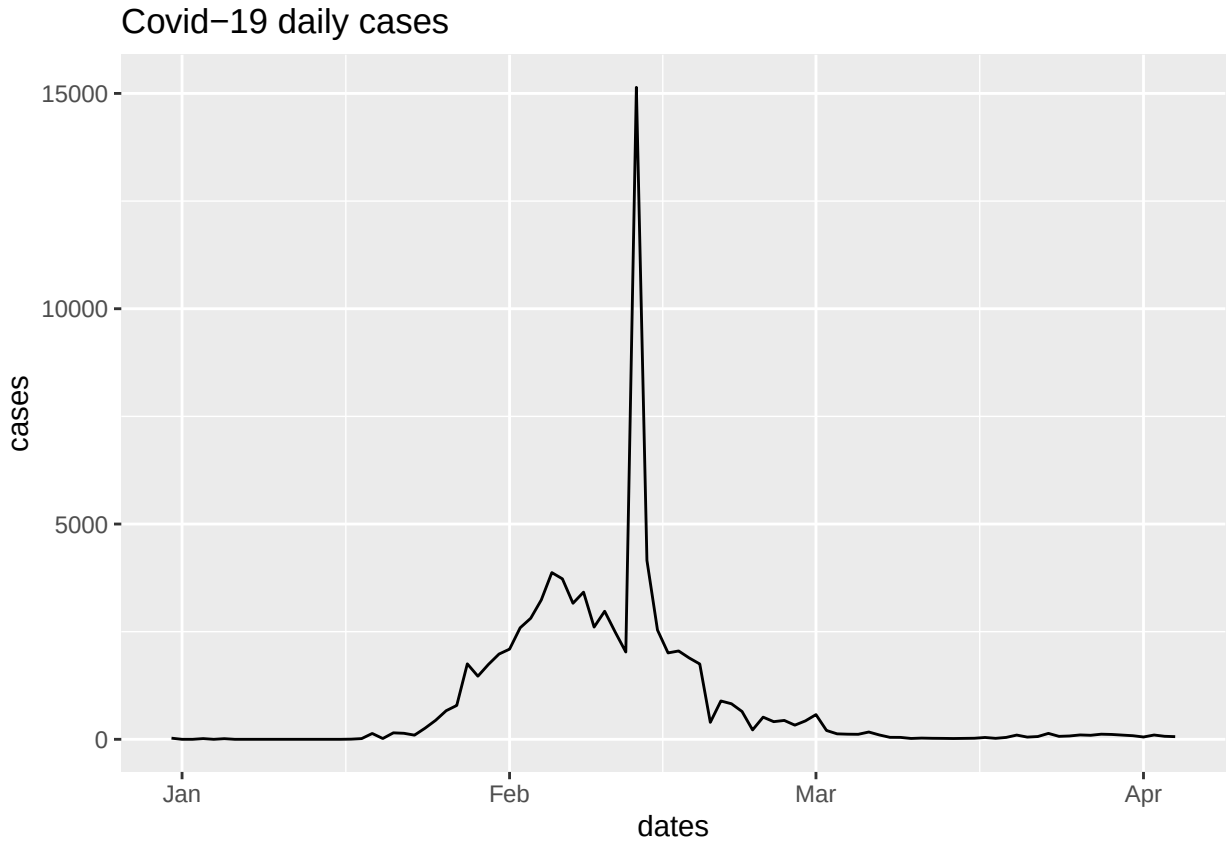
In the variables `c_cases` and `c_deaths` we now have the accumulated cases and deaths by country.

Some country graphs

Let's create some nicer graphs to describe the development of the pandemic in different countries. Let's first continue with the Chinese data.

First we replicate the above Figure but using the `ggplot` function which produces much nicer graphs.

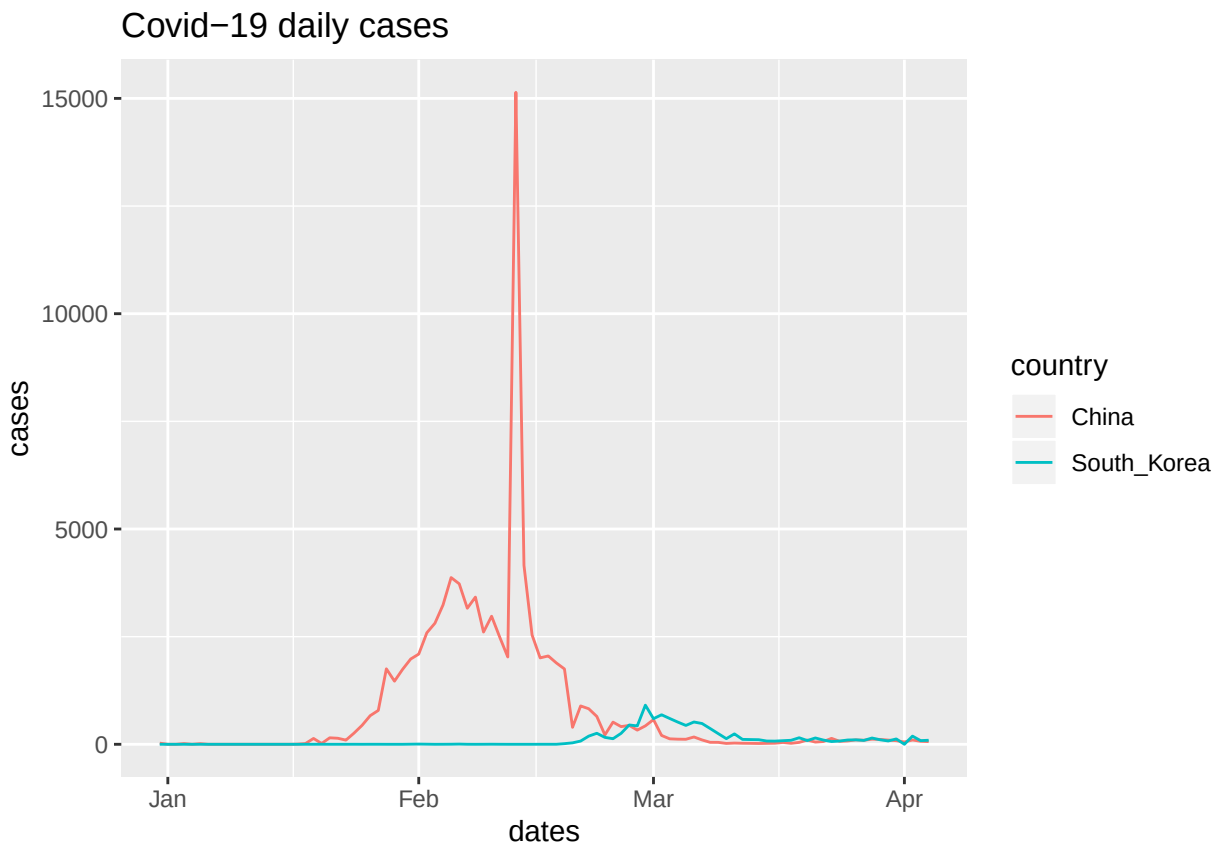
```
g1 <- ggplot(subset(data, country == "China"), aes(x=dates,y=cases)) +
      geom_line() +
      ggtitle("Covid-19 daily cases")
g1
```



As you can see we didn't create a special Chinese dataset first, but we used the subset function instead.

Let's overlay the daily cases for two countries, say China and South Korea.

```
sel_countries <- c("China", "South_Korea")
g2 <- ggplot(subset(data, country %in% sel_countries),
             aes(x=dates,y=cases, color = country)) +
      geom_line() +
      ggtitle("Covid-19 daily cases")
g2
```



Here you can see the much praised ability by South Korea to suppress the numbers of infections effectively. However, you might argue that

> HOw do these look like if you were to look at per capita infection rates?

Some country comparisons

Let's calculate some country wide statistics