EDA ON COUNTRY VACCINATIONS DATASET

PES University

The Aggregators

```r
library(readr)
country_vaccinations <- read_csv("E:/SEM 5/E1 CS312 DA/DA PROJECT/country_vaccinations.csv")
```

```
## Rows: 33358 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (5): country, iso_code, vaccines, source_name, source_website
## dbl  (9): total_vaccinations, people_vaccinated, people_fully_vaccinated, da...
## date (1): date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
country_vaccinations <- country_vaccinations[,c("country", "total_vaccinations", "date", "people_vaccina
```

```r
dim(country_vaccinations)
```

```
## [1] 33358     8
```

```r
sum(is.na(country_vaccinations))
```

```
## [1] 64207
```

```r
summary(is.na(country_vaccinations))
```

```
##   country         total_vaccinations    date          people_vaccinated
##  Mode :logical   Mode :logical      Mode :logical   Mode :logical
##  FALSE:33358     FALSE:18619        FALSE:33358     FALSE:17754
##                  TRUE :14739                        TRUE :15604
##  daily_vaccinations_raw people_vaccinated_per_hundred
##  Mode :logical          Mode :logical
##  FALSE:15356            FALSE:17754
##  TRUE :18002            TRUE :15604
##  daily_vaccinations_per_million  vaccines
##  Mode :logical                  Mode :logical
##  FALSE:33100                    FALSE:33358
##  TRUE :258
```

```r
sapply(country_vaccinations, function(x) sum(is.na(x)))
```

```
##                         country             total_vaccinations
##                               0                          14739
##                            date              people_vaccinated
##                               0                          15604
##          daily_vaccinations_raw  people_vaccinated_per_hundred
##                           18002                          15604
## daily_vaccinations_per_million                        vaccines
##                             258                              0
```

```r
var1 <- unique(country_vaccinations[,c("country","date")])
dim(var1)
```

```
## [1] 33358     2
```

The data-set we are working on here has 86512 ROWS and 8 COLUMNS.

It has a very sizable number of missing values, here 184790 observations across the data-set.

Data inconsistency prevails as long as missing values are not treated properly.

Duplicates are also looked into and resolved due to the combined uniqueness of two attributes in this particular data-set

```r
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```r
country_vaccinations$date <- as.Date(country_vaccinations$date)
country_vaccinations$date <- as.Date(country_vaccinations$date)
country_vaccinations$total_vaccinations[is.na(country_vaccinations$total_vaccinations)==T] <- 0
country_vaccinations$people_vaccinated[is.na(country_vaccinations$people_vaccinated)==T] <- 0
country_vaccinations$daily_vaccinations_raw[is.na(country_vaccinations$daily_vaccinations_raw)==T] <- 0
country_vaccinations$people_vaccinated_per_hundred[is.na(country_vaccinations$people_vaccinated_per_hund
country_vaccinations$daily_vaccinations_per_million[is.na(country_vaccinations$daily_vaccinations_per_mi
head <- country_vaccinations[sample(1:nrow(country_vaccinations),5), ]
head[order(head$date),]
```

```
## # A tibble: 5 x 8
##   country             total~1 date       peopl~2 daily~3 peopl~4 daily~5 vacci~6
##   <chr>                 <dbl> <date>       <dbl>   <dbl>   <dbl>   <dbl> <chr>
## 1 Turks and Caicos I~       0 2021-03-06       0       0     0      8058 Pfizer~
## 2 Portugal            4090614 2021-05-10 2966108   66805    29.1    8233 Johnso~
## 3 Slovakia            1905260 2021-05-11 1301332   31254    23.8    5738 Modern~
## 4 Grenada                   0 2021-06-05       0       0     0      1733 Oxford~
## 5 Northern Cyprus           0 2021-06-19       0       0     0      5489 Oxford~
```

```
## # ... with abbreviated variable names 1: total_vaccinations,
## #   2: people_vaccinated, 3: daily_vaccinations_raw,
## #   4: people_vaccinated_per_hundred, 5: daily_vaccinations_per_million,
## #   6: vaccines
```

```
country_vaccinations$month <- month(country_vaccinations$date)
country_vaccinations$weekday <- weekdays(country_vaccinations$date)
country_vaccinations$percent_people <- country_vaccinations$people_vaccinated_per_hundred/100
numcol_country_vaccinations <- country_vaccinations[,c('total_vaccinations','people_vaccinated','daily_
```

Missing values have been filled with zeroes as no other metric is suitable.

This is done to ensure completeness and help us with our further observations.

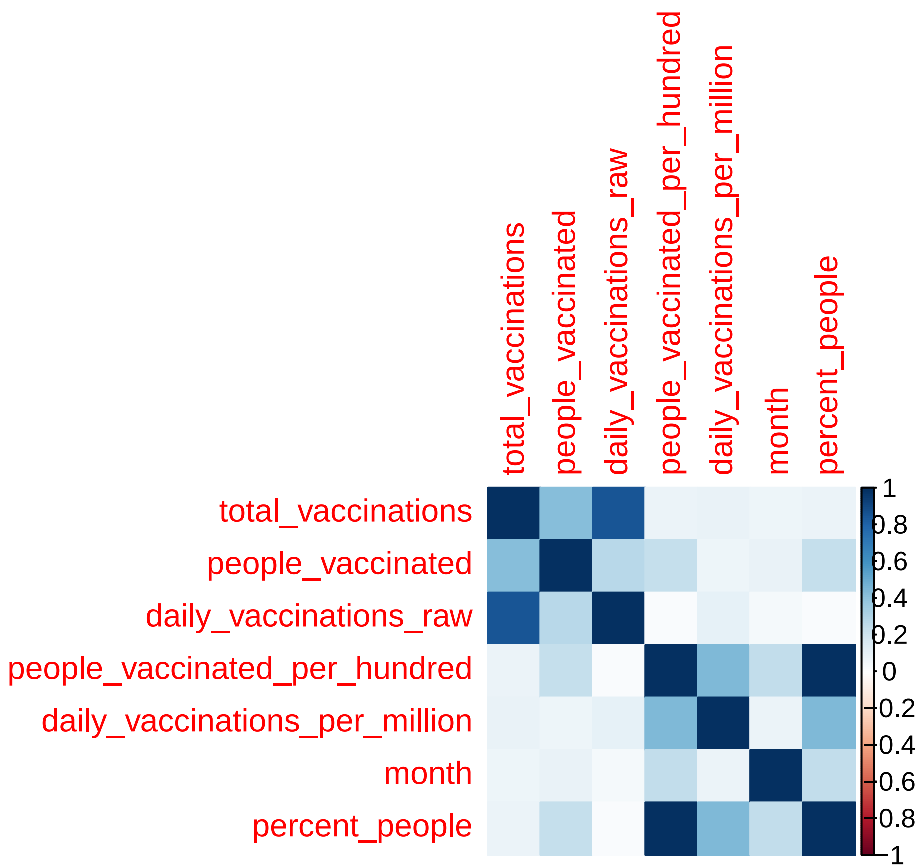
```
library(ggpubr)
```

```
## Loading required package: ggplot2
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
M = cor(numcol_country_vaccinations)
corrplot(M, method = 'color')
```



The correlation plot can be observed to say there is no negative correlation between any of the attributes.

percent_people and people_vaccinated_per_hundred is very strongly correlated.

Most attributes that depend on people or attributes that directly contribute to another attribute (eg: people_vaccinated and total_vaccinations) show high correlation.

COMMENTED CODE:

```
#library(fpp2)
 #autoplot(ts(numcol_country_vaccinations$total_vaccinations))
 #autoplot(ts(numcol_country_vaccinations$people_vaccinated))
 #autoplot(ts(numcol_country_vaccinations$daily_vaccinations_raw))
 #autoplot(ts(numcol_country_vaccinations$people_vaccinated_per_hundred))
 #autoplot(ts(numcol_country_vaccinations$daily_vaccinations_per_million))
 #autoplot(ts(numcol_country_vaccinations$month))

 #autoplot(ts(numcol_country_vaccinations$percent_people))

 #tsoutliers(numcol_country_vaccinations$total_vaccinations)

 #tsoutliers(numcol_country_vaccinations$people_vaccinated)

#tsoutliers(numcol_country_vaccinations$daily_vaccinations_raw)

 #tsoutliers(numcol_country_vaccinations$people_vaccinated_per_hundred)

 #tsoutliers(numcol_country_vaccinations$daily_vaccinations_per_million)

 #tsoutliers(numcol_country_vaccinations$month)

 #tsoutliers(numcol_country_vaccinations$percent_people)


 #autoplot(tsclean(ts((numcol_country_vaccinations$total_vaccinations))), series="clean", color='red',


 #autoplot(tsclean(ts((numcol_country_vaccinations$people_vaccinated))), series="clean", color='red', l


 #autoplot(tsclean(ts((numcol_country_vaccinations$daily_vaccinations_raw))), series="clean", color='re


 #autoplot(tsclean(ts((numcol_country_vaccinations$daily_vaccinations_raw))), series="clean", color='re


 #autoplot(tsclean(ts((numcol_country_vaccinations$people_vaccinated_per_hundred))), series="clean", co


 #autoplot(tsclean(ts((numcol_country_vaccinations$daily_vaccinations_per_million))), series="clean", c

 #autoplot(tsclean(ts((numcol_country_vaccinations$month))), series="clean", color='red', lwd=0.9) +aut

#autoplot(tsclean(ts((numcol_country_vaccinations$percent_people))), series="clean", color='red', lwd=0
```

A block of code has been commented above which identifies and caps the outliers that fall outside a certain range of values.

CONCLUSION:

Outliers were identified by transforming into time series data but could not be replaced by a suitable metric since this

data-set comprises of real time data which is necessary for our study.

Hence we will not be addressing them as outliers thus making the outlier count equal to 0.

```
numcol_country_vaccinations.pca <- prcomp(numcol_country_vaccinations[,c(1:7)],
                   center = TRUE,
                   scale. = TRUE)

summary(numcol_country_vaccinations.pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5     PC6       PC7
## Standard deviation     1.6132 1.3811 0.9674 0.9036 0.77730 0.36553 1.709e-13
## Proportion of Variance 0.3718 0.2725 0.1337 0.1166 0.08631 0.01909 0.000e+00
## Cumulative Proportion  0.3718 0.6443 0.7780 0.8946 0.98091 1.00000 1.000e+00
```

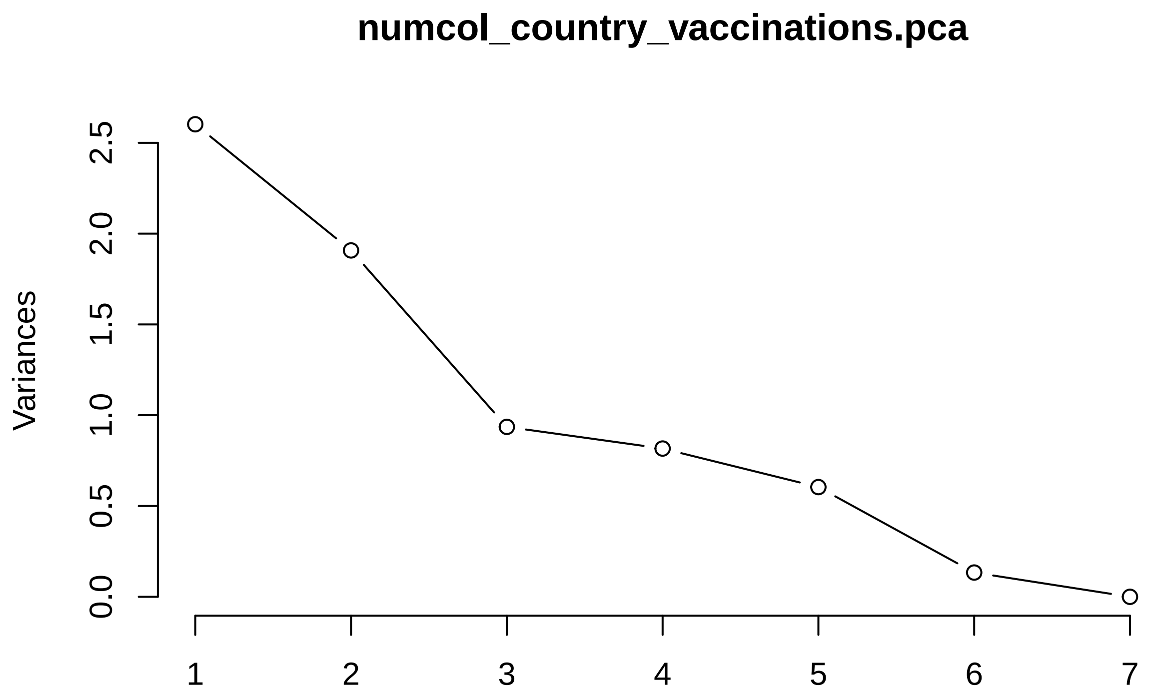
Proportion of variance for all 7 numeric principal components is low and PCA would not be the best option.

Other transformations also do not seem fit due to the nature of this data-set.

```
str(numcol_country_vaccinations.pca)
```

```
## List of 5
##  $ sdev    : num [1:7] 1.613 1.381 0.967 0.904 0.777 ...
##  $ rotation: num [1:7, 1:7] 0.313 0.311 0.269 0.528 0.355 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:7] "total_vaccinations" "people_vaccinated" "daily_vaccinations_raw" "people_vacci
##   .. ..$ : chr [1:7] "PC1" "PC2" "PC3" "PC4" ...
##  $ center  : Named num [1:7] 7.48e+06 3.29e+06 1.08e+05 1.08e+01 3.45e+03 ...
##   ..- attr(*, "names")= chr [1:7] "total_vaccinations" "people_vaccinated" "daily_vaccinations_raw"
##  $ scale   : Named num [1:7] 5.63e+07 1.68e+07 8.72e+05 1.86e+01 4.53e+03 ...
##   ..- attr(*, "names")= chr [1:7] "total_vaccinations" "people_vaccinated" "daily_vaccinations_raw"
##  $ x       : num [1:33358, 1:7] -1.32 -1.31 -1.31 -1.31 -1.31 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:7] "PC1" "PC2" "PC3" "PC4" ...
##  - attr(*, "class")= chr "prcomp"
```

```
plot.numcol_country_vaccinations.pca <- plot(numcol_country_vaccinations.pca, type="l")
```




```
plot.numcol_country_vaccinations.pca
```

```
## NULL
```

In the screeplot above, the ‘arm-bend’ represents a decrease in cumulative contribution.

The above plot shows the bend at the third principal component.

```
library(fpp2)
```

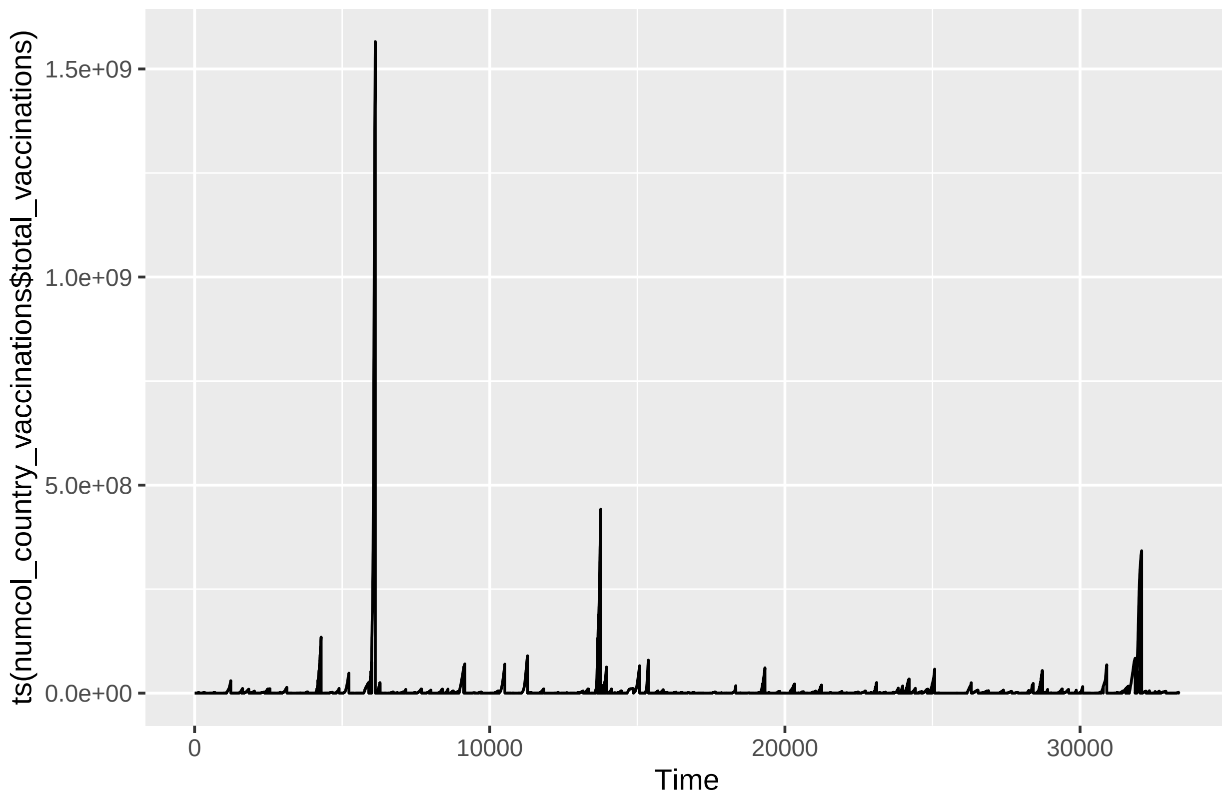
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## -- Attaching packages ---------------------------------------------- fpp2 2.4 --
```

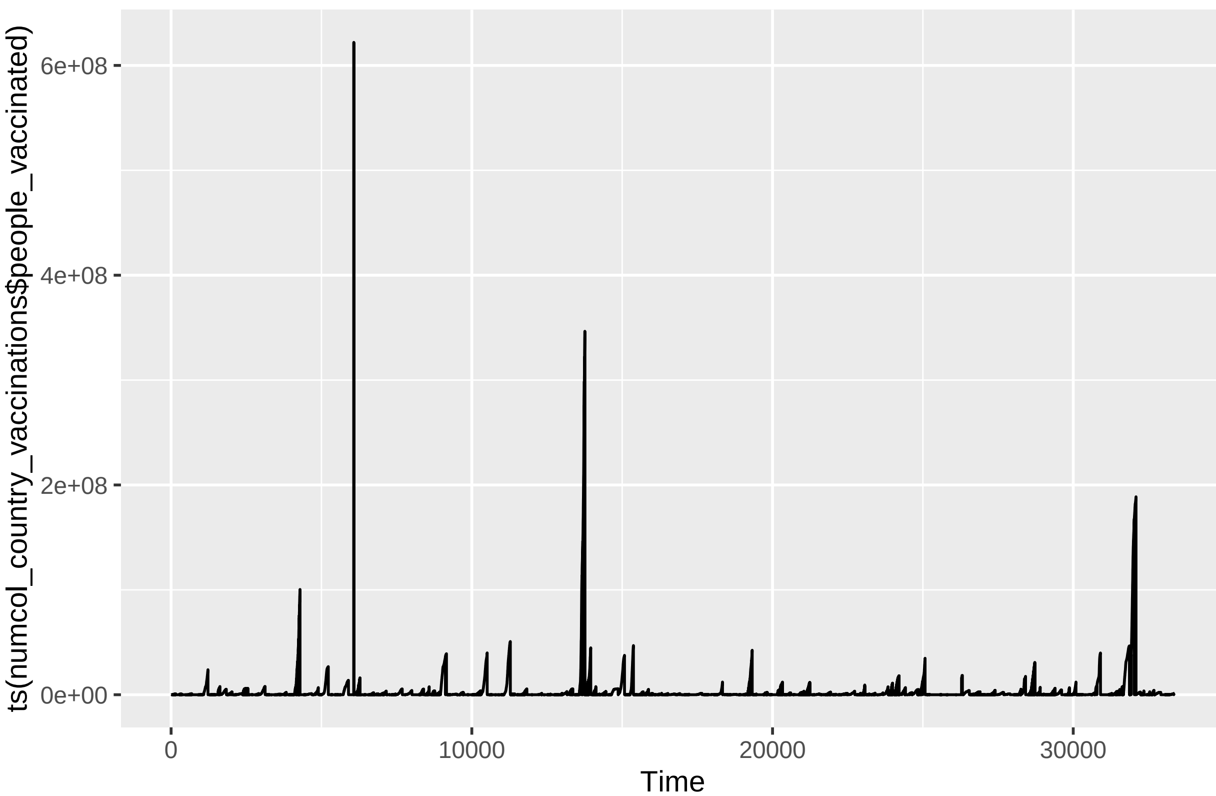
```
## v forecast  8.17.0     v expsmooth 2.3
## v fma       2.4
```

```
## -- Conflicts ------------------------------------------------- fpp2_conflicts --
## x forecast::gghistogram() masks ggpubr::gghistogram()
```

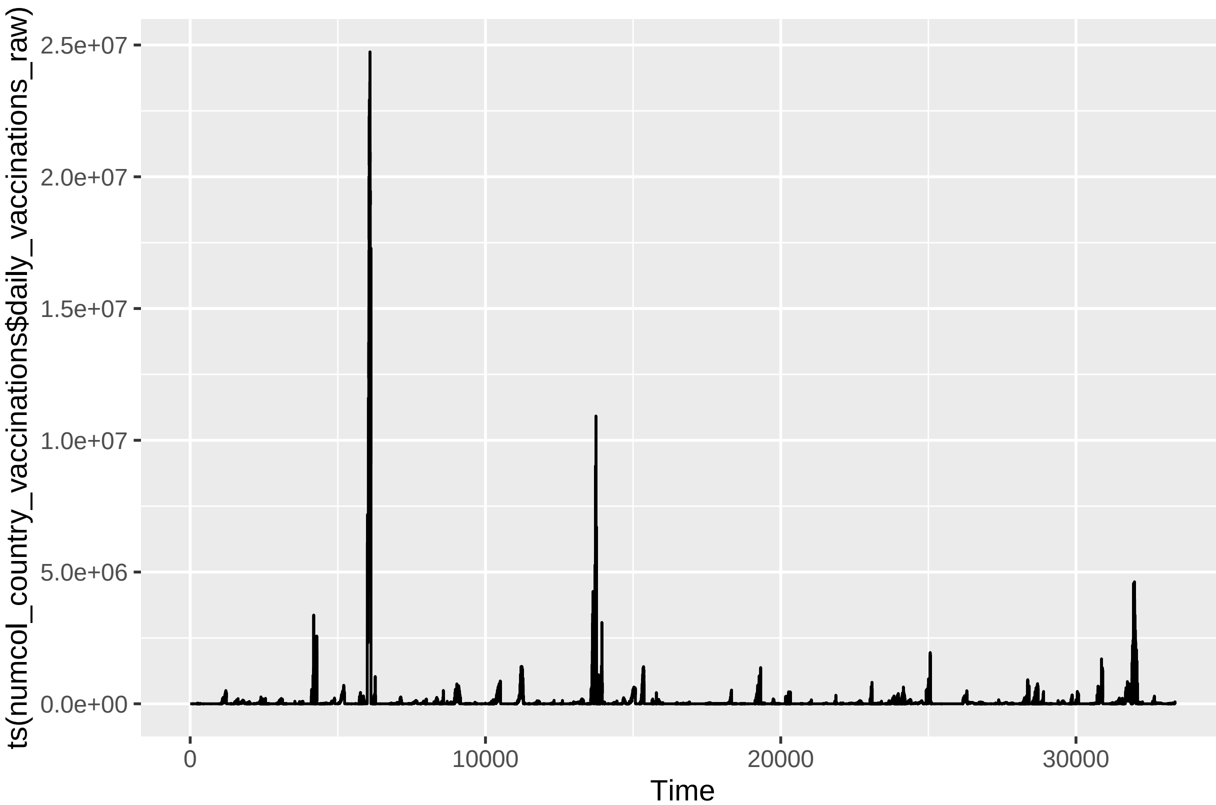
```
autoplot(ts(numcol_country_vaccinations$total_vaccinations))
```



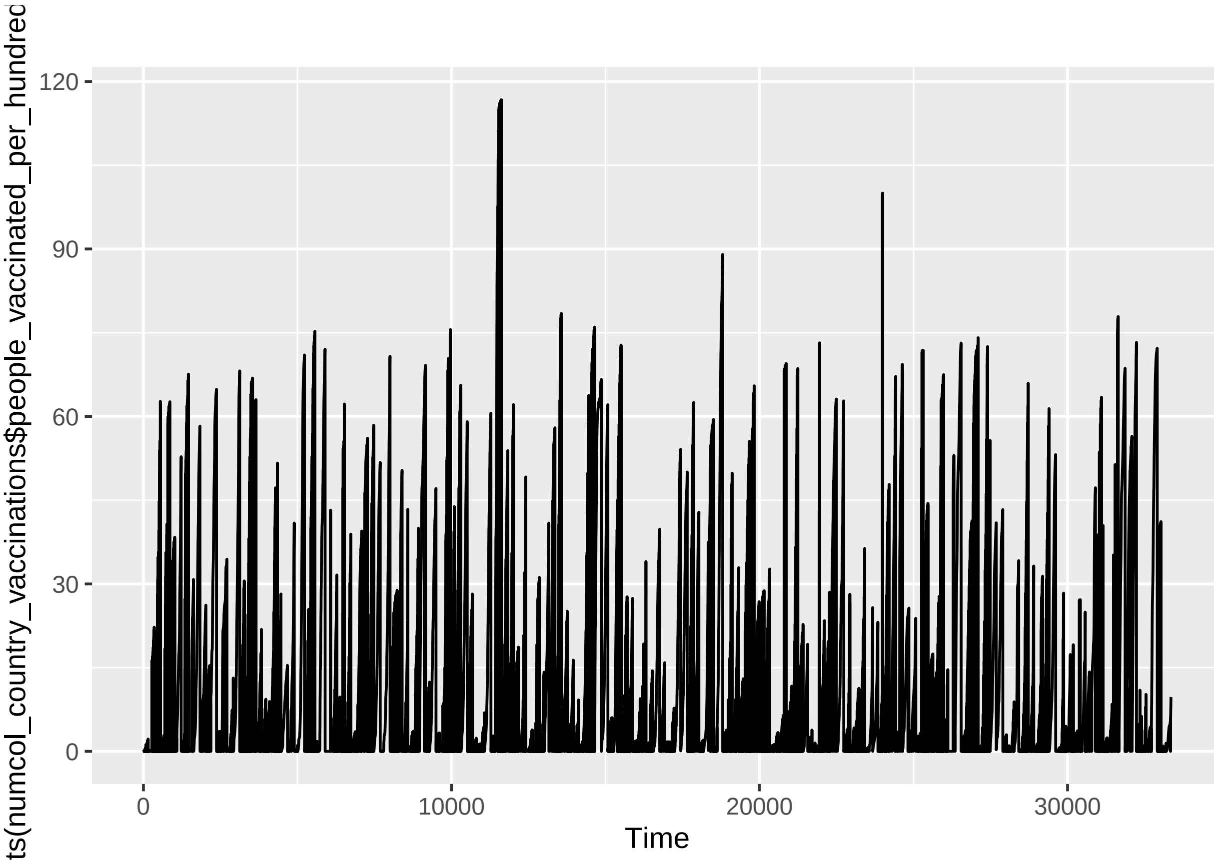

```
autoplot(ts(numcol_country_vaccinations$people_vaccinated))
```



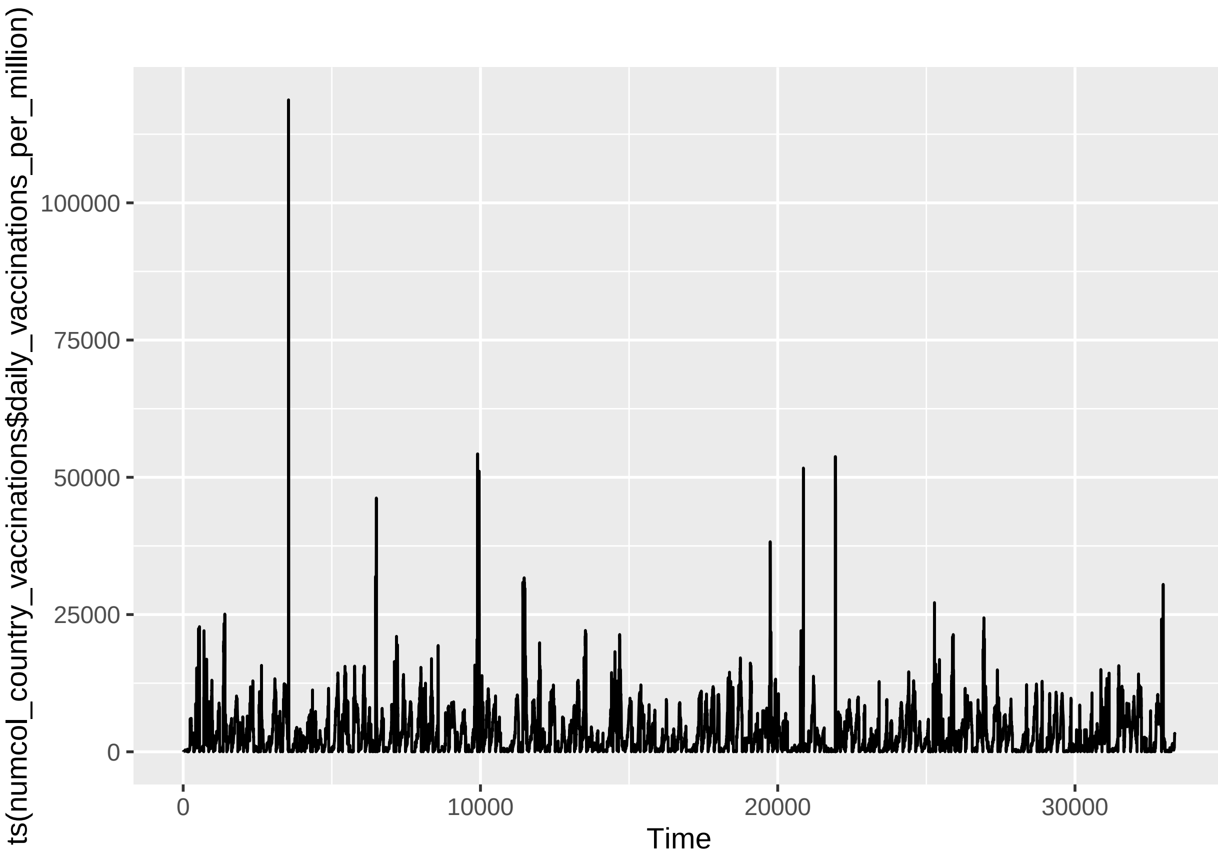

```
autoplot(ts(numcol_country_vaccinations$daily_vaccinations_raw))
```



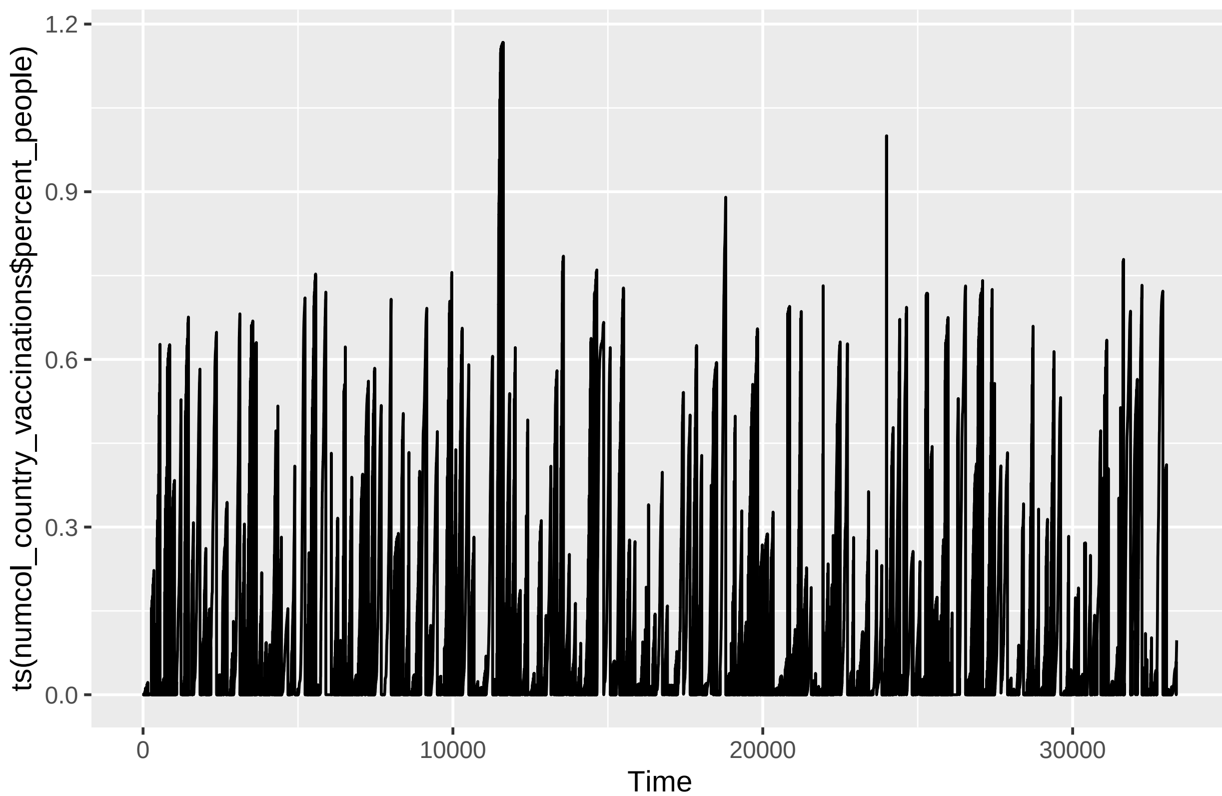

```
autoplot(ts(numcol_country_vaccinations$people_vaccinated_per_hundred))
```




```
autoplot(ts(numcol_country_vaccinations$daily_vaccinations_per_million))
```




```
autoplot(ts(numcol_country_vaccinations$percent_people))
```



The plots above suggest that there are outliers but we will not be treating them for our analysis.

This variation in values is what makes the base of our study.