Aggregating total confirmed cases

Bbosa Robert

4/28/2020

Table of Contents

# Install necessary packages

# Install the necessary packages

library(tidyverse)

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

## v ggplot2 3.3.0 v purrr 0.3.4  
## v tibble 3.0.1 v dplyr 0.8.5  
## v tidyr 1.1.0 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)  
library(hrbrthemes)

## 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

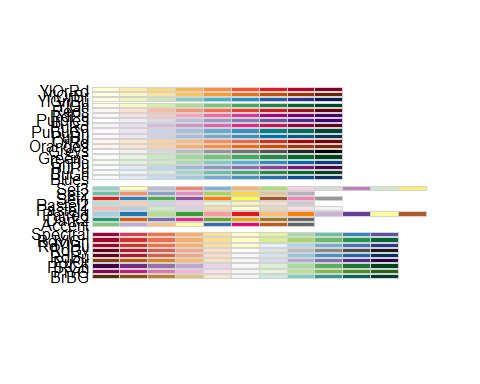
library(directlabels)  
library(knitr)  
library(pastecs)

##   
## Attaching package: 'pastecs'

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

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

library (RColorBrewer)  
display.brewer.all()



# Read the file

corona\_global <- read.csv("~/COVID-19/COVID-19\_Data/coronavirus.csv",   
 header = TRUE, stringsAsFactors = FALSE)  
attach(corona\_global)  
summary(corona\_global)

## Province.State Country.Region Lat Long   
## Length:116180 Length:116180 Min. :-51.796 Min. :-135.00   
## Class :character Class :character 1st Qu.: 6.611 1st Qu.: -14.45   
## Mode :character Mode :character Median : 22.300 Median : 21.76   
## Mean : 20.714 Mean : 24.35   
## 3rd Qu.: 40.143 3rd Qu.: 81.00   
## Max. : 71.707 Max. : 178.06   
## date cases type   
## Length:116180 Min. :-10034.0 Length:116180   
## Class :character 1st Qu.: 0.0 Class :character   
## Mode :character Median : 0.0 Mode :character   
## Mean : 110.8   
## 3rd Qu.: 3.0   
## Max. : 94305.0

names(corona\_global)

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

str(corona\_global)

## 'data.frame': 116180 obs. of 7 variables:  
## $ Province.State: chr "" "" "" "" ...  
## $ Country.Region: chr "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...  
## $ Lat : num 33 33 33 33 33 33 33 33 33 33 ...  
## $ Long : num 65 65 65 65 65 65 65 65 65 65 ...  
## $ 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

# Transfor 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': 116180 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 <- tbl\_df(corona\_global)  
class(corona\_global)

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

str(corona\_global)

## tibble [116,180 x 4] (S3: tbl\_df/tbl/data.frame)  
## $ country: chr [1:116180] "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...  
## $ date : Date[1:116180], format: "2020-01-22" "2020-01-23" ...  
## $ cases : int [1:116180] 0 0 0 0 0 0 0 0 0 0 ...  
## $ type : chr [1:116180] "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 2020-06-12 0 recovered 79510  
## 2 China 2020-06-13 0 recovered 79510  
## 3 China 2020-06-14 0 recovered 79510  
## 4 China 2020-06-15 0 recovered 79510  
## 5 China 2020-06-16 0 recovered 79510  
## 6 China 2020-06-17 0 recovered 79510

# 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 2020-06-12 0 recovered 51  
## 2 Zimbabwe 2020-06-13 3 recovered 54  
## 3 Zimbabwe 2020-06-14 0 recovered 54  
## 4 Zimbabwe 2020-06-15 0 recovered 54  
## 5 Zimbabwe 2020-06-16 8 recovered 62  
## 6 Zimbabwe 2020-06-17 1 recovered 63

# Compute the 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 2020-06-12 11 confirmed 343 23.8  
## 2 Zimbabwe 2020-06-13 13 confirmed 356 24.7  
## 3 Zimbabwe 2020-06-14 27 confirmed 383 26.5  
## 4 Zimbabwe 2020-06-15 4 confirmed 387 26.8  
## 5 Zimbabwe 2020-06-16 4 confirmed 391 27.1  
## 6 Zimbabwe 2020-06-17 10 confirmed 401 27.8

str(confirmed\_cases)

## tibble [1,480 x 6] (S3: grouped\_df/tbl\_df/tbl/data.frame)  
## $ country : chr [1:1480] "Angola" "Angola" "Angola" "Angola" ...  
## $ date : Date[1:1480], format: "2020-01-22" "2020-01-23" ...  
## $ cases : int [1:1480] 0 0 0 0 0 0 0 0 0 0 ...  
## $ type : chr [1:1480] "confirmed" "confirmed" "confirmed" "confirmed" ...  
## $ total\_cases : int [1:1480] 0 0 0 0 0 0 0 0 0 0 ...  
## $ infection\_rate: num [1:1480] 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 of 10  
## .. ..$ : int [1:148] 1 2 3 4 5 6 7 8 9 10 ...  
## .. ..$ : int [1:148] 149 150 151 152 153 154 155 156 157 158 ...  
## .. ..$ : int [1:148] 297 298 299 300 301 302 303 304 305 306 ...  
## .. ..$ : int [1:148] 445 446 447 448 449 450 451 452 453 454 ...  
## .. ..$ : int [1:148] 593 594 595 596 597 598 599 600 601 602 ...  
## .. ..$ : int [1:148] 741 742 743 744 745 746 747 748 749 750 ...  
## .. ..$ : int [1:148] 889 890 891 892 893 894 895 896 897 898 ...  
## .. ..$ : int [1:148] 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046 ...  
## .. ..$ : int [1:148] 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 ...  
## .. ..$ : int [1:148] 1333 1334 1335 1336 1337 1338 1339 1340 1341 1342 ...  
## ..- attr(\*, ".drop")= logi TRUE

summary(confirmed\_cases$infection\_rate)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0000 0.7618 29.9542 6.9895 1391.7017

stat.desc(confirmed\_cases)

## country date cases type total\_cases infection\_rate  
## nbr.val NA 1.480000e+03 1480.000000 NA 1.480000e+03 1.480000e+03  
## nbr.null NA 0.000000e+00 1011.000000 NA 6.300000e+02 6.300000e+02  
## nbr.na NA 0.000000e+00 0.000000 NA 0.000000e+00 0.000000e+00  
## min NA 1.828300e+04 -6.000000 NA 0.000000e+00 0.000000e+00  
## max NA 1.843000e+04 4302.000000 NA 8.041200e+04 1.391702e+03  
## range NA 1.470000e+02 4308.000000 NA 8.041200e+04 1.391702e+03  
## sum NA 2.716762e+07 84285.000000 NA 1.622753e+06 4.433216e+04  
## median NA 1.835650e+04 0.000000 NA 7.000000e+00 7.618247e-01  
## mean NA 1.835650e+04 56.949324 NA 1.096455e+03 2.995416e+01  
## SE.mean NA 1.110905e+00 8.848960 NA 1.696585e+02 3.131856e+00  
## CI.mean NA 2.179118e+00 17.357847 NA 3.327969e+02 6.143352e+00  
## var NA 1.826484e+03 115890.045436 NA 4.260034e+07 1.451661e+04  
## std.dev NA 4.273739e+01 340.426270 NA 6.526894e+03 1.204849e+02  
## coef.var NA 2.328188e-03 5.977705 NA 5.952725e+00 4.022309e+00

# 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 2020-06-12 11 0 1.17  
## 2 Zimbabwe 2020-06-13 13 0 1.12  
## 3 Zimbabwe 2020-06-14 27 0 1.04  
## 4 Zimbabwe 2020-06-15 4 0 1.03  
## 5 Zimbabwe 2020-06-16 4 0 1.02  
## 6 Zimbabwe 2020-06-17 10 0 1

str(death\_cases)

## tibble [1,480 x 5] (S3: grouped\_df/tbl\_df/tbl/data.frame)  
## $ country : chr [1:1480] "Angola" "Angola" "Angola" "Angola" ...  
## $ date : Date[1:1480], format: "2020-01-22" "2020-01-23" ...  
## $ confirmed : int [1:1480] 0 0 0 0 0 0 0 0 0 0 ...  
## $ death : int [1:1480] 0 0 0 0 0 0 0 0 0 0 ...  
## $ fatality\_rate: num [1:1480] 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 of 10  
## .. ..$ : int [1:148] 1 2 3 4 5 6 7 8 9 10 ...  
## .. ..$ : int [1:148] 149 150 151 152 153 154 155 156 157 158 ...  
## .. ..$ : int [1:148] 297 298 299 300 301 302 303 304 305 306 ...  
## .. ..$ : int [1:148] 445 446 447 448 449 450 451 452 453 454 ...  
## .. ..$ : int [1:148] 593 594 595 596 597 598 599 600 601 602 ...  
## .. ..$ : int [1:148] 741 742 743 744 745 746 747 748 749 750 ...  
## .. ..$ : int [1:148] 889 890 891 892 893 894 895 896 897 898 ...  
## .. ..$ : int [1:148] 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046 ...  
## .. ..$ : int [1:148] 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 ...  
## .. ..$ : int [1:148] 1333 1334 1335 1336 1337 1338 1339 1340 1341 1342 ...  
## ..- attr(\*, ".drop")= logi TRUE

stat.desc(death\_cases)

## country date confirmed death fatality\_rate  
## nbr.val NA 1.480000e+03 1480.000000 1480.0000000 850.0000000  
## nbr.null NA 0.000000e+00 1011.000000 1375.0000000 284.0000000  
## nbr.na NA 0.000000e+00 0.000000 0.0000000 630.0000000  
## min NA 1.828300e+04 -6.000000 0.0000000 0.0000000  
## max NA 1.843000e+04 4302.000000 88.0000000 33.3300000  
## range NA 1.470000e+02 4308.000000 88.0000000 33.3300000  
## sum NA 2.716762e+07 84285.000000 1711.0000000 2924.4200000  
## median NA 1.835650e+04 0.000000 0.0000000 1.1550000  
## mean NA 1.835650e+04 56.949324 1.1560811 3.4404941  
## SE.mean NA 1.110905e+00 8.848960 0.1878514 0.1836550  
## CI.mean NA 2.179118e+00 17.357847 0.3684836 0.3604711  
## var NA 1.826484e+03 115890.045436 52.2264674 28.6697967  
## std.dev NA 4.273739e+01 340.426270 7.2267882 5.3544184  
## coef.var NA 2.328188e-03 5.977705 6.2511084 1.5562934

# 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)

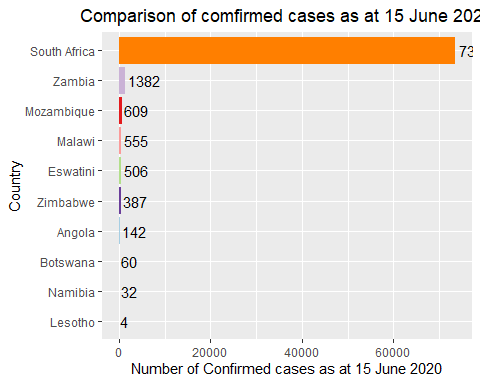
## # A tibble: 6 x 5  
## # Groups: country [1]  
## country date confirmed recovered recovery\_rate  
## <chr> <date> <int> <int> <dbl>  
## 1 Zimbabwe 2020-06-12 11 0 14.9  
## 2 Zimbabwe 2020-06-13 13 3 15.2  
## 3 Zimbabwe 2020-06-14 27 0 14.1  
## 4 Zimbabwe 2020-06-15 4 0 14.0  
## 5 Zimbabwe 2020-06-16 4 8 15.9  
## 6 Zimbabwe 2020-06-17 10 1 15.7

stat.desc(recovered\_cases)

## country date confirmed recovered recovery\_rate  
## nbr.val NA 1.480000e+03 1480.000000 1480.000000 8.500000e+02  
## nbr.null NA 0.000000e+00 1011.000000 1233.000000 1.780000e+02  
## nbr.na NA 0.000000e+00 0.000000 0.000000 6.300000e+02  
## min NA 1.828300e+04 -6.000000 0.000000 0.000000e+00  
## max NA 1.843000e+04 4302.000000 2907.000000 8.750000e+01  
## range NA 1.470000e+02 4308.000000 2907.000000 8.750000e+01  
## sum NA 2.716762e+07 84285.000000 46141.000000 2.436942e+04  
## median NA 1.835650e+04 0.000000 0.000000 2.585500e+01  
## mean NA 1.835650e+04 56.949324 31.176351 2.866991e+01  
## SE.mean NA 1.110905e+00 8.848960 5.372062 7.956803e-01  
## CI.mean NA 2.179118e+00 17.357847 10.537671 1.561731e+00  
## var NA 1.826484e+03 115890.045436 42711.388081 5.381411e+02  
## std.dev NA 4.273739e+01 340.426270 206.667337 2.319787e+01  
## coef.var NA 2.328188e-03 5.977705 6.628978 8.091365e-01

# Plot a bar chart

SA\_countries %>%   
 filter(country %in% c("Eswatini","Angola","Botswana","Malawi","South Africa","Mozambique","Zambia", "Zimbabwe", "Namibia","Lesotho"),type=="confirmed", date=="2020-06-15" ) %>%   
 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 15 June 2020") +  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 xlab("Country")+  
 ylab("Number of Confirmed cases as at 15 June 2020")

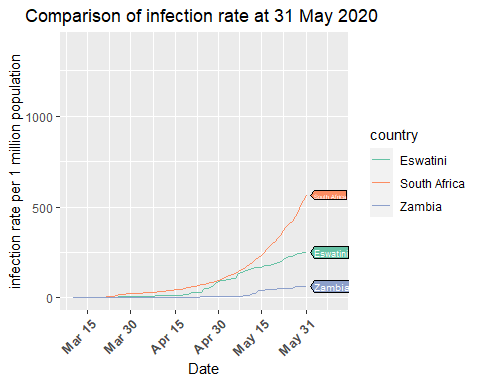


ggsave("confirmed\_ctg1.png",  
 width =30, height = 15,units = "cm",dpi = 70)

# Plot a line graph as at 31 May for SA, Eswatini and Zambia

confirmed\_cases %>%  
 filter(country %in% c("Eswatini","Zambia","South Africa"),date<"2020-06-01") %>%  
 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 May 2020 ") +  
 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","2020-06-10")),  
 date\_labels = ("%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"))) +  
 theme(axis.text.x = element\_text(angle = 45, vjust = 1, hjust = 1,   
 size = 10, face = "bold"))

## Warning: Removed 144 row(s) containing missing values (geom\_path).



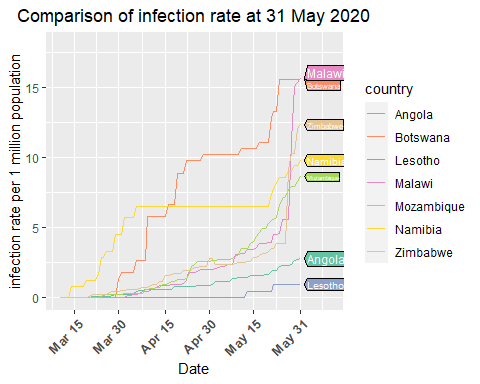
ggsave("confirmed\_ctg2.png", #other supported files;- "eps", "ps", "tex" # (pictex),"pdf","jpeg", "tiff","bmp", "svg" or "wmf".  
width =30, height = 15,  
units = "cm", # other options c("in", "cm", "mm")  
dpi = 70)

## Warning: Removed 144 row(s) containing missing values (geom\_path).

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

confirmed\_cases %>%  
 filter(country %in% c("Angola","Botswana","Lesotho","Malawi","Zimbabwe","Namibia","Mozambique"), date<"2020-06-01")%>%  
 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 May 2020") +  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 ylab("infection rate per 1 million population")+  
 xlab("Date")+  
 scale\_y\_continuous(limits = c(0,18)) +  
 scale\_x\_date(limits = as.Date(c("2020-03-10","2020-06-10")),  
 date\_labels = ("%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"))) +  
 theme(axis.text.x = element\_text(angle = 45, vjust = 1, hjust = 1,   
 size = 10, face = "bold"))

## Warning: Removed 336 row(s) containing missing values (geom\_path).

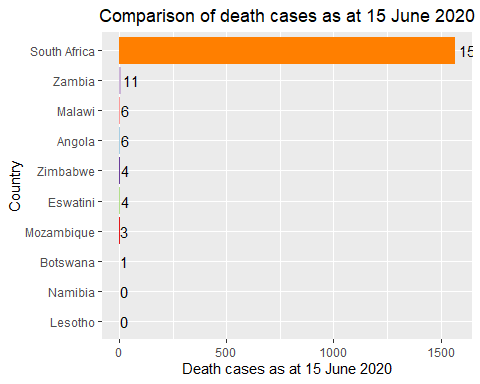


ggsave("confirmed\_ctg3.png",  
 width =30,height = 15,units = "cm", dpi = 70)

## Warning: Removed 336 row(s) containing missing values (geom\_path).

# Plot a bar graph for the deaths

SA\_countries %>%   
 filter(country %in% c("Eswatini","Angola","Botswana","Malawi","South Africa","Mozambique","Zambia", "Zimbabwe", "Namibia","Lesotho"),type=="death", date=="2020-06-15" ) %>%   
 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 15 June 2020") +  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 xlab("Country")+  
 ylab("Death cases as at 15 June 2020")

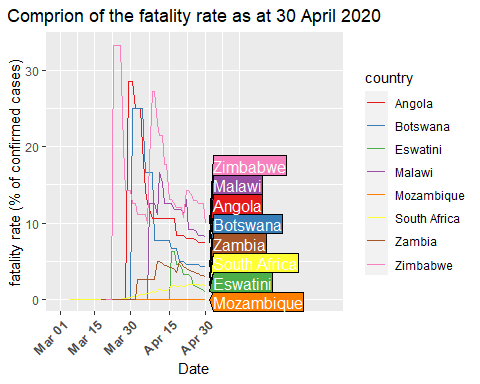


ggsave("death\_ctg1.png",  
 width =30, height = 15,units = "cm",dpi = 70)

# Plot line graph for fatality rate as at 30 April

death\_cases %>%  
 filter(country %in% c("Angola","Botswana","Malawi","Zimbabwe","Mozambique","Zambia","Eswatini", "South Africa"),date<"2020-05-01") %>%  
 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 April 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","2020-06-20")),  
 date\_labels = ("%b %d"),  
 breaks = as.Date(c("2020-03-01","2020-03-15","2020-03-30",  
 "2020-04-15","2020-04-30"))) +  
 theme(axis.text.x = element\_text(angle = 45, vjust = 1,   
 hjust = 1, size = 10, face = "bold"))

## Warning: Removed 466 row(s) containing missing values (geom\_path).



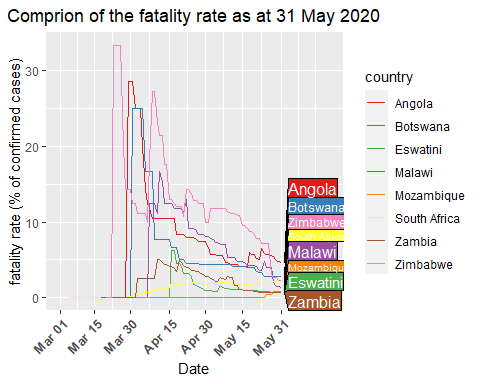
ggsave("death\_ctg2.png",  
 width =30, height = 15,units = "cm",dpi = 70)

## Warning: Removed 466 row(s) containing missing values (geom\_path).

# Plot the line graph for the fatality rate as at 31 May

death\_cases %>%  
 filter(country %in% c("Angola","Botswana","Malawi","Zimbabwe","Mozambique","Zambia","Eswatini", "South Africa"),date<"2020-06-01") %>%  
 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 May 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","2020-06-20")),  
 date\_labels = ("%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"))) +  
 theme(axis.text.x = element\_text(angle = 45, vjust = 1,   
 hjust = 1, size = 10, face = "bold"))

## Warning: Removed 466 row(s) containing missing values (geom\_path).



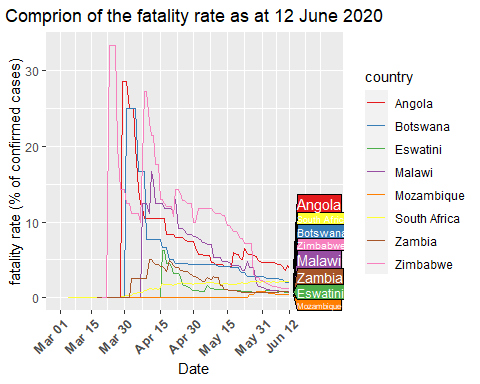
ggsave("death\_ctg3.png",  
 width =30, height = 15,units = "cm",dpi = 70)

## Warning: Removed 466 row(s) containing missing values (geom\_path).

# Plot the fatality graph as at 12 June

death\_cases %>%  
 filter(country %in% c("Angola","Botswana","Malawi","Zimbabwe","Mozambique","Zambia","Eswatini", "South Africa"),date<"2020-06-13") %>%  
 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 12 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","2020-06-30")),  
 date\_labels = ("%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","2020-06-12"))) +  
 theme(axis.text.x = element\_text(angle = 45, vjust = 1,   
 hjust = 1, size = 10, face = "bold"))

## Warning: Removed 466 row(s) containing missing values (geom\_path).

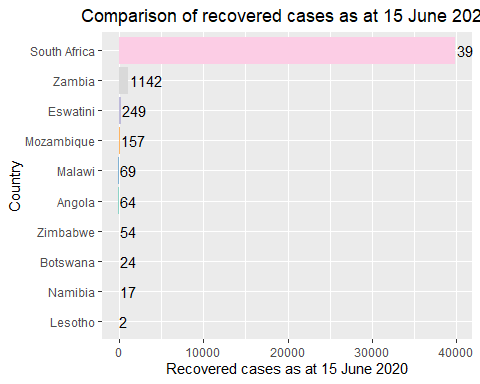


ggsave("death\_ctg4.png",  
 width =30, height = 15,units = "cm",dpi = 70)

## Warning: Removed 466 row(s) containing missing values (geom\_path).

# Plot the bar chart for recoveries

SA\_countries %>%  
 filter(country %in% c("Eswatini","Mozambique","Angola","Malawi","Namibia","South Africa",  
 "Zambia","Botswana","Zimbabwe","Lesotho"),  
 type=="recovered", date=="2020-06-15" ) %>%   
 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 15 June 2020")+  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 xlab("Country")+  
 ylab("Recovered cases as at 15 June 2020")

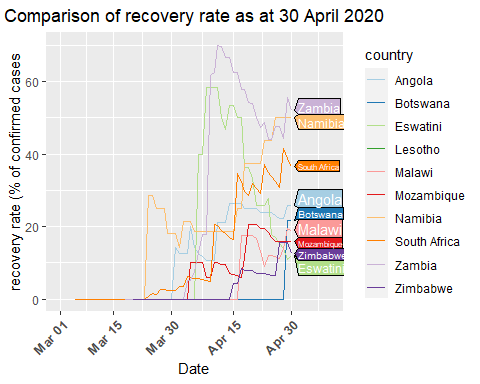


ggsave("recovered\_ctg1.png",  
 width =30, height = 15,units = "cm",dpi = 70)

# Plot a line graph for recovery rate as at 30 April

recovered\_cases %>%  
 filter(country %in% c("Eswatini","Mozambique","Angola","Malawi","Namibia","South Africa",  
 "Zambia","Botswana","Zimbabwe", "Lesotho"),date<"2020-05-01") %>%   
 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 April 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","2020-05-10")),  
 date\_labels = ("%b %d"),  
 breaks = as.Date(c("2020-03-01","2020-03-15","2020-03-30",  
 "2020-04-15","2020-04-30"))) +  
 theme(axis.text.x = element\_text(angle = 45, vjust = 1,   
 hjust = 1, size = 10, face = "bold"))

## Warning: Removed 618 row(s) containing missing values (geom\_path).



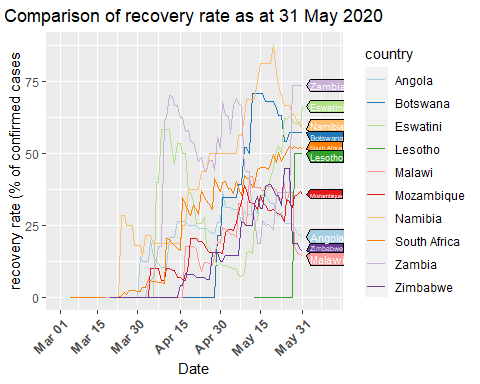
ggsave("recovered\_ctg2.png",  
 width =30, height = 15,units = "cm",dpi = 70)

## Warning: Removed 618 row(s) containing missing values (geom\_path).

# Plot the recovery rate as at 31 May

recovered\_cases %>%  
 filter(country %in% c("Eswatini","Mozambique","Angola","Malawi","Namibia","South Africa",  
 "Zambia","Botswana","Zimbabwe", "Lesotho"), date<"2020-06-01") %>%   
 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 May 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","2020-06-10")),  
 date\_labels = ("%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"))) +  
 theme(axis.text.x = element\_text(angle = 45, vjust = 1,   
 hjust = 1, size = 10, face = "bold"))

## Warning: Removed 630 row(s) containing missing values (geom\_path).



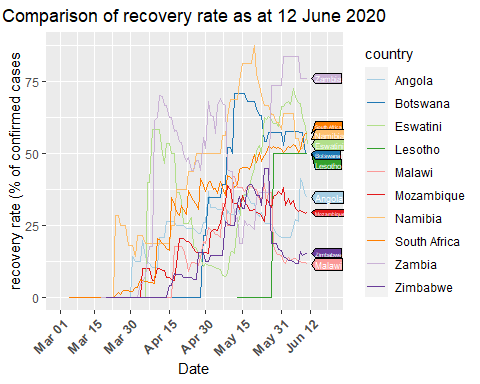
ggsave("recovered\_ctg3.png",  
 width =30, height = 15,units = "cm",dpi = 70)

## Warning: Removed 630 row(s) containing missing values (geom\_path).

# Plot the line graph of the recovery rata as at 12 June

recovered\_cases %>%  
 filter(country %in% c("Eswatini","Mozambique","Angola","Malawi","Namibia","South Africa",  
 "Zambia","Botswana","Zimbabwe", "Lesotho"), date<"2020-06-12") %>%   
 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 12 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","2020-06-20")),  
 date\_labels = ("%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","2020-06-12"))) +  
 theme(axis.text.x = element\_text(angle = 45, vjust = 1,   
 hjust = 1, size = 10, face = "bold"))

## Warning: Removed 630 row(s) containing missing values (geom\_path).



ggsave("recovered\_ctg4.png",  
 width =30, height = 15,units = "cm",dpi = 70)

## Warning: Removed 630 row(s) containing missing values (geom\_path).

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