Práctica 2: Analítica de datos sobre Covid-19

Patricia García Suarez[*] Adrián Arnaiz-Rodríguez[**]

9/6/2020

Índice

[*]Perfil Github:, https://github.com/Kadatashi
[**]Perfil Github:, https://github.com/AdrianArnaiz/

[1]

[1]Bibliogrfía al final del documento

1. Enlances de interés

Repositorio de Github: https://github.com/AdrianArnaiz/scrap_uoc

DOI de Zenodo (Base de datos):

- Versión 1.0 (datos del 30 de Marzo al 10 de Abril): 10.5281/zenodo.3748050.
- Versión 1.1 (datos del 30 de Marzo al 4 de Mayo): 10.5281/zenodo.3784400.
- Version final

Link a Zenodo: https://zenodo.org/record/3748050#.XpD5w8gzZ9A

2. Importación de librerías

```
library(dplyr)
library(ggplot2)
library(knitr)
library(kableExtra)
```

3. To do's

To Do's

- Seguir pasos marcados en el enunciado
- Explicar bien que vamos a hacer y porque (todo el rollo de las Series temporales, autocorrelaciones)
- Crear el dataset, o los diferentes datasets según los análisis.
  - Sacar población, densidad (o extendion), continente y Marsh political risk index.
    - Mirar en eurostat. INSEE, Institut national de la statistique et des études économiques. Espérance de vie, 2013.
    - Otra fuente aqui
- Explicar bien el dataset
- Limpiar dataset (nulos, no creo que tenga sentido normalizar. De todas maneras podemos explicar que depende del análisis se normalizará o no. Así hacemos al principiosólo eliminación de nulos y en cada analisis decidimos si se normaliza o no)
  - Comprobar normalidad y homogeneidad de varianza: también cuando toque.
- Realizar cada analisis:
  - Contrastes hipótesis:
    - Contraste proporciones Esp-Ita, Esp-Port, Esp-ALe
    - Contraste anova entre continentes
  - Correlaciones:
    - Correlación entre proporcion de tests y proporcion de contagiados.
    - Correlacion entre variación de indice y numero de casos-O-fallecidos.
  - Regresión:
    - Time Series Forecasting: ARIMA
    - Proporción a x días ~ densidad + %gente mayor + etc

4. Introducción

4.1. Contexto

En el contexto de obtenciónd el dataset explicado en la anterior práctica, nos gustaría realizar un análisis de varios aspectos de la pandemia por **Covid-19**. En este contexto es importante analizar con rigor diferentes aspectos de la pandemia, para dar lugar a consluiones basaddas en esos análisis.

4.2. Análisis de datos sobre el COVID

To Do Primero de todo, nos gustaría explicar la **complicación de las series temporales a la hora de realizar diferentes contrastes de hipótesis, correlaciones o predicciones**.

4.2.1. Contrastes de hipótesis

Nos gustaría realizar contrastes de hipótesis sobre la diferencia de afectación entre países o continentes.

- La primera complicación esque las *series temporales son muestras con gran autocorrelación en sus datos*, en la que cada observación es muy dependiente de las observaciones temporales anteriores, tienen un orden. Esto rompe el principio de independencia entre observaciones necesario para realizar la mayoría de los análisis estadísticos. Otro problema son los datos a comparar. Es decir, normalmente tenemos una muestra de datos asumiendo en primer lugar independencia (y después normalidad, además de que si lo comparamos con otra muestra también suponemos igualdad de varianzas), por ejemplo, una muestra de pesos y alturas de bebés. No podemos tratar nuestra serie temporal como una muestra de pesos, ya que en nuestra serie temporal las observaciones a lo largo del tiempo no son independientes. Por ello, no podemos realizar un típico contraste de hipótesis, tenemos que buscar otro enfoque.

- Por otro lado, **para hacer un contraste de hipótesis se comparan distribuciones**. Si comparamos un dato de un dia concreto en el tiempo para dos países, son dos puntos individuales, que no tendrán varianza. Esto **lo resolvemos realizando contrastes de hipótesis sobre la proporción** Citar modulo estadistica. Consideramos el contagio o no como una variable que proviene de una distribución de Bernouilli con posibilidad de contagio p y posibilidad de no contagiarse de $1-p$. Por ello compararemos las proporciones de contagio de dos países.

- Cuando tomamos el dato de proporción de contagiados en un país, (i.e., si queremos hacer el contraste que acabamos de explicar para analizar si la proporción de contagiados en España e Italia se puede considerar igual o es diferente), no podemos considerar la serie temporal como la muestra y hacer la media, por que no tendría sentido (no tiene sentido relaizar la media de una serie temporal para ver la proporción de afectados). **Deberemos elegir un punto en el tiempo** para ver la proporción de contagios ese día. Es **muy importante que para hacer un análisis justo, no debemos elegir el mismo día para los dos países**. Deberemos elegir un dñia para cada país de tal modo que esa proporción muestre la misma estapa dentro de la pandemia, es decir, **debemos tener en cuenta cuando llegó el COVID a cada país**. Por ello utilizaremos la estrategia que utilizan diferentes analisa, como por ejempo el New York Times, se elegira el Día 30 después de contabilizar 100 muertes.

- Para ver la afectación por continentes, consideraremos diferentes muestras de los diferentes continentes. Cada continente tendra n proporciones, 1 de cada país que lo componga. Recordamos que la proporción de cada país ha sido obtenida como la proporción de contagiados el Día 30 después de contabilizar 100 muertes. De este modo, **de cada continente tendremos una media de proporciones con una desviación, lo que nos permitirá relizar un análisis ANOVA**.

4.2.2. Correlaciones

REPASAR, XQ PEARSON SÍ QUE SE PUEDE

https://stats.stackexchange.com/questions/133155/how-to-use-pearson-correlation-correctly-with-time-series

Como hemos comentado, una de las principales características que tienen las series temporales, sobre todo estas series derivadas de fenómenos epidemiológicos, es el alto grado de autocorrelación interna de sus datos que anula la hipótesis de independencia en las observaciones. Cuando nosotros realizamos contrastes de hipótesis o análisis de correlación (Pearson o Spearman), suponemos independencia en los datos, que no se cumple en las series temporales. Por ejemplo, no es lo mismo analizar la correlación entre peso y altura de bebés (cada altura es independiente a las demás) que entre la serie temporal de casos de COVID y de índices económicos (los casos de COVID de un día son muy dependientes de los de ayer, autocorrelación).

Por ello, utilizaremos datos 'estáticos'. Es decir, datos de un instante de tiempo para los diferentes países. Utilizzaremos la misma estrategia que venimos comentando, elegir la proporción de casos del país el textcolor{red}{día 30 después de contabilizar 100 muertes}.

Con ello, ya podremos realizar las siguientes correlaciones:

- Correlación entre proporción de contagiados (día 30 DC) y proporción de test realizados (dia 30 DC)

- Correlación entre proporción de contagiados (día 30 DC) y variación de los índices económicos.

Nuestro objetivo es realizar análisis de los siguientes puntos:

4.2.3. Regresión

La predicción de las series temporales del covid es quizá es el tema más candente, y está poniendo en vista las grandes dificultades en la predicción de evolución de fenómenos epidemiológicos. Los principales problemas que ocurren en modelos epidemiológicos es que siguen un modelo exponencial. El fenómeno de contagio se basa en sofisticaciones del modelo SIR (con base en exponenciales). En el paper de José Cuesta https://arxiv.org/pdf/2004.08842.pdf CITAR se llega a la conclusión de que estos modelos tienen mucha incertidumbre derivada de los parámetros elegidos, lo que da lugar a muchos escenarios diferentes. Tantos escenarios de confianza diferentes y sus intervalos de confianza, hace que no sea predecible de manera óptima los fenómenos epidemiológicos, dando innumerables escenarios sólo a 4 días vista.

Esta complejidad y dificultad hace que para nosotros intentar estimar el número de casos sea una tarea muy difícil.

Sin embargo, con objetivo de aplicar algñun modelo de *Time Series Forecasting* aplicaremos modelos de predicción utilizados en otras investigaciones, como el modelo autoregresivo ARIMA, y así ver cómo estima el modelo.

- ejemplo
- https://www.medrxiv.org/content/10.1101/2020.04.18.20070631v1.full.p df
- https://www.medrxiv.org/content/10.1101/2020.03.30.20047803v1.full.p df
- +ejemplos
- COVID-19: ARIMA based time-series analysis to forecastnear future

Por otro lado, intentaremos explicar la proporción de muertes basandonos en características sociodemográficas del país. Es decir, **intentaremos explicar la variable objetivo proporción de fallecidos a travñes de las variables explicativas densidad de población, porcentaje de mayores o Marsh Political Risk Index**.

4.2.4. Analisis que realizaremos

Por lo tanto, y resumiendo, los analisis a realizar serán los siguientes

- Contrastes hipótesis:
  - Contraste proporciones Esp-Ita, Esp-Port, Esp-ALe
  - Contraste anova entre continentes
- Correlaciones:
  - Correlación entre proporcion de tests y proporcion de contagiados.
  - Correlacion entre variación de indice y numero de casos-O-fallecidos.
- Regresión:
  - Time Series Forecasting: ARIMA
  - Proporción a x días ~ densidad + %gente mayor + etc

Como es de entender, no podemos realiar estos análsisis solo con los datos de la primera práctica (cubo de dato país fecha). En el siguiente apartado, describiremos los dos diferentes dataser que utilizaremos con el objetivo de realizar estos análisis.

4.3. Descripción del dataset

4.3.1. Descripción breve general del dataset

Nuestro dataset evolucionará con respecto al de la primera práctica. Tendremos dos datasets, uno de series temporales de los datos del COVID (el de la primera práctica) y otro con datos estáticos de cada país. El motivo de tener dos datasets lo explicaremos más adelante en la sección 1 y sobre todo en la sección 1. Por ello, a parte del cubo de datos País-Dato-Fecha, obtendremos datos de cada país de:

- Total de población de cada país
- Densidad de población
- Continente
- Porcentaje de población mayor

El dataset de la primera práctica tendrá la evolución temporal de 5 datos relativos al covid por países. Guardamos los datos relativos a **contagiados, casos activos, recuperados, muertes y tests realizados**. Es decir para cada uno de los países en los que haya casos registrados, guardamos un dato al día (de manera automática) cada uno de los datos recién enumerados. Al final, reflejamos la serie temporal de cada uno de esos datos por países. Por lo tanto, resultado de la anterior práctica, tenemos **5 csv**: la variación temporal de cada tipo de dato por país (ver Figura 1).

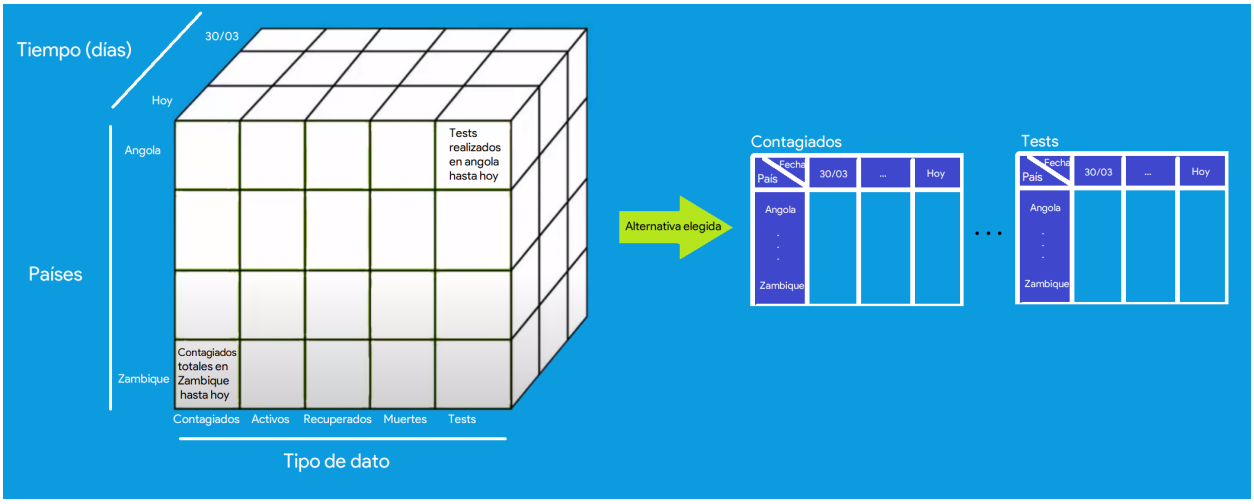

Figura 1: Representación gráfica

Por otro lado, tendremos para cada país la proporción de contagiados y fallecidos el día 30 después de contabilizar 100 muertes, acompañado del número total de la población, densidad de población, continente, porcentaje de población mayor, etc.

4.3.2. Dataset final

Tendremos **dos datasets**, cada uno de ellos usado para un diferente tipo de análisis.

- Por un lado tendremos el cubo de series temporales de los datos.
  - Lo utilizaremos para realizar Analisis de series temporales. Concretamente la regesión con ARIMA.
- Por otro lado, tendremos una tabla final con datos de los países. Le llamaremos dataset de datos estáticos.

- Lo utilizaremos para los contrastes de hipótesis, correlaciones y regresión de la proporción de muertes a través de características sociodemográficas.
- Lo explicamos más extensamente en la sección 4.2. Contiene los datos de un país de manera estática, es decir: los datos de contagiados o fallecidos el día x después de las 100 muertes, datos de densidad, población, vejez de la población del país, nivel de ingresos, etc.

País	Continente	Contag dia 40 DC	Fallec dia 40 DC	tests dia 40 DC	Densidad total	Población	%vejez	Nivel Ingresos ONU
España	Eu	n	n	n	n	n	n	st
Italia	Eu	n	n	n	n	n	n	st
Zambique	Af	n	n	n	n	n	n	st

4.4. Fuentes

Link a Worldometers - COVID-19

- Link a Worldometers - COVID-19: https://www.worldometers.info/coronavirus.
- Fuente población
- Fuente extension o densidad de poblacion
- Continente
- Nivel de ingresos del país

4.5. Cómo se ha recogido y fuentes

4.5.1. Cubo de datos Dato-Pais-Fecha

Se detalló en la anterior práctica la recogida de datos del cubo Dato-Pais-Fecha. Hicimos scrapping sobre la página de Worldometers-Coronavirus, en el script alojado en el directorio `src\Scraping_covid19.py`. En esa página tenemos **una tabla** que muestra los **valores de los datos (contagiados, activos, etc) por país en el momento actual**. Es decir, las **filas los países y las columnas los datos del momento actual**. Nuestro enfoque ha sido **automatizar el lanzamiento del scraping** para que se ejecute una vez al día y se vayan **actualizando automáticamente los csv de las series temporales de los datos por país**. Por ello, la primera fecha de la que tenemos datos es del 30/03, que fue el primer día que teníamos desarrollado el scraping y lo lanzamos. La herramienta **Travis** ha sido utilizada para automatizar el lanzamiento del script y el *autodeploy* a *Github* (*Travis* permite que, en su plataforma, una vez al día y de forma planificada y automática se ejecute el scrapping, se actualizan las tablas de datos y se haga un commit automático para actualizar los datos en el github).

- Link a (Worldometers - COVID-19)[https://www.worldometers.info/coronavirus]: https://www.worldometers.info/coronavirus.

4.5.2. Población y densidad de población

To Do También de worldometers.

4.5.3. Continente, pocentaje mayores de 65 y nivel de ingresos del pais

To Do

To Do https://data.worldbank.org/indicator/SP.POP.65UP.TO.ZS?end=2018&start=2017

csvs de datos y metadatos bajados de world bank. Son 2 ficheros csv que leeremos y les haremos join para tener los datos del país.

En el mismo csv que arriba, en los metadatos.

5. Integración y selección de datos

Como hemos comentado necesitamos dos datasets: El cubo de series temporales y el dataset de datos estáticos. El cubo de series temporales ya le tenemos realizado. Para **integrar los datos de diferentes origenes al dataset de datos estáticos realizaremos los siguientes pasos**:

1. Leer los dos CSV de **World Bank** (nuevos en esta práctica) para sacar los datos de continente, nivel de ingresos y porcentaje de población mayor de 65 años de cada país. Haremos un Join de ambos y nos quedaremos con los datos que queremos. Tendremos este DF_1.

País	Continente	%vejez	Nivel Ingresos ONU
España	Eu	n	st
Italia	Eu	n	st
Zambique	Af	n	st

2. Leer csv de series temporales de **Worldometers** (*cubo* de la primera práctica): leeremos 3 csv del *cubo* relativos a las series temporales de de contagios, fallecidos y tests. Sacamos el dato estático de cada país. Es decir, calculamos para cada país su día 40 despues de llegar la pandemia y obtenemos los datos de contagiados, fallecidos y tests ese día. Tendremos este DF_2.

País	Contagiados dia 40 DC	Fallecidos dia 40 DC	tests/1M dia 40 DC
España	n	n	n
Italia	n	n	n
Zambique	n	n	n

3. Leemos el csv de la población **Worldometers** (nuevo en esta práctica): cruzamos los datos de contagiados y fallecidos con población para poder sacar proporciones cuando se necesiten. Tendremos este DF_2.

País	Contag dia 40 DC	Fallec dia 40 DC	tests/1M dia 40 DC	Población	Densidad
España	n	n	n	n	n
Italia	n	n	n	n	n
Zambique	n	n	n	n	n

4. Hacemos join del dataset resultante de Worldometers y el de worldbank. Join de DF_1 y DF_2.

País	Continente	Contag dia 40 DC	Fallec dia 40 DC	tests/1M dia 40 DC	Densidad	Población	%vejez	Nivel Ingresos ONU
España	Eu	n	n	n	n	n	n	st
Italia	Eu	n	n	n	n	n	n	st
Zambique	Af	n	n	n	n	n	n	st

5.1. Paso 1 - Leer WorldBank y hacer Join

Leemos metadatos del país.

```
continente_ingresos <- read.csv(file="..\\csv\\WorldBankData\\Metadata_Country.csv")

continente_ingresos <- continente_ingresos %>% select("ï..Country.Code",
                                                      TableName,
                                                      Region,
                                                      IncomeGroup )

head(continente_ingresos)
```

```
##   ï..Country.Code   TableName                    Region         IncomeGroup
## 1             ABW       Aruba Latin America & Caribbean         High income
## 2             AFG Afghanistan                South Asia          Low income
## 3             AGO      Angola        Sub-Saharan Africa Lower middle income
## 4             ALB     Albania     Europe & Central Asia Upper middle income
## 5             AND     Andorra     Europe & Central Asia         High income
## 6             ARB  Arab World
```

Leemos serie de porcentaje de mayores y nos quedamos con el ultimo año con datos

```
porcentaje_mayores <- read.csv(file="..\\csv\\WorldBankData\\UP65_Percentage.csv",
                                                        sep = ",")
porcentaje_mayores <- porcentaje_mayores %>% select(Country.Code,
                                                    "ï..Country.Name", X2018)
head(porcentaje_mayores)
```

```
##   Country.Code ï..Country.Name     X2018
## 1          ABW           Aruba 13.550947
## 2          AFG     Afghanistan  2.584927
## 3          AGO          Angola  2.216374
## 4          ALB         Albania 13.744736
## 5          AND         Andorra        NA
## 6          ARB      Arab World  4.557876
```

Hacemos join entre ambos dataframes para tener los datos de porcentaje de vejez y metadatos

```
country_incomes_elderly_continent <-merge(porcentaje_mayores, continente_ingresos,
                                          by.x="Country.Code", by.y="ï..Country.Code")
country_incomes_elderly_continent <- country_incomes_elderly_continent %>%
                                                    select(Country.Code,
                                                    "ï..Country.Name",
                                                    Region, IncomeGroup,
                                                    X2018)

country_incomes_elderly_continent <- country_incomes_elderly_continent %>%
                                                    rename(Continent=Region,
                                                           Country.Name="ï..Country.Name",
                                                           UpTo65=X2018)
head(country_incomes_elderly_continent)
```

```
##   Country.Code Country.Name                 Continent         IncomeGroup
## 1          ABW        Aruba Latin America & Caribbean         High income
## 2          AFG  Afghanistan                South Asia          Low income
## 3          AGO       Angola        Sub-Saharan Africa Lower middle income
## 4          ALB      Albania     Europe & Central Asia Upper middle income
## 5          AND      Andorra     Europe & Central Asia         High income
## 6          ARB   Arab World
##      UpTo65
## 1 13.550947
## 2  2.584927
## 3  2.216374
## 4 13.744736
## 5        NA
## 6  4.557876
```

5.2. Paso 2 - Leer series temporales del dataset cubo y calcular proporcion dia 40 Despues de 100 muertes para cada pais

También limpiaremos valores Na de indice de paises (borrado de filas) y de los valores de los días. Si hay indice del país, que haya una casilla vacia (NaN significa que el número es 0, casilla vacía de la tabla de Worldometers).

Primero leemos los datos de las series temporales, que les necesitaremos para calcular el día 40 depúes de 100 muertes de cada país. Limpiamos países nulos y valores Nan.

```
total_casos <- read.csv(file="..\\csv\\covid_19_series\\TotalCases_covid19_timeserie.csv",
                                                    sep = ",")
total_muertes <- read.csv(file="..\\csv\\covid_19_series\\TotalDeaths_covid19_timeserie.csv",
                                                    sep = ",")
total_tests <- read.csv(file="..\\csv\\covid_19_series\\TotalTests_covid19_timeserie.csv",
                                                    sep = ",")
#Quitamos indices de paises vacios
total_casos <- total_casos[!is.na(total_casos$Country),]
total_muertes <- total_muertes[!is.na(total_muertes$Country),]
total_tests <- total_tests[!is.na(total_tests$Country),]

#Quitamos datos de el mundo
total_casos <- total_casos[!total_casos$Country=='World',]
total_muertes <- total_muertes[!total_muertes$Country=='World',]
total_tests <- total_tests[!total_tests$Country=='World',]

#Limpiamos los casos donde hay Nan en la medida: será 0.
total_casos[is.na(total_casos)] <- 0
total_muertes[is.na(total_muertes)] <- 0
total_tests[is.na(total_tests)] <- 0

#Mostramos ejemplos de unas pocas columnas y unas pocas filas
head(total_muertes[,1:3])
```

```
##   Country X2020.03.30.12.00.00 X2020.04.06.17.24.01
## 1     USA                 2484                 9687
## 2   Italy                10779                15887
## 3   Spain                 6803                13055
```

```
## 4 Germany                541                1608
## 5  France                2606               8078
## 6    Iran                2640               3739
```

Como hemos comentado, tenemos que obtener el dato del día 20 desde el día para el que cada país haya superado las 100 muertes, con el objetivo de comparar de forma objetiva diferentes países: misma fase de la pandemia. **Por ello para cada país buscaremos el día en el que se superan esas muertes. Realmente, para cada país, obtendremos el índice de la columna donde se pasan por primera vez los 100 fallecidos**, con el objetivo de sumarle después los 30 días y obtener nuestro dato de **día 30 después del Covid** para cada país.

- Existe un problema. Nuestro primer scrapping de la primera práctica fue el dia 30/03, y en ese día ya había varios países que superaban con creces los 100 fallecidos. Nos guataría utilizar nuestros datos y no buscar nuevos (i.e. del Jhon Hopkins University) para realizar una práctica solucionando los problemas del propio dataset. **Por ello, debemos estimar hace cuantos días se llegó a las 100 muertes**. Se estima que las muertes se doblan cada dos días. Es decir $fallecidos = (dias/2)^2 \rightarrow dias = \sqrt{fallecidos/4}$. Como queremos estimar el día de las 100 muertes **aplicaremos un suavizado a la función** sacando el 4 de la raiz: $dias_{100} = \sqrt{fallecidos}/4$. Por ello, si vemos que en el primer dia del scrapping tenemos ya mas de 200 muertes, calculamos el índice el negativo, para que después al sumarle 30 a esa fecha, nos de una columna de nuestro dataset y correspondiente al día 30 Después del Covid.

Estimación vs real en algunos ejemplos (el día es el día de Marzo donde se superaron las 100 muertes):

Pais	**Estimacion**	**Real**
Italia	$30 - (sqrt(10779)/4) =$ Día 4	día 4
España	$30 - (sqrt(6803)/4) =$ Día 9	día 13
Francia	$30 - (sqrt(2500)/4) =$ Día 17	día 16
UK	$30 - (sqrt(1228)/4) =$ Día 21	día 19
Alemania	$30 - (sqrt(771)/4) =$ Día 23	día 24
Belgica	$30 - (sqrt(431)/4) =$ Día 24	día 25

```
index_100<-function(row){
  #Inicializamos los datos del pais, sera c(nombre, idColumna20diasDC)
  res <- c(row[1], NaN)
  #Vemos el numero de columnas que hay: numero de dias+nombrepais
  ncols <- length(names(row))
  #Recorremos todos los dias(por eso desde el 2, no recorremos el nombre)
  for(idxCol in seq(2,ncols)){
    #Obtenemos los fallecidos de ese dia
    fallec = as.numeric(row[idxCol])
    #Si superan nuestro umbral de 100 establecemos el resultado, BREAK del loop y return
    #Es decir, el primer valor que pase de 100 romperá el bucle
    if(fallec>100){
      if((idxCol==2)&(fallec>200)){
        #Si ya en la primera columna hay mas de 100, estimamos los dias
        res<-c(row[1], -round((sqrt(fallec)/4))+40)
      }else{
        res<-c(row[1], as.numeric(idxCol)+40)
      }
      #Salimos del bucle para devolver el resultado segun encontramos el 1er valor >100
```

```
      break
    }
  }
  return(res)
}
```

Aplicamos nuestra funcion para obtener un **dataset con el nombre de cada país y el índice de la columna del día 20 después del Covid** (tomando como día de llegada el día que se superan las 100 muertes).

```
#Aplicamos la funcion a cada linea del df, a cada pais
death_100 <- apply(total_muertes, 1, FUN=index_100)
#Creamos el dataframe de la lista resultante
death_100 <- as.data.frame(t(death_100))
#Damos nombre a las coumnas
colnames(death_100) <- c("Pais", "idCol")
#Convertimos la columna a numerico
death_100$idCol <- as.numeric(as.character(death_100$idCol))
#Ordenamos los paises por la llegada del Covid
death_100 <- death_100[order(death_100$idCol),]
#Mostramos un ejemplo del resultado
death_100[c(1,2,4,20,50,65,100),]
```

```
##              Pais idCol
## 2           Italy    14
## 3           Spain    19
## 5          France    27
## 22        Ireland    43
## 78        Moldova    67
## 81         Kuwait    87
## 91 Faeroe Islands   NaN
```

Ahora elegiremos los datos de contagiados y fallecidos para cada país de su día correspondiente.

```
get_deaths_40_days <- function(row){

  idCol <- as.numeric(row["idCol"])
  fall <- total_muertes[total_muertes$Country==row["Pais"],idCol]
  casos <- total_casos[total_casos$Country==row["Pais"],idCol]
  test <- total_tests[total_tests$Country==row["Pais"],idCol]

  if(is.null(casos)){
    casos<-0
    fall<-0
    test<-0
  }
  return(c(row["Pais"],casos,fall,test))
}
```

```
day_40_dc <- apply(death_100, 1, FUN=get_deaths_40_days)
day_40_dc <- as.data.frame(t(as.data.frame(day_40_dc)))
colnames(day_40_dc) <- c("Pais", "CasosDia40DC", "FallDia40DC", "Tests")
```

```
day_40_dc$CasosDia40DC <- as.numeric(as.character(day_40_dc$CasosDia40DC))
day_40_dc$FallDia40DC <- as.numeric(as.character(day_40_dc$FallDia40DC))
day_40_dc$Pais <- as.character(day_40_dc$Pais)

#Vemos los países que tenemos en total (de momento son todos)
dim(day_40_dc)
```

```
## [1] 215   4
```

```
head(day_40_dc)
```

```
##       Pais CasosDia40DC FallDia40DC   Tests
## 2    Italy       147577       18849  963473
## 3    Spain       195944       20639  930230
## 201  China        82827        4632       0
## 5   France       162100       22856  463662
## 6     Iran        90481        5710  432329
## 1      USA      1010507       56803 5919847
```

Ahora, tenemos que elegir los que han llegado a esa fase de la pandemia. Es decir, hay 3 casuísicas:

- Que idcol sea menor de 58 (son las columnas que tenemos), lo que significa que el dia 40 después de las 100 muertes están en el dataset.
- Que idcol sea mayor que 58, con lo que quiere decir que el país no haya llegado al día 40 después de las 100 muertes.
- Que idCol sea Nan, lo que significa que el país no ha llegado a las 100 muertes.

A nosotros nos interesa solo el primer conjunto de paises. Por eso, en la anterior función, hemos establecido los otros dos casos con los valores casos y fallecidos a 0.

```
day_40_dc <- day_40_dc[(day_40_dc$CasosDia40DC>0),]
#Contamos los países con los que podemos realizar en análisis estático.
dim(day_40_dc)
```

```
## [1] 46  4
```

```
#Mostramos ejemplo del dataset
head(day_40_dc)
```

```
##       Pais CasosDia40DC FallDia40DC   Tests
## 2    Italy       147577       18849  963473
## 3    Spain       195944       20639  930230
## 201  China        82827        4632       0
## 5   France       162100       22856  463662
## 6     Iran        90481        5710  432329
## 1      USA      1010507       56803 5919847
```

- Destacamos que las primera columnas del dataset no se corresponden día a día. Sin embargo, al obtener los índices de las columnas todos >=14, a partir de ahí todos cumplen con una columna por día.

Con esto hemos acabado el segundo paso, el del cálculo de cada país de los contagiados, fallecidos y tests el día 40 después de los 100 fallecidos. Esto será utilizado para nuestro dataset estático (contrastes de hipótesis, alguna correlación y regresión). Este dataset tiene 46 países.

5.3. Paso 3 - Cruzar datos por país dia 40 DC con datos de población

Leemos el archivo de datos de población

```
poblacion <- read.csv(file="..\\csv\\world_population_2020.csv")
poblacion <- poblacion %>% rename(Country="Country..or.dependency.",
                                  Poblacion="Population..2020.",
                                  Densidad="Density..P.KmÂ².")
head(poblacion[order(poblacion$Country),])
```

```
##            Country Poblacion Densidad
## 37     Afghanistan  38928346       60
## 140        Albania   2877797      105
## 33         Algeria  43851044       18
## 210 American Samoa     55191      276
## 203        Andorra     77265      164
## 44          Angola  32866272       26
```

Debemos hacer un Join del dataframe de los datos de contagios, fallecidos y tests el dia 40 con el dataframe de la población y densidad obtenido de un scrapping a Worldometers. Antes de hacer el join debemos mapear algunos nombres de países (nos hemos dado cuenta del error al hacer un letf outer join y ver países con nulos).

```
day_40_dc[day_40_dc$Pais=='USA',]$Pais <-"United States"
day_40_dc[day_40_dc$Pais=='UK',]$Pais <-"United Kingdom"
day_40_dc[day_40_dc$Pais=='S. Korea',]$Pais <-"South Korea"
day_40_dc[day_40_dc$Pais=='Czechia',]$Pais <-"Czech Republic (Czechia)"
```

Realizamos el merge de las tablas. Tendremos los 46 países que están en la fase de haber pasado 40 días después de las 100 muertes, pero ahora con los datos de población y densidad añadidos.

```
covid_country<-merge(day_40_dc, poblacion, by.x="Pais", by.y="Country",all.x = TRUE)
covid_country<-covid_country[order(-covid_country$CasosDia40DC),]
head(covid_country)
```

```
##              Pais CasosDia40DC FallDia40DC   Tests Poblacion Densidad
## 46  United States      1010507       56803 5919847 331002651       36
## 37         Russia       344481        3541 8945384 145934462        9
## 40          Spain       195944       20639  930230  46754778       94
## 45 United Kingdom       171253       26771 1023824  67886011      281
## 5          Brazil       169594       11653  735224 212559417       25
## 16        Germany       165664        6866 2547052  83783942      240
```

5.4. Paso 4 - Merge de datos covid por país y metadatos (Merge 3-1)

Primero de todo vemos los países que no coinciden en el merge. Este es el paso que hemo hecho también en el anterior caso, pero que en el anterior caso hemos obviado contarle.

```
aux<-merge(covid_country, country_incomes_elderly_continent, by.x="Pais",
           by.y="Country.Name",all.x = TRUE)
aux[is.na(aux$Country.Code),]
```

```
##                         Pais CasosDia40DC FallDia40DC   Tests Poblacion Densidad
## 10 Czech Republic (Czechia)         8721         304  387127  10708981      139
## 14                    Egypt        14229         680  135000 102334404      103
## 21                     Iran        90481        5710  432329  83992949       52
## 37                   Russia       344481        3541 8945384 145934462        9
## 39              South Korea        10936         258  695920  51269185      527
##    Country.Code Continent IncomeGroup UpTo65
## 10         <NA>      <NA>        <NA>     NA
## 14         <NA>      <NA>        <NA>     NA
## 21         <NA>      <NA>        <NA>     NA
## 37         <NA>      <NA>        <NA>     NA
## 39         <NA>      <NA>        <NA>     NA
```

Mapeamos los nombres para que coincidan

```
country_incomes_elderly_continent$Country.Name <-
  as.character(country_incomes_elderly_continent$Country.Name)
country_incomes_elderly_continent[country_incomes_elderly_continent$Country.Name==
                          'Czech Republic',]$Country.Name <-"Czech Republic (Czechia)"
country_incomes_elderly_continent[country_incomes_elderly_continent$Country.Name==
                                    'Egypt, Arab Rep.',]$Country.Name <-"Egypt"
country_incomes_elderly_continent[country_incomes_elderly_continent$Country.Name==
                                    'Iran, Islamic Rep.',]$Country.Name <-"Iran"
country_incomes_elderly_continent[country_incomes_elderly_continent$Country.Name==
                                    'Russian Federation',]$Country.Name <-"Russia"
country_incomes_elderly_continent[country_incomes_elderly_continent$Country.Name==
                                    'Korea, Rep.',]$Country.Name <-"South Korea"
```

Merge final

```
country_covid_and_metadata <- merge(covid_country, country_incomes_elderly_continent,
                                    by.x="Pais", by.y="Country.Name",all.x = TRUE)
#Borramos el codigo del pais
country_covid_and_metadata <- country_covid_and_metadata %>% select(-Country.Code)
head(country_covid_and_metadata)
```

```
##        Pais CasosDia40DC FallDia40DC   Tests Poblacion Densidad
## 1   Algeria         6067         515    6500  43851044       18
## 2 Argentina        12628         467  136662  45195774       17
## 3   Austria        15997         624  344606   9006398      109
## 4   Belgium        50509        8016  474176  11589623      383
## 5    Brazil       169594       11653  735224 212559417       25
## 6    Canada        71157        5169 1169380  37742154        4
##                     Continent         IncomeGroup    UpTo65
## 1 Middle East & North Africa Upper middle income  6.362497
## 2  Latin America & Caribbean Upper middle income 11.117789
## 3      Europe & Central Asia         High income 19.001566
## 4      Europe & Central Asia         High income 18.788744
## 5  Latin America & Caribbean Upper middle income  8.922838
## 6              North America         High income 17.232007
```

En este momento ya hemos integrado todas las diferentes fuentes de datos para hacer nuestro dataset estático de datos de países. El resumen del *pipeline* de integración y transformación es el comen-

tado a principio de esta sección. Sin embargo, añadiremos una imagen ilustrativa del mismo concretamente la figura 2.

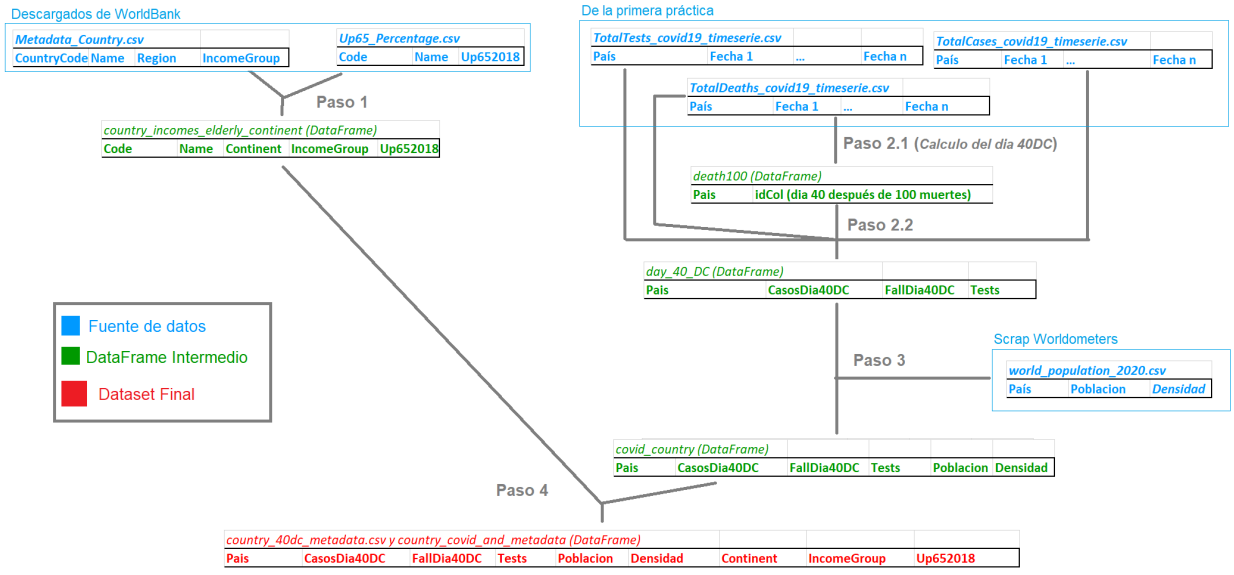

Figura 2: Pasos de la integración y selección

6. Limpieza de datos

Nos centraremos en el dataset que acabamos de crear, no en las series temporales. Aunque como hemos visto, en la anterior sección de integración hemos incluido algunas tareas como tratamiento de valores nulos.

6.1. Tipos de datos

Primero vemos si los tipos de datos de R coinciden con la naturaleza de los mismos.

```
sapply(country_covid_and_metadata, class)
```

```
##         Pais CasosDia40DC  FallDia40DC        Tests    Poblacion     Densidad
##  "character"    "numeric"    "numeric"     "factor"    "integer"    "integer"
##    Continent  IncomeGroup       UpTo65
##     "factor"     "factor"    "numeric"
```

Vemos que el único dato que no corresponde a su naturaleza es el número de tests. Para segurarnos que no hay desbordamientos debido a números grandes, también cambiamos el tipo de dato de población.

```
country_covid_and_metadata$Tests <- as.numeric(as.character(country_covid_and_metadata$Tests))
country_covid_and_metadata$Poblacion <- as.numeric(country_covid_and_metadata$Poblacion)
```

6.2. Nulos y vacíos

En la construción del nuevo dataset estático, se han incorporado los mecanismos para limpiar de nulos, con lo que en la anterior sección de integración hemos hecho tareas de ésta índole. Vemos como no hay un solo valor nulo en todo el dataframe.

```
sapply(country_covid_and_metadata, function(x) sum(is.na(x)))
```

```
##         Pais CasosDia40DC  FallDia40DC        Tests    Poblacion     Densidad
##            0            0            0            0            0            0
##    Continent  IncomeGroup       UpTo65
##            0            0            0
```

Para las columnas numéricas, valores nulos podrían ser el 0. Sin embargo, esto lo dejamos para la fase de detección de outliers.

6.3. Outliers

Los valores outliers son aquellos que se alejan de la distribución habitual de los datos. Estos outliers se pueden dar a varias causas: errores en la insercción de datos, medidas de individuos fuera de la población, o datros correctos pero que simplemente son altos. Sabiendo las características de los datos, para muchos campos tendremos vaalores altos pero totalmente explicables.

Trataremos los outliers campo por campo, empezando por los numéricos.

```r
show_outlier<-function(data){
  values <- boxplot.stats(data)$out
  idx <- which( data %in% values)
  cat("Valores extremos:", toString(values), "\n" )
  (country_covid_and_metadata[idx, ])
}
```

Vemos los **casos** que son outlier:

```r
show_outlier(country_covid_and_metadata$CasosDia40DC)
```

```
## Valores extremos: 344481, 195944, 1010507
```

```
##             Pais CasosDia40DC FallDia40DC   Tests Poblacion Densidad
## 37        Russia       344481        3541 8945384 145934462        9
## 40         Spain       195944       20639  930230  46754778       94
## 46 United States      1010507       56803 5919847 331002651       36
##               Continent         IncomeGroup   UpTo65
## 37 Europe & Central Asia Upper middle income 14.67471
## 40 Europe & Central Asia         High income 19.37851
## 46         North America         High income 15.80765
```

Estos valores pueden darse perfectamente, países en los que se han detectado muchos casos.

Vemos los **fallecidos** que son outlier:

```r
show_outlier(country_covid_and_metadata$FallDia40DC)
```

```
## Valores extremos: 11653, 22856, 18849, 20639, 26771, 56803
```

```
##              Pais CasosDia40DC FallDia40DC   Tests Poblacion Densidad
## 5          Brazil       169594       11653  735224 212559417       25
## 15         France       162100       22856  463662  65273511      119
## 24          Italy       147577       18849  963473  60461826      206
## 40          Spain       195944       20639  930230  46754778       94
## 45 United Kingdom       171253       26771 1023824  67886011      281
## 46  United States      1010507       56803 5919847 331002651       36
##                    Continent         IncomeGroup    UpTo65
## 5  Latin America & Caribbean Upper middle income  8.922838
## 15     Europe & Central Asia         High income 20.034625
## 24     Europe & Central Asia         High income 22.751680
## 40     Europe & Central Asia         High income 19.378508
## 45     Europe & Central Asia         High income 18.395866
## 46             North America         High income 15.807654
```

Al igual que antes son casos que se han dado, no hay fallos. Se explcia porque son los países con más afectados y que tienen un gran número de población.

Vemos los **tests** que son outlier:

```
show_outlier(country_covid_and_metadata$Tests)
```

```
## Valores extremos: 2547052, 1947041, 8945384, 1440671, 5919847
```

```
##             Pais CasosDia40DC FallDia40DC   Tests  Poblacion Densidad
## 16       Germany       165664        6866 2547052   83783942      240
## 19         India        74925        2436 1947041 1380004385      464
## 37        Russia       344481        3541 8945384  145934462        9
## 43        Turkey       139771        3841 1440671   84339067      110
## 46 United States      1010507       56803 5919847  331002651       36
##                Continent         IncomeGroup    UpTo65
## 16 Europe & Central Asia         High income 21.461962
## 19            South Asia Lower middle income  6.179956
## 37 Europe & Central Asia Upper middle income 14.674708
## 43 Europe & Central Asia Upper middle income  8.483213
## 46         North America         High income 15.807654
```

Son valores totalmente correctos, explicados porque son países grandes y que tienen la estrategia de hacer tests.

Vemos los **poblacion** que son outlier:

```
show_outlier(country_covid_and_metadata$Poblacion)
```

```
## Valores extremos: 212559417, 1439323776, 1380004385, 273523615, 220892340, 331002651
```

```
##             Pais CasosDia40DC FallDia40DC   Tests  Poblacion Densidad
## 5         Brazil       169594       11653  735224  212559417       25
## 8          China        82827        4632       0 1439323776      153
## 19         India        74925        2436 1947041 1380004385      464
## 20     Indonesia        14749        1007  169195  273523615      151
## 30      Pakistan        57705        1197  499399  220892340      287
## 46 United States      1010507       56803 5919847  331002651       36
##                     Continent         IncomeGroup    UpTo65
## 5  Latin America & Caribbean Upper middle income  8.922838
## 8        East Asia & Pacific Upper middle income 10.920884
## 19                South Asia Lower middle income  6.179956
## 20       East Asia & Pacific Lower middle income  5.857166
## 30                South Asia Lower middle income  4.312774
## 46             North America         High income 15.807654
```

Son valores correctos, correspondientes a los paises más grandes del mundo.

Vemos los **densidad** que son outlier:

```
show_outlier(country_covid_and_metadata$Densidad)
```

```
## Valores extremos: 508, 527
```

```
##           Pais CasosDia40DC FallDia40DC  Tests Poblacion Densidad
## 28 Netherlands        40236        4987 225899  17134872      508
```

```
## 39 South Korea        10936         258 695920  51269185      527
##                Continent IncomeGroup   UpTo65
## 28 Europe & Central Asia High income 19.19619
## 39   East Asia & Pacific High income 14.41856
```

Con la tónica habitual, vemos que son valores de densidad altos, pero son perfectamente correctos.

Vemos los **porcentajes de vejez** que son outlier:

```
show_outlier(country_covid_and_metadata$UpTo65)
```

```
## Valores extremos:
```

```
## [1] Pais         CasosDia40DC FallDia40DC  Tests        Poblacion
## [6] Densidad     Continent    IncomeGroup  UpTo65
## <0 rows> (or 0-length row.names)
```

Vemos que están todos dentro de los valores normales de ola muestra.

6.4. Incongruencias

Otro aspecto que hay que mirar en nuestros casos en la consistencia de los mismos. En nuestro caso, comprobaremos que el número de casos realizados es menor o igual al número de tests realizados. En caso contrario, habría algún fallo en los datos.

```
country_covid_and_metadata[country_covid_and_metadata$Tests
                           < country_covid_and_metadata$CasosDia40DC,]
```

```
##    Pais CasosDia40DC FallDia40DC Tests  Poblacion Densidad           Continent
## 8 China        82827        4632     0 1439323776      153 East Asia & Pacific
##           IncomeGroup   UpTo65
## 8 Upper middle income 10.92088
```

```
country_covid_and_metadata$Tests[country_covid_and_metadata$Tests
                           < country_covid_and_metadata$CasosDia40DC]<-10
```

En este caso vemos que China no cumple esa condición. No solo es que no cumpla la restricción, esque además los test realizados son 0. Esto significará que hay falta de datos sobre los test realizados de este país. **Imputaremos el valor basándonos en una regresión lineal de los test a través de los casos y los fallecidos.** Primero creamos el modelo:

```
model_tests <- lm(Tests CasosDia40DC+FallDia40DC, data=country_covid_and_metadata)
summary(model_tests)
```

```
##
## Call:
## lm(formula = Tests ~ CasosDia40DC + FallDia40DC, data = country_covid_and_metadata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

Cuadro 7: Head de Dataset Final

Pais	CasosDia40DC	FallDia40DC	Tests	Poblacion	Densidad	Continent	IncomeGroup	UpTo65
Algeria	6067	515	6500	43851044	18	Middle East & North Africa	Upper middle income	6.362496
Argentina	12628	467	136662	45195774	17	Latin America & Caribbean	Upper middle income	11.117789
Austria	15997	624	344606	9006398	109	Europe & Central Asia	High income	19.001566
Belgium	50509	8016	474176	11589623	383	Europe & Central Asia	High income	18.788744
Brazil	169594	11653	735224	212559417	25	Latin America & Caribbean	Upper middle income	8.922838
Canada	71157	5169	1169380	37742154	4	North America	High income	17.232007

```
## -1779882  -317163  -176700    18360  3975477
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  309275.881 139112.023   2.223   0.0315 *
## CasosDia40DC     14.920      1.674   8.914 2.51e-11 ***
## FallDia40DC    -135.323     26.231  -5.159 6.01e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 835600 on 43 degrees of freedom
## Multiple R-squared:  0.7218, Adjusted R-squared:  0.7088
## F-statistic: 55.78 on 2 and 43 DF,  p-value: 1.133e-12
```

Vemos que el resultado nos indica un R^2 de 0.72. Es decir, que el modelo explica el 72 % de la vrianza original de los datos. Además, vemos que tanto el p-valor para el modelo (para R^2), como para las dos variables, nos indican que el resultado es estadísticamente significativo con un nivel de significancia muy bajo. Por ello, consideramos el modelo suficientemente bueno par imputar los tests de China. Imputamoes el valor:

```
casos = country_covid_and_metadata[country_covid_and_metadata$Pais=='China',]$CasosDia40DC
falle = country_covid_and_metadata[country_covid_and_metadata$Pais=='China',]$FallDia40DC
newdata <- data.frame( CasosDia40DC = casos, FallDia40DC=falle)
(pr <- predict(model_tests, newdata))
```

```
##        1
## 918274.8
```

```
country_covid_and_metadata$Tests[country_covid_and_metadata$Pais=='China']<-round(pr)
```

7. Exportar datos limpios

Guardamos nuestro dataset en un csv. Mostramos el resultado de la tabla en el Cuadro 7

```
write.csv(country_covid_and_metadata, "..\\csv\\country_40dc_metadata.csv", row.names=FALSE)

head(country_covid_and_metadata) %>% kable(caption="Head de Dataset Final") %>%
  kable_styling(latex_options="scale_down")
```

8. Analisis de datos

Recordar hacer analisis de normalidad y varianza cuando toque

9. Agradecimientos

Todo * Principalmente, agradecer a la asociación *Worldometers* [Worldometers, 2020], asociación de estadísticas mundiales en tiempo real, por tener los datos actualizados de manera tan rápida y en abierto. * Después, tanto a los estudios de la *Johns Hopkins University* [Dong et al., 2020], como a la asociación *Our world in Data* de la *Oxford University* [Roser et al., 2020], por sus trabajos que nos han permitido descubrir fuentes da calidad. * Agradecer los recursos encontrados para realizar el scraping, tanto en [Lawson, 2015] como en el módulo [Subirats Mate and Calvo Gonzalez, (sf], propio de la UOC.

10. Inspiración

ToDo Hablar de J. Arenas y demás.

11. Código fuente y dataset en Zenodo

ToDo * El código fuente del scraping, actualización de datos y autoimatización mediante *Travis* se encuentra en este enlace. Además hay archivos `readmemd` en los directorios que dan más información del proyecto.

- El dataset (conjunto de 5 csv) se sube a *Zenodo*, sin embargo, cabe **destacar que debido al potencial añadido de la autoactualización con *Travis*, este dataset está en continua actualización diaria de los datos**. El dataset con mayor actualización se coresponderá al que tenemos en el repositorio.
    - **DOI de Zenodo**: 10.5281/zenodo.3748050.
    - **Link a Zenodo**: https://zenodo.org/record/3748050#.XpD5w8gzZ9A

12. Tabla de contribuciones

ToDo | Contribuciones | Firma | |———————-|————-| | Investigación previa | P.G.S, A.A.R | | Redacción respuestas | P.G.S, A.A.R | | Desarrollo de código | P.G.S, A.A.R |

ToDo

Referencias

Ensheng Dong, Hongru Du, and Lauren Gardner. An interactive web-based dashboard to track covid-19 in real time. *The Lancet infectious diseases*, 2020.

Richard Lawson. *Web scraping with Python*. Packt Publishing Ltd, 2015.

Max Roser, Hannah Ritchie, and Esteban Ortiz-Ospina. Coronavirus disease (covid-19)–statistics and research. *Our World in Data*, 2020.

Laia Subirats Mate and Mireia Calvo Gonzalez. Web scraping. Technical report, UOC, Barcelona, (sf). PID00256970.

Worldometers. Covid-19 coronavirus pandemic. https://www.worldometers.info/coronavirus/, 2020.