EDA ON COUNTRY VACCINATIONS DATASET

PES University

PES1UG20CS622, Dept. of CSE - ADITYA SUNDAR RAJ

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

```
## Rows: 86512 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.
```

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

```
dim(country_vaccinations)
```

```
## [1] 86512     8
```

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

```
## [1] 184790
```

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

```
##   country        total_vaccinations    date         people_vaccinated
##  Mode :logical   Mode :logical      Mode :logical   Mode :logical
##  FALSE:86512     FALSE:43607        FALSE:86512     FALSE:41294
##                  TRUE :42905                        TRUE :45218
##  daily_vaccinations_raw people_vaccinated_per_hundred
##  Mode :logical          Mode :logical
##  FALSE:35362            FALSE:41294
##  TRUE :51150            TRUE :45218
##  daily_vaccinations_per_million  vaccines
##  Mode :logical                  Mode :logical
##  FALSE:86213                    FALSE:86512
##  TRUE :299
```

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

```
##                         country             total_vaccinations
##                               0                          42905
##                            date              people_vaccinated
##                               0                          45218
##         daily_vaccinations_raw  people_vaccinated_per_hundred
##                           51150                          45218
## daily_vaccinations_per_million                       vaccines
##                             299                              0
```

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

```
## [1] 86512     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

```
library(lubridate)
```

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

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

```
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_vaccinat~1 date       peopl~2 daily~3 peopl~4 daily~5 vacci~6
##   <chr>                 <dbl> <date>       <dbl>   <dbl>   <dbl>   <dbl> <chr>
## 1 Guatemala             42330 2021-03-15   42330    8113    0.23     216 Modern~
## 2 Azerbaijan          2252809 2021-05-30 1352778   44735   13.2     3715 Oxford~
## 3 Lesotho                   0 2021-06-02       0       0    0        201 Johnso~
## 4 Serbia              5851158 2021-08-29 2917843    5225   42.5     3059 Oxford~
## 5 Uzbekistan                0 2022-01-20       0       0    0        116 Modern~
```

```
## # ... 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.

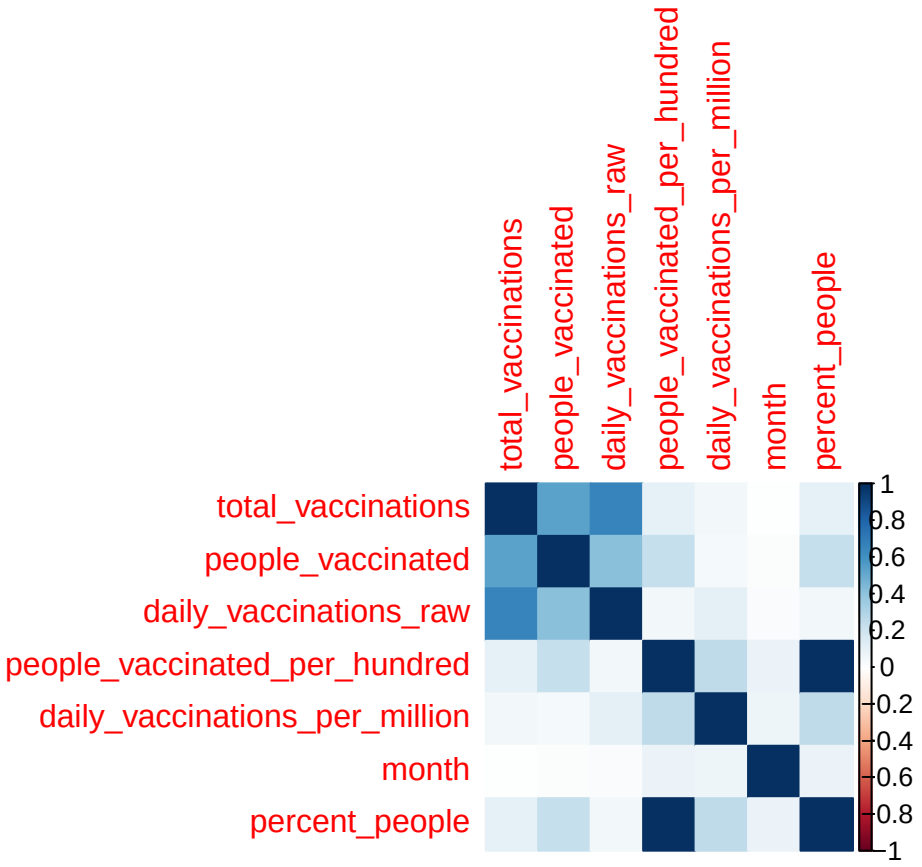
```
library(ggpubr)
```

```
## 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     1.5678 1.3329 1.0001 0.9466 0.74534 0.5600 2.839e-13
## Proportion of Variance 0.3511 0.2538 0.1429 0.1280 0.07936 0.0448 0.000e+00
## Cumulative Proportion  0.3511 0.6049 0.7478 0.8758 0.95520 1.0000 1.000e+00
```

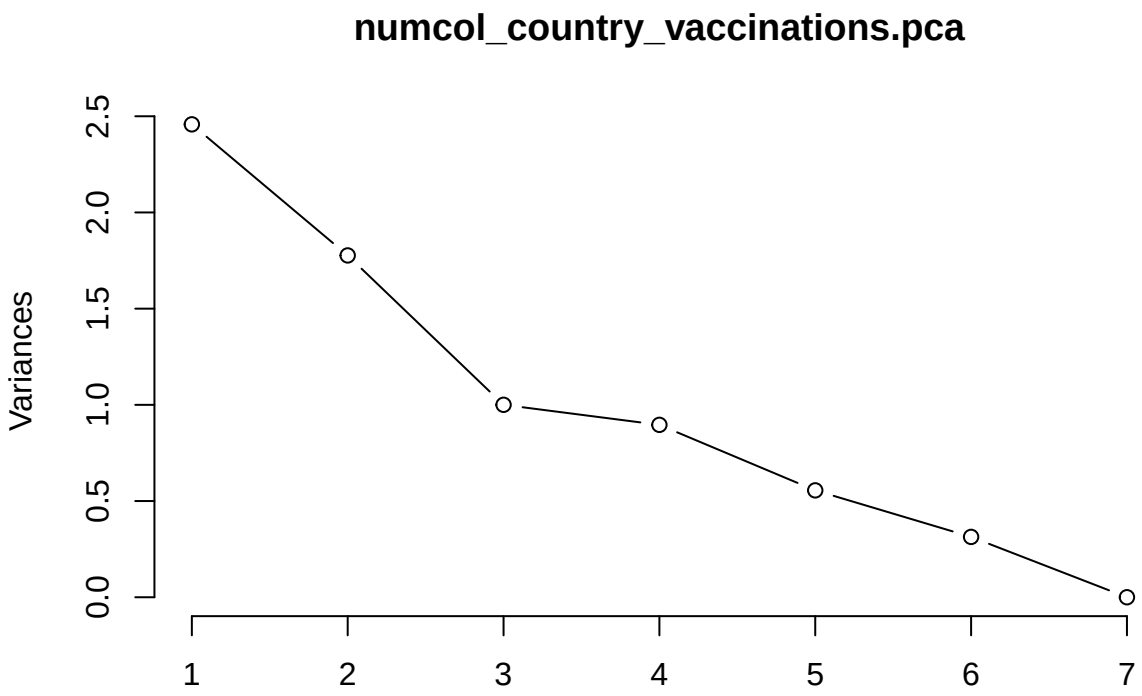
Proportion of variance for all 7 numeric principal components is low and PCA would not be the best option.

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] 1.568 1.333 1 0.947 0.745 ...
##  $ rotation: num [1:7, 1:7] 0.395 0.414 0.356 0.493 0.23 ...
##   ..- 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] 2.32e+07 8.45e+06 1.11e+05 1.95e+01 3.25e+03 ...
##   ..- attr(*, "names")= chr [1:7] "total_vaccinations" "people_vaccinated" "daily_vaccinations_raw"
##  $ scale   : Named num [1:7] 1.61e+08 4.97e+07 7.86e+05 2.88e+01 3.93e+03 ...
##   ..- attr(*, "names")= chr [1:7] "total_vaccinations" "people_vaccinated" "daily_vaccinations_raw"
##  $ x       : num [1:86512, 1:7] -1.13 -1.13 -1.13 -1.13 -1.13 ...
##   ..- 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
```

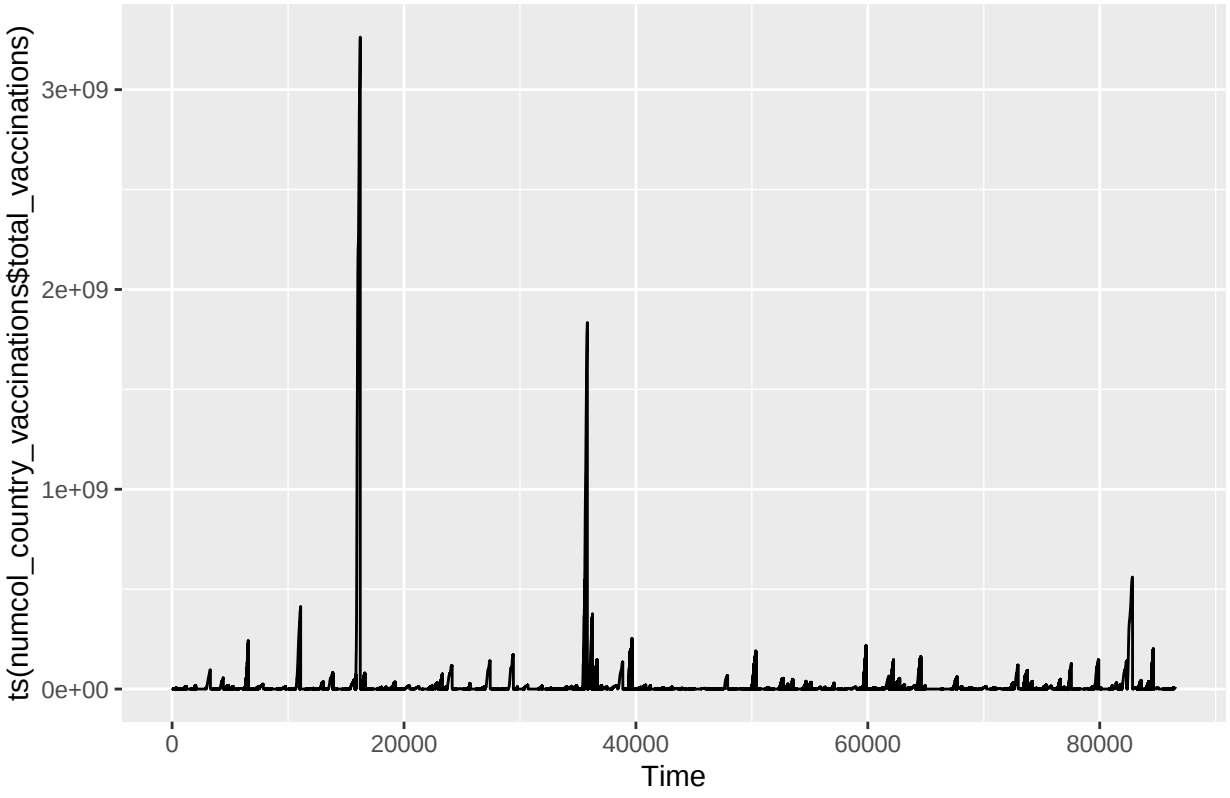
In the screeplot above, the 'arm-bend' represents a decrease in cumulative contribution.

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

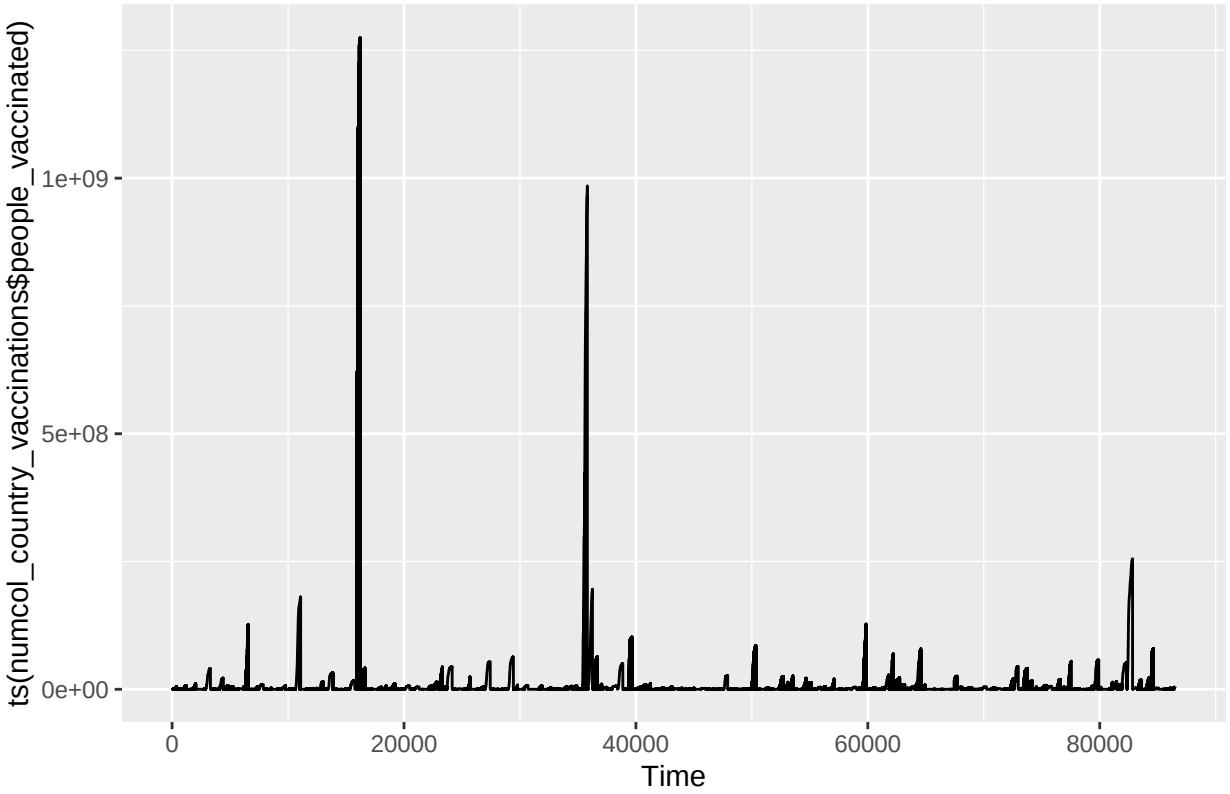
```
library(fpp2)
```

```
## 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()
```

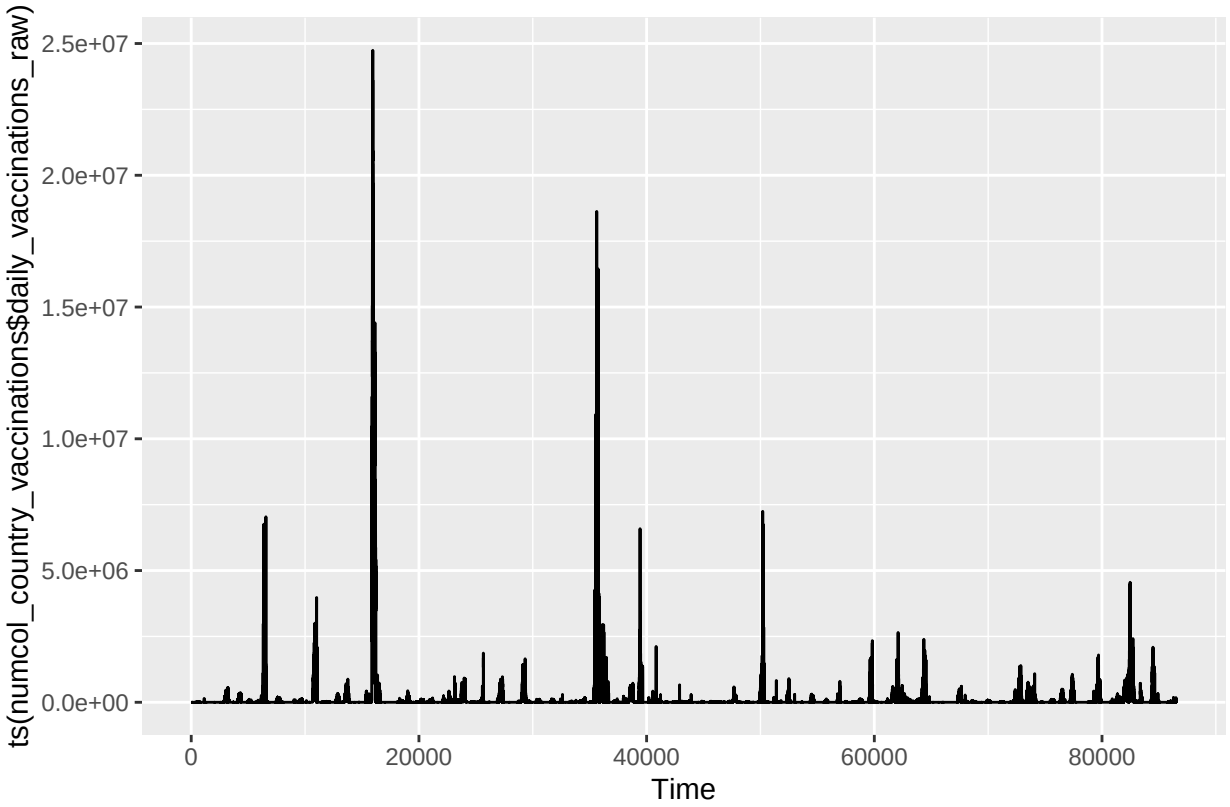
```
autoplot(ts(numcol_country_vaccinations$total_vaccinations))
```



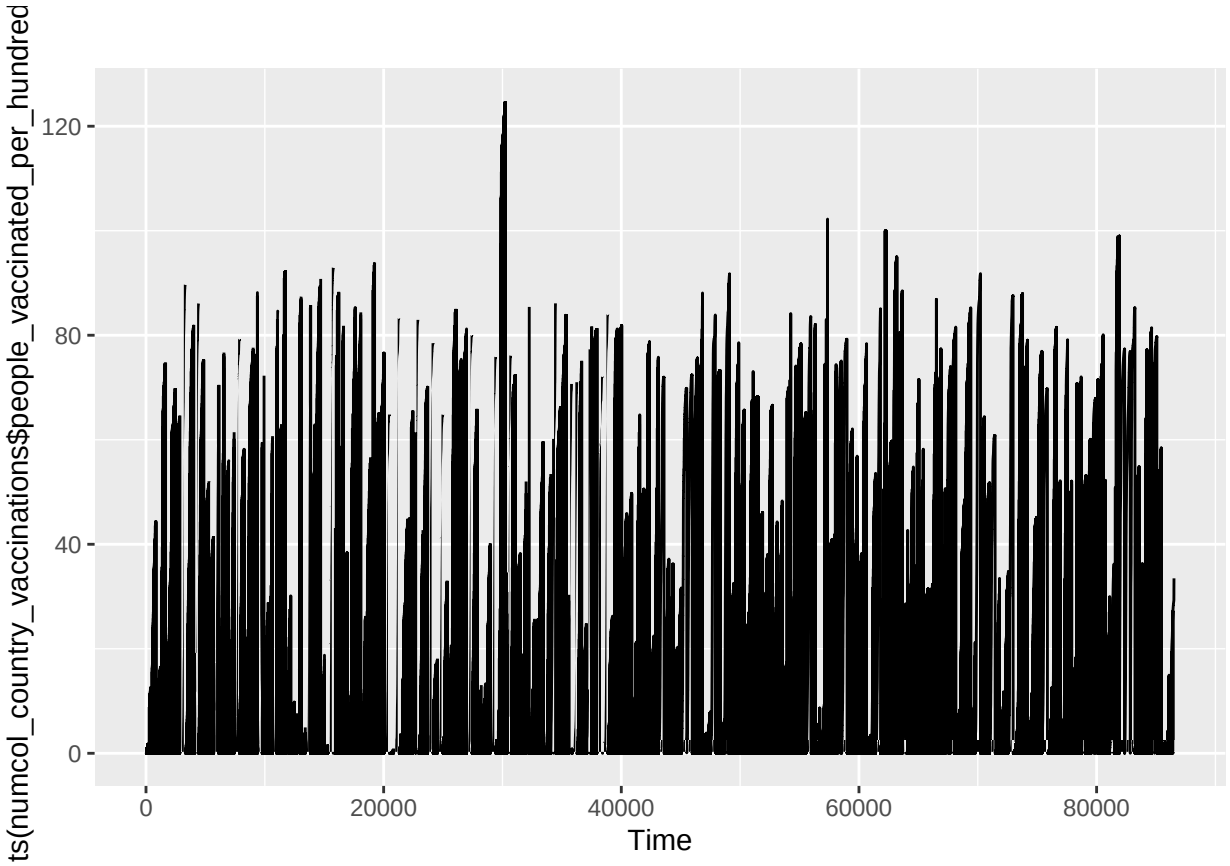

```
autoplot(ts(numcol_country_vaccinations$people_vaccinated))
```



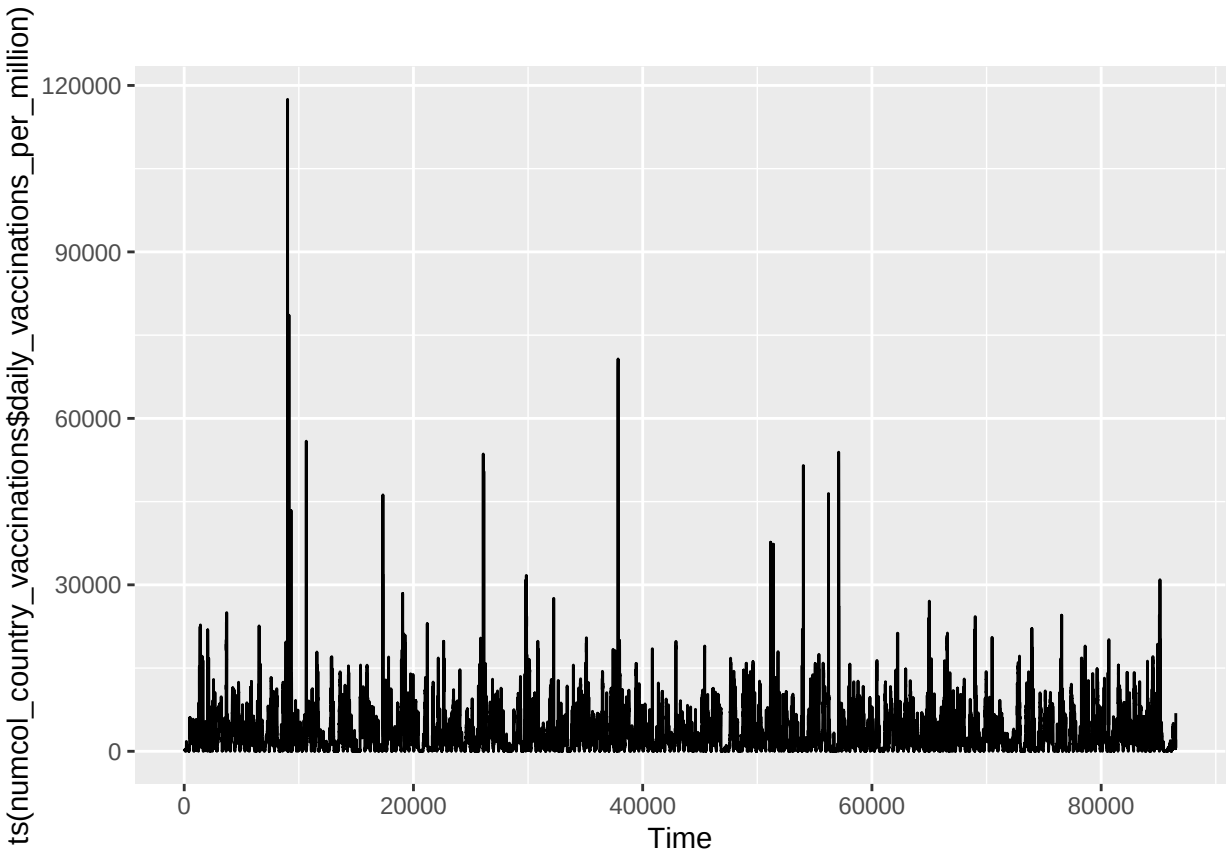

```
autoplot(ts(numcol_country_vaccinations$daily_vaccinations_raw))
```



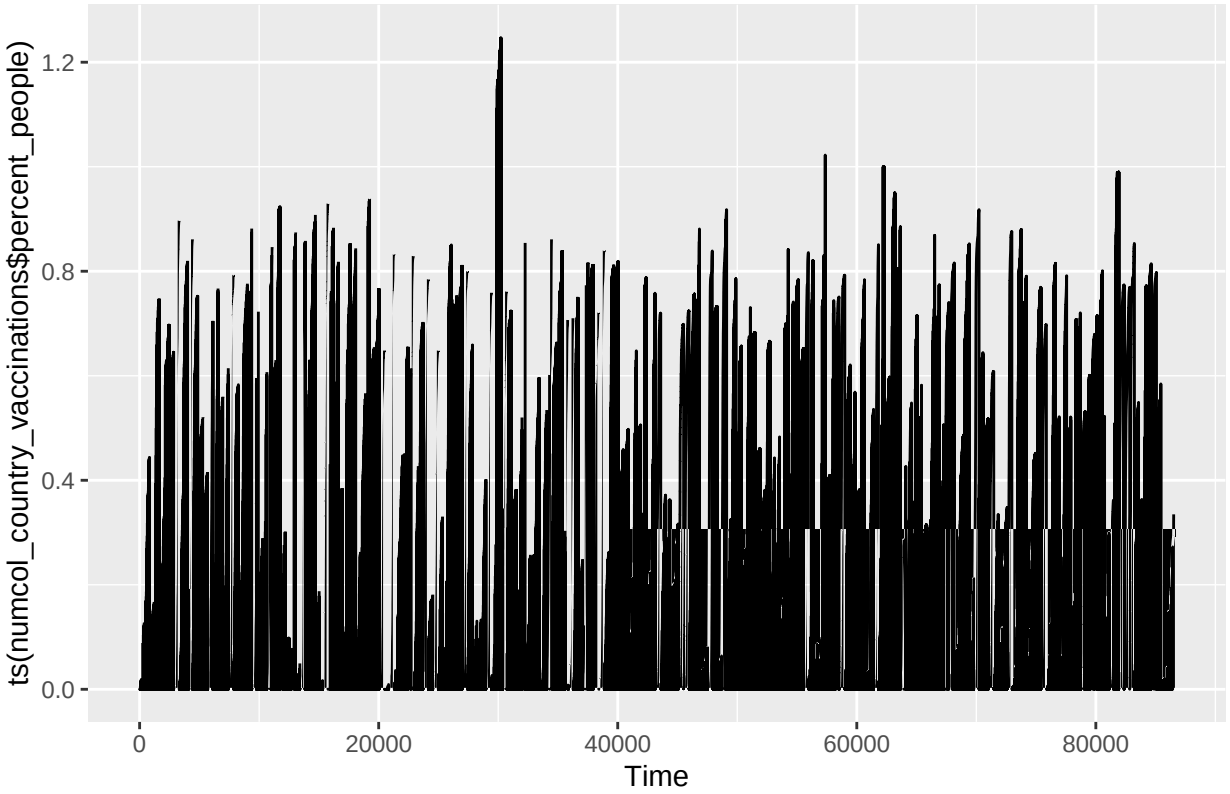

```
autoplot(ts(numcol_country_vaccinations$people_vaccinated_per_hundred))
```




```
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.