Instruction Manual

Tobias Konieczka

2023-10-04

Introduction

The covid19_vaccine data set obtained from RamiKrispin on Github as part of their "coronavirus" package was selected for analysis. According to the description, the data set comes from Johns Hopkins Centers for Civic Impact global vaccination data, and is presented in long format by defualt. It can be found here. Due to the sheer breadth of data included in this dataset, functions designed to organize and filter the data needed to be constructed. Using these functions, new questions can be asked: | What is the vaccination information in all countries in Africa? | How did the percentage of vaccinated individuals in a country change over a five day period? | Which country had the highest number of total vaccinations on a certain day? | Is there an association between the rate of vaccination and its position on the globe? These are just a few of the possible questions that can be raised and answered using

KonieTobFin Package

The analysis of the data was made possible and streamlined through the development of the **KonieTobFin** package. The functions in the package were designed to accomplish a number of data management and analytical tasks. The package can be installed from my Github and loaded in R using the following code snippet:

```
devtools::install_github("konieczkat/KonieTobFin")
library(tidyverse)
library("KonieTobFin")
```

get_data()

The get_data() Function is used to gather the vaccine dataset from github. It will fetch the coronavirus package from Github, install it onto the user's system, and attaches them to the R session. Use of this function is required for use of the other methods in the package.

```
full_vaccine_data <- get_data()
head(full_vaccine_data)
```

```
## # A tibble: 6 x 15
##   date       country_region continent_name continent_code combined_key
##   <date>     <chr>          <chr>          <chr>          <chr>
## 1 2020-12-29 Austria        Europe         EU             Austria
## 2 2020-12-29 Bahrain        Asia           AS             Bahrain
## 3 2020-12-29 Belarus        Europe         EU             Belarus
```

```
## 4 2020-12-29 Belgium        Europe         EU             Belgium
## 5 2020-12-29 Canada         North America  NA             Canada
## 6 2020-12-29 Chile          South America  SA             Chile
## # i 10 more variables: doses_admin <int>, people_at_least_one_dose <dbl>,
## #   population <dbl>, uid <dbl>, iso2 <chr>, iso3 <chr>, code3 <dbl>,
## #   fips <chr>, lat <dbl>, long <dbl>
```

The raw vaccination data is presented in tibble form and contains 15 columns, which include date, integer, double, and character types. There are also 142597 rows, as the data contains information from 195 countries between 2020-12-29 and 2023-03-09. There are 142597 observations in the dataset.

relevise()

A number of the columns in the dataset represent identifiers that will not be used during analysis. The relevise() function was designed to remove these unnecessary columns for data management purposes.

```
data <- relevise(full_vaccine_data)
head(data)
```

```
## # A tibble: 6 x 8
##   date       continent_name country_region doses_admin people_at_least_one_dose
##   <date>     <chr>          <chr>                <int>                     <dbl>
## 1 2020-12-29 Europe         Austria               2123                      2123
## 2 2020-12-29 Asia           Bahrain              55014                     55014
## 3 2020-12-29 Europe         Belarus                  0                         0
## 4 2020-12-29 Europe         Belgium                340                       340
## 5 2020-12-29 North America  Canada               59079                     59078
## 6 2020-12-29 South America  Chile                   NA                        NA
## # i 3 more variables: population <dbl>, lat <dbl>, long <dbl>
```

Seven of the original 15 rows were removed to streamline the analyses. The columns that remain include date, continent, country, total doses administered, the number of people with at least one dose, the population of the locality, as well as the latitude and longitude for each locality.

Percentage Calculation

A new column representing the percentage of the population that has received at least one dose can be appended to the modified dataset using the percent_vaccinated() function.

```
data <- percent_vaccinated(data)
head(data)
```

```
## # A tibble: 6 x 9
##   date       continent_name country_region doses_admin people_at_least_one_dose
##   <date>     <chr>          <chr>                <int>                     <dbl>
## 1 2020-12-29 Europe         Austria               2123                      2123
## 2 2020-12-29 Asia           Bahrain              55014                     55014
## 3 2020-12-29 Europe         Belarus                  0                         0
## 4 2020-12-29 Europe         Belgium                340                       340
## 5 2020-12-29 North America  Canada               59079                     59078
## 6 2020-12-29 South America  Chile                   NA                        NA
## # i 4 more variables: population <dbl>, lat <dbl>, long <dbl>,
## #   Percent_Vaccinated <dbl>
```

Dosing information could be present (as in row 1, 2, 4, and 5 of the above table), have a value of 0 (as in the third row), or missing (as given by NA in row 6). Missing data takes the form on NA throughout the dataset, but the functions have been designed to handle them accordingly.

Filter Methods

Three tibble filtration methods were developed to subset the data for three different purposes. They relate to the data's presence in spacetime.

filter_by_continent()

The filter_by_continent() method is used to filter the vaccination data by a specified country of interest. For example, the function can be used to isolate data from all countries in Africa.

```
Africa <- data %>% filter_by_continent(., "Africa")
head(Africa)
```

```
## # A tibble: 6 x 9
##   date       continent_name country_region doses_admin people_at_least_one_dose
##   <date>     <chr>          <chr>                <int>                     <dbl>
## 1 2021-01-10 Africa         Seychelles               0                         0
## 2 2021-01-11 Africa         Seychelles               0                         0
## 3 2021-01-12 Africa         Seychelles               0                         0
## 4 2021-01-13 Africa         Seychelles               0                         0
## 5 2021-01-14 Africa         Seychelles            2000                      2000
## 6 2021-01-15 Africa         Seychelles            2000                      2000
## # i 4 more variables: population <dbl>, lat <dbl>, long <dbl>,
## #   Percent_Vaccinated <dbl>
```

filter_by_continent()

The filter_by_country() method can be used to filter the vaccination data by a specified country of interest. For example, the function can be used to isolate all vaccination data from Mexico.

```
Mexico <- data %>% filter_by_country(., "Mexico")
head(Mexico)
```

```
## # A tibble: 6 x 9
##   date       continent_name country_region doses_admin people_at_least_one_dose
##   <date>     <chr>          <chr>                <int>                     <dbl>
## 1 2020-12-29 North America  Mexico                9579                      9579
## 2 2020-12-30 North America  Mexico               18529                     18529
## 3 2020-12-31 North America  Mexico               24998                     24998
## 4 2021-01-01 North America  Mexico               24998                     24998
## 5 2021-01-02 North America  Mexico               24998                     24998
## 6 2021-01-03 North America  Mexico               24998                     24998
## # i 4 more variables: population <dbl>, lat <dbl>, long <dbl>,
## #   Percent_Vaccinated <dbl>
```

filter_by_date()

The filter_by_date() function can be used to filter the vaccination data through a specified period of time. For example, the function can be used to isolate vaccination data in Mexico from January 13th, 2021, to January 17th, 2021.

```
mexicoJan <- Mexico %>% filter_by_date(., "2021-01-13", "2021-01-17")
head(mexicoJan)
```

```
## # A tibble: 5 x 9
##   date       continent_name country_region doses_admin people_at_least_one_dose
##   <date>     <chr>          <chr>                <int>                     <dbl>
## 1 2021-01-13 North America  Mexico               92879                     92879
## 2 2021-01-14 North America  Mexico              192567                    192567
## 3 2021-01-15 North America  Mexico              329983                    329983
## 4 2021-01-16 North America  Mexico              417375                    415417
## 5 2021-01-17 North America  Mexico              463246                    461025
## # i 4 more variables: population <dbl>, lat <dbl>, long <dbl>,
## #   Percent_Vaccinated <dbl>
```

Over a period of five days, 3.68146×10^5 Mexicans received at least one dose of the covid vaccine. Using this method in conjunction with the other methods allows the user to identify the global vaccination data on a given day.

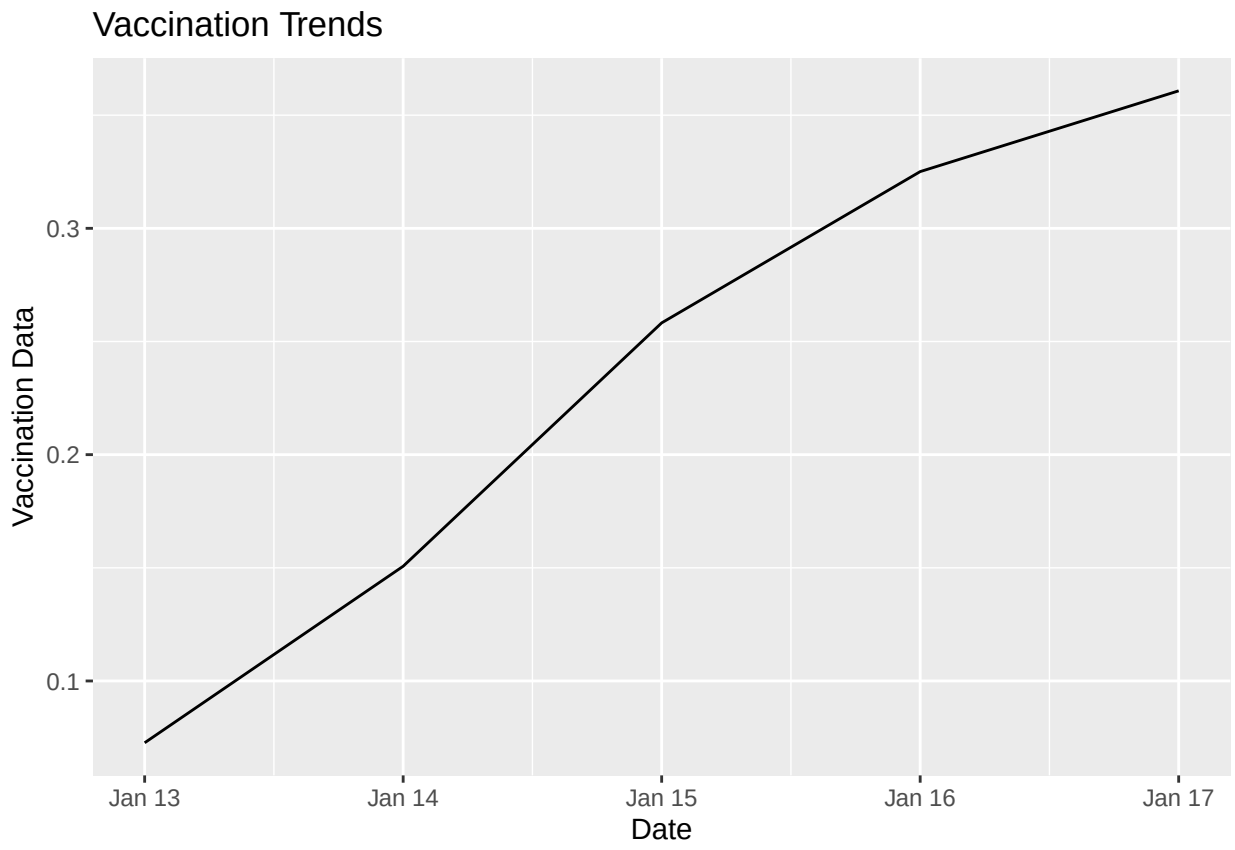
```
global_Jan_13_2021 <- data %>% filter_by_date(., "2021-01-13", "2021-01-13")
head(global_Jan_13_2021)
```

```
## # A tibble: 6 x 9
##   date       continent_name country_region doses_admin people_at_least_one_dose
##   <date>     <chr>          <chr>                <int>                     <dbl>
## 1 2021-01-13 Europe         Albania                128                       128
## 2 2021-01-13 South America  Argentina           175334                    175257
## 3 2021-01-13 Europe         Austria              52730                     52725
## 4 2021-01-13 Asia           Bahrain              97776                     97776
## 5 2021-01-13 Europe         Belarus                  0                         0
## 6 2021-01-13 Europe         Belgium              50579                     50528
## # i 4 more variables: population <dbl>, lat <dbl>, long <dbl>,
## #   Percent_Vaccinated <dbl>
```

visualyze_line()

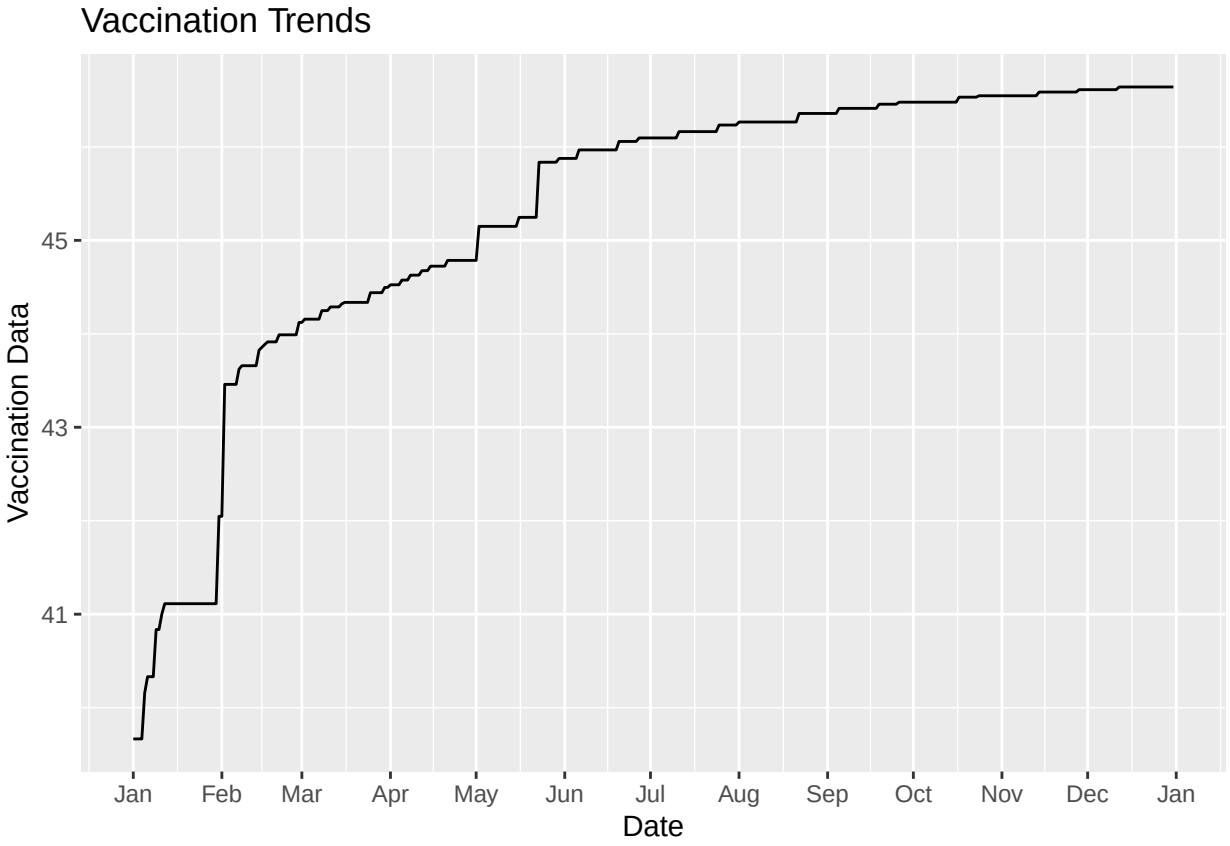
The visualyze_line() function can be used to plot a line graph representing the change in a variable over time. For example, using the vaccination info for Mexico from January 13th, 2021, to January 17th, 2021, a graph of the percentage of vaccinated people can be produced.

```
mexicoJan %>% visualyze_line(., .$Percent_Vaccinated)
```



Larger ranges can also be used, and the breaks on the x-axis can be adjusted. Let's take a look at the vaccination percentage in Albania from January 1st, 2022 to December 31st, 2022.

```
data %>% filter_by_country(.,"Albania") %>% filter_by_date(., "2022-01-01", "2022-12-31") %>% visualyze
```



We can see that there was a large spike of vaccinations in Albania in early Febraury 2022 following a plateua of vaccinations in mid to late January.

relation_to_location()

The relation_to_location() function can be used to determine the relationship between vaccination information and position in space. For example, the relationship between total doses and latitude, total doses and longitude, and total doses and longitude and latitude in every country over the course of the study.

```
data %>% relation_to_location(., .$doses_admin, .$lat, .$long)
```

```
##
## Call:
## lm(formula = var ~ lat + long + lat:long, data = data)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
##  -69393550  -32668717  -23765029  -10379068 2102507471
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.633e+07  4.134e+05  63.688  < 2e-16 ***
## lat         1.116e+05  1.371e+04   8.144 3.86e-16 ***
## long        1.050e+05  5.317e+03  19.748  < 2e-16 ***
## lat:long    4.003e+03  2.223e+02  18.009  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 115900000 on 138894 degrees of freedom
##   (3699 observations deleted due to missingness)
## Multiple R-squared:  0.009185,   Adjusted R-squared:  0.009164
## F-statistic: 429.2 on 3 and 138894 DF,  p-value: < 2.2e-16
```

The output of this call demonstrates that there is a clear association between the number of doses administered in a country and that country's location on Earth.

Future Considerations

As it stands, this package serves to simplify a very large dataset containing a large number of observations and a moderate amount of columns into a more manageable form. While the analytical functions utilized in this package were designed to work with subset forms of the orignal data, they are limited in functionality as they can only really be used with RamiKrispin's vaccine data. The visualization function in the package is only able to produce a line graph, which is limiting. Since there is a data point from every country for every day over a few year period, space and time were the defining factors in the analysis. Since I'm not a statistician, I'm sure there are seemingly obvious research questions that could be answered through some combination of my function, although the visualyze_line and relation_to_location methods are may be too specific to be applied in any way other than their current implementation. Ultimately, the functions contained within these packages are designed more with data management in mind, rather than analytical considerations. This could, of course, be due to the limited application of a number of the original columns, which were almost always identifiers. Had more quantitative data been available in the original data (i.e., what company produced the vaccine used in each country, updates to the population data for each country, mortality rates, etc) I believe that more statistically interesting/relevant questions could be asked and answered using these methods.