Covid 19 Project - HarvardX:PH125.9x Data Science Capstone

Silvane Paixao (silpai)

1/8/2021

All the visualizations and analysis in this report are as per **Jan 8th 2020 1pm AST** Note that the covid-19 data **changes over time** ##

1. Introduction

It has been 1 year that the Covid-19 pandemic has put our life and behaviors in check, there is so much to learn from and millions of questions to be asked. By having my own questions associated with the demographic aspect of the pandemic, I decided to use in this project demographic and geographical & socio-economic parameters, together with the longitudinal data of covid-19 which changed over time.

As highlighted by Mohammed (2016), major challenges that are encountered during development of a longitudinal data set are, among others , loss of information due to missing or incomplete data, a deterioration of data over time, lack of data Standardization (specially related to countries' names) and data Quality (such as missing data, incorrect data type,. . .).

This is the 2nd machine learning project requirement for the HarvardX Professional Certificate Data Science Program.

2. Methodology and Analysis

There were 3 datasets used: • *Worldtilegrid* – contains the countries region, subregion, and cartesian x,y coordinates. Note that some countries were not listed in this dataset and small islands were also missing.

Source: https://gist.githubusercontent.com/maartenzam/787498bbc07ae06b637447dbd430ea0a/raw/9a9dafafb44d8990f85243a9c7ca349acd3a0d07/worldtilegrid.csv

• *Covid-19 dataset with vaccination information* – contains covid-19 tracking data collected from the COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University (JHU); Vaccinations against COVID-19 collected by the Our World in Data team from official reports. This data is the single dose of the vaccine; demographic and socio-economic data collected from United Nations, WorldBank and other governmental agencies. Dataset is longitudinal (changes over time) and it is updated according to the countries time zones.

Source: https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/owid-covid-data.csv

• *WorldBank estimated population* – Contains projected total population for 2021 and population per age range (age 0-14, age 15-64 and age 65 and up). Since all the covid-19 vaccines so far have been approved for age 16 or older, in my analysis I will be using the range age 15 or older. One of my biggest curiosity was to know the percentage of the population vaccinated considering the approved age, not the entire population. Note that these are estimates, and there is no reduction in the number of deaths for covid-19 or any other cause.

Source: "http://databank.worldbank.org/data/download/Population-Estimates_CSV.zip"

Steps: a) Data Load and Cleaning 1. Load datasets 2. Combine the worldtilegrid with covid-19 3. Data Transformation i. Worldbank data is pivot longer, so I had to pivot wider the variables ii. Combine the 3 datasets. iii. Create new categories iv. exclude the rows with aggregated values by region b) Exploratory Analysis (EDA) were performed c) Modeling was used to create the predictions. Models were built from the training data to the test data: - Linear Models: correlation matrix, Linear regression, naive Bayes Classifier and decision tree

2.1 Data Transformation

The only loaded dataset that required transformations was the Worldbank data. Data was originally pivot longer and the parameters had to be remained before to be pivot wider. After joining, the 3 datasets, rows with aggregated values by region were eliminated, remaining only countries as observations. New grouping was created for the analysis.

```
##### a) Data Load and Cleaning

#1. Load datasets
# dataset: worldtilegrid x y
worldtilegrid <- read.csv(
  "https://gist.githubusercontent.com/maartenzam/787498bbc07ae06b637447dbd430ea0a/raw/9a9dafafb44d8990f8
  select(name, alpha.3,region,sub.region,x,y)
View(worldtilegrid)

# dataset: Covid-19 dataset with vaccination information
#Source: https://ourworldindata.org/covid-vaccinations
covid_data<-read.csv("https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/owid-covi
    select(iso_code,location,date,total_cases,new_cases, total_deaths,new_deaths,total_vaccinations,gdp_
           human_development_index, population_density, median_age,life_expectancy)
View(covid_data)

# dataset: worldbank estimated population
#Source:https://datacatalog.worldbank.org/dataset/population-estimates-and-projections
worldb <- tempfile()
download.file("http://databank.worldbank.org/data/download/Population-Estimates_CSV.zip", worldb)
worldbank_csv <- fread(text = gsub(",", "\t", readLines(unzip(worldb, "Population-EstimatesData.csv"))))
names(worldbank_csv) <- as.character(worldbank_csv[1,])                    # use 1st row as header
worldbank_csv=worldbank_csv[-c(1),]                                        # eliminate the 1st row that
names(worldbank_csv)<-str_replace_all(names(worldbank_csv), c(" " = "_" )) # replace space by _
#str(worldbank_csv)

##2.Combine covid data & tile grid x y
covid_grid<-full_join(x=covid_data, y=worldtilegrid, by=c("iso_code" ="alpha.3"))
#head(covid_grid)

# 3.    Data Transformation

# a) Pivot wider worldbank
worldbank <- worldbank_csv %>%
select("Country_Code","Country_Name","Indicator_Code","2021") %>%              # select indicator popu
filter(Indicator_Code %in% c("SP.POP.TOTL","SP.POP.1564.TO","SP.POP.65UP.TO")) %>%  # exclude SP.POP.00
spread(key = "Indicator_Code",
         value = "2021") %>%                                                    #pivot wider
rename(iso_code=Country_Code,Est_Pop_2021= SP.POP.TOTL, Pop15_64=SP.POP.1564.TO, Pop65UP=SP.POP.65UP.TO
```

```
mutate(Pop15Over=Pop15_64+Pop65UP)                                    # combine population ag
#head(worldbank)


#b) Combine covid data & worldbank estimated population age 15 and older data & tile grid x y data
# exclude row related to aggregated values
iso_code_aggregated <- data.frame(c("ARB", "CAF", "CEB", "CSS", "EAP", "EAR","EAS","ECA","ECS","EUU",
                          "FCS","HPC","INX","LAC", "LCN","LDC","LIC","LMC","LMY",
                          "LTE","MEA","MIC","MNA","NAC","OED","OSS","OWID_KOS","OWID_WRL", "PRE","PSS",
                          "PST","SAS","SSA","SSF","SST","TEA","TEC", "TLA",
                          "TMN", "TSA", "TSS", "UMC", "WLD", "OWID_KOS") )  %>%
  rename(iso_code=c..ARB....CAF....CEB....CSS....EAP....EAR....EAS....ECA....ECS...)
#iso_code_aggregated

#c) Create new categories
Pop_Vaccine_tile<-full_join(x=worldbank, y=covid_grid, by="iso_code") %>%
group_by(iso_code)%>%
mutate(
Percent_Pop15UP= percent(Pop15Over/Est_Pop_2021),
vaccination_administrated=as.numeric(total_vaccinations),
vaccination_categ= ifelse(is.na(vaccination_administrated), "Data unavailable",
            ifelse(vaccination_administrated >0 & vaccination_administrated <10000,"Single doses < 10k"
             ifelse(vaccination_administrated >=10000 & vaccination_administrated <100000,"10k >= Single
              ifelse(vaccination_administrated >=100000 & vaccination_administrated <1000000,"100k >= Si
               ifelse(vaccination_administrated >=1000000 & vaccination_administrated <5000000, "1M >= S
vaccination_status = ifelse(is.na(vaccination_administrated),"Vaccination did not start",
              ifelse(vaccination_administrated >0,"Vaccination started", "Vaccination did not start")),
Pop_percent_vaccinated_15over=vaccination_administrated/Pop15Over,
percent_vaccinated_15over=percent(vaccination_administrated/Pop15Over),
vaccination_percent_categ= ifelse(is.na(Pop_percent_vaccinated_15over),"Vaccination did not start",
        ifelse(Pop_percent_vaccinated_15over >0 & Pop_percent_vaccinated_15over <0.05,"Single doses < 5%
         ifelse(Pop_percent_vaccinated_15over >=0.05 & Pop_percent_vaccinated_15over <0.2,"5% >= Single
          ifelse(Pop_percent_vaccinated_15over >=0.2 & Pop_percent_vaccinated_15over <0.7,"20% >= Single
GDP_category=ifelse(is.na(gdp_per_capita),"Data unavailable",
       ifelse(gdp_per_capita<1000,"GDP < 1k",
        ifelse(gdp_per_capita >=1000 & gdp_per_capita <5000,"1k >= GDP < 5k",
         ifelse(gdp_per_capita >=5000 & gdp_per_capita <10000,"5k >= GDP < 10k",
          ifelse(gdp_per_capita >=10000 & gdp_per_capita <50000,"10k >= GDP < 50k",
           ifelse(gdp_per_capita >=50000 & gdp_per_capita <90000,"50k >= GDP < 90k","GDP >= 90K")))))),
Pop_density_categ= ifelse(is.na(population_density), "Data unavailable",
         ifelse(population_density >=0 & population_density <25,"Low (0 >= ppl/Km2 < 25)",
          ifelse(population_density >=25 & population_density <50,"Medium (25 >= ppl/Km2 < 50)",
           ifelse(population_density >=50 & population_density <100,"High (50 >= ppl/Km2 < 100)",
            ifelse(population_density >=100 & population_density <400,"Very High (100 >= ppl/Km2 <400)"
Life_expectancy_categ= ifelse(is.na(life_expectancy), "Data unavailable",
            ifelse(life_expectancy >=0 & life_expectancy <50,"Life expectancy < 50",
             ifelse(life_expectancy >=50 & life_expectancy <60,"50 >= Life expectancy < 60",
              ifelse(life_expectancy >=60 & life_expectancy <70,"60 >= Life expectancy < 70",
               ifelse(life_expectancy >=70 & life_expectancy <80,"70 >= Life expectancy < 80","Life exp
Total_cases_categ= ifelse(is.na(total_cases), "Data unavailable",
         ifelse(total_cases >=0 & total_cases <50000,"Total cases < 50k",
          ifelse(total_cases >=50000 & total_cases <200000,"50k >= Total cases < 200k",
           ifelse(total_cases >=200000 & total_cases <500000,"200k >= Total cases < 500k",
```

```
           ifelse(total_cases >=500000 & total_cases <1000000,"500k >= Total cases < 1M","Total cases >
Total_deaths_categ= ifelse(is.na(total_deaths), "Data unavailable",
          ifelse(total_deaths >=0 & total_deaths <100,"Total deaths < 100",
           ifelse(total_deaths >=100 & total_deaths <1000,"100 >= Total deaths < 1k",
            ifelse(total_deaths >=1000 & total_deaths <10000,"1k >= Total deaths < 10k",
             ifelse(total_deaths >=10000 & total_deaths <100000,"10k >= Total deaths< 100k","Total deat
vaccinationstatus_factor = factor(ifelse(vaccination_status == "Vaccination did not start", 0,1)),
Pop_density_factor= factor(ifelse(Pop_density_categ == "Data unavailable",0,
            ifelse(Pop_density_categ =="Low (0 >= ppl/Km2 < 25)",1,
             ifelse(Pop_density_categ == "Medium (25 >= ppl/Km2 < 50)",2,
              ifelse(Pop_density_categ =="High (50 >= ppl/Km2 < 100)",3,
               ifelse(Pop_density_categ =="Very High (100 >= ppl/Km2 <400)",4,5))))))
         ) %>%
# exclude rows with iso_code NULL
filter (location != c("International" ) & location != c("World") & iso_code != "OWID_KOS")

# d) d. exclude the rows with aggregated values by region
setDT(Pop_Vaccine_tile)
setDT(iso_code_aggregated)
Pop_Vaccine_tile[!iso_code_aggregated, on=c(iso_code = "iso_code")]
```

```
##        iso_code Country_Name Pop15_64 Pop65UP Est_Pop_2021 Pop15Over
##     1:      AFG  Afghanistan 22353000 1071000     39835000  23424000
##     2:      AFG  Afghanistan 22353000 1071000     39835000  23424000
##     3:      AFG  Afghanistan 22353000 1071000     39835000  23424000
##     4:      AFG  Afghanistan 22353000 1071000     39835000  23424000
##     5:      AFG  Afghanistan 22353000 1071000     39835000  23424000
##    ---
## 57467:      VAT         <NA>       NA      NA           NA        NA
## 57468:      VAT         <NA>       NA      NA           NA        NA
## 57469:      VAT         <NA>       NA      NA           NA        NA
## 57470:      VAT         <NA>       NA      NA           NA        NA
## 57471:      VAT         <NA>       NA      NA           NA        NA
##          location       date total_cases new_cases total_deaths new_deaths
##     1: Afghanistan 2020-02-24           1         1           NA         NA
##     2: Afghanistan 2020-02-25           1         0           NA         NA
##     3: Afghanistan 2020-02-26           1         0           NA         NA
##     4: Afghanistan 2020-02-27           1         0           NA         NA
##     5: Afghanistan 2020-02-28           1         0           NA         NA
##    ---
## 57467:     Vatican 2021-01-03          27         0           NA         NA
## 57468:     Vatican 2021-01-04          27         0           NA         NA
## 57469:     Vatican 2021-01-05          27         0           NA         NA
## 57470:     Vatican 2021-01-06          27         0           NA         NA
## 57471:     Vatican 2021-01-07          27         0           NA         NA
##        total_vaccinations gdp_per_capita human_development_index
##     1:                 NA       1803.987                   0.498
##     2:                 NA       1803.987                   0.498
##     3:                 NA       1803.987                   0.498
##     4:                 NA       1803.987                   0.498
##     5:                 NA       1803.987                   0.498
##    ---
## 57467:                 NA             NA                      NA
```

```
## 57468:                 NA             NA                      NA
## 57469:                 NA             NA                      NA
## 57470:                 NA             NA                      NA
## 57471:                 NA             NA                      NA
##        population_density median_age life_expectancy        name region
##     1:             54.422       18.6           64.83 Afghanistan   Asia
##     2:             54.422       18.6           64.83 Afghanistan   Asia
##     3:             54.422       18.6           64.83 Afghanistan   Asia
##     4:             54.422       18.6           64.83 Afghanistan   Asia
##     5:             54.422       18.6           64.83 Afghanistan   Asia
##    ---
## 57467:                 NA         NA           75.12        <NA>   <NA>
## 57468:                 NA         NA           75.12        <NA>   <NA>
## 57469:                 NA         NA           75.12        <NA>   <NA>
## 57470:                 NA         NA           75.12        <NA>   <NA>
## 57471:                 NA         NA           75.12        <NA>   <NA>
##           sub.region  x  y Percent_Pop15UP vaccination_administrated
##     1: Southern Asia 22  8             59%                        NA
##     2: Southern Asia 22  8             59%                        NA
##     3: Southern Asia 22  8             59%                        NA
##     4: Southern Asia 22  8             59%                        NA
##     5: Southern Asia 22  8             59%                        NA
##    ---
## 57467:          <NA> NA NA            <NA>                        NA
## 57468:          <NA> NA NA            <NA>                        NA
## 57469:          <NA> NA NA            <NA>                        NA
## 57470:          <NA> NA NA            <NA>                        NA
## 57471:          <NA> NA NA            <NA>                        NA
##        vaccination_categ        vaccination_status
##     1:  Data unavailable Vaccination did not start
##     2:  Data unavailable Vaccination did not start
##     3:  Data unavailable Vaccination did not start
##     4:  Data unavailable Vaccination did not start
##     5:  Data unavailable Vaccination did not start
##    ---
## 57467:  Data unavailable Vaccination did not start
## 57468:  Data unavailable Vaccination did not start
## 57469:  Data unavailable Vaccination did not start
## 57470:  Data unavailable Vaccination did not start
## 57471:  Data unavailable Vaccination did not start
##        Pop_percent_vaccinated_15over percent_vaccinated_15over
##     1:                            NA                      <NA>
##     2:                            NA                      <NA>
##     3:                            NA                      <NA>
##     4:                            NA                      <NA>
##     5:                            NA                      <NA>
##    ---
## 57467:                            NA                      <NA>
## 57468:                            NA                      <NA>
## 57469:                            NA                      <NA>
## 57470:                            NA                      <NA>
## 57471:                            NA                      <NA>
##        vaccination_percent_categ     GDP_category          Pop_density_categ
##     1: Vaccination did not start   1k >= GDP < 5k High (50 >= ppl/Km2 < 100)
```

```
##     2: Vaccination did not start   1k >= GDP < 5k High (50 >= ppl/Km2 < 100)
##     3: Vaccination did not start   1k >= GDP < 5k High (50 >= ppl/Km2 < 100)
##     4: Vaccination did not start   1k >= GDP < 5k High (50 >= ppl/Km2 < 100)
##     5: Vaccination did not start   1k >= GDP < 5k High (50 >= ppl/Km2 < 100)
##    ---
## 57467: Vaccination did not start Data unavailable           Data unavailable
## 57468: Vaccination did not start Data unavailable           Data unavailable
## 57469: Vaccination did not start Data unavailable           Data unavailable
## 57470: Vaccination did not start Data unavailable           Data unavailable
## 57471: Vaccination did not start Data unavailable           Data unavailable
##             Life_expectancy_categ Total_cases_categ Total_deaths_categ
##     1: 60 >= Life expectancy < 70 Total cases < 50k   Data unavailable
##     2: 60 >= Life expectancy < 70 Total cases < 50k   Data unavailable
##     3: 60 >= Life expectancy < 70 Total cases < 50k   Data unavailable
##     4: 60 >= Life expectancy < 70 Total cases < 50k   Data unavailable
##     5: 60 >= Life expectancy < 70 Total cases < 50k   Data unavailable
##    ---
## 57467: 70 >= Life expectancy < 80 Total cases < 50k   Data unavailable
## 57468: 70 >= Life expectancy < 80 Total cases < 50k   Data unavailable
## 57469: 70 >= Life expectancy < 80 Total cases < 50k   Data unavailable
## 57470: 70 >= Life expectancy < 80 Total cases < 50k   Data unavailable
## 57471: 70 >= Life expectancy < 80 Total cases < 50k   Data unavailable
##        vaccinationstatus_factor Pop_density_factor
##     1:                        0                  3
##     2:                        0                  3
##     3:                        0                  3
##     4:                        0                  3
##     5:                        0                  3
##    ---
## 57467:                        0                  0
## 57468:                        0                  0
## 57469:                        0                  0
## 57470:                        0                  0
## 57471:                        0                  0
```

```
View(Pop_Vaccine_tile)
```

2.2 Exploratory Data Analysis (EDA) & Visualizations

Covid-19 cases

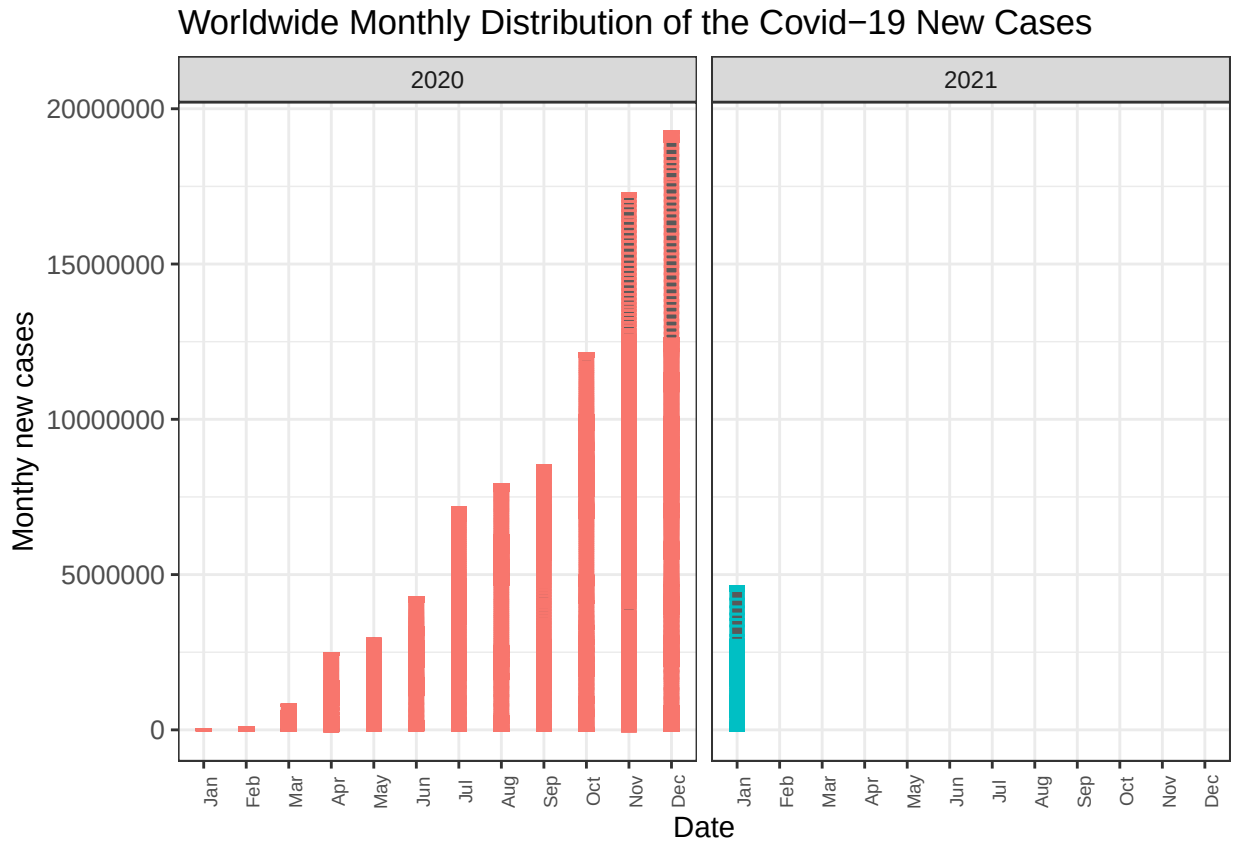
The trends: Worldwide Monthly Distribution of the Covid-19 New Cases indicated that the first 7 days of 2021 are comparable with the total new cases found during the full month of April 2020.

There was some stability in the number of new cases on the northern hemisphere during the 2020 Summer months (Jun, July, Aug). Then, it starts to increase again after September 2020. December 2020 ended with almost 20M new cases worldwide.

```
Pop_Vaccine_tile %>%
  ggplot(aes(x=lubridate::month(date, label = TRUE, abbr = TRUE),
             y=new_cases,
             group = factor(lubridate::year(date)),
             color = factor(lubridate::year(date)))) +
  geom_bar(stat="identity", width=0.3) +
```

```
  #theme_classic() +
  labs(title = "Worldwide Monthly Distribution of the Covid-19 New Cases",
       x= "Date", y= "Monthy new cases") +
  theme_bw() + theme(axis.text.x =element_text(size = rel(0.75),angle = 90),
                     axis.text.y = element_text(size = 10),
                     legend.position = "none") +
# scale_y_continuous(labels = function(x) format(x, scientific = FALSE)) + # format scientific notatio
  facet_wrap(~ lubridate::year(date))                                       # see data by year
```

```
## Warning: Removed 566 rows containing missing values (position_stack).
```



Since covid-19 is longitudinal data, the last date of the report based on total cases, total deaths and vaccination were extracted for analysis.

Note that these results may differ from the total actual published data in each country, due to the lag on the covid-19 package being updated. Also, different time zones must be considered.

```
#most recent cases
recent_totalcases<- Pop_Vaccine_tile %>% group_by(iso_code) %>%       # most recent data for total case
  slice( which.max( total_cases) ) %>% as.data.frame
View(recent_totalcases)
```

Trends: Top 15 countries with Total cases vs. administrating vaccine showed that only 66% of the countries started the vaccination rollout.United States - USA is the only country that is also on the TOP 3 of Total cases with more than 21M. The other 2 Top total cases countries are India - IND (about 10M) and Brazil

- BRA (about 8M) which did not yet start start their vaccination. These are also countries with high population size.

The Top 15 countries with Total cases that are administrating covid-19 vaccine are: - Single doses >= 5M : United States - USA - 100k >= Single doses < 1M: Poland - POL, Spain - ESP, Italy - ITA - 10k >= Single doses < 100k: Mexico - MEX, France - FRA

```
#top 15 countries total cases
Top_15_cases_countries<-recent_totalcases %>%
  top_n(15,total_cases)

table(Top_15_cases_countries$vaccination_status)
```

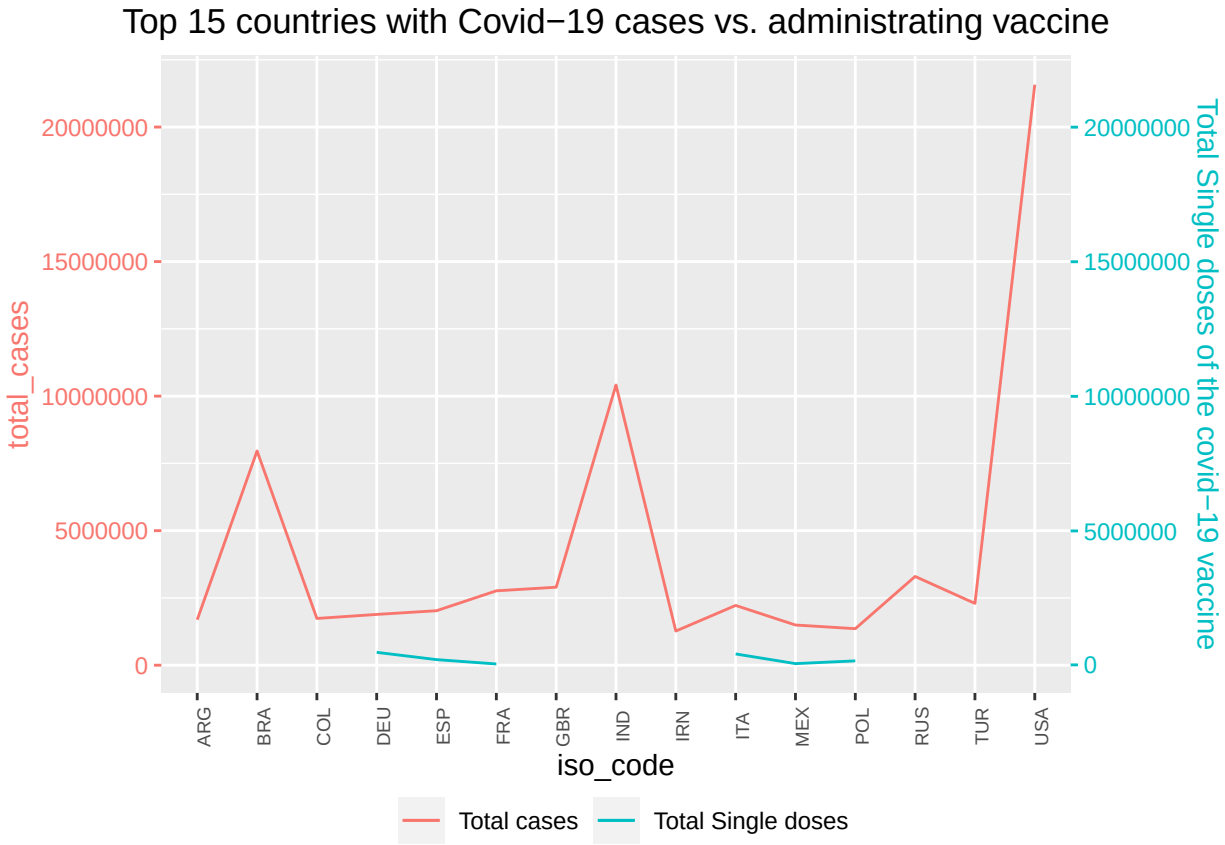
```
##
## Vaccination did not start       Vaccination started
##                          8                         7
```

```
table(Top_15_cases_countries$vaccination_categ)
```

```
##
##  100k >= Single doses < 1M 10k >= Single doses < 100k
##                          4                         2
##           Data unavailable         Single doses >= 5M
##                          8                         1
```

```
## Top 15 countries with Total cases vs. administrating vaccine
cols = hcl(c(15, 15+180), 100, 65)
# Set scale factor for second axis
#scl = with(Top_15_cases_countries, max(abs(total_cases))/max(abs(vaccination_administrated)))  #scale
scl=1
Top_15_cases_countries %>%
  ggplot(aes(x = iso_code)) +
  geom_line(aes(y = total_cases, colour = "Total cases", group=1)) +
  geom_line(aes(y = vaccination_administrated*scl, colour = "Total Single doses", group=2)) +
  scale_y_continuous(sec.axis = sec_axis(~./scl, name = "Total Single doses of the covid-19 vaccine"))+
  ggtitle("Top 15 countries with Covid-19 cases vs. administrating vaccine") +
  theme(legend.position = "bottom",
        legend.margin=margin(-5,0,0,0),
        plot.title = element_text(hjust = 0.5),
        axis.text.y.right=element_text(colour=cols[2]),
        axis.ticks.y.right=element_line(colour=cols[2]),
        axis.title.y.right=element_text(colour=cols[2]),
        axis.text.x =element_text(size = rel(0.75),angle = 90),
        axis.text.y=element_text(colour=cols[1]),
        axis.ticks.y=element_line(colour=cols[1]),
        axis.title.y=element_text(colour=cols[1])) +
  scale_colour_manual(values=cols) +
  labs(colour="")
```

```
## Warning: Removed 3 row(s) containing missing values (geom_path).
```

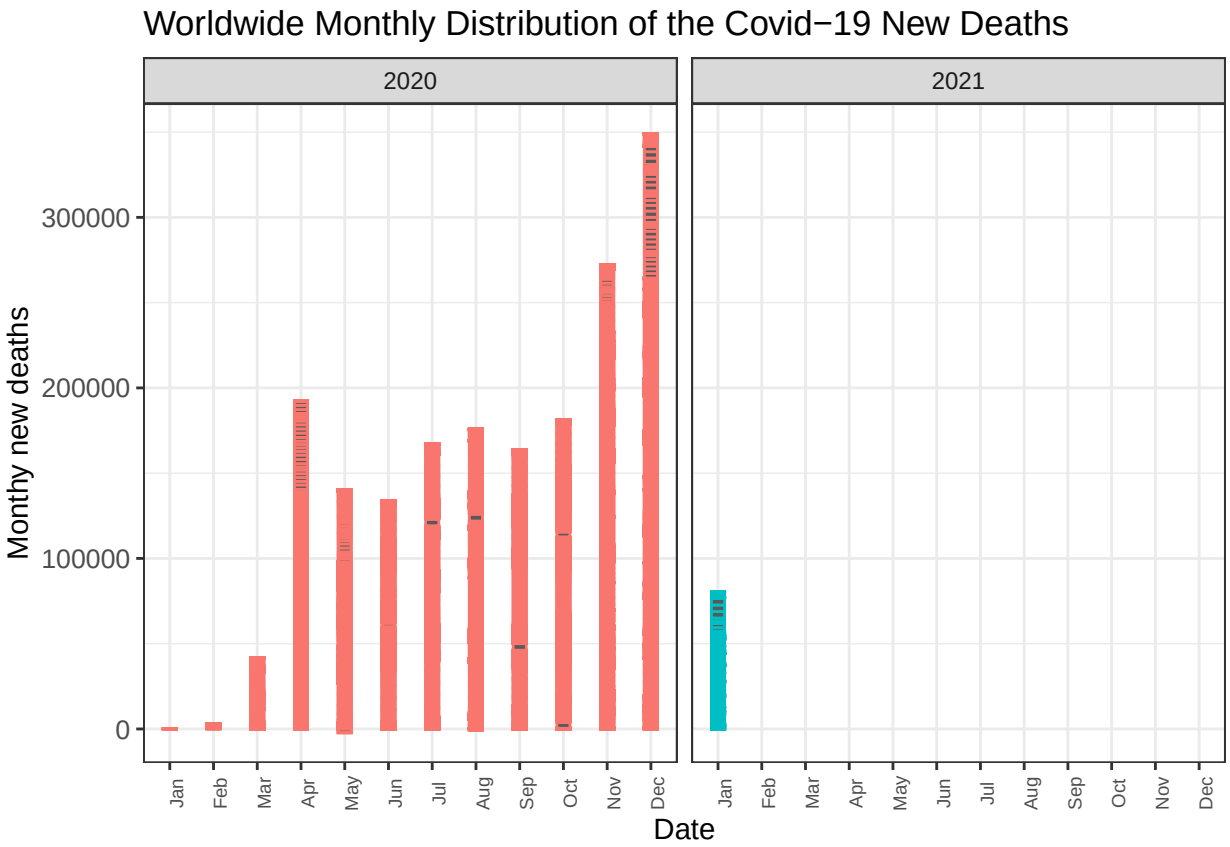


Covid-19 deaths

In 2020 there were 3 significant peaks of new deaths: one In April during the Northern hemisphere spring and 2 others at the start of Winter (November and December), being December 2020 the month with the highest number of new deaths (about 350K worldwide).

```
Pop_Vaccine_tile %>%
  ggplot(aes(x=lubridate::month(date, label = TRUE, abbr = TRUE),
             y=new_deaths,
             group = factor(lubridate::year(date)),
             color = factor(lubridate::year(date)))) +
  geom_bar(stat="identity", width=0.3) +
  #theme_classic() +
  labs(title = "Worldwide Monthly Distribution of the Covid-19 New Deaths", x= "Date", y= "Monthy new de
  theme_bw() + theme(axis.text.x =element_text(size = rel(0.75),angle = 90),
                     axis.text.y = element_text(size = 10),
                     legend.position = "none") +
  scale_y_continuous(labels = function(x) format(x, scientific = FALSE)) +      # format scientific not
  facet_wrap(~ lubridate::year(date))                                           # see data by year
```

```
## Warning: Removed 8903 rows containing missing values (position_stack).
```



It has been already 1 year since we are facing covid-19, and there are more than 86M of confirmed covid-19 cases and almost 2M deaths worldwide, although there are asymptomatic people who were not tested and continue to transmit the virus. Bendix (2020) suggests that the actual number of covid-19 cases in US could be anywhere from 5 to 20 times the numbers published, which might also be a reality for many others.

```
#most recent deaths
recent_totaldeaths<- Pop_Vaccine_tile %>% group_by(iso_code) %>%     # most recent data for total death
  slice( which.max( total_deaths) ) %>% as.data.frame
View(recent_totaldeaths)
```

Vaccination

On December 2nd2020, the Pfizer/BioNTech covid-19 vaccine was the 1st in the world to receive emergency approval in UK, days after, Baharain, Canada, Mexico and USA followed suit.

The vaccine rollout started slow, but it has already reached 3 regions and 42 countries:

The vaccine rollout started slow but already reached 3 regions and 42 countries: - Europe (28): Northern Europe is leading the rollout - Americas (6): countries are equally distributed among the sub regions - Asia (8): Western Asia is leading

```
countries_w_vaccine<- Pop_Vaccine_tile %>%
  filter (vaccination_administrated>0) %>%
  as.data.frame
#head(countries_w_vaccine)

# counts of the countries that started the covid vaccine rollout by regions, sub region
```

```r
countries_w_vaccine %>% group_by(region) %>%
  summarise(count = n_distinct(iso_code))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

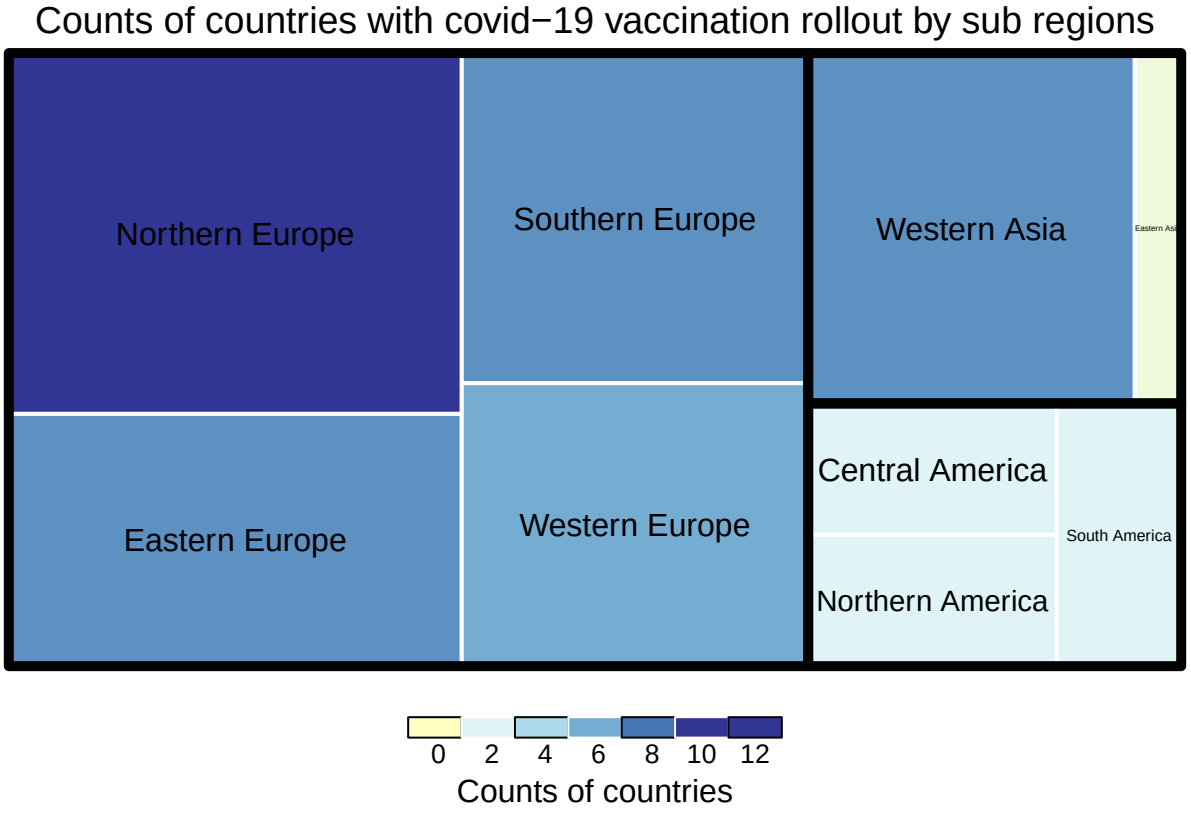
```
## # A tibble: 3 x 2
##   region   count
##   <chr>    <int>
## 1 Americas     6
## 2 Asia         8
## 3 Europe      30
```

```r
# counts of the countries that started the covid vaccine rollout by regions, sub region
n_countries_region <- countries_w_vaccine %>% group_by(region, sub.region) %>%
  summarise(count = n_distinct(iso_code))
```

```
## `summarise()` regrouping output by 'region' (override with `.groups` argument)
```

```r
#n_countries_region

# Creating the treemap
treemap(n_countries_region,
        index=c("region","sub.region"),
        vSize=c("count"),
        vColor=c("count"),
        type="value",
        range=c(0,12),
        #palette=brewer.pal(n=8,"RdYlGn"),
        algorithm="pivotSize",
        sortID="-size",
        palette="RdYlBu",
        title="Counts of countries with covid-19 vaccination rollout by sub regions",
        title.legend = "Counts of countries",
        fontsize.labels=c(0.1,12),                # size of labels. Give the size per level of aggregat
        fontcolor.labels=c("white","black"),    # Color of labels
        fontface.labels=c(2,1),                  # Font of labels: 1,2,3,4 for normal, bold, italic, bo
        #bg.labels=c("transparent"),              # Background color of labels
        align.labels=list(
          c("center", "center"),
          c("center", "center")
        ),                                   # Where to place labels in the rectangle?
        overlap.labels=0.5,                      # number between 0 and 1 that determines the tolerance
        inflate.labels=F,                        # If true, labels are bigger when rectangle is bigger.
        border.col=c("black","white"),             # Color of borders of groups, of subgroups, of subsu
        border.lwds=c(5,2)                       # Width of colors
        )
```




```
#most recent vaccination totals
recent_vaccine<- Pop_Vaccine_tile %>% group_by(iso_code) %>%          # most recent data for single dos
  slice( which.max(vaccination_administrated) ) %>% as.data.frame
View(recent_vaccine)
```

- Geographical Distribution of the Countries with vaccination rollout*

The tile grid map of the *Countries with Covid-19 vaccine administration per Single Doses intervals shows that the majority of the countries are located at sub regions North America and in Northern & Western Europe. The countries with the highest number of doses administrated: - North America: USA (Single doses >= 5M), - Northern & Western Europe: Great Britain and Northern Ireland - GBR (1M>=Single doses >= 5M) - Western Asia: Israel - ISR (1M>=Single doses >= 5M) - Eastern Asia: China - CHN (1M>=Single doses >= 5M)

```
#join x=recent_totalcases, y=recent_vaccine but only have the most recent vaccination category
vaccine_tile <-full_join(x=recent_totalcases, y=recent_vaccine, by=c("iso_code")) %>%
  select (iso_code,Country_Name.x,vaccination_administrated.y,vaccination_categ.y,vaccination_percent_ca

vaccine_tile %>% filter (is.na(x.x)) # missing coordinates can not have data represented in the map
```

```
##   iso_code     Country_Name.x vaccination_administrated.y vaccination_categ.y
## 1      AND            Andorra                          NA               <NA>
## 2      LIE      Liechtenstein                          NA               <NA>
## 3      MCO             Monaco                          NA               <NA>
## 4      PSE West Bank and Gaza                          NA               <NA>
```

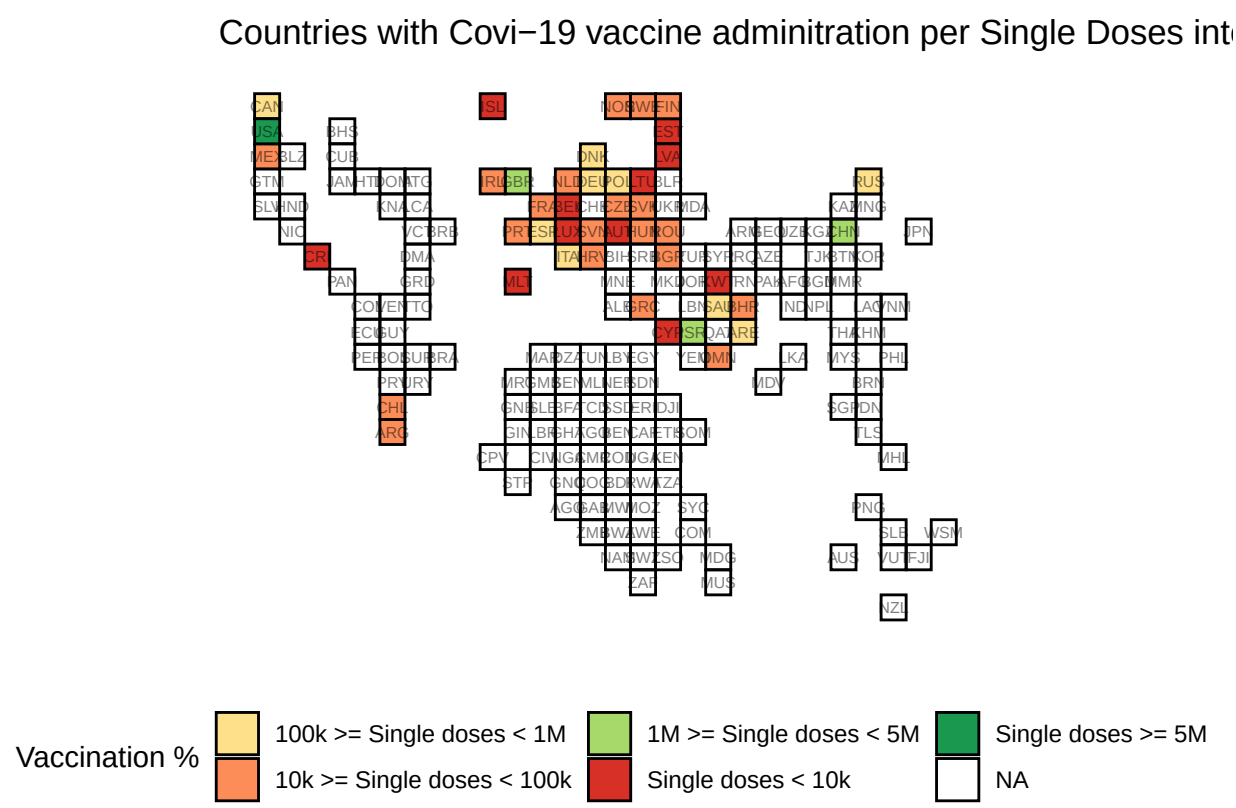
```
## 5      SMR         San Marino                          NA                <NA>
## 6      TWN               <NA>                          NA                <NA>
## 7      VAT               <NA>                          NA                <NA>
##   vaccination_percent_categ.y x.x y.x sub.region.x
## 1                        <NA>  NA  NA         <NA>
## 2                        <NA>  NA  NA         <NA>
## 3                        <NA>  NA  NA         <NA>
## 4                        <NA>  NA  NA         <NA>
## 5                        <NA>  NA  NA         <NA>
## 6                        <NA>  NA  NA         <NA>
## 7                        <NA>  NA  NA         <NA>
```

```
View(vaccine_tile)

# tile map of the countries by the counts of single doses administrated
ggplot(vaccine_tile, aes(xmin = x.x, ymin = y.x, xmax = x.x + 1, ymax = y.x + 1, fill = vaccination_cat
  geom_rect(color = "black") +
  mytheme + theme(legend.position = "bottom") +
  geom_text(aes(x = x.x, y = y.x, label = iso_code), color = "black", alpha = 0.5, nudge_x = 0.5, nudge
  scale_y_reverse() +
  scale_fill_manual(values = colors_green_red) +
  coord_equal()+
  labs(fill = "Vaccination %",
       title= "Countries with Covi-19 vaccine adminitration per Single Doses intervals",
       caption = "Countries such as Liechtenstein, Monaco, West Bank and Gaza, San Marino and small isl
```

```
## Warning: Removed 7 rows containing missing values (geom_rect).
```

```
## Warning: Removed 7 rows containing missing values (geom_text).
```

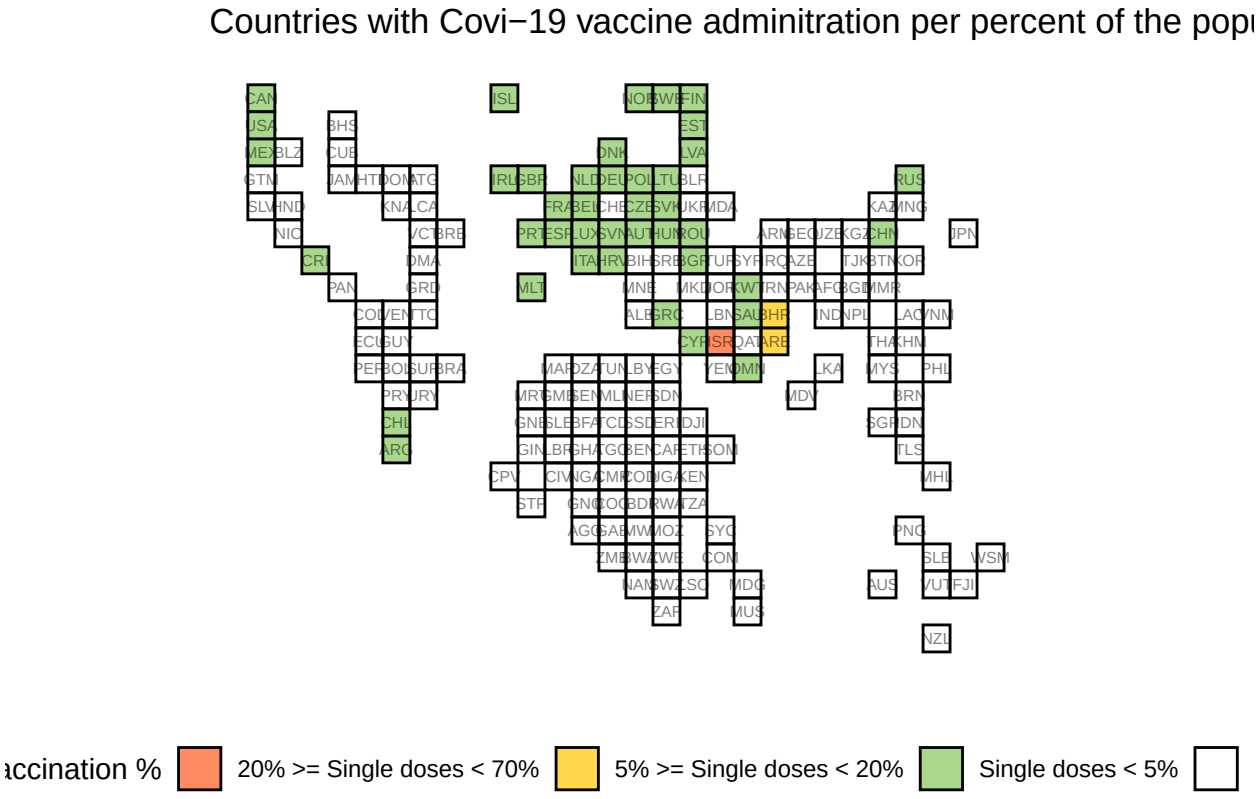


In terms of the **percent of the population age 15 or older been vaccinated**, Western Asia is leading with Israel - ISR (20% >=Single doses >= 70%) followed by United Arab Emirates - ARE and Bahrain - BHR (5% >=Single doses >= 20%).

```
# tile map of the countries by percent of the population age 15 or older
ggplot(vaccine_tile, aes(xmin = x.x, ymin = y.x, xmax = x.x + 1, ymax = y.x + 1, fill = vaccination_per
  geom_rect(color = "black") +
  mytheme +
  theme(legend.position = "bottom") +
  geom_text(aes(x = x.x, y = y.x, label = iso_code), color = "black", alpha = 0.5, nudge_x = 0.5, nudge
  scale_y_reverse() +
  coord_equal()+
  labs(fill = "Vaccination %",
       title= "Countries with Covi-19 vaccine adminitration per percent of the population age 15 or old
       caption = "Countries such as Liechtenstein, Monaco, West Bank and Gaza, San Marino and small isla
  scale_fill_manual(values = colors_tangerine_green)
```

```
## Warning: Removed 7 rows containing missing values (geom_rect).
```

```
## Warning: Removed 7 rows containing missing values (geom_text).
```



Trends: Top 15 countries administrating vaccine vs gdp per capita shows that the majority of the countries had gdp per capita in between aproximately35K and 45k Except for: United States - USA (55K), United Arab Emirates - ARE (around 65K), and China - CHN which is slightly over 15K.

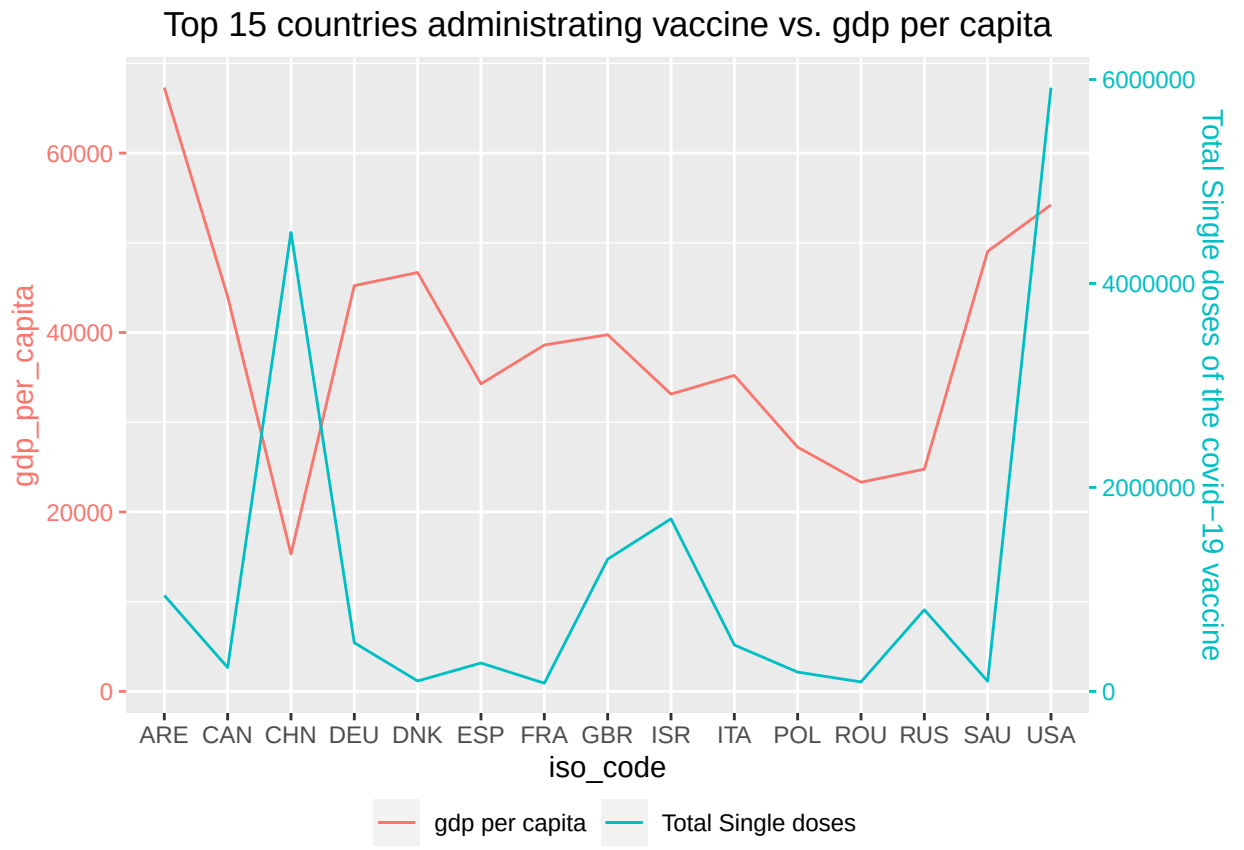
There were some downward trends compared with the vaccination rollout: high gdp per capita and lower total single dose of the vaccine (Bahrain - BHR, Canada - CAN, Denmark - DNK, Spain - ESP, Italy - ITA, Poland - POL, Romania - ROU, Saudi Arabia - SAU) SAU) and the opposite happens with China with lowest gdp but the 2nd country with the highest vaccination administration, only behind USA which is in 1st place.

```
#top 15 countries administrating the vaccine
Top_15_vac_countries<-recent_vaccine %>%
  top_n(15,vaccination_administrated)

## Top 15 countries administrating vaccine vs gdp per capita"
cols = hcl(c(15, 15+180), 100, 65)
# Set scale factor for second axis
scl = with(Top_15_vac_countries, max(abs(gdp_per_capita))/max(abs(vaccination_administrated)))  #scale

Top_15_vac_countries %>%
  ggplot(aes(x = iso_code)) +
  geom_line(aes(y = gdp_per_capita, colour = "gdp per capita", group=1)) +
  geom_line(aes(y = vaccination_administrated*scl, colour = "Total Single doses", group=2)) +
  scale_y_continuous(sec.axis = sec_axis(~./scl, name = "Total Single doses of the covid-19 vaccine"))+
  ggtitle("Top 15 countries administrating vaccine vs. gdp per capita") +
  theme(legend.position = "bottom",
        legend.margin=margin(-5,0,0,0),
```

```
        plot.title = element_text(hjust = 0.5),
        axis.text.y.right=element_text(colour=cols[2]),
        axis.ticks.y.right=element_line(colour=cols[2]),
        axis.title.y.right=element_text(colour=cols[2]),
        axis.text.y=element_text(colour=cols[1]),
        axis.ticks.y=element_line(colour=cols[1]),
        axis.title.y=element_text(colour=cols[1])) +
  scale_colour_manual(values=cols) +
  labs(colour="")
```



2.3 Modelling

The data set with combined covid-19, tile grid and population was split 80% to train_set and 20% to test_set. A validation set was also created from the test set.

```
# Validation set will be 20%
set.seed(1, sample.kind="Rounding") # if using R 3.5 or earlier, use `set.seed(1)`
```

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```
test_index <- createDataPartition(y = na.omit(Pop_Vaccine_tile$total_cases), times = 1, p = 0.2, list =
covid_train_set <- Pop_Vaccine_tile[-test_index,]
```

```
covid_test_set  <- Pop_Vaccine_tile[test_index,]

#dim(covid_train_set)
#View(covid_train_set)

#dim(covid_test_set)
#View(covid_test_set)


# Make sure iso_code and date in validation set are also in covid_train_set
validation <- covid_test_set %>%
  semi_join(covid_train_set, by = "date") %>%
  semi_join(covid_train_set, by = "iso_code")

#dim(validation)
#View(validation)
```

2.3.2 Linear Models ** Correlation Matrix**

The closer the Pearson correlation is to 1 or -1, the more perfect linear relationship it will have (direct or inverse), where "0" means no correlation. In the correlation matrix, the strongest relationships are represented in darker red and blue (excluding correction 1, variables related to themselves).

Related to Covid -19, the strongest correlations are: - total cases with total deaths - new cases with total cases and total deaths - new deaths with total cases, total deaths, new cases - total vaccinations with population age 15 and older, total cases, total deaths, new cases and new deaths

Personally, I would imagine that population density would have a strong correlation to new cases, or total cases, but the results showed almost no correlation.

** Note ** : If both covid cases & death are not updated and it is seen as **NULL** but the vaccination is updated, the ** Correlation Matrix** will be showing missing coefficient values, you will have to wait few hours until all the data is refreshed by the package owner.

```
########### using validation set

  # create dataset numeric values
  recent_vaccine_num <- covid_train_set %>% filter(total_cases!=is.na(total_cases), total_deaths!=is.na
    select(Pop15Over, total_deaths, total_cases, new_cases, new_deaths, total_vaccinations,population_
  View(recent_vaccine_num)

  #head(recent_vaccine_num)
  cormat <- round(cor(recent_vaccine_num),2)                #calculate the cor matrix
  #head(cormat)
  melted_cormat <- melt(cormat)                            # pair variables
  #head(melted_cormat)

  # Get lower triangle of the correlation matrix
  get_lower_tri<-function(cormat){
    cormat[upper.tri(cormat)] <- NA
    return(cormat)}
  # Get upper triangle of the correlation matrix
  get_upper_tri <- function(cormat){
    cormat[lower.tri(cormat)]<- NA
    return(cormat)}
```

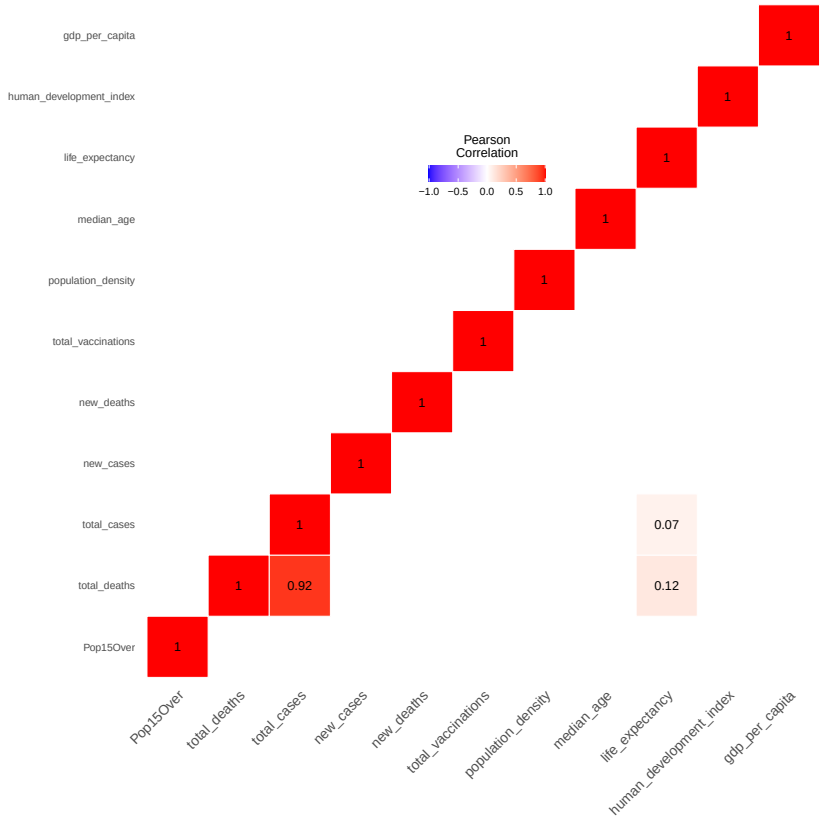
```r
upper_tri <- get_upper_tri(cormat)
upper_tri
```

```
##                         Pop15Over total_deaths total_cases new_cases new_deaths
## Pop15Over                       1           NA          NA        NA          NA
## total_deaths                   NA            1        0.92        NA          NA
## total_cases                    NA           NA        1.00        NA          NA
## new_cases                      NA           NA          NA         1          NA
## new_deaths                     NA           NA          NA        NA           1
## total_vaccinations             NA           NA          NA        NA          NA
## population_density             NA           NA          NA        NA          NA
## median_age                     NA           NA          NA        NA          NA
## life_expectancy                NA           NA          NA        NA          NA
## human_development_index        NA           NA          NA        NA          NA
## gdp_per_capita                 NA           NA          NA        NA          NA
##                         total_vaccinations population_density median_age
## Pop15Over                               NA                  NA         NA
## total_deaths                            NA                  NA         NA
## total_cases                             NA                  NA         NA
## new_cases                               NA                  NA         NA
## new_deaths                              NA                  NA         NA
## total_vaccinations                       1                  NA         NA
## population_density                      NA                   1         NA
## median_age                              NA                  NA          1
## life_expectancy                         NA                  NA         NA
## human_development_index                 NA                  NA         NA
## gdp_per_capita                          NA                  NA         NA
##                         life_expectancy human_development_index gdp_per_capita
## Pop15Over                            NA                      NA             NA
## total_deaths                       0.12                      NA             NA
## total_cases                        0.07                      NA             NA
## new_cases                            NA                      NA             NA
## new_deaths                           NA                      NA             NA
## total_vaccinations                   NA                      NA             NA
## population_density                   NA                      NA             NA
## median_age                           NA                      NA             NA
## life_expectancy                    1.00                      NA             NA
## human_development_index              NA                       1             NA
## gdp_per_capita                       NA                      NA              1
```

```r
# Melt the correlation matrix
melted_cormat <- melt(upper_tri, na.rm = TRUE)

# Create  Heatmap
ggplot(data = melted_cormat, aes(Var2, Var1, fill = value))+
  geom_tile(color = "white")+
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",      #limit = c(-1,1) as cor ran
                       name="Pearson\nCorrelation") +
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   size = 12, hjust = 1))+
  coord_fixed() +                                                          #unit on the x-axis = leng
```

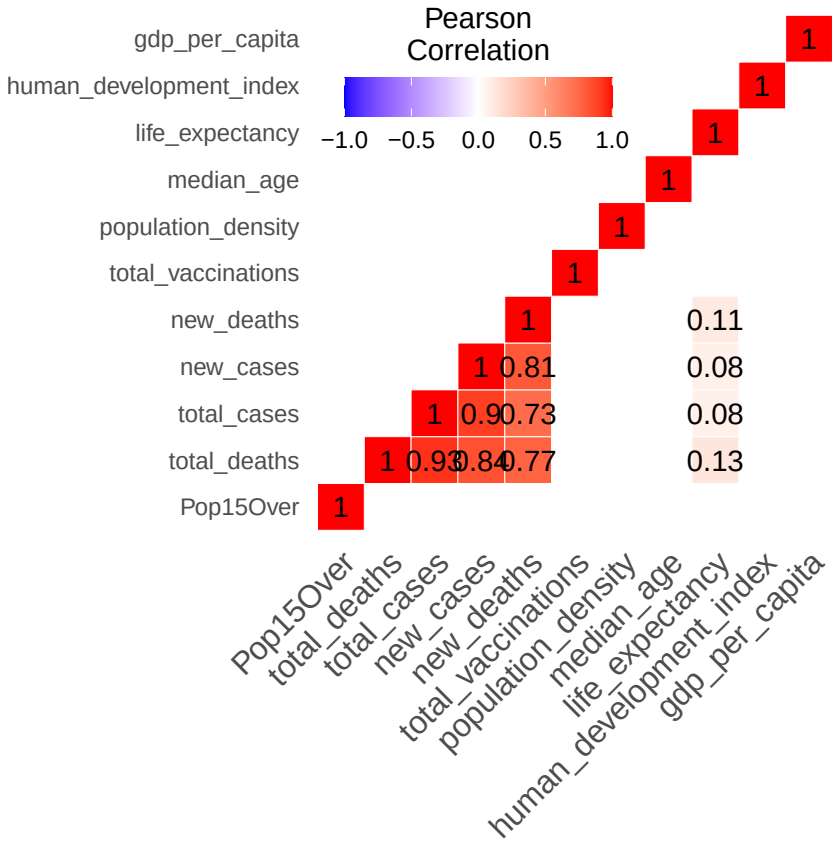
```
    geom_text(aes(Var2, Var1, label = value), color = "black", size = 4) +   # add the cor values
    theme(
      axis.title.x = element_blank(),
      axis.title.y = element_blank(),
      panel.grid.major = element_blank(),
      panel.border = element_blank(),
      panel.background = element_blank(),
      axis.ticks = element_blank(),
      legend.justification = c(1, 0),
      legend.position = c(0.6, 0.7),
      legend.direction = "horizontal")+
    guides(fill = guide_colorbar(barwidth = 7, barheight = 1,
                                 title.position = "top", title.hjust = 0.5))
```



There was no difference between the correlation matrix created using the covid_train_set and the validation set.

```
##                      Pop15Over total_deaths total_cases new_cases new_deaths
## Pop15Over                    1           NA          NA        NA         NA
## total_deaths                NA            1        0.93      0.84       0.77
## total_cases                 NA           NA        1.00      0.90       0.73
## new_cases                   NA           NA          NA      1.00       0.81
## new_deaths                  NA           NA          NA        NA       1.00
## total_vaccinations          NA           NA          NA        NA         NA
## population_density          NA           NA          NA        NA         NA
## median_age                  NA           NA          NA        NA         NA
```

```
## life_expectancy                NA           NA          NA        NA         NA
## human_development_index        NA           NA          NA        NA         NA
## gdp_per_capita                 NA           NA          NA        NA         NA
##                       total_vaccinations population_density median_age
## Pop15Over                             NA                  NA         NA
## total_deaths                          NA                  NA         NA
## total_cases                           NA                  NA         NA
## new_cases                             NA                  NA         NA
## new_deaths                            NA                  NA         NA
## total_vaccinations                     1                  NA         NA
## population_density                    NA                   1         NA
## median_age                            NA                  NA          1
## life_expectancy                       NA                  NA         NA
## human_development_index               NA                  NA         NA
## gdp_per_capita                        NA                  NA         NA
##                       life_expectancy human_development_index gdp_per_capita
## Pop15Over                          NA                      NA             NA
## total_deaths                     0.13                      NA             NA
## total_cases                      0.08                      NA             NA
## new_cases                        0.08                      NA             NA
## new_deaths                       0.11                      NA             NA
## total_vaccinations                 NA                      NA             NA
## population_density                 NA                      NA             NA
## median_age                         NA                      NA             NA
## life_expectancy                  1.00                      NA             NA
## human_development_index            NA                       1             NA
## gdp_per_capita                     NA                      NA              1
```



** Prediction 1 - Multiple linear regression **

The results of the Multiple linear regression new_deaths ~ total_vaccinations + median_age + population_density + gdp_per_capita + life_expectancy using covid_train_set showed that all of the variables were statically significant (p-value: < 0.00000000000000022, Multiple R-squared: 0.5048). Whereas when using the validation set, it showed significance (p-value: 0.000000001437, Multiple R-squared: 0.6516) but only for total_vaccinations. Population density and life expectancy were the only variables with downward results.

```
# b) multilinear regression
   set.seed(1, sample.kind = "Rounding")
```

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```
   fit_new_deaths_train<-lm(new_deaths ~ total_vaccinations + median_age + population_density  + gdp_per
                       data = covid_train_set)
   fit_new_deaths_train
```

```
##
## Call:
## lm(formula = new_deaths ~ total_vaccinations + median_age + population_density +
##     gdp_per_capita + life_expectancy, data = covid_train_set)
##
## Coefficients:
##        (Intercept)  total_vaccinations          median_age  population_density
```

```
##       1959.2625195          0.0004981         15.0508878         -0.1021454
##     gdp_per_capita     life_expectancy
##          0.0072222        -33.6323142
```

```
summary(fit_new_deaths_train)
```

```
##
## Call:
## lm(formula = new_deaths ~ total_vaccinations + median_age + population_density +
##     gdp_per_capita + life_expectancy, data = covid_train_set)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2291.10  -155.45   -42.58    64.84  2073.50
##
## Coefficients:
##                         Estimate    Std. Error t value             Pr(>|t|)
## (Intercept)        1959.26251954  834.48850676   2.348              0.01988 *
## total_vaccinations    0.00049812    0.00003577  13.927 < 0.0000000000000002 ***
## median_age           15.05088778    5.42361943   2.775              0.00605 **
## population_density   -0.10214539    0.06224938  -1.641              0.10242
## gdp_per_capita        0.00722219    0.00237339   3.043              0.00266 **
## life_expectancy     -33.63231419   10.84508209  -3.101              0.00221 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 373.3 on 196 degrees of freedom
##   (46124 observations deleted due to missingness)
## Multiple R-squared:  0.5461, Adjusted R-squared:  0.5345
## F-statistic: 47.16 on 5 and 196 DF,  p-value: < 0.00000000000000022
```

```
set.seed(1, sample.kind = "Rounding")
```

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```
fit_new_deaths_validation<-lm(new_deaths ~ total_vaccinations + median_age + population_density  + g
                    data = validation)
fit_new_deaths_validation
```

```
##
## Call:
## lm(formula = new_deaths ~ total_vaccinations + median_age + population_density +
##     gdp_per_capita + life_expectancy, data = validation)
##
## Coefficients:
##       (Intercept)  total_vaccinations          median_age  population_density
##       3832.0715100          0.0005966         10.7777598         -0.1763277
##     gdp_per_capita     life_expectancy
##          0.0043429        -52.7380952
```

```r
summary(fit_new_deaths_validation)
```

```
##
## Call:
## lm(formula = new_deaths ~ total_vaccinations + median_age + population_density +
##     gdp_per_capita + life_expectancy, data = validation)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -883.5 -227.6 -113.2  160.6 2411.5
##
## Coefficients:
##                         Estimate    Std. Error t value        Pr(>|t|)
## (Intercept)        3832.07150998 1904.22518739   2.012          0.0492 *
## total_vaccinations    0.00059656    0.00007209   8.275 0.0000000000355 ***
## median_age           10.77775975   12.54127641   0.859          0.3939
## population_density   -0.17632774    0.17355287  -1.016          0.3142
## gdp_per_capita        0.00434288    0.00577133   0.752          0.4550
## life_expectancy     -52.73809518   26.65739561  -1.978          0.0530 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 491.9 on 54 degrees of freedom
##   (11384 observations deleted due to missingness)
## Multiple R-squared:  0.6203, Adjusted R-squared:  0.5851
## F-statistic: 17.64 on 5 and 54 DF,  p-value: 0.0000000002523
```

Classifier

** Prediction 2 - naive Bayes Classifier **

This is a simple probabilistic classifier which is based on Bayes theorem. the idea is taken the (prior probability of the outcome * likelihood or probability of observing the predictor values)/ evidence or probability of the predictor variables. Laplace smoother. The Laplace smoother adds a small number to each of the counts in the frequencies for each feature, which ensures that each feature has a nonzero probability of occuring for each class.

Comparing Total cases categories of Total cases < 50k, 50k >= Total cases < 200k, 200k >= Total cases < 500k, 500k >= Total cases < 1M, Total cases >= 1M, Data unavailable with GDP_category+ Pop_density_categ+ Total_deaths_categ + vaccination_status results of the naive model indicated:

- Highest priori probabilities for Total cases < 50k (75%) followed by 50k >= Total cases < 200k (13.8%). Also the model indicated highest probabilities for:

- *GDP_category (Categorical)*: 500k >= Total cases < 1M and 10k >= gdp < 50k (92.8%). Same gdp per capita category was higher for the Total cases >= 1M (68.7%)

- *Pop_density_categ (Categorical)*: Very High (100 >= ppl/Km2 <400) and 200k >= Total cases < 500k (47.4%). Total cases >= 1M had highest probability (41.6%) for Medium (25 >= ppl/Km2 < 50) pop density.
- *Total_deaths_categ (Categorical)*: 10k >= Total deaths< 100k and 500k >= Total cases < 1M (92.3%), whereas Total cases >= 1M had 63% to 10k >= Total deaths< 100k - *vaccination_status (Bernoulli)*: very low probabilities for vaccination rollout to all categories of total cases. Total cases < 50k was 99.9% for Vaccination did not start.

```r
#create a naive bayes model and fit it on train set
model <-naive_bayes(Total_cases_categ ~ GDP_category+ Pop_density_categ+ Total_deaths_categ + vaccinati
```

```
## Warning: naive_bayes(): Feature GDP_category - zero probabilities are present.
## Consider Laplace smoothing.
```

```
## Warning: naive_bayes(): Feature Pop_density_categ - zero probabilities are
## present. Consider Laplace smoothing.
```

```
## Warning: naive_bayes(): Feature Total_deaths_categ - zero probabilities are
## present. Consider Laplace smoothing.
```

```r
model
```

```
##
## ================================== Naive Bayes ==================================
##
##  Call:
## naive_bayes.formula(formula = Total_cases_categ ~ GDP_category +
##     Pop_density_categ + Total_deaths_categ + vaccination_status,
##     data = covid_train_set, usekernel = T)
##
## ---------------------------------------------------------------------------------
##
## Laplace smoothing: 0
##
## ---------------------------------------------------------------------------------
##
##  A priori probabilities:
##
## 200k >= Total cases < 500k   500k >= Total cases < 1M
##                0.054656133                0.022557527
##  50k >= Total cases < 200k           Data unavailable
##                0.138151362                0.009908043
##          Total cases < 50k          Total cases >= 1M
##                0.750032379                0.024694556
##
## ---------------------------------------------------------------------------------
##
##  Tables:
##
## ---------------------------------------------------------------------------------
##  ::: GDP_category (Categorical)
## ---------------------------------------------------------------------------------
##
## GDP_category       200k >= Total cases < 500k 500k >= Total cases < 1M
##   10k >= GDP < 50k                0.736176935              0.923444976
##   1k >= GDP < 5k                  0.066745656              0.017224880
##   50k >= GDP < 90k                0.026856240              0.012440191
##   5k >= GDP < 10k                 0.170221169              0.046889952
##   Data unavailable                0.000000000              0.000000000
##   GDP < 1k                        0.000000000              0.000000000
```

```
##   GDP >= 90K                      0.000000000             0.000000000
##
## GDP_category       50k >= Total cases < 200k Data unavailable Total cases < 50k
##   10k >= GDP < 50k               0.637343750      0.577342048       0.413428884
##   1k >= GDP < 5k                 0.089218750      0.034858388       0.274132274
##   50k >= GDP < 90k               0.083906250      0.180827887       0.042681172
##   5k >= GDP < 10k                0.161406250      0.185185185       0.165659356
##   Data unavailable               0.000000000      0.013071895       0.060927877
##   GDP < 1k                       0.000000000      0.000000000       0.033615380
##   GDP >= 90K                     0.028125000      0.008714597       0.009555057
##
## GDP_category       Total cases >= 1M
##   10k >= GDP < 50k       0.692307692
##   1k >= GDP < 5k         0.000000000
##   50k >= GDP < 90k       0.172202797
##   5k >= GDP < 10k        0.135489510
##   Data unavailable       0.000000000
##   GDP < 1k               0.000000000
##   GDP >= 90K             0.000000000
##
## ------------------------------------------------------------------------------
##  ::: Pop_density_categ (Categorical)
## ------------------------------------------------------------------------------
##
## Pop_density_categ                 200k >= Total cases < 500k
##   Data unavailable                                0.00000000
##   Extreme High (ppl/Km2 >400)                     0.10308057
##   High (50 >= ppl/Km2 < 100)                      0.17812006
##   Low (0 >= ppl/Km2 < 25)                         0.15955766
##   Medium (25 >= ppl/Km2 < 50)                     0.08491311
##   Very High (100 >= ppl/Km2 <400)                 0.47432859
##
## Pop_density_categ                 500k >= Total cases < 1M
##   Data unavailable                              0.00000000
##   Extreme High (ppl/Km2 >400)                   0.06985646
##   High (50 >= ppl/Km2 < 100)                    0.21052632
##   Low (0 >= ppl/Km2 < 25)                       0.17416268
##   Medium (25 >= ppl/Km2 < 50)                   0.33397129
##   Very High (100 >= ppl/Km2 <400)               0.21148325
##
## Pop_density_categ                 50k >= Total cases < 200k Data unavailable
##   Data unavailable                               0.00000000       0.01307190
##   Extreme High (ppl/Km2 >400)                    0.09218750       0.23747277
##   High (50 >= ppl/Km2 < 100)                     0.24968750       0.20479303
##   Low (0 >= ppl/Km2 < 25)                        0.16312500       0.15250545
##   Medium (25 >= ppl/Km2 < 50)                    0.08671875       0.15468410
##   Very High (100 >= ppl/Km2 <400)                0.40828125       0.23747277
##
## Pop_density_categ                 Total cases < 50k Total cases >= 1M
##   Data unavailable                       0.03214759        0.00000000
##   Extreme High (ppl/Km2 >400)            0.09929200        0.12150350
##   High (50 >= ppl/Km2 < 100)             0.22868819        0.10576923
##   Low (0 >= ppl/Km2 < 25)                0.21470097        0.13986014
##   Medium (25 >= ppl/Km2 < 50)            0.12335233        0.41783217
```

```
##   Very High (100 >= ppl/Km2 <400)        0.30181891        0.21503497
##
## ------------------------------------------------------------------------------
##  ::: Total_deaths_categ (Categorical)
## ------------------------------------------------------------------------------
##
## Total_deaths_categ          200k >= Total cases < 500k 500k >= Total cases < 1M
##   100 >= Total deaths < 1k                0.003554502              0.000000000
##   10k >= Total deaths< 100k               0.356635071              0.923444976
##   1k >= Total deaths < 10k                0.639810427              0.076555024
##   Data unavailable                        0.000000000              0.000000000
##   Total deaths < 100                      0.000000000              0.000000000
##   Total deaths >= 100k                    0.000000000              0.000000000
##
## Total_deaths_categ          50k >= Total cases < 200k Data unavailable
##   100 >= Total deaths < 1k               0.243281250      0.000000000
##   10k >= Total deaths< 100k              0.035156250      0.000000000
##   1k >= Total deaths < 10k               0.697500000      0.000000000
##   Data unavailable                       0.000000000      1.000000000
##   Total deaths < 100                     0.024062500      0.000000000
##   Total deaths >= 100k                   0.000000000      0.000000000
##
## Total_deaths_categ          Total cases < 50k Total cases >= 1M
##   100 >= Total deaths < 1k        0.277787371       0.000000000
##   10k >= Total deaths< 100k       0.000000000       0.631993007
##   1k >= Total deaths < 10k        0.038767052       0.000000000
##   Data unavailable                0.194756231       0.000000000
##   Total deaths < 100              0.488689346       0.000000000
##   Total deaths >= 100k            0.000000000       0.368006993
##
## ------------------------------------------------------------------------------
##  ::: vaccination_status (Bernoulli)
## ------------------------------------------------------------------------------
##
## vaccination_status          200k >= Total cases < 500k 500k >= Total cases < 1M
##   Vaccination did not start               0.985387046              0.966507177
##   Vaccination started                     0.014612954              0.033492823
##
## vaccination_status          50k >= Total cases < 200k Data unavailable
##   Vaccination did not start              0.991406250      0.976034858
##   Vaccination started                    0.008593750      0.023965142
##
## vaccination_status          Total cases < 50k Total cases >= 1M
##   Vaccination did not start       0.999309273       0.955419580
##   Vaccination started             0.000690727       0.044580420
##
## ------------------------------------------------------------------------------
```

```
p1<-predict(model,covid_train_set)
```

```
## Warning: predict.naive_bayes(): more features in the newdata are provided as
## there are probability tables in the object. Calculation is performed based on
## features to be found in the tables.
```

```
tab1<- table(p1, covid_train_set$Total_cases_categ)
tab1
```

```
##
## p1                           200k >= Total cases < 500k
##   200k >= Total cases < 500k                        436
##   500k >= Total cases < 1M                          463
##   50k >= Total cases < 200k                        1496
##   Data unavailable                                    0
##   Total cases < 50k                                 133
##   Total cases >= 1M                                   4
##
## p1                           500k >= Total cases < 1M 50k >= Total cases < 200k
##   200k >= Total cases < 500k                      267                       114
##   500k >= Total cases < 1M                        683                       111
##   50k >= Total cases < 200k                        62                      4355
##   Data unavailable                                  0                         0
##   Total cases < 50k                                18                      1820
##   Total cases >= 1M                                15                         0
##
## p1                           Data unavailable Total cases < 50k
##   200k >= Total cases < 500k                0                 0
##   500k >= Total cases < 1M                  0                 0
##   50k >= Total cases < 200k                 0              1252
##   Data unavailable                         11                 0
##   Total cases < 50k                       448             33494
##   Total cases >= 1M                         0                 0
##
## p1                           Total cases >= 1M
##   200k >= Total cases < 500k               291
##   500k >= Total cases < 1M                 411
##   50k >= Total cases < 200k                  0
##   Data unavailable                           0
##   Total cases < 50k                          0
##   Total cases >= 1M                        442
```

```
#calculate train accuracy by dividing the numbers of correct predictions on total numbers of data point
trainacc=sum(diag(tab1))/sum(tab1)
trainacc
```

```
## [1] 0.8509476
```

```
p2<-predict(model,covid_test_set)
```

```
## Warning: predict.naive_bayes(): more features in the newdata are provided as
## there are probability tables in the object. Calculation is performed based on
## features to be found in the tables.
```

```
tab2<- table(p2, covid_test_set$Total_cases_categ)
tab2
```

```
##
## p2                          200k >= Total cases < 500k
##   200k >= Total cases < 500k                        126
##   500k >= Total cases < 1M                          115
##   50k >= Total cases < 200k                         387
##   Data unavailable                                    0
##   Total cases < 50k                                  33
##   Total cases >= 1M                                   2
##
## p2                          500k >= Total cases < 1M 50k >= Total cases < 200k
##   200k >= Total cases < 500k                       74                       33
##   500k >= Total cases < 1M                        159                       31
##   50k >= Total cases < 200k                        20                     1085
##   Data unavailable                                  0                        0
##   Total cases < 50k                                 1                      426
##   Total cases >= 1M                                 4                        0
##
## p2                          Data unavailable Total cases < 50k
##   200k >= Total cases < 500k                0                 1
##   500k >= Total cases < 1M                  0                 0
##   50k >= Total cases < 200k                 0               304
##   Data unavailable                          4                 0
##   Total cases < 50k                        96              8237
##   Total cases >= 1M                         0                 0
##
## p2                          Total cases >= 1M
##   200k >= Total cases < 500k                80
##   500k >= Total cases < 1M                 111
##   50k >= Total cases < 200k                  0
##   Data unavailable                           0
##   Total cases < 50k                          0
##   Total cases >= 1M                        115
```

```
#calculate test accuracy by dividing the numbers of correct predictions on total numbers of data points
testacc=sum(diag(tab2))/sum(tab2)
testacc
```

```
## [1] 0.8498777
```

** Prediction 3 - Decision Tree **

In order to read Decision Tree be aware that right side means (yes) and left side means (no)

At the top, it is the overall probability of be European countries (46%)

For Total cases categories: 200k >= Total cases < 500k is related to 7%, which European countries are 27%. Total cases >= 1M represents 5%, oout of America countries is 57%.

```
#Fancy decision tree

europe_americas_train <-covid_train_set %>% filter(region=="Europe" | region=="Americas")
europe_americas_test <-validation %>% filter(region=="Europe" | region=="Americas")

tree_train <- rpart( region ~ Total_cases_categ , data = europe_americas_train, method = "class",
                control = rpart.control(minsplit = 1, minbucket = 1, cp = 0.001))
tree_train
```

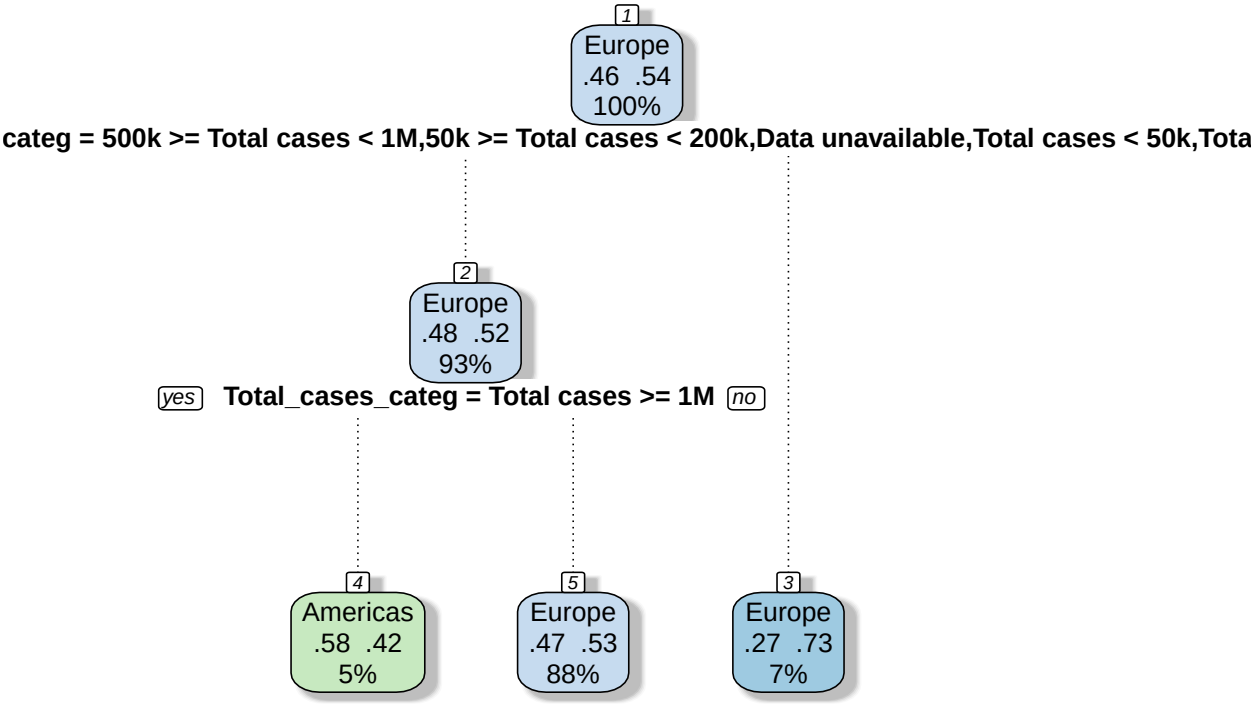
```
## n= 18842
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 18842 8729 Europe (0.4632735 0.5367265)
##   2) Total_cases_categ=500k >= Total cases < 1M,50k >= Total cases < 200k,Data unavailable,Total case
##     4) Total_cases_categ=Total cases >= 1M 944  397 Americas (0.5794492 0.4205508) *
##     5) Total_cases_categ=500k >= Total cases < 1M,50k >= Total cases < 200k,Data unavailable,Total ca
##   3) Total_cases_categ=200k >= Total cases < 500k 1345  366 Europe (0.2721190 0.7278810) *
```

```
fancyRpartPlot(tree_train,yesno = 2, cex=0.8)
```

```
tree_test <- rpart( region ~ Total_cases_categ , data = europe_americas_train, method = "class",
               control = rpart.control(minsplit = 1, minbucket = 1, cp = 0.001))
tree_test
```

```
## n= 18842
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 18842 8729 Europe (0.4632735 0.5367265)
##   2) Total_cases_categ=500k >= Total cases < 1M,50k >= Total cases < 200k,Data unavailable,Total case
##     4) Total_cases_categ=Total cases >= 1M 944  397 Americas (0.5794492 0.4205508) *
##     5) Total_cases_categ=500k >= Total cases < 1M,50k >= Total cases < 200k,Data unavailable,Total ca
##   3) Total_cases_categ=200k >= Total cases < 500k 1345  366 Europe (0.2721190 0.7278810) *
```

```
fancyRpartPlot(tree_test,yesno = 2, cex=0.8)
```



3. Conclusion Limitation of this project was related to working with longitudinal data (covid-19). As data changes over time, it was challenging to validate the data. Also, I encounter days which the covid-19 package had issue with its server, in other instances data was missing or did not properly updated.

What I notice frequently, especially on the time zone change from one day to another was that the covid-19 cases and deaths were updated, but there was a lag on the vaccination information (vice verse). So, if I am trying to retrieve a subset with the most recent date, then the data was not complete. My solution to this temporal data issues as to create the most recent date that each individual variable was published (total deaths, total cases and vaccination). I then joined these subsets, selecting just the variables that I would be plotting and were updated).

I also noticed a few changes on the covid-19 data structure itself (new columns were added), probably to accommodate the reality, specially related to the vaccination information. Vaccination administration is related to a single dose of the covid-19 vaccine. For example, it was impossible for me to know if the total vaccinations data of the day 21 (when the 2nd dose supposed to be administrated for some vaccine) was related to a single first or second dose.

There are many other geographical-socio-economic questions that I would be curious to know. This is just a beginning of my insights.

4. Reference Bendix, To know the real number of coronavirus cases in the US, China, or Italy, researchers say multiply by 10. Accesed:Apr 19, 2020, 12:50 PM https://www.businessinsider.com/real-number-of-coronavirus-cases-underreported-us-china-italy-2020-4

Hale T, Phillips T, Petherick A, Kira B, Angrist N, Aymar K, et al. Risk of Openness Index: when do government responses need to be increased or maintained? [Internet]. Version 2.0. Oxford:Blavatnik School of Government; 2020 [cited 2020 Oct 21] https://www.publichealthontario.ca/-/media/documents/ncov/research/

2020/10/research-hale-risk-of-openness-index.pdf?la=en based on https://www.bsg.ox.ac.uk/sites/default/files/2020-10/10-2020-Risk-of-Openness-Index-BSG-Research-Note.pdf

HDI https://ec.europa.eu/environment/beyond_gdp/download/factsheets/bgdp-ve-hdi.pdf

HDI wikipedia 2019 https://en.wikipedia.org/wiki/Human_Development_Index

https://www.maartenlambrechts.com/2017/10/22/tutorial-a-worldtilegrid-with-ggplot2.html https://github.com/ishaberry/Covid19Canada, https://github.com/kaerosen/tilemaps"

Naïve Bayes Classifier - https://uc-r.github.io/naive_bayes

Mohammed, R.A. Longitudinal Data Integration for a Tracking System for Health Professionals. Masters Thesis. UNIVERSITY OF NEW BRUNSWICK, 2016