Southern\_Africa\_Countries\_COVID\_19

Bbosa Robert

29/04/2021

Table of Contents

# Install necessary packages

install.packages("ggpubr")  
install.packages("hrbrthemes")  
install.packages("directlabels")  
install.packages("pastecs")  
tinytex::install\_tinytex()

library(tidyverse)

## -- Attaching packages ------------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2 v purrr 0.3.4  
## v tibble 3.0.3 v dplyr 1.0.2  
## v tidyr 1.1.2 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.5.0

## -- Conflicts ---------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(plotly)

##   
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':  
##   
## last\_plot

## The following object is masked from 'package:stats':  
##   
## filter

## The following object is masked from 'package:graphics':  
##   
## layout

library(ggplot2)  
library(dplyr)  
library(viridis)

## Loading required package: viridisLite

library(patchwork)  
library(ggpubr)

## Warning: package 'ggpubr' was built under R version 4.0.5

library(hrbrthemes)

## Warning: package 'hrbrthemes' was built under R version 4.0.5

## NOTE: Either Arial Narrow or Roboto Condensed fonts are required to use these themes.

## Please use hrbrthemes::import\_roboto\_condensed() to install Roboto Condensed and

## if Arial Narrow is not on your system, please see https://bit.ly/arialnarrow

library(directlabels)  
library(knitr)  
library(pastecs)

## Warning: package 'pastecs' was built under R version 4.0.5

##   
## Attaching package: 'pastecs'

## The following objects are masked from 'package:dplyr':  
##   
## first, last

## The following object is masked from 'package:tidyr':  
##   
## extract

library (RColorBrewer)

# Read the file into R

corona\_global <- read.csv("C:/Users/uganda/OneDrive - BBOSA ROBERT/COVID-19/COVID-19\_Data/Data/coronavirus.csv",   
 header = TRUE, stringsAsFactors = FALSE)  
attach(corona\_global)  
summary(corona\_global)

## Province.State Country.Region Lat Long   
## Length:375030 Length:375030 Min. :-51.80 Min. :-178.12   
## Class :character Class :character 1st Qu.: 4.86 1st Qu.: -14.45   
## Mode :character Mode :character Median : 21.51 Median : 21.75   
## Mean : 20.07 Mean : 24.79   
## 3rd Qu.: 40.14 3rd Qu.: 85.24   
## Max. : 71.71 Max. : 178.06   
## NA's :2315 NA's :2315   
## date cases type   
## Length:375030 Min. :-6298082 Length:375030   
## Class :character 1st Qu.: 0 Class :character   
## Mode :character Median : 0 Mode :character   
## Mean : 639   
## 3rd Qu.: 32   
## Max. : 1123456   
##

names(corona\_global)

## [1] "Province.State" "Country.Region" "Lat" "Long"   
## [5] "date" "cases" "type"

str(corona\_global)

## 'data.frame': 375030 obs. of 7 variables:  
## $ Province.State: chr "" "" "" "" ...  
## $ Country.Region: chr "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...  
## $ Lat : num 33.9 33.9 33.9 33.9 33.9 ...  
## $ Long : num 67.7 67.7 67.7 67.7 67.7 ...  
## $ date : chr "2020-01-22" "2020-01-23" "2020-01-24" "2020-01-25" ...  
## $ cases : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ type : chr "confirmed" "confirmed" "confirmed" "confirmed" ...

# Eliminate the unnecessary columns

corona\_global <- select(corona\_global, -Province.State, -Lat, -Long)  
class(corona\_global)

## [1] "data.frame"

head(corona\_global)

## Country.Region date cases type  
## 1 Afghanistan 2020-01-22 0 confirmed  
## 2 Afghanistan 2020-01-23 0 confirmed  
## 3 Afghanistan 2020-01-24 0 confirmed  
## 4 Afghanistan 2020-01-25 0 confirmed  
## 5 Afghanistan 2020-01-26 0 confirmed  
## 6 Afghanistan 2020-01-27 0 confirmed

# Transform date from “Character format to date format”

corona\_global <- rename(corona\_global, country = Country.Region)  
myformat <- "%Y-%m-%d"  
corona\_global$date <- as.Date(corona\_global$date, myformat)  
str(corona\_global)

## 'data.frame': 375030 obs. of 4 variables:  
## $ country: chr "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...  
## $ date : Date, format: "2020-01-22" "2020-01-23" ...  
## $ cases : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ type : chr "confirmed" "confirmed" "confirmed" "confirmed" ...

# Check if the date has changed

corona\_global <- tibble::as\_tibble(corona\_global)  
class(corona\_global)

## [1] "tbl\_df" "tbl" "data.frame"

str(corona\_global)

## tibble [375,030 x 4] (S3: tbl\_df/tbl/data.frame)  
## $ country: chr [1:375030] "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...  
## $ date : Date[1:375030], format: "2020-01-22" "2020-01-23" ...  
## $ cases : int [1:375030] 0 0 0 0 0 0 0 0 0 0 ...  
## $ type : chr [1:375030] "confirmed" "confirmed" "confirmed" "confirmed" ...

# Include a column for totals

corona\_global\_total <- corona\_global %>%  
 group\_by(country, type ) %>%  
 mutate(total\_cases = cumsum(cases))  
head(corona\_global\_total)

## # A tibble: 6 x 5  
## # Groups: country, type [1]  
## country date cases type total\_cases  
## <chr> <date> <int> <chr> <int>  
## 1 Afghanistan 2020-01-22 0 confirmed 0  
## 2 Afghanistan 2020-01-23 0 confirmed 0  
## 3 Afghanistan 2020-01-24 0 confirmed 0  
## 4 Afghanistan 2020-01-25 0 confirmed 0  
## 5 Afghanistan 2020-01-26 0 confirmed 0  
## 6 Afghanistan 2020-01-27 0 confirmed 0

tail(corona\_global\_total)

## # A tibble: 6 x 5  
## # Groups: country, type [1]  
## country date cases type total\_cases  
## <chr> <date> <int> <chr> <int>  
## 1 China 2021-04-23 0 recovered 97121  
## 2 China 2021-04-24 0 recovered 97121  
## 3 China 2021-04-25 0 recovered 97121  
## 4 China 2021-04-26 0 recovered 97121  
## 5 China 2021-04-27 0 recovered 97121  
## 6 China 2021-04-28 2 recovered 97123

# Select Southern Africa countries

SA\_countries <- corona\_global\_total %>%   
 filter(country %in% c("Angola","Botswana","Eswatini","Malawi","Mozambique",  
 "Lesotho","Namibia","South Africa","Zambia","Zimbabwe"))  
   
head(SA\_countries)

## # A tibble: 6 x 5  
## # Groups: country, type [1]  
## country date cases type total\_cases  
## <chr> <date> <int> <chr> <int>  
## 1 Angola 2020-01-22 0 confirmed 0  
## 2 Angola 2020-01-23 0 confirmed 0  
## 3 Angola 2020-01-24 0 confirmed 0  
## 4 Angola 2020-01-25 0 confirmed 0  
## 5 Angola 2020-01-26 0 confirmed 0  
## 6 Angola 2020-01-27 0 confirmed 0

tail(SA\_countries)

## # A tibble: 6 x 5  
## # Groups: country, type [1]  
## country date cases type total\_cases  
## <chr> <date> <int> <chr> <int>  
## 1 Zimbabwe 2021-04-23 21 recovered 35094  
## 2 Zimbabwe 2021-04-24 7 recovered 35101  
## 3 Zimbabwe 2021-04-25 22 recovered 35123  
## 4 Zimbabwe 2021-04-26 26 recovered 35149  
## 5 Zimbabwe 2021-04-27 331 recovered 35480  
## 6 Zimbabwe 2021-04-28 37 recovered 35517

# Compute the cumulative infection rate

confirmed\_cases <- SA\_countries %>%   
 filter(type=="confirmed") %>%  
 group\_by(country,type) %>%  
 mutate(infection\_rate=ifelse(country=="Angola",  
 total\_cases\*(1000000/30809762),  
 ifelse(country=="Botswana",  
 total\_cases\*(1000000/2254125),  
 ifelse(country=="Eswatini",   
 total\_cases\*(1000000/1136192),  
 ifelse(country=="Lesotho",  
 total\_cases\*(1000000/2108132),  
 ifelse(country=="Malawi",  
 total\_cases\*(1000000/18143315),  
 ifelse(country=="Mozambique",  
 total\_cases\*(1000000/29495962),  
 ifelse(country=="Namibia",  
 total\_cases\*(1000000/2448255),  
 ifelse(country=="South Africa",  
 total\_cases\*(1000000/57779622),  
 ifelse(country=="Zambia",  
 total\_cases\*(1000000/17351822),  
 total\_cases\*(1000000/14439018)))))))))))   
head(confirmed\_cases)

## # A tibble: 6 x 6  
## # Groups: country, type [1]  
## country date cases type total\_cases infection\_rate  
## <chr> <date> <int> <chr> <int> <dbl>  
## 1 Angola 2020-01-22 0 confirmed 0 0  
## 2 Angola 2020-01-23 0 confirmed 0 0  
## 3 Angola 2020-01-24 0 confirmed 0 0  
## 4 Angola 2020-01-25 0 confirmed 0 0  
## 5 Angola 2020-01-26 0 confirmed 0 0  
## 6 Angola 2020-01-27 0 confirmed 0 0

tail(confirmed\_cases)

## # A tibble: 6 x 6  
## # Groups: country, type [1]  
## country date cases type total\_cases infection\_rate  
## <chr> <date> <int> <chr> <int> <dbl>  
## 1 Zimbabwe 2021-04-23 27 confirmed 38045 2635.  
## 2 Zimbabwe 2021-04-24 19 confirmed 38064 2636.  
## 3 Zimbabwe 2021-04-25 22 confirmed 38086 2638.  
## 4 Zimbabwe 2021-04-26 16 confirmed 38102 2639.  
## 5 Zimbabwe 2021-04-27 62 confirmed 38164 2643.  
## 6 Zimbabwe 2021-04-28 27 confirmed 38191 2645.

str(confirmed\_cases)

## tibble [4,630 x 6] (S3: grouped\_df/tbl\_df/tbl/data.frame)  
## $ country : chr [1:4630] "Angola" "Angola" "Angola" "Angola" ...  
## $ date : Date[1:4630], format: "2020-01-22" "2020-01-23" ...  
## $ cases : int [1:4630] 0 0 0 0 0 0 0 0 0 0 ...  
## $ type : chr [1:4630] "confirmed" "confirmed" "confirmed" "confirmed" ...  
## $ total\_cases : int [1:4630] 0 0 0 0 0 0 0 0 0 0 ...  
## $ infection\_rate: num [1:4630] 0 0 0 0 0 0 0 0 0 0 ...  
## - attr(\*, "groups")= tibble [10 x 3] (S3: tbl\_df/tbl/data.frame)  
## ..$ country: chr [1:10] "Angola" "Botswana" "Eswatini" "Lesotho" ...  
## ..$ type : chr [1:10] "confirmed" "confirmed" "confirmed" "confirmed" ...  
## ..$ .rows : list<int> [1:10]   
## .. ..$ : int [1:463] 1 2 3 4 5 6 7 8 9 10 ...  
## .. ..$ : int [1:463] 464 465 466 467 468 469 470 471 472 473 ...  
## .. ..$ : int [1:463] 927 928 929 930 931 932 933 934 935 936 ...  
## .. ..$ : int [1:463] 1390 1391 1392 1393 1394 1395 1396 1397 1398 1399 ...  
## .. ..$ : int [1:463] 1853 1854 1855 1856 1857 1858 1859 1860 1861 1862 ...  
## .. ..$ : int [1:463] 2316 2317 2318 2319 2320 2321 2322 2323 2324 2325 ...  
## .. ..$ : int [1:463] 2779 2780 2781 2782 2783 2784 2785 2786 2787 2788 ...  
## .. ..$ : int [1:463] 3242 3243 3244 3245 3246 3247 3248 3249 3250 3251 ...  
## .. ..$ : int [1:463] 3705 3706 3707 3708 3709 3710 3711 3712 3713 3714 ...  
## .. ..$ : int [1:463] 4168 4169 4170 4171 4172 4173 4174 4175 4176 4177 ...  
## .. ..@ ptype: int(0)   
## ..- attr(\*, ".drop")= logi TRUE

summary(confirmed\_cases$infection\_rate)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000 8.801 517.427 3120.440 3195.965 27318.455

stat.desc(confirmed\_cases)

## country date cases type total\_cases infection\_rate  
## nbr.val NA 4.630000e+03 4.630000e+03 NA 4.630000e+03 4.630000e+03  
## nbr.null NA 0.000000e+00 1.478000e+03 NA 6.300000e+02 6.300000e+02  
## nbr.na NA 0.000000e+00 0.000000e+00 NA 0.000000e+00 0.000000e+00  
## min NA 1.828300e+04 -6.000000e+00 NA 0.000000e+00 0.000000e+00  
## max NA 1.874500e+04 2.198000e+04 NA 1.578450e+06 2.731845e+04  
## range NA 4.620000e+02 2.198600e+04 NA 1.578450e+06 2.731845e+04  
## sum NA 8.571982e+07 1.962108e+06 NA 3.427470e+08 1.444763e+07  
## median NA 1.851400e+04 2.050000e+01 NA 5.577500e+03 5.174266e+02  
## mean NA 1.851400e+04 4.237814e+02 NA 7.402743e+04 3.120440e+03  
## SE.mean NA 1.964472e+00 2.542823e+01 NA 3.845016e+03 8.374313e+01  
## CI.mean NA 3.851301e+00 4.985146e+01 NA 7.538063e+03 1.641765e+02  
## var NA 1.786786e+04 2.993735e+06 NA 6.845060e+10 3.246978e+07  
## std.dev NA 1.336707e+02 1.730241e+03 NA 2.616307e+05 5.698226e+03  
## coef.var NA 7.219980e-03 4.082863e+00 NA 3.534239e+00 1.826097e+00

# Computer the Fatality rate

death\_cases <- SA\_countries %>%   
 filter(type %in% c("death", "confirmed")) %>%   
 select(-total\_cases) %>%   
 spread(type, cases) %>%   
 mutate(fatality\_rate=ifelse(country=="Angola",  
 round((cumsum(death)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="Botswana",  
 round((cumsum(death)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="Eswatini",  
 round((cumsum(death)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="Lesotho",  
 round((cumsum(death)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="Malawi",  
 round((cumsum(death)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="Mozambique",  
 round((cumsum(death)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="Namibia",  
 round((cumsum(death)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="South Africa",  
 round((cumsum(death)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="Zambia",  
 round((cumsum(death)/cumsum(confirmed))\*100,digits=2),  
 round((cumsum(death)/cumsum(confirmed))\*100,  
 digits=2)))))))))))  
  
head(death\_cases)

## # A tibble: 6 x 5  
## # Groups: country [1]  
## country date confirmed death fatality\_rate  
## <chr> <date> <int> <int> <dbl>  
## 1 Angola 2020-01-22 0 0 NaN  
## 2 Angola 2020-01-23 0 0 NaN  
## 3 Angola 2020-01-24 0 0 NaN  
## 4 Angola 2020-01-25 0 0 NaN  
## 5 Angola 2020-01-26 0 0 NaN  
## 6 Angola 2020-01-27 0 0 NaN

tail(death\_cases)

## # A tibble: 6 x 5  
## # Groups: country [1]  
## country date confirmed death fatality\_rate  
## <chr> <date> <int> <int> <dbl>  
## 1 Zimbabwe 2021-04-23 27 1 4.09  
## 2 Zimbabwe 2021-04-24 19 0 4.09  
## 3 Zimbabwe 2021-04-25 22 1 4.09  
## 4 Zimbabwe 2021-04-26 16 3 4.09  
## 5 Zimbabwe 2021-04-27 62 5 4.1   
## 6 Zimbabwe 2021-04-28 27 0 4.1

str(death\_cases)

## tibble [4,630 x 5] (S3: grouped\_df/tbl\_df/tbl/data.frame)  
## $ country : chr [1:4630] "Angola" "Angola" "Angola" "Angola" ...  
## $ date : Date[1:4630], format: "2020-01-22" "2020-01-23" ...  
## $ confirmed : int [1:4630] 0 0 0 0 0 0 0 0 0 0 ...  
## $ death : int [1:4630] 0 0 0 0 0 0 0 0 0 0 ...  
## $ fatality\_rate: num [1:4630] NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN ...  
## - attr(\*, "groups")= tibble [10 x 2] (S3: tbl\_df/tbl/data.frame)  
## ..$ country: chr [1:10] "Angola" "Botswana" "Eswatini" "Lesotho" ...  
## ..$ .rows : list<int> [1:10]   
## .. ..$ : int [1:463] 1 2 3 4 5 6 7 8 9 10 ...  
## .. ..$ : int [1:463] 464 465 466 467 468 469 470 471 472 473 ...  
## .. ..$ : int [1:463] 927 928 929 930 931 932 933 934 935 936 ...  
## .. ..$ : int [1:463] 1390 1391 1392 1393 1394 1395 1396 1397 1398 1399 ...  
## .. ..$ : int [1:463] 1853 1854 1855 1856 1857 1858 1859 1860 1861 1862 ...  
## .. ..$ : int [1:463] 2316 2317 2318 2319 2320 2321 2322 2323 2324 2325 ...  
## .. ..$ : int [1:463] 2779 2780 2781 2782 2783 2784 2785 2786 2787 2788 ...  
## .. ..$ : int [1:463] 3242 3243 3244 3245 3246 3247 3248 3249 3250 3251 ...  
## .. ..$ : int [1:463] 3705 3706 3707 3708 3709 3710 3711 3712 3713 3714 ...  
## .. ..$ : int [1:463] 4168 4169 4170 4171 4172 4173 4174 4175 4176 4177 ...  
## .. ..@ ptype: int(0)   
## ..- attr(\*, ".drop")= logi TRUE

stat.desc(death\_cases)

## country date confirmed death fatality\_rate  
## nbr.val NA 4.630000e+03 4.630000e+03 4630.0000000 4.000000e+03  
## nbr.null NA 0.000000e+00 1.478000e+03 2723.0000000 3.270000e+02  
## nbr.na NA 0.000000e+00 0.000000e+00 0.0000000 6.300000e+02  
## min NA 1.828300e+04 -6.000000e+00 -34.0000000 0.000000e+00  
## max NA 1.874500e+04 2.198000e+04 844.0000000 3.333000e+01  
## range NA 4.620000e+02 2.198600e+04 878.0000000 3.333000e+01  
## sum NA 8.571982e+07 1.962108e+06 61978.0000000 9.391250e+03  
## median NA 1.851400e+04 2.050000e+01 0.0000000 2.000000e+00  
## mean NA 1.851400e+04 4.237814e+02 13.3861771 2.347812e+00  
## SE.mean NA 1.964472e+00 2.542823e+01 0.8584884 4.300469e-02  
## CI.mean NA 3.851301e+00 4.985146e+01 1.6830463 8.431316e-02  
## var NA 1.786786e+04 2.993735e+06 3412.3204829 7.397613e+00  
## std.dev NA 1.336707e+02 1.730241e+03 58.4150707 2.719855e+00  
## coef.var NA 7.219980e-03 4.082863e+00 4.3638352 1.158464e+00

# Compute the Recovery rate

recovered\_cases <- SA\_countries %>%   
 filter(type %in% c("recovered","confirmed")) %>%   
 select(-total\_cases) %>%   
 spread(type, cases) %>%  
 mutate(recovery\_rate=ifelse(country=="Angola",  
 round((cumsum(recovered)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="Botswana",  
 round((cumsum(recovered)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="Eswatini",  
 round((cumsum(recovered)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="Lesotho",  
 round((cumsum(recovered)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="Malawi",  
 round((cumsum(recovered)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="Mozambique",  
 round((cumsum(recovered)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="Namibia",  
 round((cumsum(recovered)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="South Africa",  
 round((cumsum(recovered)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="Zambia",  
 round((cumsum(recovered)/cumsum(confirmed))\*100,digits=2),  
 round((cumsum(recovered)/cumsum(confirmed))\*100,  
 digits=2)))))))))))  
  
names(recovered\_cases)

## [1] "country" "date" "confirmed" "recovered"   
## [5] "recovery\_rate"

head(recovered\_cases)

## # A tibble: 6 x 5  
## # Groups: country [1]  
## country date confirmed recovered recovery\_rate  
## <chr> <date> <int> <int> <dbl>  
## 1 Angola 2020-01-22 0 0 NaN  
## 2 Angola 2020-01-23 0 0 NaN  
## 3 Angola 2020-01-24 0 0 NaN  
## 4 Angola 2020-01-25 0 0 NaN  
## 5 Angola 2020-01-26 0 0 NaN  
## 6 Angola 2020-01-27 0 0 NaN

tail(recovered\_cases)

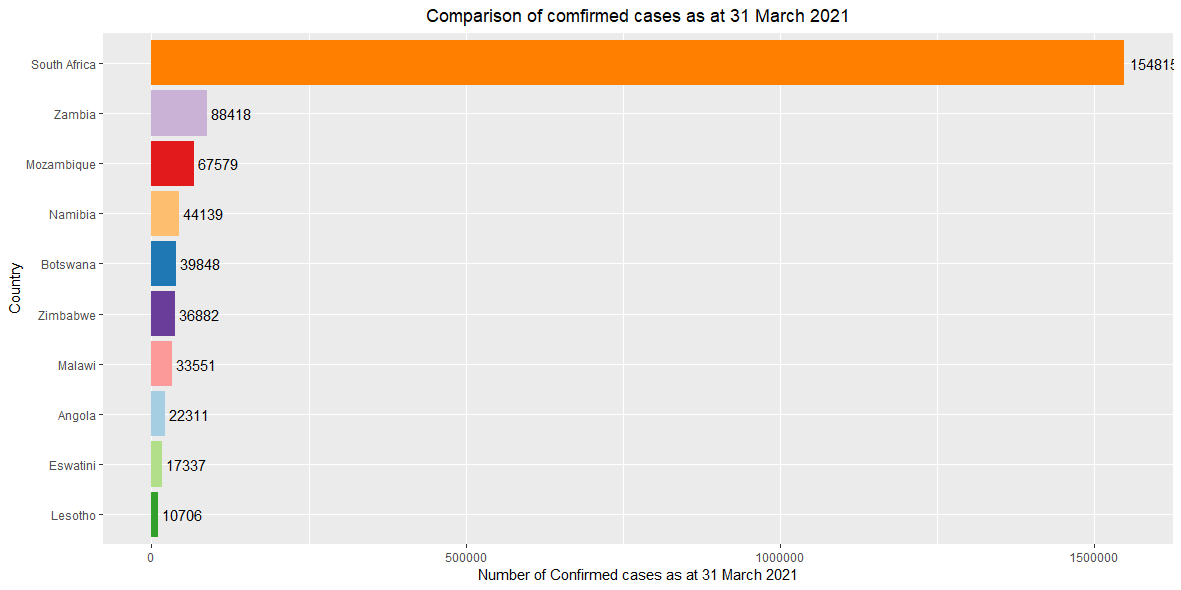
## # A tibble: 6 x 5  
## # Groups: country [1]  
## country date confirmed recovered recovery\_rate  
## <chr> <date> <int> <int> <dbl>  
## 1 Zimbabwe 2021-04-23 27 21 92.2  
## 2 Zimbabwe 2021-04-24 19 7 92.2  
## 3 Zimbabwe 2021-04-25 22 22 92.2  
## 4 Zimbabwe 2021-04-26 16 26 92.2  
## 5 Zimbabwe 2021-04-27 62 331 93.0  
## 6 Zimbabwe 2021-04-28 27 37 93

stat.desc(recovered\_cases)

## country date confirmed recovered recovery\_rate  
## nbr.val NA 4.630000e+03 4.630000e+03 4.630000e+03 4.000000e+03  
## nbr.null NA 0.000000e+00 1.478000e+03 1.909000e+03 1.780000e+02  
## nbr.na NA 0.000000e+00 0.000000e+00 0.000000e+00 6.300000e+02  
## min NA 1.828300e+04 -6.000000e+00 -3.070000e+02 0.000000e+00  
## max NA 1.874500e+04 2.198000e+04 4.585800e+04 9.814000e+01  
## range NA 4.620000e+02 2.198600e+04 4.616500e+04 9.814000e+01  
## sum NA 8.571982e+07 1.962108e+06 1.863627e+06 2.402409e+05  
## median NA 1.851400e+04 2.050000e+01 7.000000e+00 6.446500e+01  
## mean NA 1.851400e+04 4.237814e+02 4.025112e+02 6.006023e+01  
## SE.mean NA 1.964472e+00 2.542823e+01 2.649911e+01 4.791405e-01  
## CI.mean NA 3.851301e+00 4.985146e+01 5.195089e+01 9.393825e-01  
## var NA 1.786786e+04 2.993735e+06 3.251200e+06 9.183026e+02  
## std.dev NA 1.336707e+02 1.730241e+03 1.803108e+03 3.030351e+01  
## coef.var NA 7.219980e-03 4.082863e+00 4.479647e+00 5.045520e-01

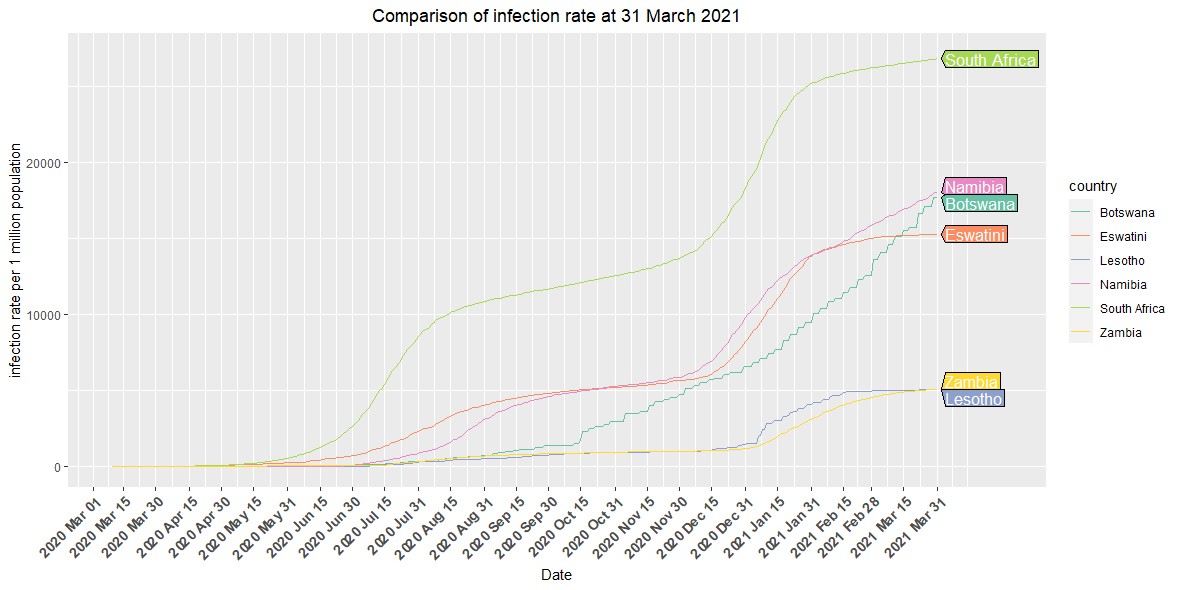
# Plot a bar chart of comfirmed cases as at 31 March 2021

mypath <- file.path("C:","Users","uganda", "OneDrive - BBOSA ROBERT","COVID-19","COVID-19\_Data","Output", "confirmed\_ctg1.png")  
png(file= mypath, width =30, height = 15,units = "cm",pointsize =12, res =100)  
  
SA\_countries %>%   
 filter(country %in% c("Eswatini","Angola","Botswana","Malawi","South Africa","Mozambique","Zambia", "Zimbabwe", "Namibia","Lesotho"),type=="confirmed", date=="2021-03-31" ) %>%   
 ggplot( aes(country, total\_cases, fill=country)) +  
 geom\_bar(aes(reorder(country, total\_cases),total\_cases),  
 stat= "identity", show.legend = FALSE) +  
 geom\_text(aes(label= total\_cases), vjust=0.5, hjust=-0.1, colour="black") +  
 scale\_fill\_brewer(palette = "Paired") +  
 coord\_flip() +  
 ggtitle("Comparison of comfirmed cases as at 31 March 2021") +  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 xlab("Country")+  
 ylab("Number of Confirmed cases as at 31 March 2021")  
  
 dev.off()



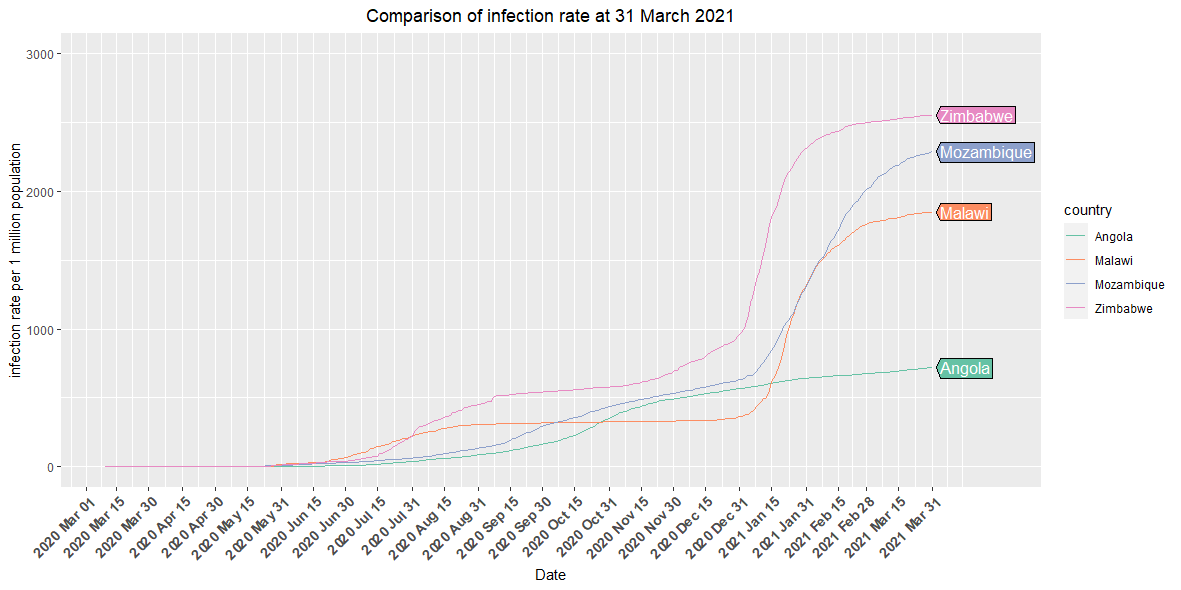
# Plot a line graph as at 31 March 2021 for Eswatini ,Zambia ,South Africa , Botswana ,Lesotho,Namibia

mypath <- file.path("C:","Users","uganda", "OneDrive - BBOSA ROBERT","COVID-19","COVID-19\_Data","Output", "confirmed\_ctg2.png")  
png(file= mypath, width =30, height = 15,units = "cm",pointsize =12, res =100)  
  
confirmed\_cases %>%  
 filter(country %in% c("Eswatini","Zambia","South Africa","Botswana","Lesotho",  
 "Namibia"),date<"2021-04-01") %>%  
 ggplot( aes(x=date, y=infection\_rate, group=country, color=country)) +  
 geom\_line() +  
 geom\_dl(aes(label = country),   
 method = list(dl.trans(x = x + 0.1), "last.polygons")) +  
 scale\_color\_brewer(palette = "Set2") +  
 ggtitle("Comparison of infection rate at 31 March 2021") +  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 ylab("infection rate per 1 million population")+  
 xlab("Date")+  
scale\_y\_continuous(limits = c(0,ymax= max(confirmed\_cases$infection\_rate))) +  
 scale\_x\_date(limits = as.Date(c("2020-03-10","2021-04-30")),  
 date\_labels = ("%Y %b %d"),  
 breaks = as.Date(c("2020-03-01","2020-03-15","2020-03-30",  
 "2020-04-15","2020-04-30","2020-05-15",  
 "2020-05-31","2020-06-15","2020-06-30", "2020-07-15","2020-07-31","2020-08-15", "2020-08-31","2020-09-15","2020-09-30",  
 "2020-10-15","2020-10-31","2020-11-15",  
 "2020-11-30","2020-12-15","2020-12-31",  
 "2021-01-15","2021-01-31","2021-02-15",  
 "2021-02-28","2021-03-15","2021-03-31"))) +  
 theme(axis.text.x = element\_text(angle = 45, vjust = 1, hjust = 1,   
 size = 10, face = "bold"))  
dev.off()



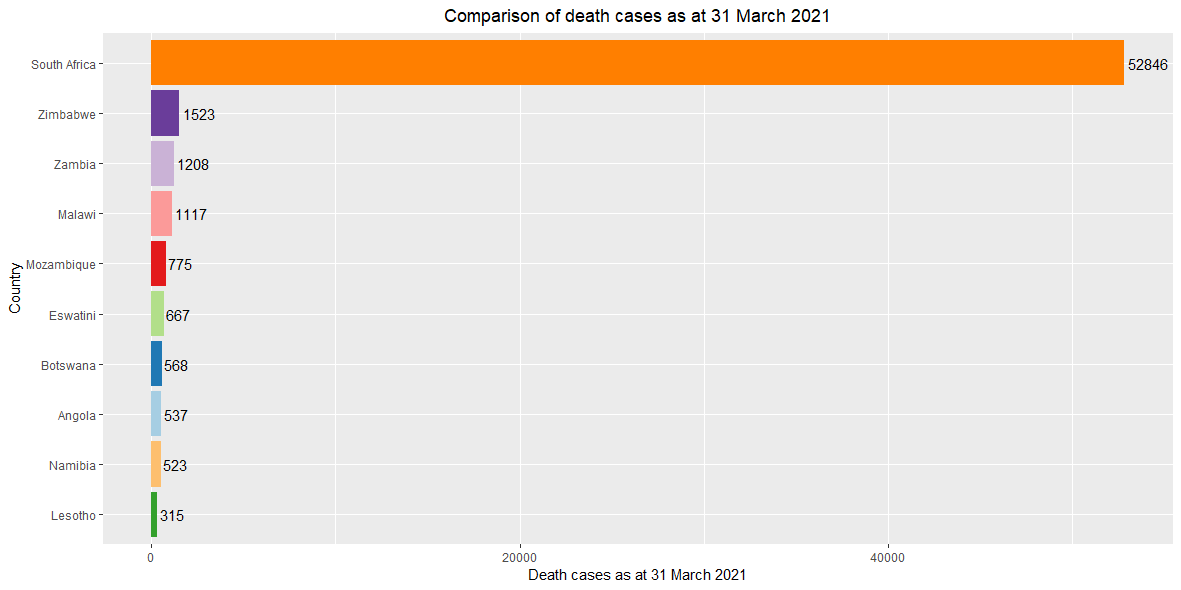
# Plot a line graph for as at 31 March for other countries

mypath <- file.path("C:","Users","uganda", "OneDrive - BBOSA ROBERT","COVID-19","COVID-19\_Data","Output", "confirmed\_ctg3.png")  
png(file= mypath, width =30, height = 15,units = "cm",pointsize =12, res =100)  
  
confirmed\_cases %>%  
 filter(country %in% c("Angola","Malawi","Zimbabwe","Mozambique"), date<"2021-04-01")%>%  
 ggplot( aes(x=date, y=infection\_rate, group=country, color=country)) +  
 geom\_line() +   
 geom\_dl(aes(label = country),   
 method = list(dl.trans(x = x + 0.1), "last.polygons")) +  
 scale\_color\_brewer(palette = "Set2") +  
 ggtitle("Comparison of infection rate at 31 March 2021") +  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 ylab("infection rate per 1 million population")+  
 xlab("Date")+  
 scale\_y\_continuous(limits = c(0,3000)) +  
 scale\_x\_date(limits = as.Date(c("2020-03-10","2021-04-30")),  
 date\_labels = ("%Y %b %d"),  
 breaks = as.Date(c("2020-03-01","2020-03-15","2020-03-30",  
 "2020-04-15","2020-04-30","2020-05-15",  
 "2020-05-31","2020-06-15","2020-06-30", "2020-07-15","2020-07-31","2020-08-15", "2020-08-31","2020-09-15","2020-09-30",  
 "2020-10-15","2020-10-31","2020-11-15",  
 "2020-11-30","2020-12-15","2020-12-31",  
 "2021-01-15","2021-01-31","2021-02-15",  
 "2021-02-28","2021-03-15","2021-03-31"))) +  
 theme(axis.text.x = element\_text(angle = 45, vjust = 1, hjust = 1,   
 size = 10, face = "bold"))  
   
dev.off()



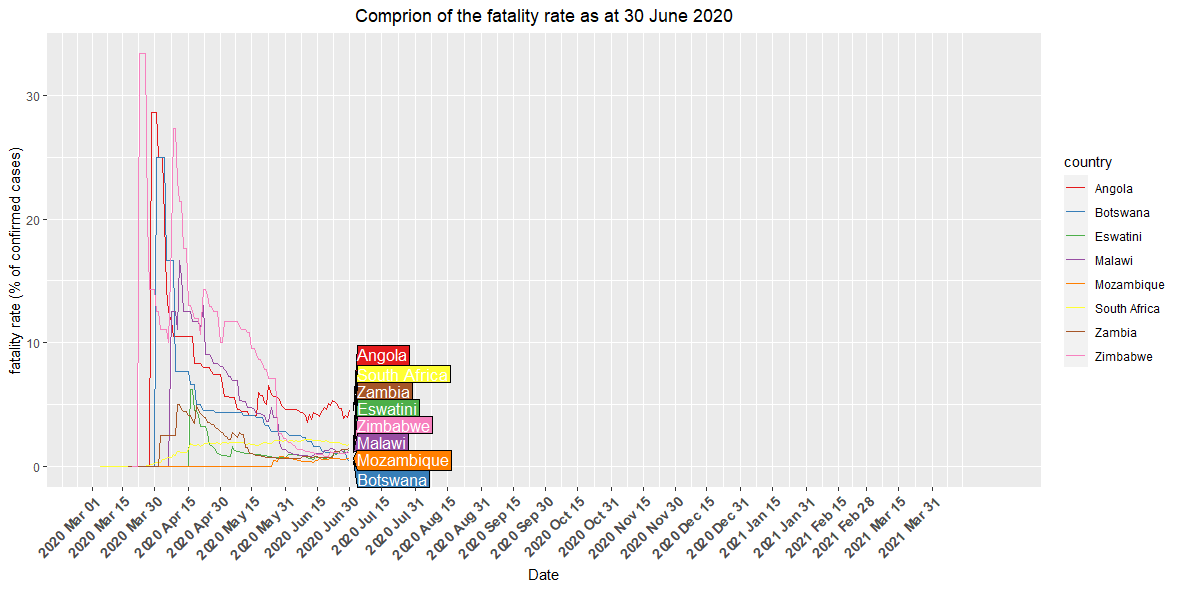
# Plot a bar graph for the deaths

mypath <- file.path("C:","Users","uganda", "OneDrive - BBOSA ROBERT","COVID-19","COVID-19\_Data","Output", "death\_ctg1.png")  
png(file= mypath, width =30, height = 15,units = "cm",pointsize =12, res =100)  
  
SA\_countries %>%   
 filter(country %in% c("Eswatini","Angola","Botswana","Malawi","South Africa","Mozambique","Zambia", "Zimbabwe", "Namibia","Lesotho"),type=="death", date=="2021-03-31" ) %>%   
 ggplot( aes(country, total\_cases, fill=country)) +  
 geom\_bar(aes(reorder(country, total\_cases),total\_cases),  
 stat= "identity", show.legend = FALSE) +  
 geom\_text(aes(label= total\_cases), vjust=0.5, hjust=-0.1,colour="black") +  
 scale\_fill\_brewer(palette = "Paired") +  
 coord\_flip() +  
 ggtitle("Comparison of death cases as at 31 March 2021") +  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 xlab("Country")+  
 ylab("Death cases as at 31 March 2021")  
  
dev.off()



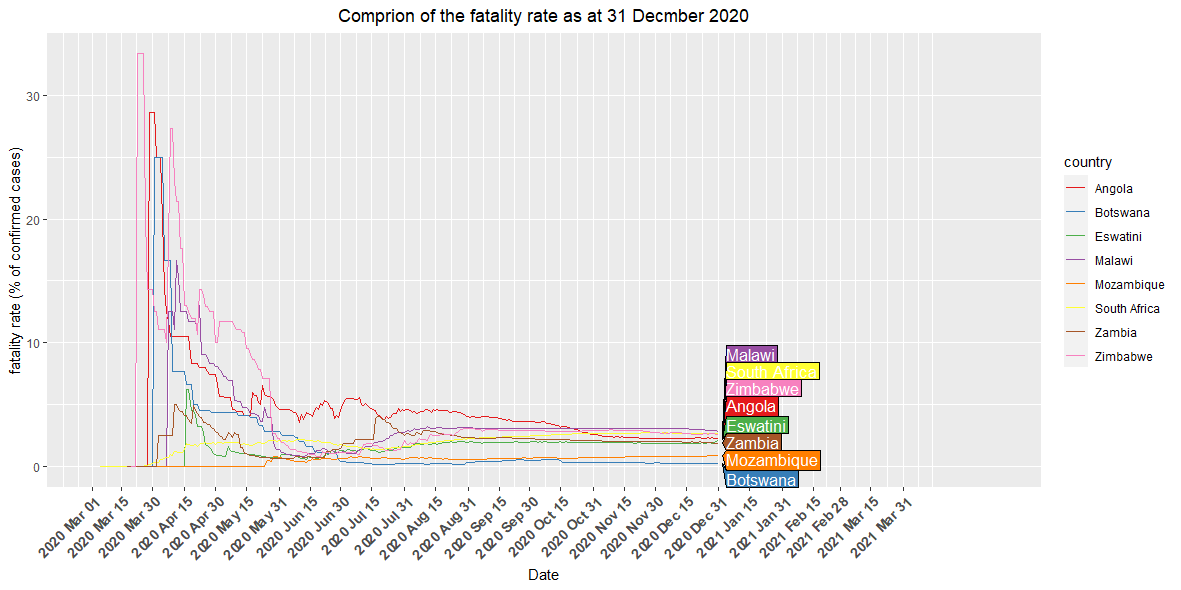
# Plot line graph for fatality rate as at 30 June 2020

mypath <- file.path("C:","Users","uganda", "OneDrive - BBOSA ROBERT","COVID-19","COVID-19\_Data","Output", "death\_ctg2.png")  
png(file= mypath, width =30, height = 15,units = "cm",pointsize =12, res =100)  
  
death\_cases %>%  
 filter(country %in% c("Angola","Botswana","Malawi","Zimbabwe","Mozambique","Zambia","Eswatini", "South Africa"),date<"2020-07-01") %>%  
 ggplot( aes(x=date, y=fatality\_rate, group=country, color=country)) +  
 geom\_line() +  
 geom\_dl(aes(label = country),  
 method = list(dl.trans(x = x + 0.1), "last.polygons")) +  
 scale\_color\_brewer(palette = "Set1") +  
 ggtitle("Comprion of the fatality rate as at 30 June 2020") +  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 ylab("fatality rate (% of confirmed cases)") +  
 xlab("Date") +  
 scale\_y\_continuous(limits = c(ymin= 0,   
 ymax= max(death\_cases$fatality\_rate)))+  
 scale\_x\_date(limits = as.Date(c("2020-03-01","2021-04-30")),  
 date\_labels = ("%Y %b %d"),  
 breaks = as.Date(c("2020-03-01","2020-03-15","2020-03-30",  
 "2020-04-15","2020-04-30","2020-05-15",  
 "2020-05-31","2020-06-15","2020-06-30", "2020-07-15","2020-07-31","2020-08-15", "2020-08-31","2020-09-15","2020-09-30",  
 "2020-10-15","2020-10-31","2020-11-15",  
 "2020-11-30","2020-12-15","2020-12-31",  
 "2021-01-15","2021-01-31","2021-02-15",  
 "2021-02-28","2021-03-15","2021-03-31"))) +  
 theme(axis.text.x = element\_text(angle = 45, vjust = 1,   
 hjust = 1, size = 10, face = "bold"))  
  
dev.off()



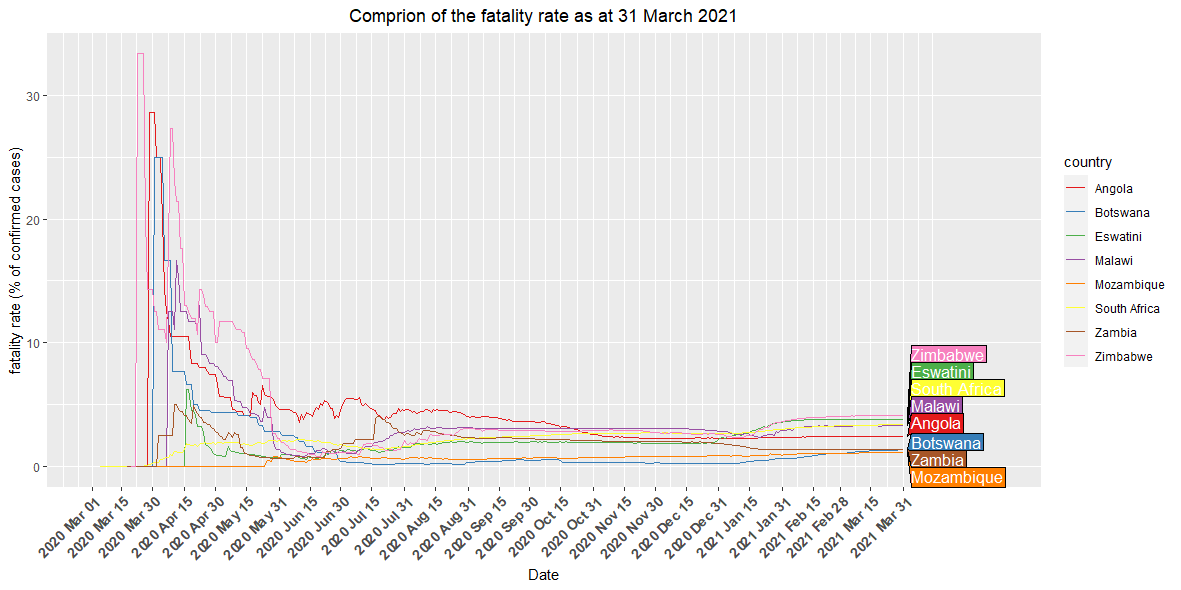
# Plot the line graph for the fatality rate as at 31 December 2020

mypath <- file.path("C:","Users","uganda", "OneDrive - BBOSA ROBERT","COVID-19","COVID-19\_Data","Output", "death\_ctg3.png")  
png(file= mypath, width =30, height = 15,units = "cm",pointsize =12, res =100)  
  
death\_cases %>%  
 filter(country %in% c("Angola","Botswana","Malawi","Zimbabwe","Mozambique","Zambia","Eswatini", "South Africa"),date<"2021-01-01") %>%  
 ggplot( aes(x=date, y=fatality\_rate, group=country, color=country)) +  
 geom\_line() +  
 geom\_dl(aes(label = country),  
 method = list(dl.trans(x = x + 0.1), "last.polygons")) +  
 scale\_color\_brewer(palette = "Set1") +  
 ggtitle("Comprion of the fatality rate as at 31 Decmber 2020") +  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 ylab("fatality rate (% of confirmed cases)") +  
 xlab("Date") +  
 scale\_y\_continuous(limits = c(ymin= 0,   
 ymax= max(death\_cases$fatality\_rate)))+  
 scale\_x\_date(limits = as.Date(c("2020-03-01","2021-05-15")),  
 date\_labels = ("%Y %b %d"),  
 breaks = as.Date(c("2020-03-01","2020-03-15","2020-03-30",  
 "2020-04-15","2020-04-30","2020-05-15",  
 "2020-05-31","2020-06-15","2020-06-30", "2020-07-15","2020-07-31","2020-08-15", "2020-08-31","2020-09-15","2020-09-30",  
 "2020-10-15","2020-10-31","2020-11-15",  
 "2020-11-30","2020-12-15","2020-12-31",  
 "2021-01-15","2021-01-31","2021-02-15",  
 "2021-02-28","2021-03-15","2021-03-31"))) +  
 theme(axis.text.x = element\_text(angle = 45, vjust = 1,   
 hjust = 1, size = 10, face = "bold"))  
  
dev.off()



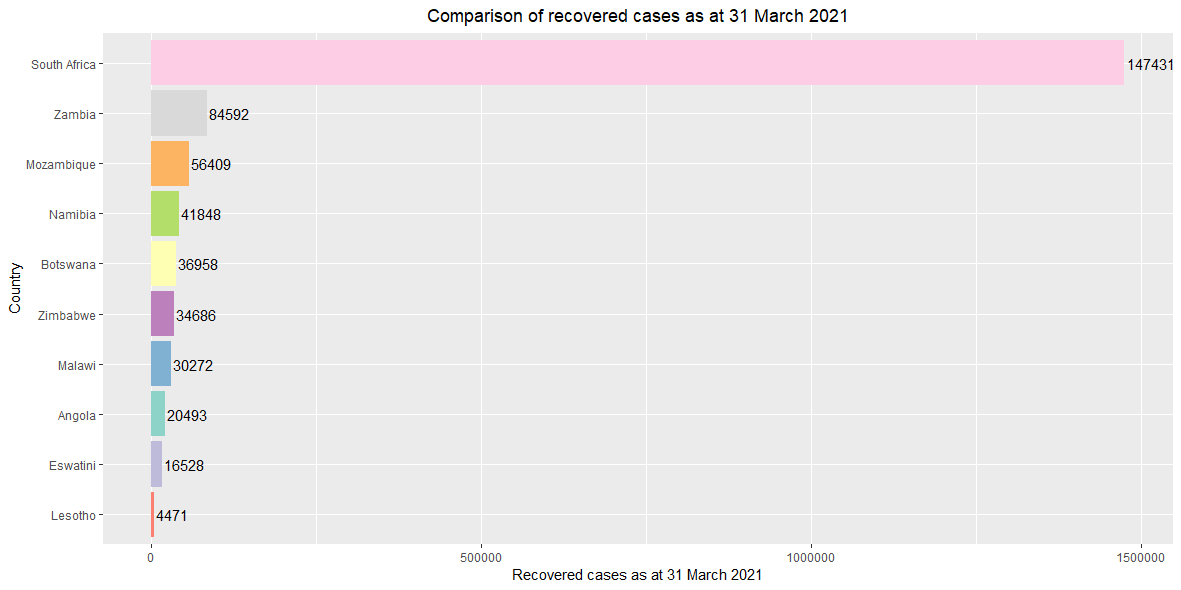
# Plot the fatality graph as at 31 March 2021

mypath <- file.path("C:","Users","uganda", "OneDrive - BBOSA ROBERT","COVID-19","COVID-19\_Data","Output", "death\_ctg4.png")  
png(file= mypath, width =30, height = 15,units = "cm",pointsize =12, res =100)  
  
death\_cases %>%  
 filter(country %in% c("Angola","Botswana","Malawi","Zimbabwe","Mozambique","Zambia","Eswatini", "South Africa"),date<"2021-04-01") %>%  
 ggplot( aes(x=date, y=fatality\_rate, group=country, color=country)) +  
 geom\_line() +  
 geom\_dl(aes(label = country),  
 method = list(dl.trans(x = x + 0.1), "last.polygons")) +  
 scale\_color\_brewer(palette = "Set1") +  
 ggtitle("Comprion of the fatality rate as at 31 March 2021") +  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 ylab("fatality rate (% of confirmed cases)") +  
 xlab("Date") +  
 scale\_y\_continuous(limits = c(ymin= 0,   
 ymax= max(death\_cases$fatality\_rate)))+  
 scale\_x\_date(limits = as.Date(c("2020-03-01","2021-05-15")),  
 date\_labels = ("%Y %b %d"),  
 breaks = as.Date(c("2020-03-01","2020-03-15","2020-03-30",  
 "2020-04-15","2020-04-30","2020-05-15",  
 "2020-05-31","2020-06-15","2020-06-30", "2020-07-15","2020-07-31","2020-08-15", "2020-08-31","2020-09-15","2020-09-30",  
 "2020-10-15","2020-10-31","2020-11-15",  
 "2020-11-30","2020-12-15","2020-12-31",  
 "2021-01-15","2021-01-31","2021-02-15",  
 "2021-02-28","2021-03-15","2021-03-31"))) +  
 theme(axis.text.x = element\_text(angle = 45, vjust = 1,   
 hjust = 1, size = 10, face = "bold"))  
  
dev.off()



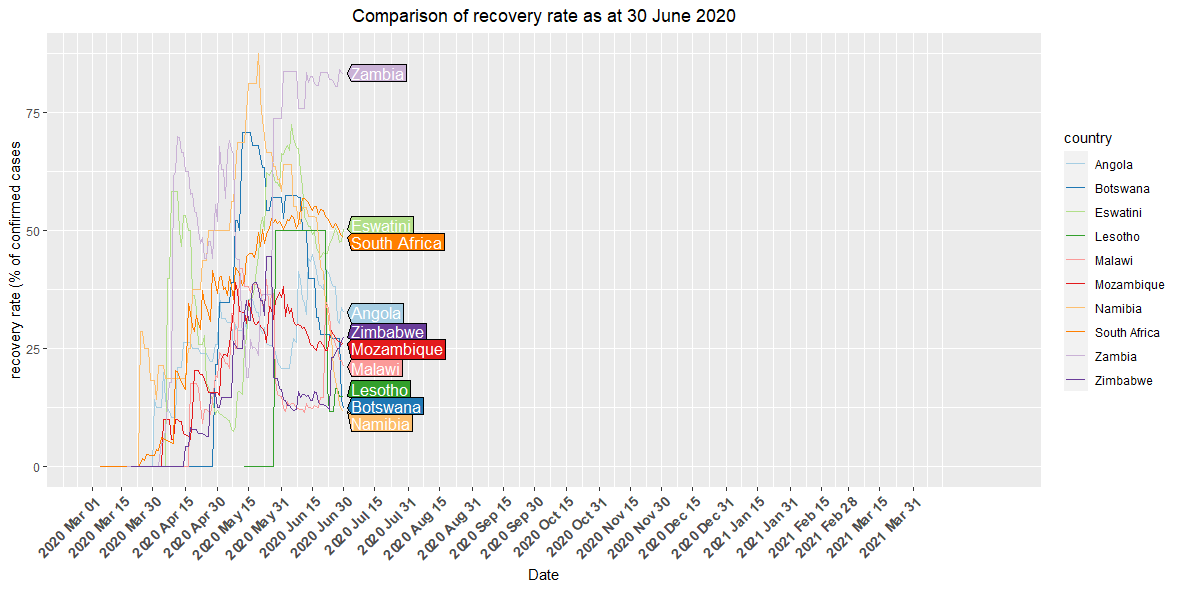
# Plot the bar chart for recoveries

mypath <- file.path("C:","Users","uganda", "OneDrive - BBOSA ROBERT","COVID-19","COVID-19\_Data","Output", "recovered\_ctg1.png")  
png(file= mypath, width =30, height = 15,units = "cm",pointsize =12, res =100)  
  
SA\_countries %>%  
 filter(country %in% c("Eswatini","Mozambique","Angola","Malawi",  
 "Namibia","South Africa","Zambia","Botswana",  
 "Zimbabwe","Lesotho"),  
 type=="recovered", date=="2021-03-31" ) %>%   
 ggplot( aes(country, total\_cases, fill=country)) +  
 geom\_bar(aes(reorder(country, total\_cases),total\_cases),  
 stat= "identity", show.legend = FALSE) +  
 geom\_text(aes(label=total\_cases), vjust=0.5, hjust=-0.05, colour="black") +  
 scale\_fill\_brewer(palette = "Set3") +  
 coord\_flip() +  
 ggtitle("Comparison of recovered cases as at 31 March 2021")+  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 xlab("Country")+  
 ylab("Recovered cases as at 31 March 2021")  
  
dev.off()



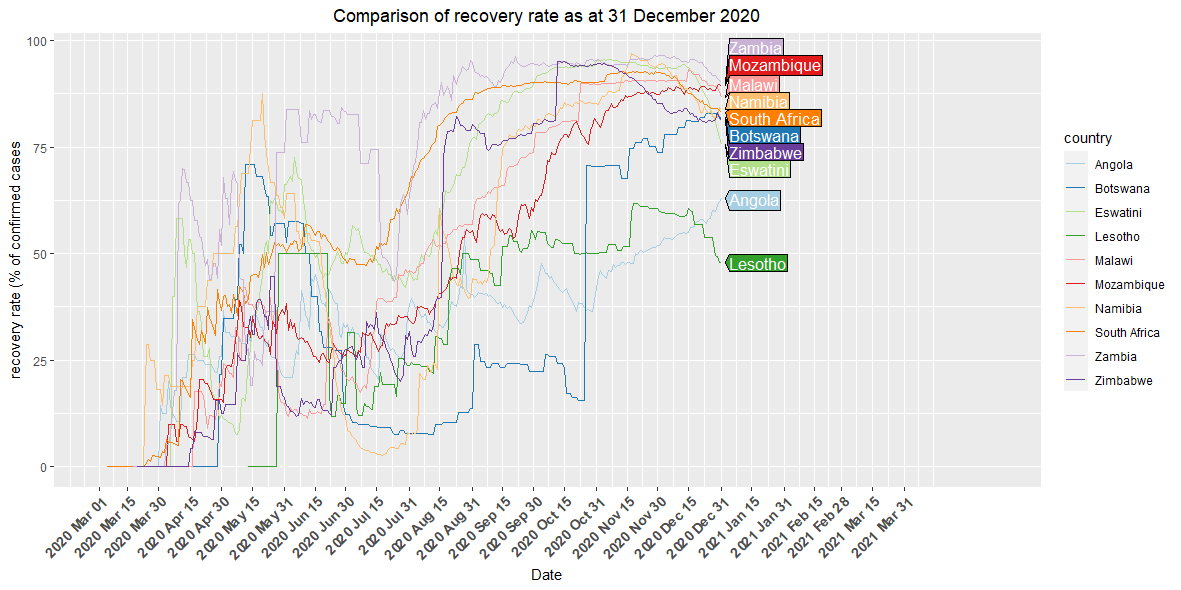
# Plot a line graph for recovery rate as at 30 June 2020

mypath <- file.path("C:","Users","uganda", "OneDrive - BBOSA ROBERT","COVID-19","COVID-19\_Data","Output", "recovered\_ctg2.png")  
png(file= mypath, width =30, height = 15,units = "cm",pointsize =12, res =100)  
  
recovered\_cases %>%  
 filter(country %in% c("Eswatini","Mozambique","Angola","Malawi","Namibia",  
 "South Africa","Zambia","Botswana","Zimbabwe",  
 "Lesotho"),  
 date<"2020-07-01") %>%   
 ggplot( aes(x=date, y=recovery\_rate, group=country, color=country)) +  
 geom\_dl(aes(label = country),   
 method = list(dl.trans(x = x + 0.1), "last.polygons")) +  
 geom\_line() +  
 scale\_color\_brewer(palette = "Paired") +  
 ggtitle("Comparison of recovery rate as at 30 June 2020")+  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 xlab("Date")+  
 ylab("recovery rate (% of confirmed cases") +  
 scale\_y\_continuous(limits = c(ymin= 0,   
 ymax= max(recovered\_cases$recovery\_rate)))+  
 scale\_x\_date(limits = as.Date(c("2020-03-01","2021-05-10")),  
 date\_labels = ("%Y %b %d"),  
 breaks = as.Date(c("2020-03-01","2020-03-15","2020-03-30",  
 "2020-04-15","2020-04-30","2020-05-15",  
 "2020-05-31","2020-06-15","2020-06-30", "2020-07-15","2020-07-31","2020-08-15", "2020-08-31","2020-09-15","2020-09-30",  
 "2020-10-15","2020-10-31","2020-11-15",  
 "2020-11-30","2020-12-15","2020-12-31",  
 "2021-01-15","2021-01-31","2021-02-15",  
 "2021-02-28","2021-03-15","2021-03-31"))) +  
 theme(axis.text.x = element\_text(angle = 45, vjust = 1,   
 hjust = 1, size = 10, face = "bold"))   
  
dev.off()



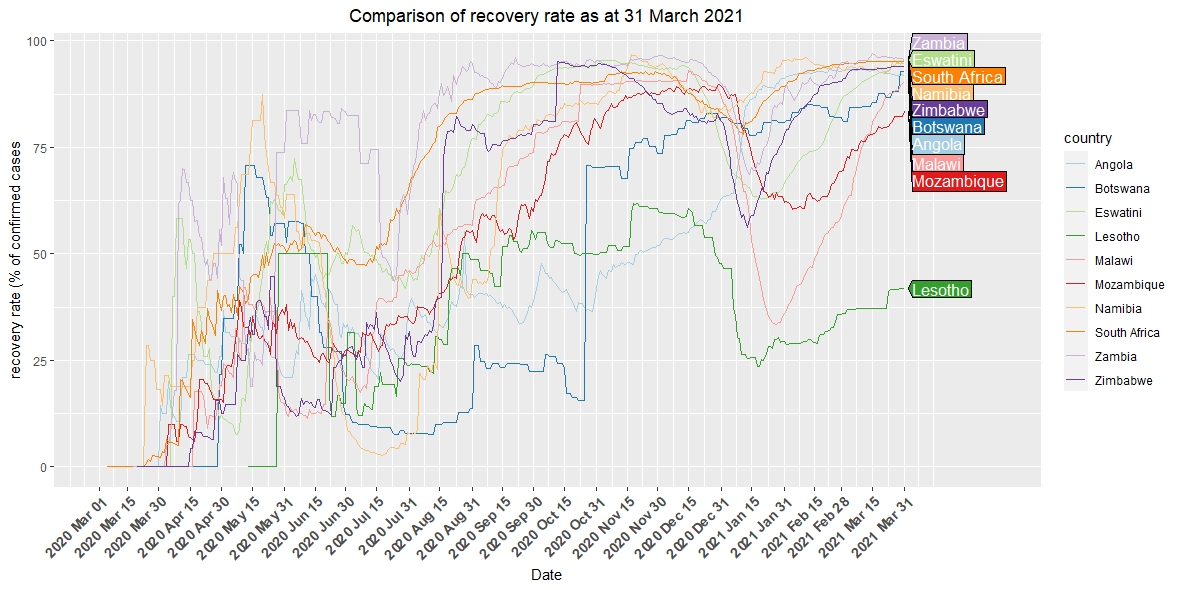
# Plot the recovery rate as at 31 Decmber 2020

mypath <- file.path("C:","Users","uganda", "OneDrive - BBOSA ROBERT","COVID-19","COVID-19\_Data","Output", "recovered\_ctg3.png")  
png(file= mypath, width =30, height = 15,units = "cm",pointsize =12, res =100)  
  
recovered\_cases %>%  
 filter(country %in% c("Eswatini","Mozambique","Angola","Malawi","Namibia",  
 "South Africa","Zambia","Botswana","Zimbabwe",  
 "Lesotho"),   
 date<"2021-01-01") %>%   
 ggplot( aes(x=date, y=recovery\_rate, group=country, color=country)) +  
 geom\_dl(aes(label = country),   
 method = list(dl.trans(x = x + 0.1), "last.polygons")) +  
 geom\_line() +  
 scale\_color\_brewer(palette = "Paired") +  
 ggtitle("Comparison of recovery rate as at 31 December 2020")+  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 xlab("Date")+  
 ylab("recovery rate (% of confirmed cases") +  
 scale\_y\_continuous(limits = c(ymin= 0,   
 ymax= max(recovered\_cases$recovery\_rate)))+  
 scale\_x\_date(limits = as.Date(c("2020-03-01","2021-05-15")),  
 date\_labels = ("%Y %b %d"),  
 breaks = as.Date(c("2020-03-01","2020-03-15","2020-03-30",  
 "2020-04-15","2020-04-30","2020-05-15",  
 "2020-05-31","2020-06-15","2020-06-30", "2020-07-15","2020-07-31","2020-08-15", "2020-08-31","2020-09-15","2020-09-30",  
 "2020-10-15","2020-10-31","2020-11-15",  
 "2020-11-30","2020-12-15","2020-12-31",  
 "2021-01-15","2021-01-31","2021-02-15",  
 "2021-02-28","2021-03-15","2021-03-31"))) +  
 theme(axis.text.x = element\_text(angle = 45, vjust = 1,   
 hjust = 1, size = 10, face = "bold"))   
  
dev.off()



# Plot the line graph of the recovery rata as at 31 March 2021

mypath <- file.path("C:","Users","uganda", "OneDrive - BBOSA ROBERT","COVID-19","COVID-19\_Data","Output", "recovered\_ctg4.png")  
png(file= mypath, width =30, height = 15,units = "cm",pointsize =12, res =100)  
  
recovered\_cases %>%  
 filter(country %in% c("Eswatini","Mozambique","Angola","Malawi","Namibia",  
 "South Africa","Zambia","Botswana","Zimbabwe",  
 "Lesotho"),   
 date<"2021-04-01") %>%   
 ggplot( aes(x=date, y=recovery\_rate, group=country, color=country)) +  
 geom\_dl(aes(label = country),   
 method = list(dl.trans(x = x + 0.1), "last.polygons")) +  
 geom\_line() +  
 scale\_color\_brewer(palette = "Paired") +  
 ggtitle("Comparison of recovery rate as at 31 March 2021")+  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 xlab("Date")+  
 ylab("recovery rate (% of confirmed cases") +  
 scale\_y\_continuous(limits = c(ymin= 0,   
 ymax= max(recovered\_cases$recovery\_rate)))+  
 scale\_x\_date(limits = as.Date(c("2020-03-01","2021-05-15")),  
 date\_labels = ("%Y %b %d"),  
 breaks = as.Date(c("2020-03-01","2020-03-15","2020-03-30",  
 "2020-04-15","2020-04-30","2020-05-15",  
 "2020-05-31","2020-06-15","2020-06-30", "2020-07-15","2020-07-31","2020-08-15", "2020-08-31","2020-09-15","2020-09-30",  
 "2020-10-15","2020-10-31","2020-11-15",  
 "2020-11-30","2020-12-15","2020-12-31",  
 "2021-01-15","2021-01-31","2021-02-15",  
 "2021-02-28","2021-03-15","2021-03-31"))) +  
 theme(axis.text.x = element\_text(angle = 45, vjust = 1,   
 hjust = 1, size = 10, face = "bold"))   
  
dev.off()



writexl::write\_xlsx(x = confirmed\_cases, path = "C:/Users/uganda/OneDrive - BBOSA ROBERT/COVID-19/COVID-19\_Data/Data/confirmed\_cases.xlsx", col\_names = TRUE)  
  
write.csv(confirmed\_cases, "C:/Users/uganda/OneDrive - BBOSA ROBERT/COVID-19/COVID-19\_Data/Data/confirmed\_cases.csv", row.names = FALSE)  
  
writexl::write\_xlsx(x = recovered\_cases, path = "C:/Users/uganda/OneDrive - BBOSA ROBERT/COVID-19/COVID-19\_Data/Data/recovered\_cases.xlsx", col\_names = TRUE)  
  
write.csv(recovered\_cases, "C:/Users/uganda/OneDrive - BBOSA ROBERT/COVID-19/COVID-19\_Data/Data/recovered\_cases.csv", row.names = FALSE)  
  
  
writexl::write\_xlsx(x = death\_cases, path = "C:/Users/uganda/OneDrive - BBOSA ROBERT/COVID-19/COVID-19\_Data/Data/death\_cases.xlsx",   
 col\_names = TRUE)  
  
write.csv(death\_cases, "C:/Users/uganda/OneDrive - BBOSA ROBERT/COVID-19/COVID-19\_Data/Data/death\_cases.csv",   
 row.names = FALSE)