Untitled

Joshelyn Intriago

18 de mayo de 2021

loc<-read.csv("locations.csv",header=T,sep=",")  
owid <-read.csv("owid-covid-data.csv",header=T,sep=",")  
who <- read.csv("WHO COVID-19 global table data May 10th 2021 at 3.48.04 AM.csv", header=T,sep=",")  
vac <- read.csv("vaccinations-by-manufacturer.csv")  
contact <-read.csv("covid-contact-tracing.csv",header=T,sep=",")  
test\_p <- read.csv("covid-19-testing-policy.csv")  
vac\_p <- read.csv("covid-vaccination-policy.csv")  
income <- read.csv("income-support-covid.csv")  
school <- read.csv("school-closures-covid.csv")  
work <- read.csv("workplace-closures-covid.csv")  
home <- read.csv("stay-at-home-covid.csv")

#### vamos a recodificar la variable continente porque contiene valores en blancos que correspondes a determinados continentes

## [1] 87099 59

## 'data.frame': 87099 obs. of 59 variables:  
## $ iso\_code : Factor w/ 219 levels "ABW","AFG","AGO",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ continent : Factor w/ 7 levels "","Africa","Asia",..: 3 3 3 3 3 3 3 3 3 3 ...  
## $ location : Factor w/ 219 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ date : Factor w/ 496 levels "2020-01-01","2020-01-02",..: 55 56 57 58 59 60 61 62 63 64 ...  
## $ total\_cases : num 1 1 1 1 1 1 1 1 2 4 ...  
## $ new\_cases : num 1 0 0 0 0 0 0 0 1 2 ...  
## $ new\_cases\_smoothed : num NA NA NA NA NA 0.143 0.143 0 0.143 0.429 ...  
## $ total\_deaths : num NA NA NA NA NA NA NA NA NA NA ...  
## $ new\_deaths : num NA NA NA NA NA NA NA NA NA NA ...  
## $ new\_deaths\_smoothed : num NA NA NA NA NA 0 0 0 0 0 ...  
## $ total\_cases\_per\_million : num 0.026 0.026 0.026 0.026 0.026 0.026 0.026 0.026 0.051 0.103 ...  
## $ new\_cases\_per\_million : num 0.026 0 0 0 0 0 0 0 0.026 0.051 ...  
## $ new\_cases\_smoothed\_per\_million : num NA NA NA NA NA 0.004 0.004 0 0.004 0.011 ...  
## $ total\_deaths\_per\_million : num NA NA NA NA NA NA NA NA NA NA ...  
## $ new\_deaths\_per\_million : num NA NA NA NA NA NA NA NA NA NA ...  
## $ new\_deaths\_smoothed\_per\_million : num NA NA NA NA NA 0 0 0 0 0 ...  
## $ reproduction\_rate : num NA NA NA NA NA NA NA NA NA NA ...  
## $ icu\_patients : num NA NA NA NA NA NA NA NA NA NA ...  
## $ icu\_patients\_per\_million : num NA NA NA NA NA NA NA NA NA NA ...  
## $ hosp\_patients : num NA NA NA NA NA NA NA NA NA NA ...  
## $ hosp\_patients\_per\_million : num NA NA NA NA NA NA NA NA NA NA ...  
## $ weekly\_icu\_admissions : num NA NA NA NA NA NA NA NA NA NA ...  
## $ weekly\_icu\_admissions\_per\_million : num NA NA NA NA NA NA NA NA NA NA ...  
## $ weekly\_hosp\_admissions : num NA NA NA NA NA NA NA NA NA NA ...  
## $ weekly\_hosp\_admissions\_per\_million : num NA NA NA NA NA NA NA NA NA NA ...  
## $ new\_tests : num NA NA NA NA NA NA NA NA NA NA ...  
## $ total\_tests : num NA NA NA NA NA NA NA NA NA NA ...  
## $ total\_tests\_per\_thousand : num NA NA NA NA NA NA NA NA NA NA ...  
## $ new\_tests\_per\_thousand : num NA NA NA NA NA NA NA NA NA NA ...  
## $ new\_tests\_smoothed : num NA NA NA NA NA NA NA NA NA NA ...  
## $ new\_tests\_smoothed\_per\_thousand : num NA NA NA NA NA NA NA NA NA NA ...  
## $ positive\_rate : num NA NA NA NA NA NA NA NA NA NA ...  
## $ tests\_per\_case : num NA NA NA NA NA NA NA NA NA NA ...  
## $ tests\_units : Factor w/ 5 levels "","people tested",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ total\_vaccinations : num NA NA NA NA NA NA NA NA NA NA ...  
## $ people\_vaccinated : num NA NA NA NA NA NA NA NA NA NA ...  
## $ people\_fully\_vaccinated : num NA NA NA NA NA NA NA NA NA NA ...  
## $ new\_vaccinations : num NA NA NA NA NA NA NA NA NA NA ...  
## $ new\_vaccinations\_smoothed : num NA NA NA NA NA NA NA NA NA NA ...  
## $ total\_vaccinations\_per\_hundred : num NA NA NA NA NA NA NA NA NA NA ...  
## $ people\_vaccinated\_per\_hundred : num NA NA NA NA NA NA NA NA NA NA ...  
## $ people\_fully\_vaccinated\_per\_hundred : num NA NA NA NA NA NA NA NA NA NA ...  
## $ new\_vaccinations\_smoothed\_per\_million: num NA NA NA NA NA NA NA NA NA NA ...  
## $ stringency\_index : num 8.33 8.33 8.33 8.33 8.33 ...  
## $ population : num 38928341 38928341 38928341 38928341 38928341 ...  
## $ population\_density : num 54.4 54.4 54.4 54.4 54.4 ...  
## $ median\_age : num 18.6 18.6 18.6 18.6 18.6 18.6 18.6 18.6 18.6 18.6 ...  
## $ aged\_65\_older : num 2.58 2.58 2.58 2.58 2.58 ...  
## $ aged\_70\_older : num 1.34 1.34 1.34 1.34 1.34 ...  
## $ gdp\_per\_capita : num 1804 1804 1804 1804 1804 ...  
## $ extreme\_poverty : num NA NA NA NA NA NA NA NA NA NA ...  
## $ cardiovasc\_death\_rate : num 597 597 597 597 597 ...  
## $ diabetes\_prevalence : num 9.59 9.59 9.59 9.59 9.59 9.59 9.59 9.59 9.59 9.59 ...  
## $ female\_smokers : num NA NA NA NA NA NA NA NA NA NA ...  
## $ male\_smokers : num NA NA NA NA NA NA NA NA NA NA ...  
## $ handwashing\_facilities : num 37.7 37.7 37.7 37.7 37.7 ...  
## $ hospital\_beds\_per\_thousand : num 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...  
## $ life\_expectancy : num 64.8 64.8 64.8 64.8 64.8 ...  
## $ human\_development\_index : num 0.511 0.511 0.511 0.511 0.511 0.511 0.511 0.511 0.511 0.511 ...

## [1] Asia Europe Africa North America  
## [6] South America Oceania   
## Levels: Africa Asia Europe North America Oceania South America

## [1] AFG OWID\_AFR ALB DZA AND AGO AIA ATG   
## [9] ARG ARM ABW OWID\_ASI AUS AUT AZE BHS   
## [17] BHR BGD BRB BLR BEL BLZ BEN BMU   
## [25] BTN BOL BIH BWA BRA BRN BGR BFA   
## [33] BDI KHM CMR CAN CPV CYM CAF TCD   
## [41] CHL CHN COL COM COG CRI CIV HRV   
## [49] CUB CUW CYP CZE COD DNK DJI DMA   
## [57] DOM ECU EGY SLV GNQ ERI EST SWZ   
## [65] ETH OWID\_EUR OWID\_EUN FRO FLK FJI FIN FRA   
## [73] GAB GMB GEO DEU GHA GIB GRC GRL   
## [81] GRD GTM GGY GIN GNB GUY HTI HND   
## [89] HKG HUN ISL IND IDN OWID\_INT IRN IRQ   
## [97] IRL IMN ISR ITA JAM JPN JEY JOR   
## [105] KAZ KEN OWID\_KOS KWT KGZ LAO LVA LBN   
## [113] LSO LBR LBY LIE LTU LUX MAC MDG   
## [121] MWI MYS MDV MLI MLT MHL MRT MUS   
## [129] MEX FSM MDA MCO MNG MNE MSR MAR   
## [137] MOZ MMR NAM NRU NPL NLD NZL NIC   
## [145] NER NGA OWID\_NAM MKD OWID\_CYN NOR OWID\_OCE OMN   
## [153] PAK PSE PAN PNG PRY PER PHL POL   
## [161] PRT QAT ROU RUS RWA SHN KNA LCA   
## [169] VCT WSM SMR STP SAU SEN SRB SYC   
## [177] SLE SGP SVK SVN SLB SOM ZAF OWID\_SAM  
## [185] KOR SSD ESP LKA SDN SUR SWE CHE   
## [193] SYR TWN TJK TZA THA TLS TGO TON   
## [201] TTO TUN TUR TCA UGA UKR ARE GBR   
## [209] USA URY UZB VUT VAT VEN VNM OWID\_WRL  
## [217] YEM ZMB ZWE   
## 219 Levels: ABW AFG AGO AIA ALB AND ARE ARG ARM ATG AUS AUT AZE BDI BEL ... ZWE

### geo

### primero vamos a obtener la lista de países en la data para generar coordenadas con ayuda de google maps

owid <- owid %>% select(location)  
owid <- owid[!duplicated(owid), ]  
owid <- as.data.frame(owid)  
df <- owid %>% filter(!owid %in% c("Africa", "Asia" , "Europe" , "European Union" , "International" ,  
 "North America" , "Oceania" , "South America" , "World"))  
  
df$owid <- as.character(df$owid)  
register\_google(key = "AIzaSyCj2lcM7XmXotibRN6jA9jrdYMN8Fw2thQ")  
df2 <- mutate\_geocode(df, owid)

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Afghanistan&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Albania&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Algeria&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Andorra&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Angola&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Anguilla&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Antigua+and+Barbuda&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Argentina&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Armenia&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Aruba&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Australia&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Austria&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Azerbaijan&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Bahamas&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Bahrain&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Bangladesh&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Barbados&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Belarus&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Belgium&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Belize&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Benin&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Bermuda&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Bhutan&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Bolivia&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Bosnia+and+Herzegovina&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Botswana&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Brazil&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Brunei&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Bulgaria&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Burkina+Faso&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Burundi&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Cambodia&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Cameroon&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Canada&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Cape+Verde&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Cayman+Islands&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Central+African+Republic&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Chad&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Chile&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=China&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Colombia&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Comoros&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Congo&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Costa+Rica&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Cote+d'Ivoire&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Croatia&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Cuba&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Curacao&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Cyprus&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Czechia&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Democratic+Republic+of+Congo&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Denmark&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Djibouti&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Dominica&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Dominican+Republic&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Ecuador&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Egypt&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=El+Salvador&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Equatorial+Guinea&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Eritrea&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Estonia&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Eswatini&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Ethiopia&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Faeroe+Islands&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Falkland+Islands&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Fiji&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Finland&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=France&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Gabon&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Gambia&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Georgia&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Germany&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Ghana&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Gibraltar&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Greece&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Greenland&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Grenada&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Guatemala&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Guernsey&key=xxx

## "Guernsey" not uniquely geocoded, using "guernsey"

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Guinea&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Guinea-Bissau&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Guyana&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Haiti&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Honduras&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Hong+Kong&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Hungary&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Iceland&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=India&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Indonesia&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Iran&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Iraq&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Ireland&key=xxx

## "Ireland" not uniquely geocoded, using "ireland"

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Isle+of+Man&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Israel&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Italy&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Jamaica&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Japan&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Jersey&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Jordan&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Kazakhstan&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Kenya&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Kosovo&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Kuwait&key=xxx

## "Kuwait" not uniquely geocoded, using "kuwait"

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Kyrgyzstan&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Laos&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Latvia&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Lebanon&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Lesotho&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Liberia&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Libya&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Liechtenstein&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Lithuania&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Luxembourg&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Macao&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Madagascar&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Malawi&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Malaysia&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Maldives&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Mali&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Malta&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Marshall+Islands&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Mauritania&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Mauritius&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Mexico&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Micronesia+(country)&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Moldova&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Monaco&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Mongolia&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Montenegro&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Montserrat&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Morocco&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Mozambique&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Myanmar&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Namibia&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Nauru&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Nepal&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Netherlands&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=New+Zealand&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Nicaragua&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Niger&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Nigeria&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=North+Macedonia&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Northern+Cyprus&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Norway&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Oman&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Pakistan&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Palestine&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Panama&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Papua+New+Guinea&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Paraguay&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Peru&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Philippines&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Poland&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Portugal&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Qatar&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Romania&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Russia&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Rwanda&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Saint+Helena&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Saint+Kitts+and+Nevis&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Saint+Lucia&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Saint+Vincent+and+the+Grenadines&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Samoa&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=San+Marino&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Sao+Tome+and+Principe&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Saudi+Arabia&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Senegal&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Serbia&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Seychelles&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Sierra+Leone&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Singapore&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Slovakia&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Slovenia&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Solomon+Islands&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Somalia&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=South+Africa&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=South+Korea&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=South+Sudan&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Spain&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Sri+Lanka&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Sudan&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Suriname&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Sweden&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Switzerland&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Syria&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Taiwan&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Tajikistan&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Tanzania&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Thailand&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Timor&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Togo&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Tonga&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Trinidad+and+Tobago&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Tunisia&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Turkey&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Turks+and+Caicos+Islands&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Uganda&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Ukraine&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=United+Arab+Emirates&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=United+Kingdom&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=United+States&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Uruguay&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Uzbekistan&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Vanuatu&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Vatican&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Venezuela&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Vietnam&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Yemen&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Zambia&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Zimbabwe&key=xxx

head(df2)

## owid lon lat  
## 1 Afghanistan 67.709953 33.93911  
## 2 Albania 20.168331 41.15333  
## 3 Algeria 1.659626 28.03389  
## 4 Andorra 1.521801 42.50628  
## 5 Angola 17.873887 -11.20269  
## 6 Anguilla -63.068615 18.22055

#write.csv(df2, "C:/Users/andre/Documents/primersemestre uoc/visualizacion/a8/shiny a8/data/coordenadas.csv", row.names = F)

#### 

##### procesamiento de la base locations

str(loc)

## 'data.frame': 196 obs. of 6 variables:  
## $ location : Factor w/ 196 levels "Afghanistan",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ iso\_code : Factor w/ 196 levels "ABW","AFG","AGO",..: 2 5 50 6 3 4 10 8 9 1 ...  
## $ vaccines : Factor w/ 34 levels "CanSino, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V",..: 12 18 34 13 12 12 12 22 34 26 ...  
## $ last\_observation\_date: Factor w/ 35 levels "2021-02-19","2021-03-23",..: 18 34 1 29 19 18 18 35 5 34 ...  
## $ source\_name : Factor w/ 120 levels "Cayman Islands Government",..: 17 87 87 18 87 87 87 87 87 19 ...  
## $ source\_website : Factor w/ 190 levels "http://covid19.ncema.gov.ae/en",..: 77 89 125 165 110 188 135 2 79 166 ...

dim(loc)

## [1] 196 6

head(loc)

## location iso\_code vaccines  
## 1 Afghanistan AFG Oxford/AstraZeneca  
## 2 Albania ALB Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V  
## 3 Algeria DZA Sputnik V  
## 4 Andorra AND Oxford/AstraZeneca, Pfizer/BioNTech  
## 5 Angola AGO Oxford/AstraZeneca  
## 6 Anguilla AIA Oxford/AstraZeneca  
## last\_observation\_date source\_name  
## 1 2021-04-22 Government of Afghanistan  
## 2 2021-05-08 Ministry of Health  
## 3 2021-02-19 Ministry of Health  
## 4 2021-05-03 Government of Andorra  
## 5 2021-04-23 Ministry of Health  
## 6 2021-04-22 Ministry of Health  
## source\_website  
## 1 https://reliefweb.int/report/afghanistan/afghanistan-strategic-situation-report-covid-19-no-95-22-april-2021  
## 2 https://twitter.com/GManastirliu/status/1391104178743828484  
## 3 https://www.echoroukonline.com/%d9%84%d9%82%d8%a7%d8%ad-%d9%83%d9%88%d8%b1%d9%88%d9%86%d8%a7-%d8%b9%d8%b6%d9%88-%d8%a8%d8%a7%d9%84%d9%84%d8%ac%d9%86%d8%a9-%d8%a7%d9%84%d8%b9%d9%84%d9%85%d9%8a%d8%a9-%d9%8a%d9%83%d8%b4%d9%81-%d8%b9/  
## 4 https://www.govern.ad/covid19\_newsletter/  
## 5 https://www.angop.ao/en/noticias/saude/covid-19-angola-com-456.349-cidadaos-vacinados/  
## 6 https://www.travelpulse.com/news/impacting-travel/new-covid-19-case-prompts-14-day-anguilla-closure.html

##separar los fabricantes de las vacunas en columnas y luego aplicar un gather para que se coloquen en filas independientes  
data\_split = separate(loc, 'vaccines', paste("vaccines", 1:5, sep=""), sep=",", extra="drop")

## Warning: Expected 5 pieces. Missing pieces filled with `NA` in 193 rows [1, 2,  
## 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].

data\_split <- data\_split %>% gather(`vaccines1`, `vaccines2`, `vaccines3`, `vaccines4`, `vaccines5`, key = "tipo", value = "manufactured")  
data\_split <- data\_split[!is.na(data\_split$manufactured),]  
data\_split$manufactured <- trimws(data\_split$manufactured)  
unique(data\_split$manufactured)

## [1] "Oxford/AstraZeneca" "Sputnik V" "Pfizer/BioNTech"   
## [4] "Johnson&Johnson" "Moderna" "Sinopharm/Beijing"   
## [7] "Covaxin" "CanSino" "EpiVacCorona"   
## [10] "Sinovac" "Sinopharm/Wuhan"

table(data\_split$manufactured)

##   
## CanSino Covaxin EpiVacCorona Johnson&Johnson   
## 2 1 1 15   
## Moderna Oxford/AstraZeneca Pfizer/BioNTech Sinopharm/Beijing   
## 42 148 93 38   
## Sinopharm/Wuhan Sinovac Sputnik V   
## 2 24 33

#write.csv(data\_split, "C:/Users/andre/Documents/primersemestre uoc/visualizacion/a8/shiny a8/data/new\_locations.csv", row.names = F)

### procesamiento para who

dim(who)

## [1] 238 13

str(who)

## 'data.frame': 238 obs. of 13 variables:  
## $ ï..Name : Factor w/ 238 levels "Afghanistan",..: 82 228 98 29 73 220 177 213 105 204 ...  
## $ WHO.Region : Factor w/ 8 levels "","Africa","Americas",..: 1 3 7 3 5 5 5 5 5 5 ...  
## $ Cases...cumulative.total : int 157289118 32300609 22296414 15082449 5676293 5016141 4880262 4433094 4102921 3559222 ...  
## $ Cases...cumulative.total.per.100000.population : num 2015 9758 1616 7096 8727 ...  
## $ Cases...newly.reported.in.last.7.days : int 5444758 298281 2738957 423438 122487 166733 57007 14560 67304 23271 ...  
## $ Cases...newly.reported.in.last.7.days.per.100000.population : num 69.7 90.1 198.5 199.2 188.3 ...  
## $ Cases...newly.reported.in.last.24.hours : int 786943 43193 403738 78886 20745 18052 8419 2047 10174 0 ...  
## $ Deaths...cumulative.total : int 3277272 575322 242362 419114 105544 42746 113326 127603 122694 78726 ...  
## $ Deaths...cumulative.total.per.100000.population : num 42 173.8 17.6 197.2 162.3 ...  
## $ Deaths...newly.reported.in.last.7.days : int 89748 4785 26820 15333 1550 2242 2464 79 1661 159 ...  
## $ Deaths...newly.reported.in.last.7.days.per.100000.population: num 1.15 1.45 1.94 7.21 2.38 ...  
## $ Deaths...newly.reported.in.last.24.hours : int 12994 804 4092 2165 205 281 334 5 224 0 ...  
## $ Transmission.Classification : Factor w/ 7 levels "","Clusters of cases",..: 1 3 2 3 3 3 2 3 2 3 ...

table(who$Transmission.Classification)

##   
## Clusters of cases Community transmission   
## 1 46 147   
## No cases Not applicable Pending   
## 22 1 3   
## Sporadic cases   
## 18

### procesamiento vacunas

dim(vac)

## [1] 3642 4

str(vac)

## 'data.frame': 3642 obs. of 4 variables:  
## $ location : Factor w/ 10 levels "Chile","Czechia",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ date : Factor w/ 137 levels "2020-12-24","2020-12-25",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ vaccine : Factor w/ 5 levels "Johnson&Johnson",..: 4 4 4 4 4 4 4 4 4 4 ...  
## $ total\_vaccinations: int 420 5198 8338 8649 8649 8649 8649 8649 8649 8649 ...

table(vac$vaccine)

##   
## Johnson&Johnson Moderna Oxford/AstraZeneca Pfizer/BioNTech   
## 411 1026 819 1290   
## Sinovac   
## 96

table(vac$location)

##   
## Chile Czechia France Germany Iceland   
## 243 355 356 528 516   
## Italy Latvia Lithuania Romania United States   
## 369 312 365 309 289

aggregate(total\_vaccinations ~ vaccine, FUN = sum, data = vac)

## vaccine total\_vaccinations  
## 1 Johnson&Johnson 327143078  
## 2 Moderna 6357036895  
## 3 Oxford/AstraZeneca 645152271  
## 4 Pfizer/BioNTech 10061113807  
## 5 Sinovac 695965590

aggregate(total\_vaccinations ~ location, FUN = sum, data = vac)

## location total\_vaccinations  
## 1 Chile 786371644  
## 2 Czechia 156904278  
## 3 France 1040924692  
## 4 Germany 1332366563  
## 5 Iceland 6495033  
## 6 Italy 990668415  
## 7 Latvia 13502251  
## 8 Lithuania 49912765  
## 9 Romania 277454541  
## 10 United States 13431811459

### procresamiento seguimiento de contactos

contact$Day <- as.Date(contact$Day)  
zzzzm <- contact %>% select(Entity, Day, contact\_tracing) %>% group\_by(Entity, contact\_tracing) %>% summarise(lasted = max(Day, na.rm = T))

## `summarise()` has grouped output by 'Entity'. You can override using the `.groups` argument.

sss <- zzzzm %>% filter(lasted == max(lasted))  
sss$Entity <- as.character(sss$Entity)  
df2<-read.csv("coordenadas.csv",header=T,sep=",")  
sssv <- sss %>% left\_join(df2, by = c("Entity" = "owid"))  
sssv$contact\_tracing <- as.character(sssv$contact\_tracing)  
sssv <- sssv[!is.na(sssv$lon),]  
sssv$contact\_tracing[sssv$contact\_tracing == "0"] <- "Sin seguimiento"  
sssv$contact\_tracing[sssv$contact\_tracing == "1"] <- "Seguimiento limitado"  
sssv$contact\_tracing[sssv$contact\_tracing == "2"] <- "Seguimiento integral"  
sssv$contact\_tracing <- as.factor(sssv$contact\_tracing)  
table(sssv$contact\_tracing)

##   
## Seguimiento integral Seguimiento limitado Sin seguimiento   
## 96 59 24

#write.csv(sssv, "C:/Users/andre/Documents/primersemestre uoc/visualizacion/a8/shiny a8/data/covid-contact-tracing.csv", row.names = F)

## procesamiento test policy

test\_p$Day <- as.Date(test\_p$Day)  
zzzzm <- test\_p %>% select(Entity, Day, testing\_policy) %>% group\_by(Entity, testing\_policy) %>% summarise(lasted = max(Day, na.rm = T))

## `summarise()` has grouped output by 'Entity'. You can override using the `.groups` argument.

sss <- zzzzm %>% filter(lasted == max(lasted))  
sssv <- sss %>% left\_join(df2, by = c("Entity" = "owid"))  
sssv <- sssv[!is.na(sssv$lon),]  
sssv$testing\_policy <- as.factor(sssv$testing\_policy)  
sssv$testing\_policy <- as.character(sssv$testing\_policy)  
sssv$testing\_policy[sssv$testing\_policy == "0"] <- "Sin políticas"  
sssv$testing\_policy[sssv$testing\_policy == "1"] <- "Con sintomas y ciertos grupos"  
sssv$testing\_policy[sssv$testing\_policy == "2"] <- "Cualquiera con sintomas"  
sssv$testing\_policy[sssv$testing\_policy == "3"] <- "Pruebas abiertas"  
sssv$testing\_policy <- as.factor(sssv$testing\_policy)  
table(sssv$testing\_policy)

##   
## Con sintomas y ciertos grupos Cualquiera con sintomas   
## 33 64   
## Pruebas abiertas Sin políticas   
## 81 1

#write.csv(sssv, "C:/Users/andre/Documents/primersemestre uoc/visualizacion/a8/shiny a8/data/covid-19-testing-policy.csv", row.names = F)

### procesamiento vacunas policy

###   
vac\_p$Day <- as.Date(vac\_p$Day)  
zzzzm <- vac\_p %>% select(Entity, Day, vaccination\_policy) %>% group\_by(Entity, vaccination\_policy) %>% summarise(lasted = max(Day, na.rm = T))

## `summarise()` has grouped output by 'Entity'. You can override using the `.groups` argument.

sss <- zzzzm %>% filter(lasted == max(lasted))  
sssv <- sss %>% left\_join(df2, by = c("Entity" = "owid"))  
sssv <- sssv[!is.na(sssv$lon),]  
sssv$vaccination\_policy <- as.factor(sssv$vaccination\_policy)  
sssv$vaccination\_policy <- as.character(sssv$vaccination\_policy)  
sssv$vaccination\_policy[sssv$vaccination\_policy == "0"] <- "Sin políticas"  
sssv$vaccination\_policy[sssv$vaccination\_policy == "1"] <- "Con sintomas y ciertos grupos"  
sssv$vaccination\_policy[sssv$vaccination\_policy == "2"] <- "Cualquiera con sintomas"  
sssv$vaccination\_policy[sssv$vaccination\_policy == "3"] <- "Pruebas abiertas"  
sssv$vaccination\_policy <- as.factor(sssv$vaccination\_policy)  
table(sssv$vaccination\_policy)

##   
## 4 5   
## 38 18   
## Con sintomas y ciertos grupos Cualquiera con sintomas   
## 16 29   
## Pruebas abiertas Sin políticas   
## 62 17

#write.csv(sssv, "C:/Users/andre/Documents/primersemestre uoc/visualizacion/a8/shiny a8/data/covid-vaccination-policy.csv", row.names = F)

## procesamiento income

##### ----  
income$Day <- as.Date(income$Day)  
zzzzm <- income %>% select(Entity, Day, income\_support) %>% group\_by(Entity, income\_support) %>% summarise(lasted = max(Day, na.rm = T))

## `summarise()` has grouped output by 'Entity'. You can override using the `.groups` argument.

sss <- zzzzm %>% filter(lasted == max(lasted))  
sssv <- sss %>% left\_join(df2, by = c("Entity" = "owid"))  
sssv <- sssv[!is.na(sssv$lon),]  
sssv$income\_support <- as.factor(sssv$income\_support)  
sssv$income\_support <- as.character(sssv$income\_support)  
sssv$income\_support[sssv$income\_support == "0"] <- "Sin apoyo"  
sssv$income\_support[sssv$income\_support == "1"] <- "Cubre<50% del salario perdido"  
sssv$income\_support[sssv$income\_support == "2"] <- "Cubre> 50% del salario perdido"  
sssv$income\_support <- as.factor(sssv$income\_support)  
table(sssv$income\_support)

##   
## Cubre<50% del salario perdido Cubre> 50% del salario perdido   
## 77 43   
## Sin apoyo   
## 59

#write.csv(sssv, "C:/Users/andre/Documents/primersemestre uoc/visualizacion/a8/shiny a8/data/income-support-covid.csv", row.names = F)

### procesamiento cierre de escuelas

#### -----  
school$Day <- as.Date(school$Day)  
zzzzm <- school %>% select(Entity, Day, school\_closures) %>% group\_by(Entity, school\_closures) %>% summarise(lasted = max(Day, na.rm = T))

## `summarise()` has grouped output by 'Entity'. You can override using the `.groups` argument.

sss <- zzzzm %>% filter(lasted == max(lasted))  
sssv <- sss %>% left\_join(df2, by = c("Entity" = "owid"))  
sssv <- sssv[!is.na(sssv$lon),]  
sssv$school\_closures <- as.factor(sssv$school\_closures)  
sssv$school\_closures <- as.character(sssv$school\_closures)  
sssv$school\_closures[sssv$school\_closures == "0"] <- "Sin medidas"  
sssv$school\_closures[sssv$school\_closures == "1"] <- "Recomendado"  
sssv$school\_closures[sssv$school\_closures == "2"] <- "Requerido-algunos niveles"  
sssv$school\_closures[sssv$school\_closures == "3"] <- "Requerido-para todos"  
sssv$school\_closures <- as.factor(sssv$school\_closures)  
table(sssv$school\_closures)

##   
## Recomendado Requerido-algunos niveles Requerido-para todos   
## 58 42 45   
## Sin medidas   
## 35

#write.csv(sssv, "C:/Users/andre/Documents/primersemestre uoc/visualizacion/a8/shiny a8/data/school-closures-covid.csv", row.names = F)

## procesamiento cierre de trabajo

#### ----  
work$Day <- as.Date(work$Day)  
zzzzm <- work %>% select(Entity, Day, workplace\_closures) %>% group\_by(Entity, workplace\_closures) %>% summarise(lasted = max(Day, na.rm = T))

## `summarise()` has grouped output by 'Entity'. You can override using the `.groups` argument.

sss <- zzzzm %>% filter(lasted == max(lasted))  
sssv <- sss %>% left\_join(df2, by = c("Entity" = "owid"))  
sssv <- sssv[!is.na(sssv$lon),]  
sssv$workplace\_closures <- as.factor(sssv$workplace\_closures)  
sssv$workplace\_closures <- as.character(sssv$workplace\_closures)  
sssv$workplace\_closures[sssv$workplace\_closures == "0"] <- "Sin medidas"  
sssv$workplace\_closures[sssv$workplace\_closures == "1"] <- "Recomendado"  
sssv$workplace\_closures[sssv$workplace\_closures == "2"] <- "Requerido-algunos"  
sssv$workplace\_closures[sssv$workplace\_closures == "3"] <- "Requerido-para todos menos trabajos claves"  
sssv$workplace\_closures <- as.factor(sssv$workplace\_closures)  
table(sssv$workplace\_closures)

##   
## Recomendado   
## 40   
## Requerido-algunos   
## 88   
## Requerido-para todos menos trabajos claves   
## 25   
## Sin medidas   
## 27

#write.csv(sssv, "C:/Users/andre/Documents/primersemestre uoc/visualizacion/a8/shiny a8/data/workplace-closures-covid.csv", row.names = F)

## procesamiento requermiento de quedarse en casa

home$Day <- as.Date(home$Day)  
zzzzm <- home %>% select(Entity, Day, stay\_home\_requirements) %>% group\_by(Entity, stay\_home\_requirements) %>% summarise(lasted = max(Day, na.rm = T))

## `summarise()` has grouped output by 'Entity'. You can override using the `.groups` argument.

sss <- zzzzm %>% filter(lasted == max(lasted))  
sssv <- sss %>% left\_join(df2, by = c("Entity" = "owid"))  
sssv <- sssv[!is.na(sssv$lon),]  
sssv$stay\_home\_requirements <- as.factor(sssv$stay\_home\_requirements)  
sssv$stay\_home\_requirements <- as.character(sssv$stay\_home\_requirements)  
sssv$stay\_home\_requirements[sssv$stay\_home\_requirements == "0"] <- "Sin medidas"  
sssv$stay\_home\_requirements[sssv$stay\_home\_requirements == "1"] <- "Recomendado"  
sssv$stay\_home\_requirements[sssv$stay\_home\_requirements == "2"] <- "Casos esenciales"  
sssv$stay\_home\_requirements[sssv$stay\_home\_requirements == "3"] <- "Todos menos pocas excepciones"  
sssv$stay\_home\_requirements <- as.factor(sssv$stay\_home\_requirements)  
table(sssv$stay\_home\_requirements)

##   
## Casos esenciales Recomendado   
## 77 47   
## Sin medidas Todos menos pocas excepciones   
## 45 11

#write.csv(sssv, "C:/Users/andre/Documents/primersemestre uoc/visualizacion/a8/shiny a8/data/stay\_home\_requirements.csv", row.names = F)