Engineering the Future

Sample Report

Created by Diego Velandia
Last compile date: 14/02/2022

Code is available for free access through: https://github.com/Dialvec/r_shiny_latex_report # Introduction Sample datasets used as test subject for the present report describes evolution of confirmed positive cases, deceases and recovered patients along the current Covid-19 epidemics. Datasets are publicly available thanks to the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University.

Datasets come in separate .csv documents and they can be downloaded through CSSE public repository on github.

- Positive cases
- Confirmed deceases
- Recovered patients

R Markdown file is written in a way that users do not have to scroll into dense monolithic code to search for a specific code snippet to modify. Instead, individual file management allow users to quickly navigate, understand, modify or even change the coding structure to make a easy to maintain reporting tool.

```
source("RFunctions/DataSource.R")

df_positives_raw = get_positive_cases_df()
df_deceases_raw  = get_deceases_cases_df()
df_recovered_raw = get_recovered_cases_df()
```

R's source() function allows remote calling of functions, parameters and from .R and .Rmd

Data Exploration

Each dataset comes from a separate source. Therefore we begin by exploring the structure of each dataset.

Shape and variable type

A simple function to describe the size of the data frame can tell us valuable information regarding how to analize and understand the data frame structure.\

Positive Cases

```
## row count: 282
## column count: 758
##  -  character :  2
##  -  integer :  754
##  -  numeric :  2
```

Deceased Cases

```
## row count: 282
## column count: 755
##  -  character :  2
##  -  integer :  751
##  -  numeric :  2
```

Recovered Cases

```
## row count: 267
## column count: 755
##  -  character :  2
##  -  integer :  751
##  -  numeric :  2
```

I can be evidenced that all Datasets contain an amount of columns that can not be easily displayed. Using pre built descriptive methods like summary(df) might be overwhelming due to the high amount of columns in each data frame. Let's begin with the less common data types within the datasets.\

Character and numeric type columns

Since all Datasets share a common property of having 2 character and 2 numeric columns, it is possible that all Datasets have the same 4 columns (fields). We can verify these two columns on each case to verify the hypothesis and, then, start exploring the integer type columns.

```
## [1] "From positive cases dataset"

## columns of type: character
## Province.State
## Country.Region

## [1] "----------------------------"

## [1] "From deceases cases dataset"

## columns of type: character
## Province.State
## Country.Region

## [1] "----------------------------"

## [1] "From recovered cases dataset"

## columns of type: character
## Province.State
## Country.Region
```

Similar scenario can be expected regarding the numeric type variables.

```
## [1] "From positive cases dataset"
```

```
## columns of type: numeric
## Lat
## Long
```

```
## [1] "--------------------------"
```

```
## [1] "From deceases cases dataset"
```

```
## columns of type: numeric
## Lat
## Long
```

```
## [1] "--------------------------"
```

```
## [1] "From recovered cases dataset"
```

```
## columns of type: numeric
## Lat
## Long
```

From this structure we have two valuable insights: Geographic identifiers are suitable candidates as key for data joining; and time series are stored column-wise for each country and state, which are represented over rows.

Data completion

To verify data completion, we can use a function to verify either if a data frame column contains null values or empty strings. Keep in mind that empty strings are not considered as **na** values.

Positive Cases

```
##                 na_count
## Province.State       195
## Lat                    2
## Long                   2
```

Deceased Cases

```
##                 na_count
## Province.State       195
## Lat                    2
## Long                   2
```

Recovered Cases

```
##                 na_count
## Province.State       196
## Lat                    1
## Long                   1
```

As far as we have obtained insights from the data, **Province.State** is not available for all countries, which adds an extra challenge of analyze and properly aggregate countries with States or Provinces in order to standardize the geographic basis.

*Countries with **Province.State** variable not empty*

```
##         positives       deceases      recovered
## 1       Australia      Australia      Australia
## 2          Canada         Canada          China
## 3           China          China        Denmark
## 4         Denmark        Denmark         France
## 5          France         France    Netherlands
## 6     Netherlands    Netherlands    New Zealand
## 7     New Zealand    New Zealand United Kingdom
## 8 United Kingdom United Kingdom      Australia
```

It is visible that all data frames share almost the same list of countries divided by State. Recovered cases data frame does not include Canada as a state-divided country.This requires a deeper understanding of the State structure for each country before merging them as a single country.

Exploratory plots