Universidad del Desarrollo

Certamen 1

Ramo: Taller de Inteligencia de Negocios
Profesor: Mauricio Herrera
Integrantes: Jan Frese y Jorge Ramírez

Para la realización de este certamen, antes que todo, se decidió instalar e implementar todas las librerías necesarias para que ciertos comandos necesarios puedan funcionar de buena manera. Además, para obtener las visualizaciones de solamente las variables y gráficos a desarrollar, se decidió borrar las variables y los gráficos que tenía guardado el programa, para así no confundirlas con las variables y gráficos a ocupar en el certamen.

```
 rm(list=ls())
graphics.off()
#install.packages("readxl")
#install.packages("dplyr")
#install.packages("httr")
#install.packages("lubridate")
#install.packages("ggplot2")
#install.packages("RColorBrewer")
#install.packages("rgdal")
#install.packages("leaflet")
#install.packages("hrbrthemes")
library(hrbrthemes)
library(readxl)
library(dplyr)
library(httr)
library(lubridate)
library(ggplot2)
library(RColorBrewer)
library(rgdal)
library(leaflet)
```

TEMA 1:
VISUALIZACIÓN Y ANÁLISIS EXPLORATORIO DE LOS DATOS

Para realizar esta parte del certamen, se decidió implementar los códigos propuestos, para poder desarrollar este primer tema sin inconvenientes.
El código que se mostrará a continuación va a representar un estudio realizado de la evolución del COVID-19 en el mundo. Para poder descargar estos datos del estudio, se decidió trabajar con una página web que presenta las variables requeridas.

```
# TEMA 1: VISUALIZACIÓN Y ANÁLISIS EXPLORATORIO DE LOS DATOS
setwd('C:/Users/pc/Desktop/DATA/') # Fijar el directorio de trabajo
getwd()
dir()

# 1. Descargar el mapa del mundo:
download.file("http://thematicmapping.org/downloads/TM_WORLD_BORDERS_SIMPL-0.3.zip" ,
              destfile="C:/Users/pc/Desktop//DATA/world_shape_file.zip")
system("unzip world_shape_file.zip")
world_spdf <- readOGR(dsn= getwd(),layer="TM_WORLD_BORDERS_SIMPL-0.3",
verbose=FALSE)

# Limpiamos el dataset
world_spdf@data$POP2005[ which(world_spdf@data$POP2005 == 0)] = NA
```

```
world_spdf@data$POP2005 <- as.numeric(as.character(world_spdf@data$POP2005)) / 1000000
%>% round(2)

# 2. Descargar reportes actualizados de COVID19 y cargarlos en R para su posterior análisis:
url=paste("https://www.ecdc.europa.eu/sites/default/files/documents/COVID-19-geographic-
disbtribution-worldwide-",
        ymd(format(Sys.time(), "%Y-%m-%d")), ".xlsx", sep = "")
# Nota: Dado que el reporte se entrega al final del día en caso de que no esté listo se deberá restar
# un día, esto se hace restando por -days(1). En este caso el url a emplear es:
#url=paste("https://www.ecdc.europa.eu/sites/default/files/documents/COVID-19-geographic-
disbtribution-worldwide-", ymd(format(Sys.time(), "%Y-%m-%d")-days(1)), ".xlsx", sep = "")

# Con el siguiente código descargamos el reporte excel y lo guardamos en un
#archivo temporal
GET(url, authenticate(":", ":", type="ntlm"),
    write_disk(tf <- tempfile(fileext = ".xlsx")))

#Cargamos la data desde el archivo temporal tf
data <- read_excel(tf)

# Limpieza de los datos.
#Modificamos algunos nombres de variables:
names(data)[7]="Countries_and_territories"
names(data)[9]="Country_Code"
data$dateRep=ymd(data$dateRep)
names(data)[9]="ISO3"
head(data)
data$Countries_and_territories[data$Countries_and_territories=='United_States_of_America']='United
States'
#glimpse(data)
#Con esto tenemos el reporte de casos de COVID19 descargado.

### Inciso a ###
#A partir de "data" construya un nuevo conjunto de datos que agrupe por país (use la variable
"ISO3") y
# calcule los casos totales, fallecidos, y los casos del día. Recomendación: Use el paquete dplyr, y su
# comando group_by(SO3).

datos1=world_spdf@data
datos1$ISO3=as.character(datos1$ISO3)
datos2=data%>%group_by(ISO3)%>%summarise(Casos_totales=sum(cases),Fallecidos=sum(death
s))
head(datos2)
datos3=filter(data,dateRep==ymd(format(Sys.time(), "%Y-%m-%d")))

### Inciso b ###
#Cruce los datos obtenidos con el conjunto datos1=world_spdf@data"

P=left_join(datos1, datos2, by = "ISO3")
Q=left_join(datos1, datos3, by = "ISO3")
```

```
### Inciso c ###
# Agregue las columnas a la data a world_spdf@data

world_spdf@data$Casos_totales=P$Casos_totales
world_spdf@data$Fallecidos=P$Fallecidos
world_spdf@data$Fecha=Q$dateRep
world_spdf@data$Casos_hoy=Q$cases
world_spdf@data$Poblacion=round(Q$popData2019/1000000,2)
#head(world_spdf@data)
```

Actividad 1:

Inciso a:

Para este inciso se decidió realizar el código mediante el propuesto en el enunciado del certamen. Primero se determinó el valor mínimo y máximo del número de fallecidos en todo el mundo. Como el valor mínimo de fallecidos en el mundo es mayor o igual a 0 y menor que 500.000, se decidió definir los distintos tonos de colores según los siguientes intervalos:

Intervalos:
[0, 10[
[10, 1.000[
[1.000, 10.000[
[10.000, 50.000[
[50.000, 100.000[
[100.000, 500.000]

El color seleccionado para representar la cantidad de fallecidos en el mundo según los intervalos propuestos fue el color rojo. Si la cantidad de personas fallecidas en un país corresponde a un valor ubicado en el primer intervalo, su color va a ser rojo claro. Mientras mayor sea el número de fallecidos, más oscuro será el rojo del país. El mapamundi mencionado anteriormente será representado por la **fig.0**.

```
max(datos2$Fallecidos)
min(datos2$Fallecidos)
mis_bins1 <- c(0,10,1000,10000,50000,100000,500000)
mi_paleta1 <- colorBin( palette="Reds", domain=world_spdf@data$Fallecidos,
                        na.color="transparent", bins=mis_bins1)
# Preparación del texto que saldrá en los poput
mi_texto1 <- paste(
  "País: ", world_spdf@data$NAME,"<br/>",
  "Fallecidos: ", world_spdf@data$Fallecidos, "<br/>",
  "Número de Casos: ", world_spdf@data$Casos_totales, "<br/>",
  "Número de Casos hoy : ", world_spdf@data$Casos_hoy, "<br/>",
  "Población: ", world_spdf@data$Poblacion,
  sep="") %>%
  lapply(htmltools::HTML)
# El mapa
leaflet(world_spdf) %>%
```

```
  addTiles()  %>%
  setView( lat=10, lng=0 , zoom=2) %>%
  addPolygons(
    fillColor = ~mi_paleta1(Fallecidos),
    stroke=TRUE,
    fillOpacity = 0.9,
    color="white",
    weight=0.3,
    label = mi_texto1,
    labelOptions = labelOptions(
      style = list("font-weight" = "normal", padding = "3px 8px"),
      textsize = "13px",
      direction = "auto"
    )
  ) %>%
  addLegend( pal=mi_paleta1,
             values=~Casos_totales, opacity=0.9, title = "Número de fallecidos", position = "bottomleft" )
```

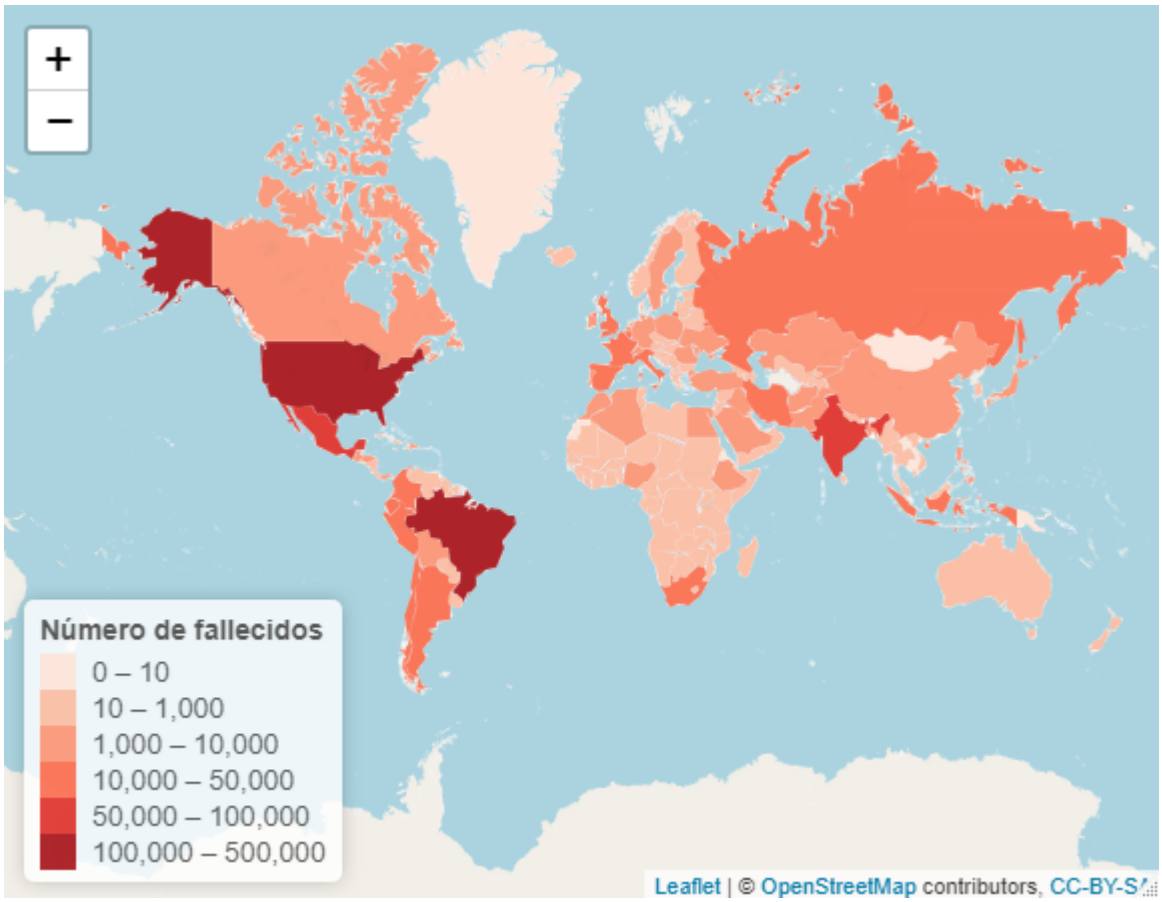

fig.0

Inciso b:

Para este inciso se decidió realizar un código similar al del inciso anterior. Se definió el color rojo y sus distintos tonos para representar los casos diarios de cada país. Como el número de casos diarios varían de manera similar a la cantidad de fallecidos, se optó por trabajar con los mismos intervalos del inciso anterior.
El mapamundi obtenido mediante el código realizado será representado por la **fig.1** expuesta tras el código, en donde el rojo oscuro representa los valores de casos diarios comprendidos en el mayor intervalo y el rojo más claro hace referencia a los países que poseen un número de casos diarios comprendidos en el menor intervalo.

```
mis_bins <- c(0,10,1000,10000,50000,100000,500000)
mi_paleta <- colorBin( palette="Reds", domain=world_spdf@data$Casos_hoy,
                       na.color="transparent", bins=mis_bins)
# Preparación del texto que saldrá en los poput
mi_texto <- paste(
  "País: ", world_spdf@data$NAME,"<br/>",
  "Fallecidos: ", world_spdf@data$Fallecidos, "<br/>",
  "Número de Casos: ", world_spdf@data$Casos_totales, "<br/>",
  "Número de Casos hoy : ", world_spdf@data$Casos_hoy, "<br/>",
  "Población: ", world_spdf@data$Poblacion,
  sep="") %>%
  lapply(htmltools::HTML)
# El mapa
leaflet(world_spdf) %>%
  addTiles()  %>%
  setView( lat=10, lng=0 , zoom=2) %>%
  addPolygons(
    fillColor = ~mi_paleta(Casos_hoy),
    stroke=TRUE,
    fillOpacity = 0.9,
    color="white",
    weight=0.3,
    label = mi_texto,
    labelOptions = labelOptions(
      style = list("font-weight" = "normal", padding = "3px 8px"),
      textsize = "13px",
      direction = "auto"
    )
  ) %>%
  addLegend( pal=mi_paleta,
             values=~Casos_hoy, opacity=0.9, title = "Número de caso de hoy", position = "bottomleft" )
```

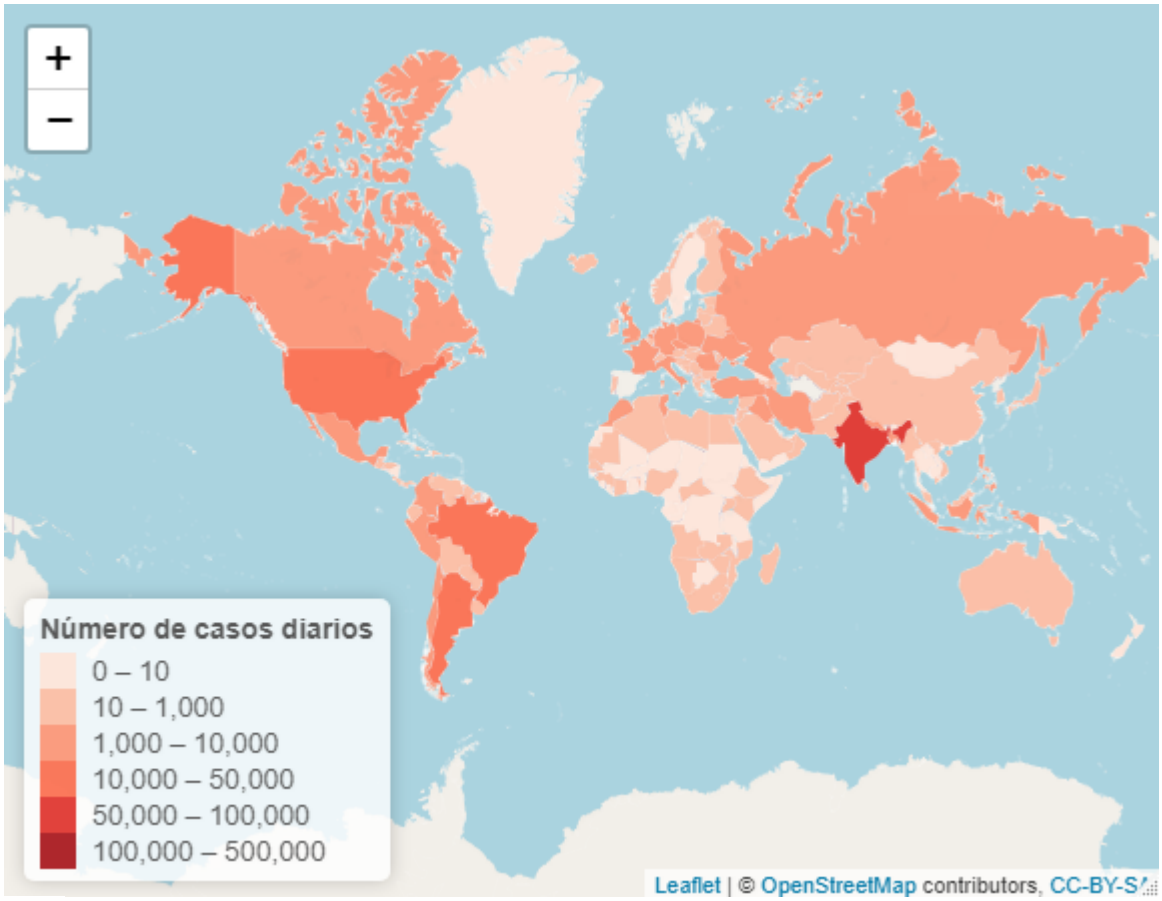

fig.1

Actividad 2:

Inciso a:

Para realizar la gráfica para el número de fallecidos en Chile de los últimos días, rescatamos los datos correspondientes a “Chile” con una frecuencia de 5 días, obteniendo de esta manera la **fig 2.**

```
#ACTIVIDAD 2:
#         A) Obtenga una grafica para el número de fallecidos
#         B) Obtenga una gráfica similar para el caso de USA (Utilice la opción
# scale_x_date(date_breaks = "5 day") para una mejor visualización)
##########################################################################################
######

# A)
data_Chile=filter(data,Countries_and_territories=='Chile')
p1<- data_Chile %>%
  ggplot( aes(x=dateRep, y=deaths)) +
  geom_area(fill="#69b3a2", alpha=0.5) +
  geom_line(color="#69b3a2") +
  ylab("Número de fallecidos") +
```

```
  theme_ipsum()+scale_x_date(date_breaks = "5 day")+
  theme(axis.text.x=element_text(angle=60, hjust=1))+geom_point(size=1,color='Darkred')
p1
```

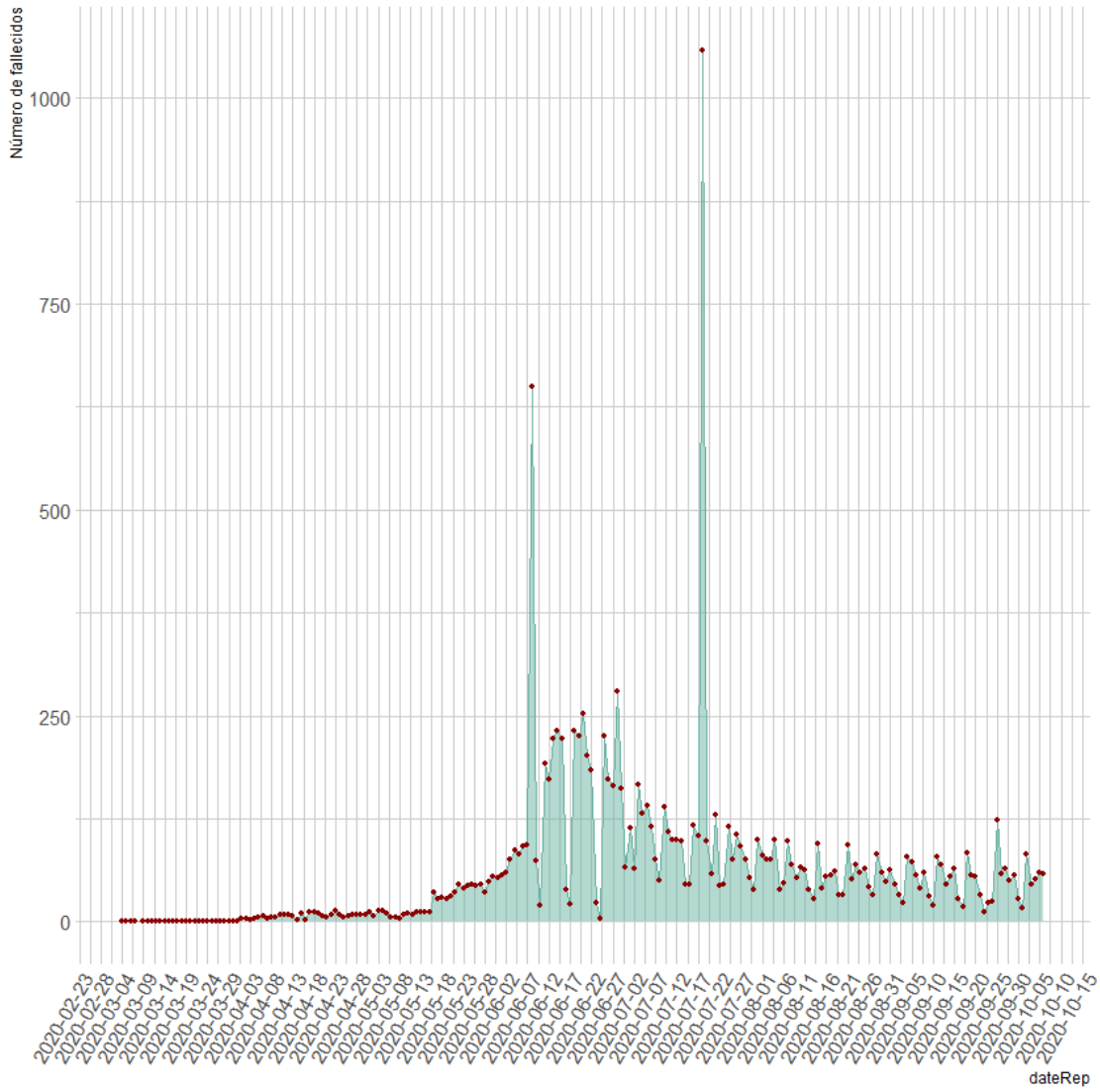

fig 2.

Por lo visto en la **fig 2**, se observa que la tendencia de fallecidos a la fecha tiende a disminuir, sin embargo, se aprecian 2 datos atípicos con un alza relevante respecto a la media de fallecidos, correspondientes a los días 02/06/2020 y 17/07/2020. Por otra parte, se observa que la llegada del virus (o los primeros casos observados) se registró entre los días 29/03/2020 y 04/03/2020.

.

Inciso b:

Para observar de una manera similar los datos de fallecidos correspondientes a Estados Unidos, se filtra la data correspondiente a este país y se grafican los datos cada 5 días (**fig 3**).

```
# B)
data_USA=filter(data,Countries_and_territories=="United States")
pUSA <- data_USA %>%
  ggplot( aes(x=dateRep, y=deaths)) +
  geom_area(fill="#69b3a2", alpha=0.5) +
  geom_line(color="#69b3a2") +
  ylab("Número de fallecidos") +
  theme_ipsum()+scale_x_date(date_breaks = "5 day")+
  theme(axis.text.x=element_text(angle=60, hjust=1))+geom_point(size=1,color='Darkblue')
```

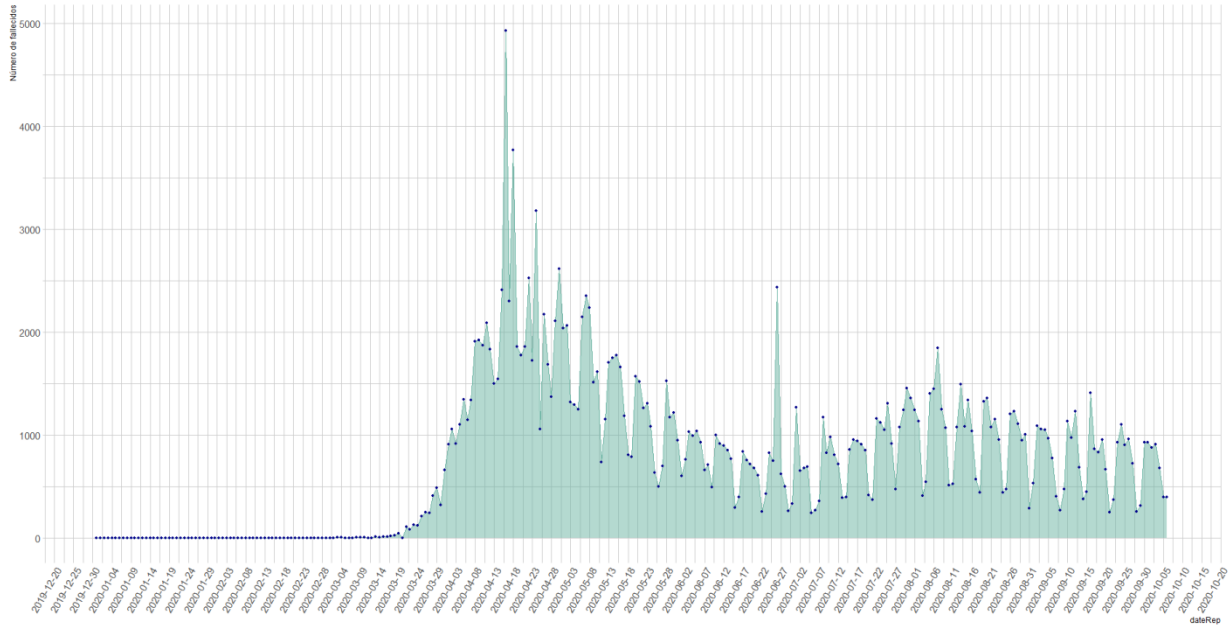

fig 3.

Respecto a Estados Unidos, la llegada del virus (o los primeros casos avistados) ocurrió entre el 28/02/2020 y 04/03/2020, mientras que los fallecidos sufrieron una dispersión considerable en la semana del 29/03/2020, registrando un máximo de casi 5.000 fallecidos cerca de la tercera semana de abril. Desde ese entonces la línea de tendencia de fallecidos fue en disminución hasta el día 27/06/2020, donde se registró un alza de 2.500 fallecidos y aumentaron los casos durante un mes. Desde la segunda semana de agosto los fallecidos registrados van en declive.

Finalmente, se ocupa el código dado en el certamen para graficar las líneas de tendencias de los países con más casos (China, Italia, Estados Unidos y España) **fig 4**.

```
# Grafique las lineas de tendencias de los países  con más casos:
a=c('China','Italy','United States','Spain')
ggplot(data=filter(data,Countries_and_territories%in%a),
```

```
    aes(x=dateRep,y=cases,color=Countries_and_territories))+
geom_line()+
geom_point() +
#facet_wrap(~Countries_and_territories,nrow=5)+
#scale_x_date(date_labels = "%m-%Y")+
scale_x_date(date_breaks = "5 day")+
theme(axis.text.x=element_text(angle=60, hjust=1))
```

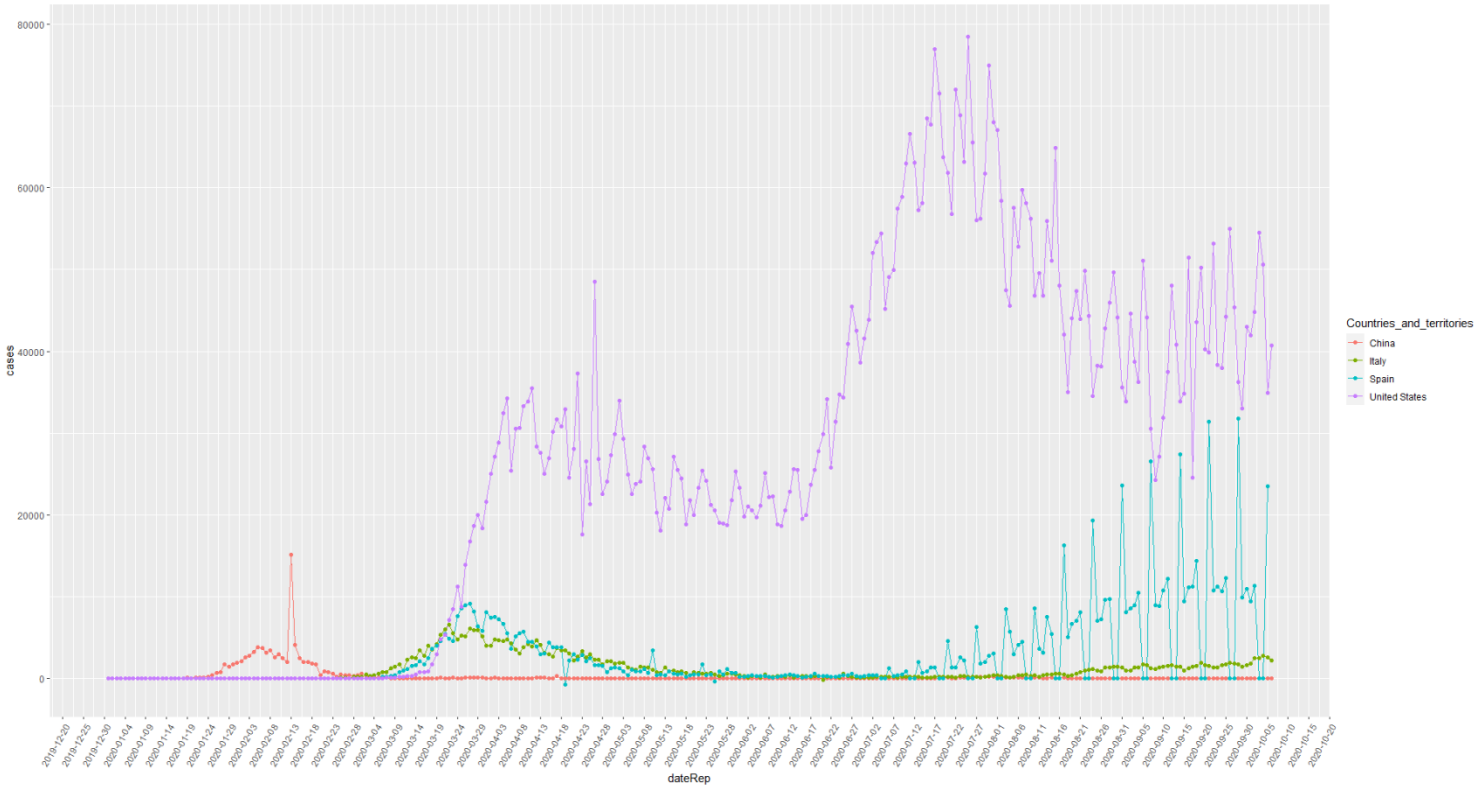

fig 4.

Las líneas de tendencia de fallecidos de estos cuatro países, indica que fue China el impulsor del virus, registrando casi 35.000 fallecidos la segunda semana de abril, lograron contener el virus de una buena manera, ya que hasta la fecha su línea de tendencia no ha aumentado. Por otra parte, Estados Unidos, no logra regular los números de fallecidos en su país, teniendo una diferencia significativa con Italia y España, que registran sus primeros fallecidos con solo 2-3 semanas de diferencia, aunque en España se observó un alza significativa de fallecidos desde la segunda semana de julio, al igual que Italia desde la tercera semana de agosto, pero mucho más contenida.

Actividad 3:

Se pide construir una gráfica de casos diarios para los países de la región. Sin embargo, como no se especifica qué región se desea analizar, se optó por analizar los casos correspondientes a los países pertenecientes a la región “América latina y el Caribe”.
Para esto se construye un diagrama de barras con los casos totales por país hasta la fecha correspondientes a la región elegida (**fig 5**).

```
#ACTIVIDAD: 3 Construya una gráfica de casos diarios para los países de la región.
##############################################################################################
dataAmerica <- data[data$continentExp=="America",]
unique(dataAmerica$Countries_and_territories)
dataAmerica$cases[which(dataAmerica$cases < 0)] <- 0
America <-
c("Antigua_and_Barbuda","Bahamas","Bolivia","Chile","Costa_Rica","Dominica","Grenada",
          "Jamaica","Panama","Dominican_Republic","Trinidad_and_Tobago","Venezuela","Argentina",
          "Barbados","Brasil","Colombia","Cuba","Ecuador","Guyana","Mexico","Peru","Suriname",
          "Uruguay")
ggplot(data=filter(dataAmerica,Countries_and_territories%in%America),
      aes(x=dateRep,y=cases,color=Countries_and_territories))+
  geom_line()+
  geom_point() +
  #facet_wrap(~Countries_and_territories,nrow=5)+
  #scale_x_date(date_labels = "%m-%Y")+
  scale_x_date(date_breaks = "5 day")+
  theme(axis.text.x=element_text(angle=60, hjust=1))
```

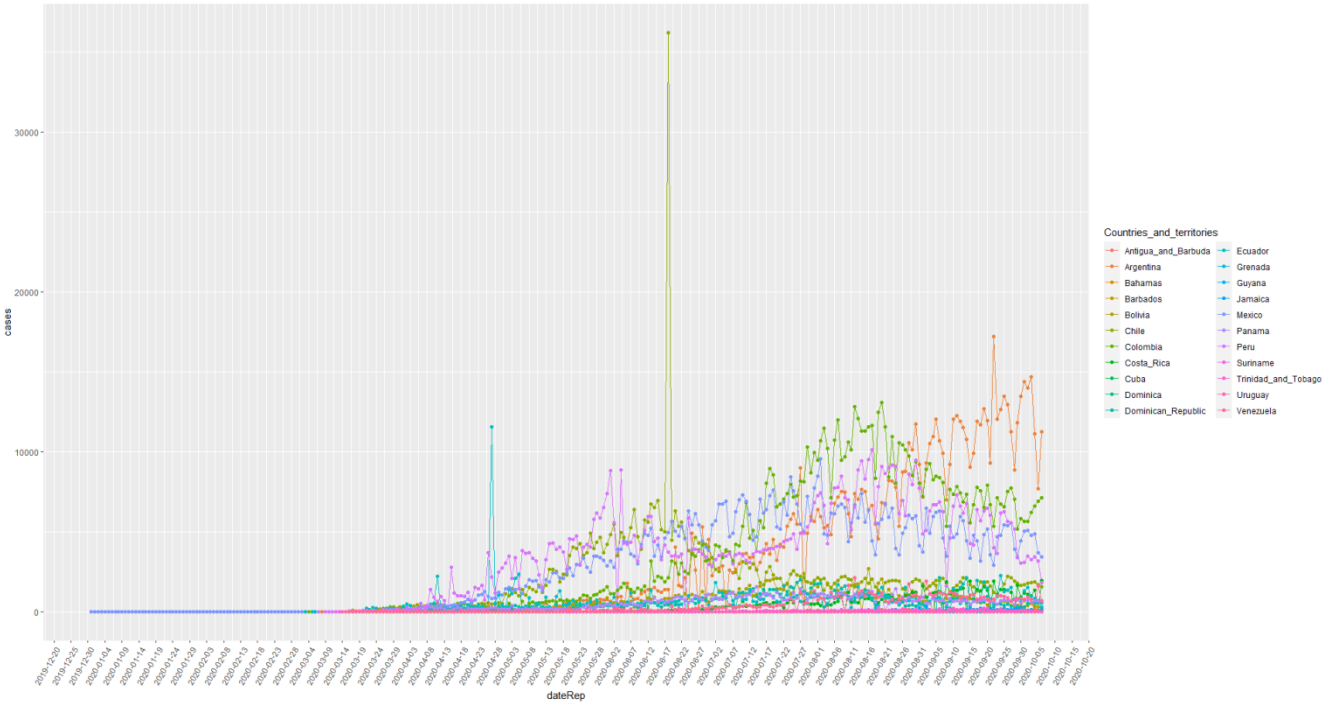

fig 5.

Por lo general se observa que el comienzo de los registros de casos en la región aconteció en simultáneo con una diferencia de unas dos semanas, Ecuador y Perú fueron los primeros dos países con más casos registrados, Chile tuvo el alza de casos registrados cerca del 17/06/2020, mientras que los casos en México y Argentina venían en alza desde esa fecha. Actualmente Argentina, Chile, México y Perú son los países de la región con más casos registrados.

Actividad 4:

Inciso a:

Se pide construir un diagrama de barras (**fig 6.**) con los fallecidos por países indicados en el conjunto a:
'Chile','Peru','Argentina','Brasil','México','Bolivia','Ecuador','Uruguay',
'China','Italy','Spain','Iran',"United States",'Germany'.

```
#ACTIVIDAD 4 : A) Construir un diagrama de barras con los fallecidos por países (los indicados
#                en el conjunto a)
#              B) Construir un diagrama de barras con los casos del día de hoy por países
################################################################################################
##############

# A)
datos4=data%>%group_by(Countries_and_territories)%>%summarise(Casos_totales=sum(cases),Fa
llecidos=sum(deaths))
a=c('Chile','Peru','Argentina','Brazil','Mexico','Bolivia','Ecuador','Uruguay',
    'China','Italy','Spain','Iran',"United States",'Germany')
p=ggplot(data=filter(datos4,Countries_and_territories%in%a),
         aes(x=Countries_and_territories,y=Fallecidos,fill=Countries_and_territories))+
  geom_bar(stat = "identity")+
  geom_text(aes(label=Fallecidos), position=position_dodge(width=0.9), vjust=-0.25)
p+theme(axis.text.x=element_text(angle=60, hjust=1))
```

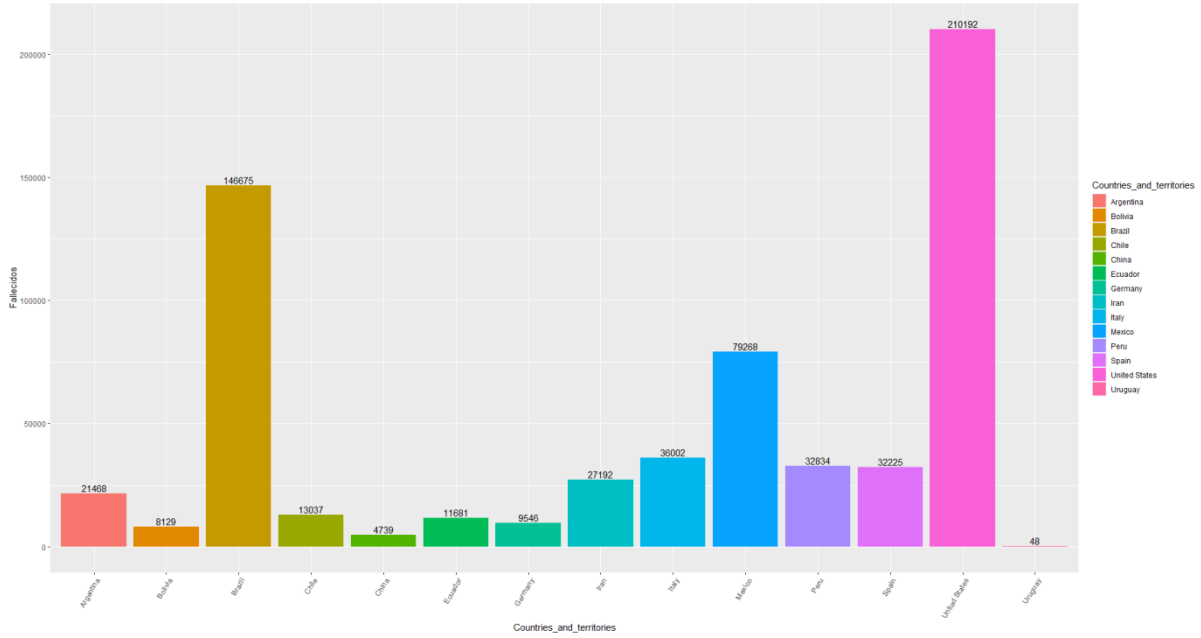

fig 6.

Dentro de este conjunto, Estados Unidos es el país con la mayor tasa de fallecidos hasta la fecha (210.192), seguido por Brasil (146.675) y México (79.268). El país más controlado en los casos y por tanto fallecidos es Uruguay con solamente 48 observaciones. Italia, Perú y España comparten el número de fallecidos con alrededor de 35.000 casos.

Inciso b:

Se pide construir un diagrama de barras con los casos del día de hoy (06/10/2020) para el conjunto anteriormente mencionado, por lo que se recurre a rescatar los datos de la fecha actual con el comando "*hoy <- ymd(format(Sys.time(), "%Y-%m-%d"))*" para luego realizar el gráfico (**fig 7.**)

```
# B)
hoy <- ymd(format(Sys.time(), "%Y-%m-%d"))
hoy
datos5=data%>%group_by(Countries_and_territories)%>%summarise(Casos_hoy=cases[data$date
Rep==hoy])
datos5 <- na.omit(datos5)
a1=c('Chile','Peru','Argentina','Brazil','Mexico','Bolivia','Ecuador','Uruguay',
   'China','Italy','Spain','Iran',"United States",'Germany')

datos5. <- datos5 %>%
          group_by(Countries_and_territories) %>%
            slice(1)

p=ggplot(data=filter(datos5.,Countries_and_territories%in%a1),
       aes(x=Countries_and_territories,y=Casos_hoy,fill=Countries_and_territories))+
  geom_bar(stat = "identity")+
  geom_text(aes(label=Casos_hoy), position=position_dodge(width=0.9), vjust=-0.25)
```

p+theme(axis.text.x=element_text(angle=60, hjust=1))

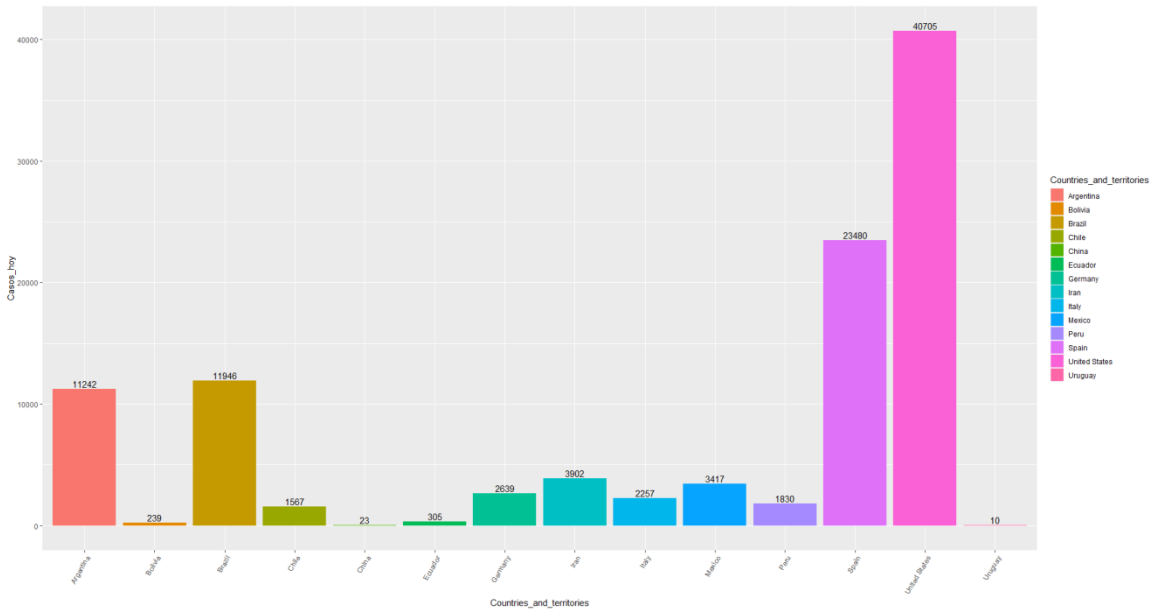

fig 7.

Estados Unidos lidera los casos diarios, seguido por España con la mitad de casos (en comparación con Estados Unidos) y luego Brasil junto con Argentina con la mitad de casos diarios en comparación con España.
Uruguay y China son los países con menores casos diarios actuales.

Actividad 5:

Conociendo la población de España, se selecciona el dato correspondiente a la población de Chile e Italia, se creó el diagrama de curvas solicitado para evaluar la evolución de la pandemia en los primeros 209 días **fig 8.**

```
#ACTIVIDAD 5: Conociendo que la población de España (Spain) es 46723749, agrege
# la curva correspondiente a este país en el diagrama anterior para comparar así
# la evolución de la pandemia en estos tres países: Chile, Italia y España.
# Comente la gráfica obtenida.
##########################################################################################
######

pop_ch=18729160
pop_italy=60431283
pop_Spain=46723749

s1=as.Date("2020-02-22")
s2=s1+days(n-1)
data_Italy=filter(data,Countries_and_territories=='Italy')
dat_italy=filter(data_Italy,dateRep<=s2&dateRep>=s1)

dat_italy_1=select(dat_italy,cases)
dat_italy_1$dias=seq(n,1,by=-1)
dat_italy_1$Pais='Italy'
```

```
head(dat_italy_1)

a=c('Chile')
data_Chile=filter(data,Countries_and_territories%in%a)
(s=min(data_Chile$dateRep))
(q=max(data_Chile$dateRep))
(n=interval(s,q)/days(1))
dat_Chile=select(data_Chile,cases)
dat_Chile$Pais='Chile'
#Generamos una nueva variable que cuente los días desde la primera fecha de reporte
dat_Chile$dias=seq(n,1,by=-1)
head(dat_Chile)

b=c('Spain')
data_Spain=filter(data,Countries_and_territories%in%b)
(s1=min(data_Spain$dateRep))
(q1=max(data_Spain$dateRep))
(n1=interval(s1,q1)/days(1))
dat_Spain=select(data_Spain,cases)
dat_Spain$Pais='Spain'
#Generamos una nueva variable que cuente los días desde la primera fecha de reporte
dat_Spain$dias=seq(n1,0,by=-1)
head(dat_Spain)

dat_Chile =mutate(dat_Chile[-which.max(dat_Chile$cases),],tasa=100000/pop_ch*cases)
dat_italy_1=mutate(dat_italy_1,tasa=100000/pop_italy*cases)
dat_Spain= mutate(dat_Spain,tasa=100000/pop_Spain*cases)
datos=rbind(dat_Chile,dat_italy_1,dat_Spain[64:273,])
ggplot(datos)+geom_point(aes(x=dias,y=tasa,color=Pais),size=2)+
  geom_line(aes(x=dias,y=tasa,color=Pais),size=1)
```

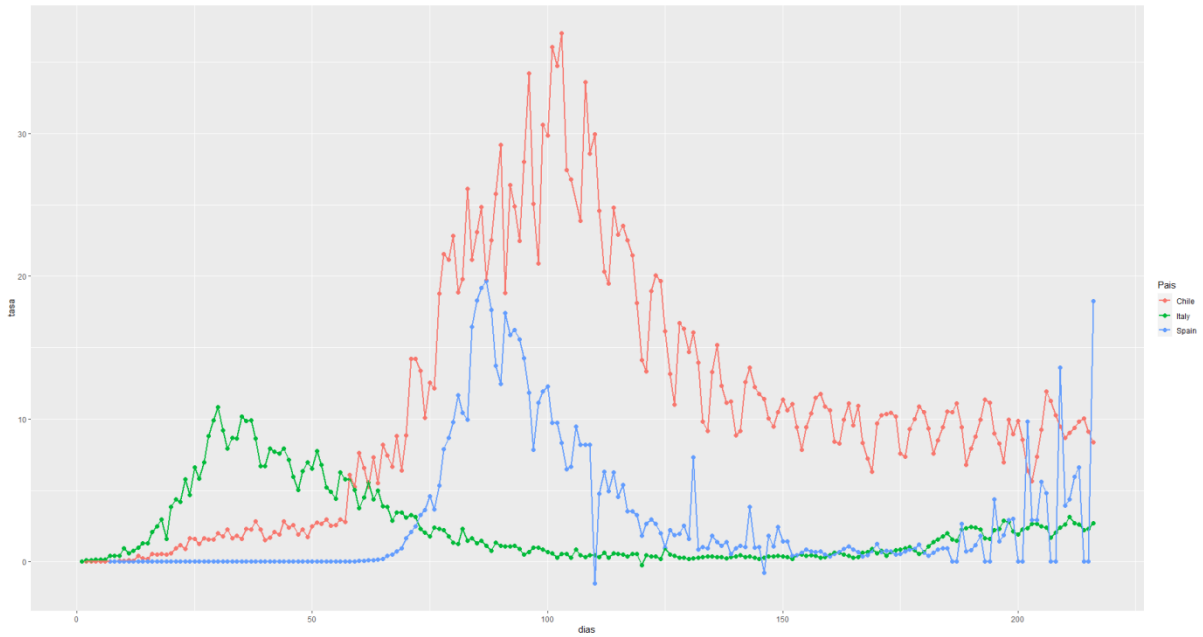

fig 8.

Como España tenía más días de registro, se optó por recortar a los primeros 209 días con tal de obtener una mejor comparación. Se puede Observar que Chile tuvo una mayor tasa de casos registrados en el país cerca de la mitad de los días de observación, mientras que en Italia el comienzo fue más brusco, logró ser rápidamente contenida, aunque actualmente está volviendo a un alza. En el Caso de España, fue parecido a Chile, un brusco aumento alrededor del día 90, pero similar a la tendencia final de Italia, donde comenzó un alza en la tasa dentro de los últimos 20 días.

TEMA 2:
ÍNDICES DE MOVILIDAD EN LAS COMUNAS DE SANTIAGO.

Para el desarrollo de este tema, se descargó la base de datos “IndiceDeMovilidad.csv”, dado en las instrucciones del certamen, con tal de realizar las actividades pertinentes.

Actividad 6:

Inciso a:

Se solicita graficar los índices de movilidad a través del tiempo para las comunas de Santiago, Las Condes y Puente Alto. Para esto se seleccionaron todos los datos correspondientes a cada una de las comunas mencionadas para crear un diagrama de curvas (**fig 9**). Para las tres comunas, se generaron variables con tal de contar los días desde la primera fecha de reporte.

```
# A)
unique(data1$Comuna)
data1$Fecha <- ymd(data1$Fecha)
a2=c('Puente Alto')
data_PA=filter(data1,Comuna%in%a2)
data_PA <- na.omit(data_PA)
(s2=min(data_PA$Fecha))
(q2=max(data_PA$Fecha))
(n2=interval(s2,q2)/days(1))
dat_PA=select(data_PA,IM)
dat_PA$Comuna='Puente Alto'
#Generamos una nueva variable que cuente los días desde la primera fecha de reporte
dat_PA$dias=seq(n2,1,by=-1)
head(dat_PA)

a3=c('Santiago')
data_Stgo=filter(data1,Comuna%in%a3)
dat_Stgo=select(data_Stgo,IM)
dat_Stgo$Comuna='Santiago'
(s3=min(data_Stgo$Fecha))
(q3=max(data_Stgo$Fecha))
(n3=interval(s3,q3)/days(1))
```

```
#Generamos una nueva variable que cuente los días desde la primera fecha de reporte
dat_Stgo$dias=seq(n3,1,by=-1)
head(dat_Stgo)

a4=c('Las Condes')
data_LC=filter(data1,Comuna%in%a4)
dat_LC=select(data_LC,IM)
dat_LC$Comuna='Las Condes'
(s4=min(data_LC$Fecha))
(q4=max(data_LC$Fecha))
(n4=interval(s4,q4)/days(1))
#Generamos una nueva variable que cuente los días desde la primera fecha de reporte
dat_LC$dias=seq(n4,1,by=-1)
head(dat_LC)

datosComunas <- rbind(dat_LC,dat_PA,dat_Stgo)

ggplot(datosComunas)+geom_point(aes(x=dias,y=IM,color=Comuna),size=2)+
  geom_line(aes(x=dias,y=IM,color=Comuna),size=1)
```

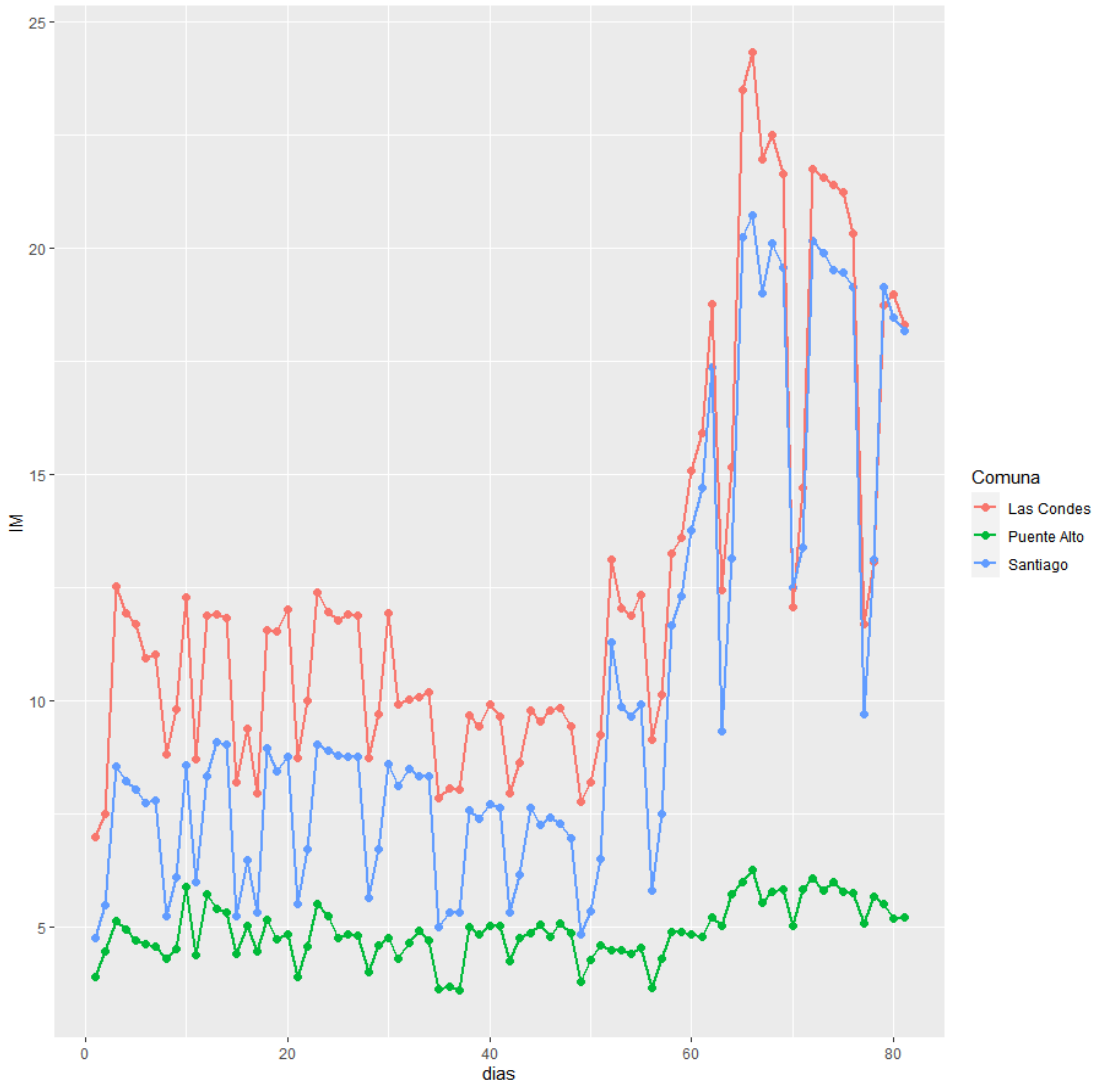

fig 9.

Observando lo entregado por la **fig 9**, se puede concluir que Puente Alto ha sido constante en su tasa de movilidad en los últimos 80 días y con un índice de movilidad mucho menor que las comunas de Las Condes y Santiago, que han tenido un comportamiento similar en su índice de movilidad del día 50 al 80, mientras que antes de estos días, Las Condes mantenía una diferencia de unos 5 puntos en su índice de movilidad respecto a la comuna de Santiago.

Inciso b:

Se pide observar a través de un análisis realizado con una prueba t-student si hubo cambios en los índices de movilidad promedio antes de los cierres de colegios (23/03/2020) y después de la cuarentena (26/03/2020) en la comuna de Santiago. Dado que los cierres de colegios se realizaron el 23/03/2020, se optó por recopilar los datos necesarios hasta el día 22/03/2020. Además, como se pide recopilar los datos después de la cuarentena, se optó por seleccionar estos datos a partir del 27/03/2020.

```
# B)
Stgo=c('Santiago')
data_Stgo1=filter(data1,Comuna%in%Stgo)
Antes <- data_Stgo1[1:26,]
Despues <- data_Stgo1[31:81,]
t.test(Antes$IM,Despues$IM)
```

Tras haber realizado la prueba t-student de los datos en estudio, obtuvimos un p-valor del test igual a $4{,}102 * 10^{-10}$, muy cercano a cero y, consecuentemente, menor al nivel de significancia (0,05) por lo que podemos concluir que sí existen diferencias significativas en el cambio de los índices de movilidad anterior al cierre de colegios y posterior a la cuarentena en la comuna de Santiago.

TEMA 3:
ANÁLISIS DE COMPONENTES PRINCIPALES.

Para el desarrollo de este tema, se descargó la base de datos “temperature.csv”, dado en las instrucciones del certamen con tal de realizar las actividades pertinentes. Esta base de datos contiene los registros de temperaturas en distintas ciudades europeas (principalmente capitales) medidas en grados Celsius. Además, esta base de datos contiene datos geográficos tales como “Latitud” y “Longitud” y “Área”.

Actividad 7:

Se solicita realizar un análisis de componentes principales con tal de encontrar patrones de agrupación en la data “temperature.csv” y hallar (si existe) una relación entre estos patrones con el área y cercanía geográfica de las ciudades.

Primero realizamos el cálculo de los componentes principales. A través del porcentaje de varianza explicada por cada componente principal (**fig 10.**), podemos apreciar que el primer componente principal (PC1) explica casi un 80% de la data, mientras que el segundo componente principal (PC2) cercano a un 15%. Para una mejor visualización, se creó un gráfico de barras y uno de tortas, se usará el de tortas (**fig 11.**) para mejor comprensión del caso.

```
data2 <- read.csv("C:/Users/jorge/OneDrive/Escritorio/SEM 6/TIN/Certamen1/temperature.csv")
#################################################################################################
#ACTIVIDAD 7
#Con ayuda de Análisis de Componentes Principales, encuentre patrones de agrupación en la data de temperaturas y halle (si existe) una relación de estos patrones con el área y cercanía geográfica de las ciudades.
#Comente sobre los patrones hallados con detalle.
#Para complementar el estudio, visualice los datos en un mapa. Por ejemplo, puede emplear:
#################################################################################################

# Transformar a header las filas (primera variable)
row.names(data2)=data2$X
data2$X=NULL

# calcular componentes principales
pcdata2 <- prcomp(data2[,-17], scale=T)

# analisis de componentes principales
names(pcdata2)
pcdata2$center # valores promedios de los componentes principales
pcdata2$rotation # vectores propios colgados para cada componente (matriz gamma) / se diseña la ecuacion
pcdata2$x # nueva tabla evaluado con los valores (sirve para crear los diagramas) / teorema espectral
pcdata2$sdev # desviacion estandar de los componentes principales
pcdata2$scale # valores promedios divididos por las desviaciones estandar (variables escaladas)

# porcentajes de varianza explicada por cada componente          principal
pcdata2.var <- (pcdata2$sdev)^2
pve <- pcdata2.var/sum(pcdata2.var)
pve
plot(cumsum(pve), xlab="Componente Principal", ylab="Proporcion acumulada de varianza explicada", ylim=c(0,1),type='b')

# grafico de barras componentes principales
```

```
library(ggplot2)
df <- data.frame(PC=c("PC1", "PC2", "PC3", "PC4", "PC5", "PC6", "PC7", "PC8", "PC9","PC10",
                      "PC11", "PC12", "PC13", "PC14", "PC15", "PC16"),
                 pve=pve)
ggplot(data=df,aes(x=PC,y=pve,fill=PC))+
  geom_bar(stat = "identity", show.legend = F)+
  geom_text(aes(label=pve), position=position_dodge(width=0.9), vjust=-0.25,size=3)

# grafico torta
library(plotly)
plot_ly(df, labels = ~PC, values = ~pve, type = "pie") %>%
  layout(title = "PCA para Temperaturas",
         xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),
         yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE))
```

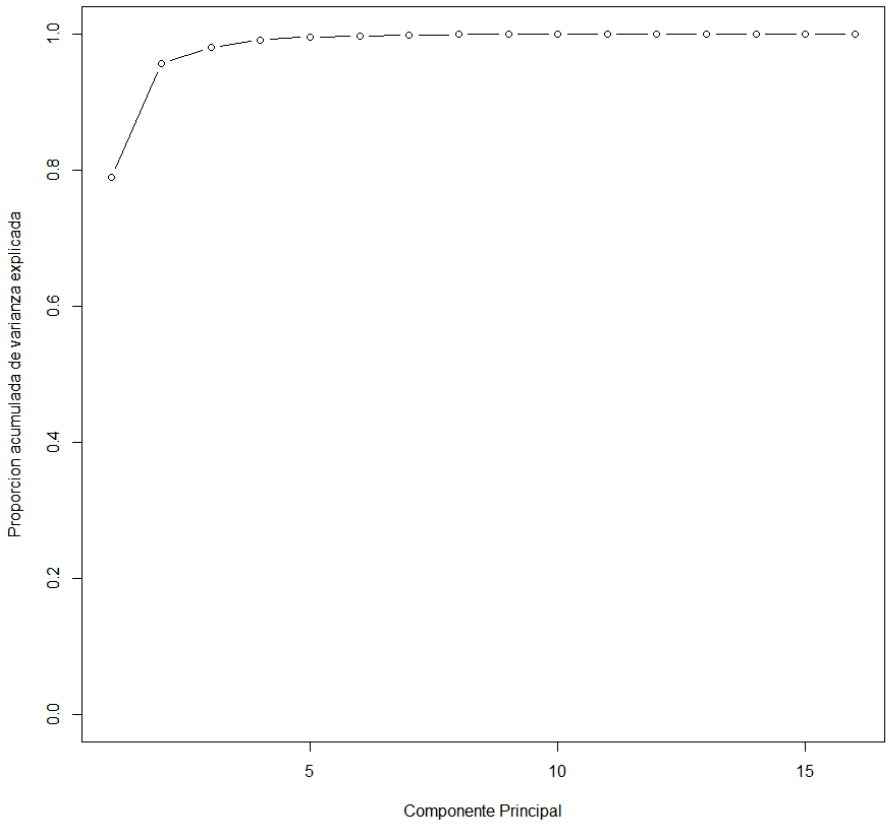

fig 10.

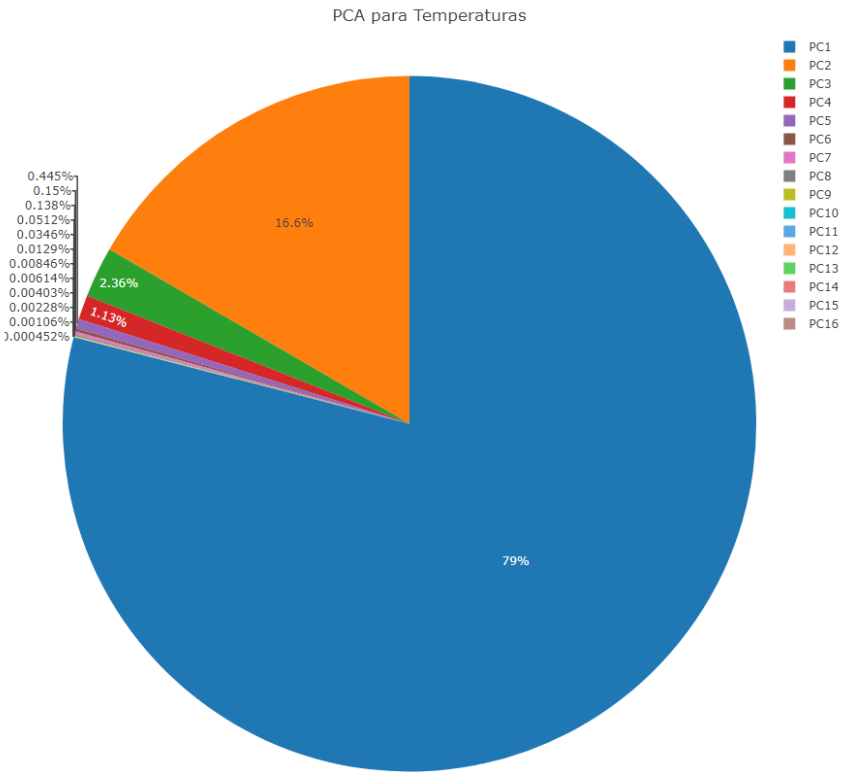

fig 11.

Como PC1= 79% y PC2=16,6%, nuestro análisis se basará en la relevancia de estos dos componentes, ya que estos explican aproximadamente el 95% de la base de datos. Luego, se procede a crear un diagrama de dispersión de los componentes principales seleccionados (**fig 12.**). Es importante mencionar que, como el primer componente principal (PC1) explica sobre el 70% de los datos, bastaría con analizar sólo este para encontrar los patrones solicitados por el ejercicio. Sin embargo, se preferirá trabajar con los dos primeros, para tener una mayor objetividad sobre la base.

```
# diagrama dispersión componentes principales
par(mfrow = c(1, 1))
y <- pcdata2$x
par(mfrow=c(1,1))
plot(y[,c(1,2)],type="n",main="PC1 vs. PC2")
text(y[,c(1,2)],row.names(data2),cex=0.7)

# diagrama biplot
biplot(pcdata2)
```

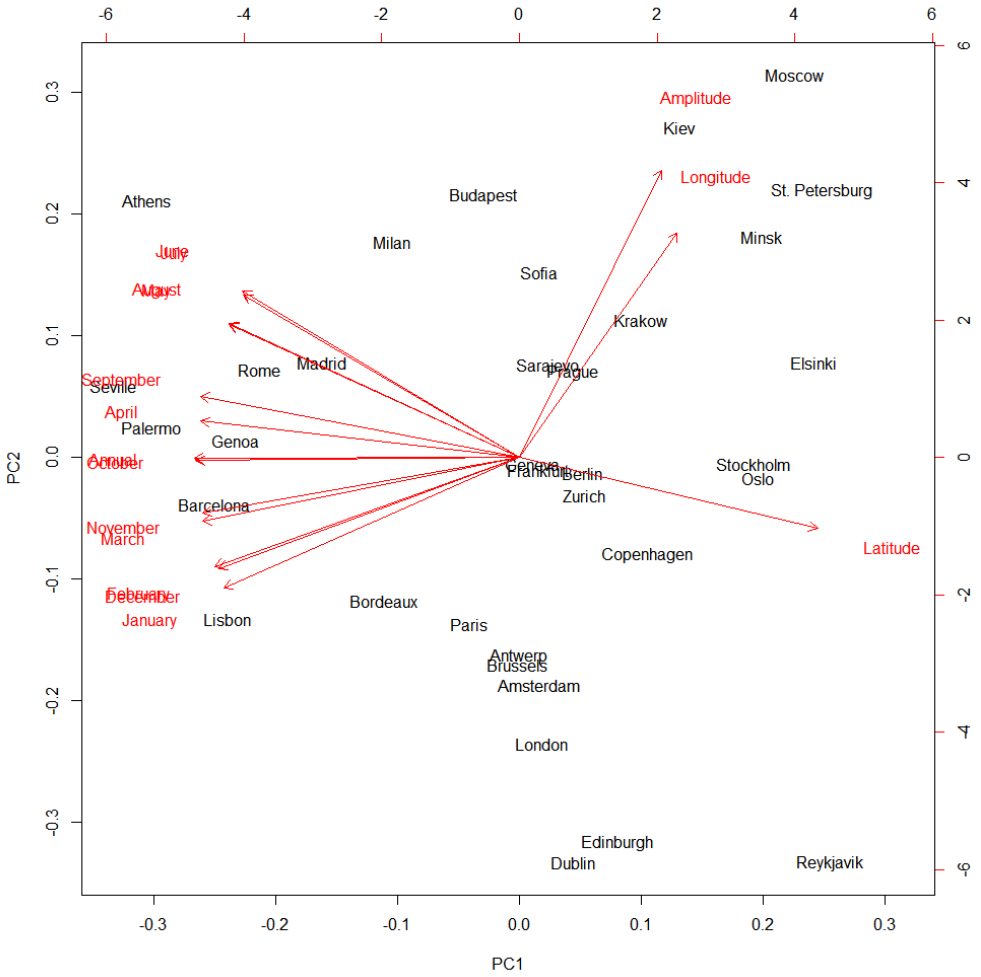

fig 12.

A partir del “biplot” de la **fig 12.** se puede apreciar que la significancia de la variable de “Latitud” en el PC1 explica que las ciudades de las regiones del Este son las más frías, a su vez relacionado con la “Longitud”, que mientras más al Norte se encuentra la ciudad menor será su temperatura promedio, por la parte de PC2, los meses más fríos en Europa son diciembre, enero y febrero, mientras que los más calientes son junio, julio y agosto.
Por lo tanto, las ciudades con temperaturas promedio más altas, se encuentran principalmente, tanto en el Sur, como al Suroeste del continente.
En lo que se refiere a la significancia de PC1 y PC2, las variables “Latitud” y “Longitud” son las más relevantes en el estudio de variación de temperatura, ya que el primer componente principal explica un 80% de los datos, mientras que PC2 un 16% como se mencionó anteriormente.
Lo anterior puede verse demostrado en el mapa (**fig 13.**), indicando las temperaturas promedio anuales para cada ciudad hallada en él.

```
# reconocimiento de patrones separando los grupos en clusters
# determina la distancia entre las observaciones
pcdata2.dist <- dist(y)
# aplica un método jerárquico para construir clusters:
```

```
pcdata2.hclust <- hclust(pcdata2.dist,method='ward.D')
# indica que usaremos 4 clusters
grupos.4 <- cutree(pcdata2.hclust,4)
datos <- data.frame(data2,grupos.4=grupos.4)
# Graficamos PC1 vs. PC2
plot(y[, 1], y[, 2], pch =datos$grupos.4, col = datos$grupos.4,
    xlab = "PC1", ylab = "PC2", main = "Primero vs. Segundo PC",
    cex.lab = 1.2, cex.axis = 1.2, cex.main = 1.8,cex=1.5)
text(y[,c(1,2)],row.names(datos),cex=0.7)

#library(leaflet)

m <- leaflet() %>%

  addTiles() %>%

  addMarkers(lng=datos$Longitude, lat=datos$Latitude, label=datos$Annual,

          labelOptions = labelOptions(noHide = T,

                                      textsize = "12px",opacity = 0.7))
```

m

fig 13.