Geo_COVID19

Ximena Fernández

14 de abril de 2020

1 Geographical evolution of COVID19

La idea es hacer un análisis de datos de índole geográfico sobre información sobre los casos de Covid19.

```
[69]: options(warn=-1)
```

1.0.1 Bueno, ya, dame la data

Los datos están en el siguiente link, que se actualiza a diario! Así que pueden ir jugando con este análisis a medida que pasa el tiempo.

```
[70]: #importamos tabla
      corona <- read.csv("./novel-corona-virus-2019-dataset/covid_19_data.csv",␣
       ↪'stringsAsFactors'=FALSE)
      head(corona)
```

	SNo	ObservationDate	Province.State	Country.Region	Last.Update	Confirmed	Deaths	Recovered
	<int>	<chr>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>
1	1	01/22/2020	Anhui	Mainland China	1/22/2020 17:00	1	0	0
2	2	01/22/2020	Beijing	Mainland China	1/22/2020 17:00	14	0	0
3	3	01/22/2020	Chongqing	Mainland China	1/22/2020 17:00	6	0	0
4	4	01/22/2020	Fujian	Mainland China	1/22/2020 17:00	1	0	0
5	5	01/22/2020	Gansu	Mainland China	1/22/2020 17:00	0	0	0
6	6	01/22/2020	Guangdong	Mainland China	1/22/2020 17:00	26	0	0

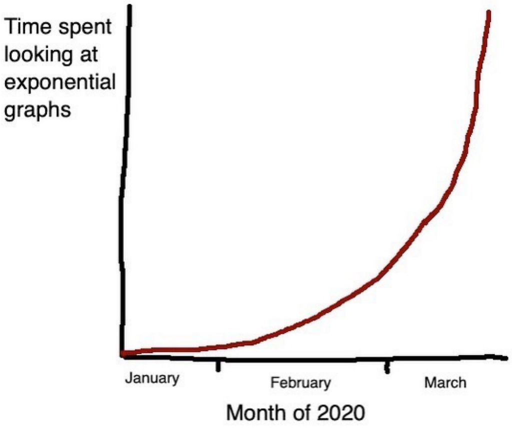
```
[71]: #arreglamos algunas columnas y datos
      corona$ObservationDate <- as.Date(corona$ObservationDate, format = "%m/%d/%y")
      corona$'Country.Region'[corona$'Country.Region'  == 'Mainland China'] <- 'China'
      corona$'Country.Region'[corona$'Country.Region'  == 'US'] <- 'United States of␣
       ↪America'
```

```
[72]: head(corona)
```

	SNo	ObservationDate	Province.State	Country.Region	Last.Update	Confirmed	Deaths	Recovered
	<int>	<date>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>
1	1	2020-01-22	Anhui	China	1/22/2020 17:00	1	0	0
2	2	2020-01-22	Beijing	China	1/22/2020 17:00	14	0	0
3	3	2020-01-22	Chongqing	China	1/22/2020 17:00	6	0	0
4	4	2020-01-22	Fujian	China	1/22/2020 17:00	1	0	0
5	5	2020-01-22	Gansu	China	1/22/2020 17:00	0	0	0
6	6	2020-01-22	Guangdong	China	1/22/2020 17:00	26	0	0

1.1 Trends

Queremos estudiar la serie temporal corespondiente a los casos acumulados de Covid-19



1.1.1 Evolución de casos confirmados de coronavirus por día

[73]:
```
library(dplyr)
```

'dplyr' es como el pandas de R. Permite trabajar con data frames de manera eficiente.

Documentación: dplyr

[74]:
```
corona %>%
  group_by(ObservationDate) %>%
  summarise(TotalConfirmed = sum(Confirmed)) -> confirmed
```

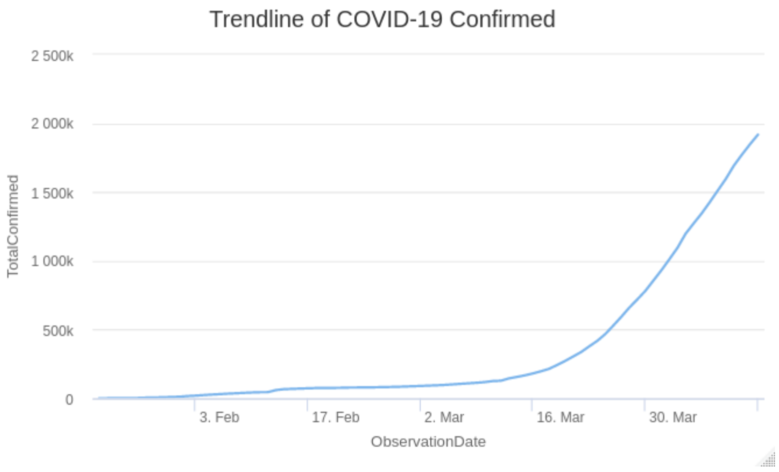
[75]:
```
head(confirmed)
```

ObservationDate <date>	TotalConfirmed <dbl>
2020-01-22	555
2020-01-23	653
2020-01-24	941
2020-01-25	1438
2020-01-26	2118
2020-01-27	2927

[77]:
```
library(highcharter)
```

Documentación highcharter

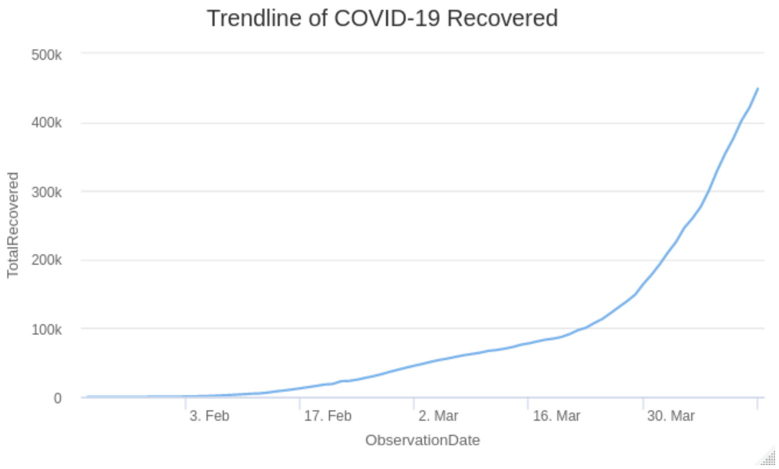
[78]:
```
confirmed %>%
  hchart("line", hcaes(x = ObservationDate, y = TotalConfirmed)) %>%
  hc_title(text = "Trendline of COVID-19 Confirmed")
```



1.1.2 Evolución de casos recuperados de coronavirus por día

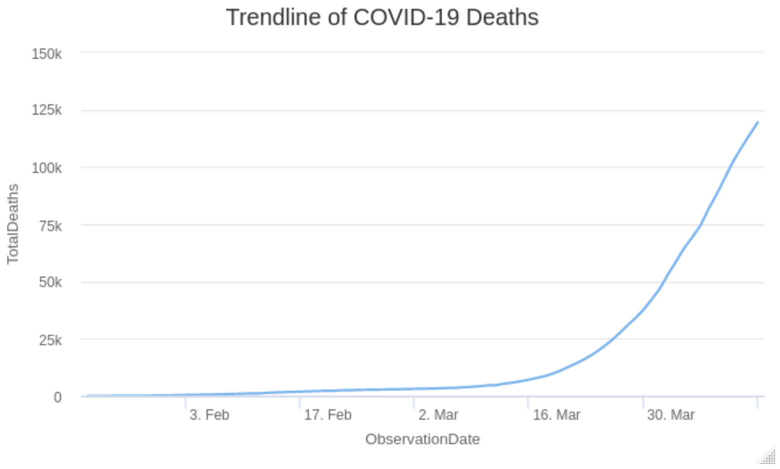
[79]:
```
corona %>%
  group_by(ObservationDate) %>%
  summarise(TotalRecovered = sum(Recovered)) -> recovered
```

[80]:
```
recovered %>%
  hchart("line",hcaes(x = ObservationDate, y = TotalRecovered)) %>%
  hc_title(text = "Trendline of COVID-19 Recovered")
```



1.1.3 Evolución de casos fallecidos de coronavirus por día

[81]:
```
corona %>%
  group_by(ObservationDate) %>%
  summarise(TotalDeaths = sum(Deaths)) -> deaths

deaths %>%
  hchart("line", hcaes(x = ObservationDate, y = TotalDeaths)) %>%
  hc_title(text = "Trendline of COVID-19 Deaths")# %>%
#hc_add_theme(hc_theme_538())
```



1.1.4 La placa de Worldometer

[82]:
```
#filtramos el último día
corona_latest <- corona %>%
filter(ObservationDate %in% max(corona$ObservationDate))
head(corona_latest)
```

	SNo <int>	ObservationDate <date>	Province.State <chr>	Country.Region <chr>	Last.Update <chr>	Confirmed <dbl>	Deaths <dbl>	Recovered <dbl>
1	14492	2020-04-13		Afghanistan	2020-04-13 23:15:42	665	21	32
2	14493	2020-04-13		Albania	2020-04-13 23:15:42	467	23	232
3	14494	2020-04-13		Algeria	2020-04-13 23:15:42	1983	313	601
4	14495	2020-04-13		Andorra	2020-04-13 23:15:42	646	29	128
5	14496	2020-04-13		Angola	2020-04-13 23:15:42	19	2	4
6	14497	2020-04-13		Antigua and Barbuda	2020-04-13 23:15:42	23	2	0

[83]:
```
#sumamos confirmados, recuperados y fallecidos
confirmed <- sum(corona_latest$Confirmed)
recovered <- sum(corona_latest$Recovered)
deaths <- sum(corona_latest$Deaths)
```

[84]:
```
cat('TOTAL CONFIRMED:', confirmed, '\n')
cat('TOTAL RECOVERED: ', recovered, '\n')
cat('TOTAL DEATHS:     ', deaths, '\n')
```

```
TOTAL CONFIRMED: 1917320
TOTAL RECOVERED:  448655
TOTAL DEATHS:     119483
```

1.2 Análisis por país

[85]:

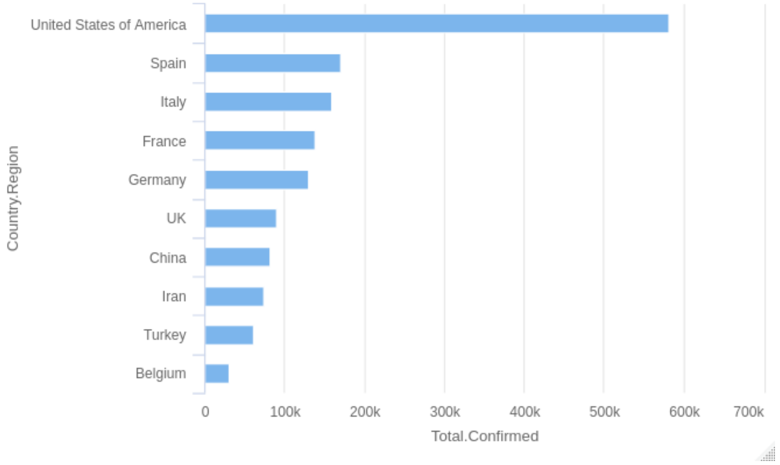
```
corona_latest %>%
  filter(!Country.Region %in% 'Others') %>%
  group_by(Country.Region) %>%
  summarise(Total.Confirmed = sum(Confirmed)) %>%
  filter(Total.Confirmed > 0) %>%
  mutate(Log.Total.Confirmed = log(Total.Confirmed)) -> countries_confirmed

head(countries_confirmed %>% arrange(desc(Total.Confirmed)))
```

Country.Region <chr>	Total.Confirmed <dbl>	Log.Total.Confirmed <dbl>
United States of America	580619	13.27185
Spain	170099	12.04414
Italy	159516	11.97990
France	137875	11.83410
Germany	130072	11.77584
UK	89570	11.40278

1.2.1 Rankings

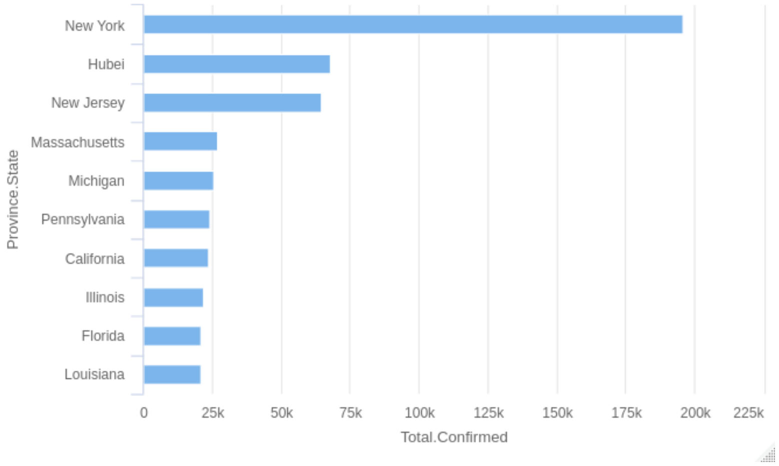
[86]:

```
countries_confirmed %>%
  arrange(desc(Total.Confirmed)) %>%
  head(10) %>%
  hchart("bar",hcaes(x = 'Country.Region',  y =Total.Confirmed)) #%>%
  #hc_add_theme(hc_theme_darkunica())
```



[87]:

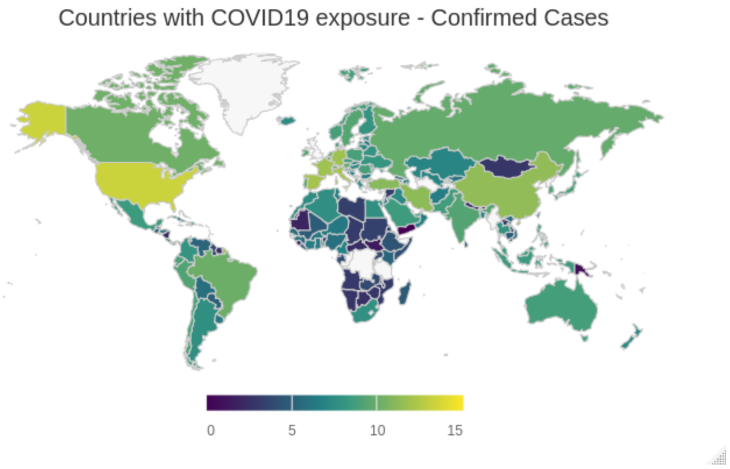
```
corona_latest %>%
  filter(!Province.State %in% c('Others', '')) %>%
  group_by(Province.State) %>%
  summarise(Total.Confirmed = sum(Confirmed)) %>%
  arrange(desc(Total.Confirmed)) %>%
  head(10) %>%
  hchart("bar",hcaes(x = 'Province.State',  y =Total.Confirmed)) #%>%
  #hc_add_theme(hc_theme_darkunica())
```



1.2.2 Algunos Mapitas

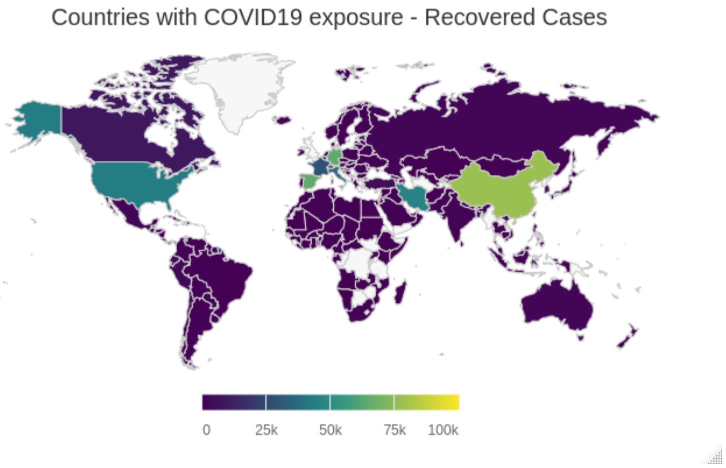
[88]:

```
data(worldgeojson, package = "highcharter")

highchart() %>%
  hc_add_series_map(worldgeojson, countries_confirmed, value = 'Log.Total.
↪Confirmed', joinBy = c('name','Country.Region'))  %>%
  hc_colorAxis(stops = color_stops()) %>%
  hc_title(text = "Countries with COVID19 exposure - Confirmed Cases")
```

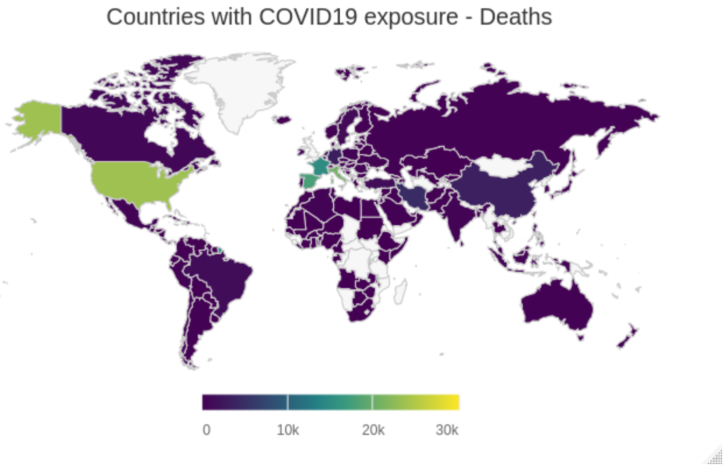


[89]:

```
corona_latest %>%
  group_by(Country.Region) %>%
  summarise(Total.Recovered = sum(Recovered)) %>%
  filter(Total.Recovered > 0)  -> countries_recovered

highchart() %>%
  hc_add_series_map(worldgeojson, countries_recovered, value = 'Total.
↪Recovered', joinBy = c('name','Country.Region'))  %>%
  hc_colorAxis(stops = color_stops()) %>%
  hc_title(text = "Countries with COVID19 exposure - Recovered Cases")
```


Countries with COVID19 exposure - Recovered Cases

```
[90]: corona_latest %>%
        group_by(Country.Region) %>%
        summarise(Total.Deaths = sum(Deaths)) %>%
        filter(Total.Deaths > 0) -> countries_deaths

      highchart() %>%
        hc_add_series_map(worldgeojson, countries_deaths, value = 'Total.Deaths',␣
      ↪joinBy = c('name','Country.Region'))  %>%
        hc_colorAxis(stops = color_stops()) %>%
        hc_title(text = "Countries with COVID19 exposure - Deaths")
```


Countries with COVID19 exposure - Deaths

Se ve que EEUU le saca una vuelta al resto de los países, pero esto puede tener que ver con su población y entonces los casos confirmados de cada país no están a escalas comparables.

***Estrategia:* tener en cuenta la población de cada país** Fuente de datos: UNData, *'Population, surface area and density'* dataset

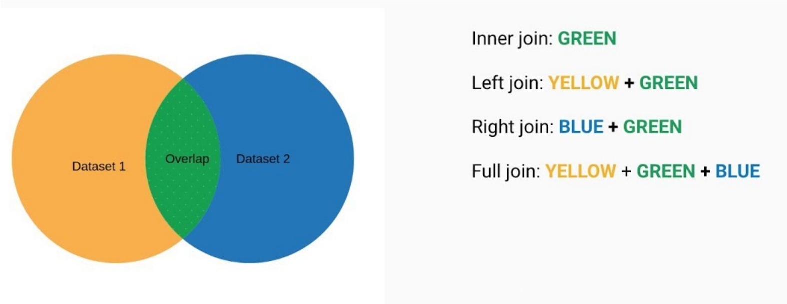
```
[91]: population <- read.csv("./WPP2019_TotalPopulationBySex.csv")
```

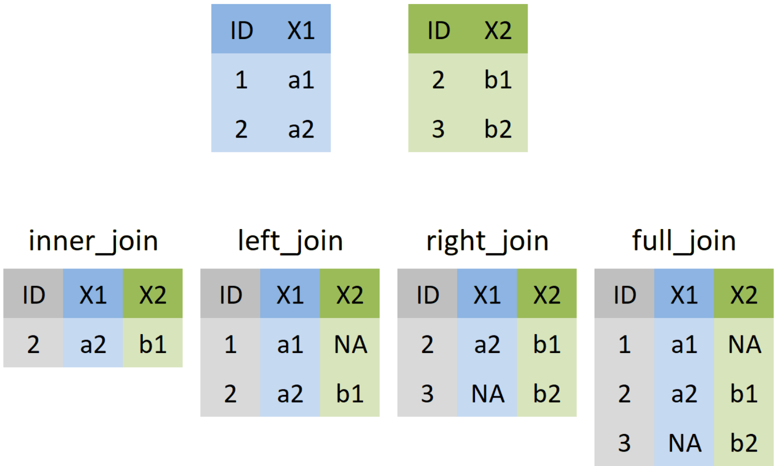
```
[92]: head(population)
```

	LocID <int>	Location <fct>	VarID <int>	Variant <fct>	Time <int>	MidPeriod <dbl>	PopMale <dbl>	PopFemale <dbl>	PopTotal <dbl>	PopDensity <dbl>
1	4	Afghanistan	2	Medium	1950	1950.5	4099.243	3652.874	7752.117	11.874
2	4	Afghanistan	2	Medium	1951	1951.5	4134.756	3705.395	7840.151	12.009
3	4	Afghanistan	2	Medium	1952	1952.5	4174.450	3761.546	7935.996	12.156
4	4	Afghanistan	2	Medium	1953	1953.5	4218.336	3821.348	8039.684	12.315
5	4	Afghanistan	2	Medium	1954	1954.5	4266.484	3884.832	8151.316	12.486
6	4	Afghanistan	2	Medium	1955	1955.5	4318.945	3952.047	8270.992	12.669

```
[93]: population %>%
      filter(Time == '2019', PopTotal>1000) -> population_2019
```

Hacemos un join.





ID	X1
1	a1
2	a2

ID	X2
2	b1
3	b2

inner_join

ID	X1	X2
2	a2	b1

left_join

ID	X1	X2
1	a1	NA
2	a2	b1

right_join

ID	X1	X2
2	a2	b1
3	NA	b2

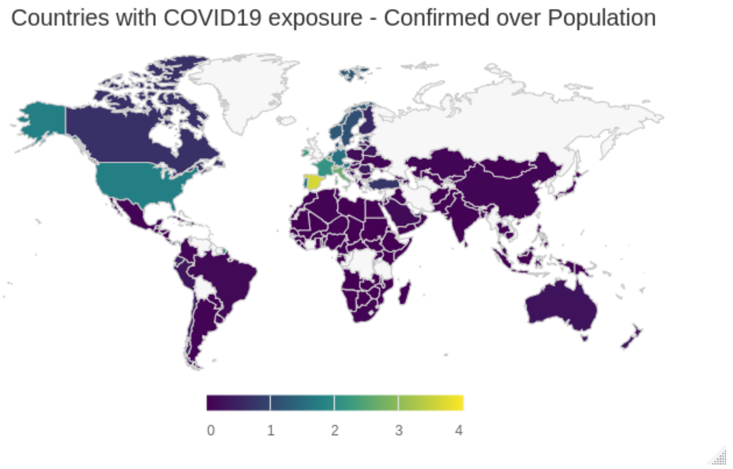
full_join

ID	X1	X2
1	a1	NA
2	a2	b1
3	NA	b2

```
[58]: population_2019 %>%
      select(Location, PopTotal) %>%
      right_join(countries_confirmed, by = c("Location" = "Country.Region")) %>%
      mutate(Confirmed.over.Population = Total.Confirmed/PopTotal)  %>%
      arrange(desc(Confirmed.over.Population)) -> countries_confirmed_pop

      head(countries_confirmed_pop, 10)
```

	Location	PopTotal	Total.Confirmed	Log.Total.Confirmed	Confirmed.over.Population
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	Spain	46736.782	170099	12.044136	3.639510
2	Switzerland	8591.361	25688	10.153779	2.989980
3	Belgium	11539.326	30589	10.328396	2.650848
4	Italy	60550.092	159516	11.979900	2.634447
5	Ireland	4882.498	10647	9.273033	2.180646
6	France	65129.731	137875	11.834103	2.116929
7	United States of America	329064.917	580619	13.271850	1.764451
8	Portugal	10226.178	16934	9.737079	1.655946
9	Austria	8955.108	14041	9.549737	1.567932
10	Netherlands	17097.123	26710	10.192793	1.562251

```
[56]: data(worldgeojson, package = "highcharter")

      highchart() %>%
        hc_add_series_map(worldgeojson, countries_confirmed_pop, value = 'Confirmed.
       ↪over.Population', joinBy = c('name','Location'))  %>%
        hc_colorAxis(stops = color_stops()) %>%
        hc_title(text = "Countries with COVID19 exposure - Confirmed over Population")
```



Podemos ver ahora que la situación crítica se concentra en Europa.

2 Conclusiones

La exploración datos es una manera de abordar la búqueda de información a partir de una base de datos. Creo que tiene dos patas importantes: formularnos las preguntas correctas y pertinentes, y tratar de responderlas de la manera más simple pero acertada posible. Para ello, los gráficos son una manera increíble de visualizar muchos dato de manera agregada. La manera de graficar es muy personal. Les dejo aquí una página donde pueden encontrar la que más les guste R Graph Gallery.