Eastern Africa

Bbosa Robert

6/18/2020

Table of Contents

# 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

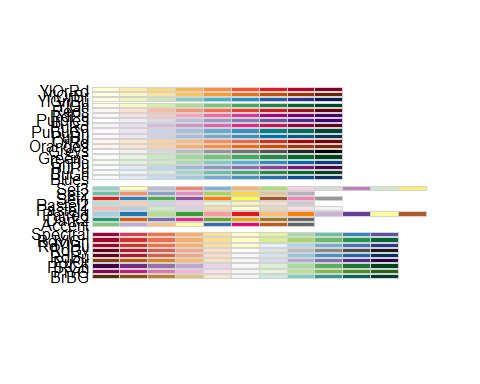
library(directlabels)  
library(knitr)  
library(pastecs)

##   
## 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

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

## Province.State Country.Region Lat Long   
## Length:115395 Length:115395 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:115395 Min. :-10034 Length:115395   
## Class :character 1st Qu.: 0 Class :character   
## Mode :character Median : 0 Mode :character   
## Mean : 109   
## 3rd Qu.: 3   
## Max. : 94305

names(EA\_global)

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

str(EA\_global)

## 'data.frame': 115395 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

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

## [1] "data.frame"

head(EA\_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”

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

## 'data.frame': 115395 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

EA\_global <- tbl\_df(EA\_global)  
class(EA\_global)

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

str(EA\_global)

## tibble [115,395 x 4] (S3: tbl\_df/tbl/data.frame)  
## $ country: chr [1:115395] "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...  
## $ date : Date[1:115395], format: "2020-01-22" "2020-01-23" ...  
## $ cases : int [1:115395] 0 0 0 0 0 0 0 0 0 0 ...  
## $ type : chr [1:115395] "confirmed" "confirmed" "confirmed" "confirmed" ...

# Include a column for totals

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

## # A tibble: 6 x 5  
## # Groups: country, type [1]  
## country date cases type total\_cases  
## <chr> <date> <int> <chr> <int>  
## 1 China 2020-06-11 0 recovered 79493  
## 2 China 2020-06-12 0 recovered 79493  
## 3 China 2020-06-13 0 recovered 79493  
## 4 China 2020-06-14 0 recovered 79493  
## 5 China 2020-06-15 0 recovered 79493  
## 6 China 2020-06-16 0 recovered 79493

# Select Eastern Africa countries

EA\_countries <- EA\_global\_total %>%   
 filter(country %in% c("Burundi","Djibouti","Eritrea","Ethiopia","Kenya",  
 "Rwanda","Somalia", "South Sudan", "Tanzania", "Uganda"))  
   
head(EA\_countries)

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

tail(EA\_countries)

## # A tibble: 6 x 5  
## # Groups: country, type [1]  
## country date cases type total\_cases  
## <chr> <date> <int> <chr> <int>  
## 1 Uganda 2020-06-11 42 recovered 161  
## 2 Uganda 2020-06-12 0 recovered 161  
## 3 Uganda 2020-06-13 58 recovered 219  
## 4 Uganda 2020-06-14 21 recovered 240  
## 5 Uganda 2020-06-15 59 recovered 299  
## 6 Uganda 2020-06-16 52 recovered 351

# Compute the infection rate

confirmed\_cases\_EA <- EA\_countries %>%   
 filter(type=="confirmed") %>%  
 group\_by(country,type) %>%  
 mutate(infection\_rate=ifelse(country=="Burundi",  
 total\_cases\*(1000000/11576000),  
 ifelse(country=="Djibouti",  
 total\_cases\*(1000000/986000),  
 ifelse(country=="Eritrea",   
 total\_cases\*(1000000/5310000),  
 ifelse(country=="Ethiopia",  
 total\_cases\*(1000000/110136000),  
 ifelse(country=="Kenya",  
 total\_cases\*(1000000/52215000),  
 ifelse(country=="Rwanda",  
 total\_cases\*(1000000/12794000),  
 ifelse(country=="Somalia",  
 total\_cases\*(1000000/15636000),  
 ifelse(country=="South Sudan",  
 total\_cases\*(1000000/13263000),  
 ifelse(country=="Tanzania",  
 total\_cases\*(1000000/60914000),  
 total\_cases\*(1000000/45712000)))))))))))   
head(confirmed\_cases\_EA)

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

tail(confirmed\_cases\_EA)

## # A tibble: 6 x 6  
## # Groups: country, type [1]  
## country date cases type total\_cases infection\_rate  
## <chr> <date> <int> <chr> <int> <dbl>  
## 1 Uganda 2020-06-11 14 confirmed 679 14.9  
## 2 Uganda 2020-06-12 7 confirmed 686 15.0  
## 3 Uganda 2020-06-13 8 confirmed 694 15.2  
## 4 Uganda 2020-06-14 2 confirmed 696 15.2  
## 5 Uganda 2020-06-15 9 confirmed 705 15.4  
## 6 Uganda 2020-06-16 19 confirmed 724 15.8

str(confirmed\_cases\_EA)

## tibble [1,470 x 6] (S3: grouped\_df/tbl\_df/tbl/data.frame)  
## $ country : chr [1:1470] "Burundi" "Burundi" "Burundi" "Burundi" ...  
## $ date : Date[1:1470], format: "2020-01-22" "2020-01-23" ...  
## $ cases : int [1:1470] 0 0 0 0 0 0 0 0 0 0 ...  
## $ type : chr [1:1470] "confirmed" "confirmed" "confirmed" "confirmed" ...  
## $ total\_cases : int [1:1470] 0 0 0 0 0 0 0 0 0 0 ...  
## $ infection\_rate: num [1:1470] 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] "Burundi" "Djibouti" "Eritrea" "Ethiopia" ...  
## ..$ type : chr [1:10] "confirmed" "confirmed" "confirmed" "confirmed" ...  
## ..$ .rows :List of 10  
## .. ..$ : int [1:147] 1 2 3 4 5 6 7 8 9 10 ...  
## .. ..$ : int [1:147] 148 149 150 151 152 153 154 155 156 157 ...  
## .. ..$ : int [1:147] 295 296 297 298 299 300 301 302 303 304 ...  
## .. ..$ : int [1:147] 442 443 444 445 446 447 448 449 450 451 ...  
## .. ..$ : int [1:147] 589 590 591 592 593 594 595 596 597 598 ...  
## .. ..$ : int [1:147] 736 737 738 739 740 741 742 743 744 745 ...  
## .. ..$ : int [1:147] 883 884 885 886 887 888 889 890 891 892 ...  
## .. ..$ : int [1:147] 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 ...  
## .. ..$ : int [1:147] 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186 ...  
## .. ..$ : int [1:147] 1324 1325 1326 1327 1328 1329 1330 1331 1332 1333 ...  
## ..- attr(\*, ".drop")= logi TRUE

summary(confirmed\_cases\_EA$infection\_rate)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000 0.000 0.625 104.541 8.356 4603.448

stat.desc(confirmed\_cases\_EA)

## country date cases type total\_cases infection\_rate  
## nbr.val NA 1.470000e+03 1470.0000000 NA 1.470000e+03 1.470000e+03  
## nbr.null NA 0.000000e+00 930.0000000 NA 5.790000e+02 5.790000e+02  
## nbr.na NA 0.000000e+00 0.0000000 NA 0.000000e+00 0.000000e+00  
## min NA 1.828300e+04 -104.0000000 NA 0.000000e+00 0.000000e+00  
## max NA 1.842900e+04 323.0000000 NA 4.539000e+03 4.603448e+03  
## range NA 1.460000e+02 427.0000000 NA 4.539000e+03 4.603448e+03  
## sum NA 2.698332e+07 18557.0000000 NA 4.678600e+05 1.536750e+05  
## median NA 1.835600e+04 0.0000000 NA 1.500000e+01 6.252931e-01  
## mean NA 1.835600e+04 12.6238095 NA 3.182721e+02 1.045408e+02  
## SE.mean NA 1.107148e+00 0.9267192 NA 1.884639e+01 1.374357e+01  
## CI.mean NA 2.171760e+00 1.8178341 NA 3.696871e+01 2.695911e+01  
## var NA 1.801892e+03 1262.4485818 NA 5.221241e+05 2.776619e+05  
## std.dev NA 4.244870e+01 35.5309524 NA 7.225816e+02 5.269364e+02  
## coef.var NA 2.312525e-03 2.8145983 NA 2.270326e+00 5.040483e+00

# Fatality rate

death\_cases\_EA <- EA\_countries %>%   
 filter(type %in% c("death", "confirmed")) %>%   
 select(-total\_cases) %>%   
 spread(type, cases) %>%   
 mutate(fatality\_rate=ifelse(country=="Burundi",  
 round((cumsum(death)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="Djibouti",  
 round((cumsum(death)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="Eritrea",  
 round((cumsum(death)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="Ethiopia",  
 round((cumsum(death)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="Kenya",  
 round((cumsum(death)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="Rwanda",  
 round((cumsum(death)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="Somalia",  
 round((cumsum(death)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="South Sudan",  
 round((cumsum(death)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="Tanzania",  
 round((cumsum(death)/cumsum(confirmed))\*100,digits=2),  
 round((cumsum(death)/cumsum(confirmed))\*100,  
 digits=2)))))))))))  
  
head(death\_cases\_EA)

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

tail(death\_cases\_EA)

## # A tibble: 6 x 5  
## # Groups: country [1]  
## country date confirmed death fatality\_rate  
## <chr> <date> <int> <int> <dbl>  
## 1 Uganda 2020-06-11 14 0 0  
## 2 Uganda 2020-06-12 7 0 0  
## 3 Uganda 2020-06-13 8 0 0  
## 4 Uganda 2020-06-14 2 0 0  
## 5 Uganda 2020-06-15 9 0 0  
## 6 Uganda 2020-06-16 19 0 0

str(death\_cases\_EA)

## tibble [1,470 x 5] (S3: grouped\_df/tbl\_df/tbl/data.frame)  
## $ country : chr [1:1470] "Burundi" "Burundi" "Burundi" "Burundi" ...  
## $ date : Date[1:1470], format: "2020-01-22" "2020-01-23" ...  
## $ confirmed : int [1:1470] 0 0 0 0 0 0 0 0 0 0 ...  
## $ death : int [1:1470] 0 0 0 0 0 0 0 0 0 0 ...  
## $ fatality\_rate: num [1:1470] 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] "Burundi" "Djibouti" "Eritrea" "Ethiopia" ...  
## ..$ .rows :List of 10  
## .. ..$ : int [1:147] 1 2 3 4 5 6 7 8 9 10 ...  
## .. ..$ : int [1:147] 148 149 150 151 152 153 154 155 156 157 ...  
## .. ..$ : int [1:147] 295 296 297 298 299 300 301 302 303 304 ...  
## .. ..$ : int [1:147] 442 443 444 445 446 447 448 449 450 451 ...  
## .. ..$ : int [1:147] 589 590 591 592 593 594 595 596 597 598 ...  
## .. ..$ : int [1:147] 736 737 738 739 740 741 742 743 744 745 ...  
## .. ..$ : int [1:147] 883 884 885 886 887 888 889 890 891 892 ...  
## .. ..$ : int [1:147] 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 ...  
## .. ..$ : int [1:147] 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186 ...  
## .. ..$ : int [1:147] 1324 1325 1326 1327 1328 1329 1330 1331 1332 1333 ...  
## ..- attr(\*, ".drop")= logi TRUE

stat.desc(death\_cases\_EA)

## country date confirmed death fatality\_rate  
## nbr.val NA 1.470000e+03 1470.0000000 1.470000e+03 8.910000e+02  
## nbr.null NA 0.000000e+00 930.0000000 1.319000e+03 4.030000e+02  
## nbr.na NA 0.000000e+00 0.0000000 0.000000e+00 5.790000e+02  
## min NA 1.828300e+04 -104.0000000 0.000000e+00 0.000000e+00  
## max NA 1.842900e+04 323.0000000 8.000000e+00 2.000000e+01  
## range NA 1.460000e+02 427.0000000 8.000000e+00 2.000000e+01  
## sum NA 2.698332e+07 18557.0000000 3.510000e+02 1.620440e+03  
## median NA 1.835600e+04 0.0000000 0.000000e+00 4.300000e-01  
## mean NA 1.835600e+04 12.6238095 2.387755e-01 1.818676e+00  
## SE.mean NA 1.107148e+00 0.9267192 2.260245e-02 9.379989e-02  
## CI.mean NA 2.171760e+00 1.8178341 4.433652e-02 1.840948e-01  
## var NA 1.801892e+03 1262.4485818 7.509801e-01 7.839391e+00  
## std.dev NA 4.244870e+01 35.5309524 8.665911e-01 2.799891e+00  
## coef.var NA 2.312525e-03 2.8145983 3.629313e+00 1.539522e+00

# Recovery rate

recovered\_cases\_EA <- EA\_countries %>%   
 filter(type %in% c("recovered","confirmed")) %>%   
 select(-total\_cases) %>%   
 spread(type, cases) %>%  
 mutate(recovery\_rate=ifelse(country=="Burundi",  
 round((cumsum(recovered)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="Djibouti",  
 round((cumsum(recovered)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="Eritrea",  
 round((cumsum(recovered)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="Ethiopia",  
 round((cumsum(recovered)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="Kenya",  
 round((cumsum(recovered)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="Rwanda",  
 round((cumsum(recovered)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="Somalia",  
 round((cumsum(recovered)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="South Sudan",  
 round((cumsum(recovered)/cumsum(confirmed))\*100,digits=2),  
 ifelse(country=="Tanzania", round((cumsum(recovered)/cumsum(confirmed))\*100,digits=2),  
 round((cumsum(recovered)/cumsum(confirmed))\*100,  
 digits=2)))))))))))  
  
  
head(recovered\_cases\_EA)

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

tail(recovered\_cases\_EA)

## # A tibble: 6 x 5  
## # Groups: country [1]  
## country date confirmed recovered recovery\_rate  
## <chr> <date> <int> <int> <dbl>  
## 1 Uganda 2020-06-11 14 42 23.7  
## 2 Uganda 2020-06-12 7 0 23.5  
## 3 Uganda 2020-06-13 8 58 31.6  
## 4 Uganda 2020-06-14 2 21 34.5  
## 5 Uganda 2020-06-15 9 59 42.4  
## 6 Uganda 2020-06-16 19 52 48.5

str(recovered\_cases\_EA)

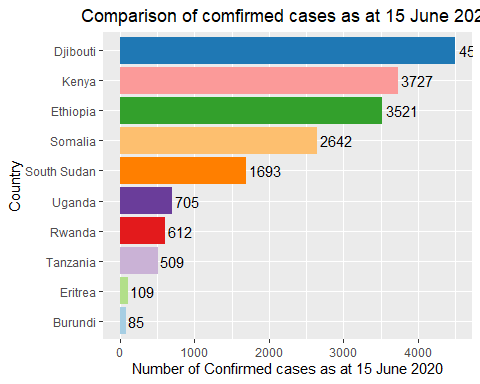
## tibble [1,470 x 5] (S3: grouped\_df/tbl\_df/tbl/data.frame)  
## $ country : chr [1:1470] "Burundi" "Burundi" "Burundi" "Burundi" ...  
## $ date : Date[1:1470], format: "2020-01-22" "2020-01-23" ...  
## $ confirmed : int [1:1470] 0 0 0 0 0 0 0 0 0 0 ...  
## $ recovered : int [1:1470] 0 0 0 0 0 0 0 0 0 0 ...  
## $ recovery\_rate: num [1:1470] 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] "Burundi" "Djibouti" "Eritrea" "Ethiopia" ...  
## ..$ .rows :List of 10  
## .. ..$ : int [1:147] 1 2 3 4 5 6 7 8 9 10 ...  
## .. ..$ : int [1:147] 148 149 150 151 152 153 154 155 156 157 ...  
## .. ..$ : int [1:147] 295 296 297 298 299 300 301 302 303 304 ...  
## .. ..$ : int [1:147] 442 443 444 445 446 447 448 449 450 451 ...  
## .. ..$ : int [1:147] 589 590 591 592 593 594 595 596 597 598 ...  
## .. ..$ : int [1:147] 736 737 738 739 740 741 742 743 744 745 ...  
## .. ..$ : int [1:147] 883 884 885 886 887 888 889 890 891 892 ...  
## .. ..$ : int [1:147] 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 ...  
## .. ..$ : int [1:147] 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186 ...  
## .. ..$ : int [1:147] 1324 1325 1326 1327 1328 1329 1330 1331 1332 1333 ...  
## ..- attr(\*, ".drop")= logi TRUE

stat.desc(recovered\_cases\_EA)

## country date confirmed recovered recovery\_rate  
## nbr.val NA 1.470000e+03 1470.0000000 1470.0000000 8.910000e+02  
## nbr.null NA 0.000000e+00 930.0000000 1107.0000000 1.910000e+02  
## nbr.na NA 0.000000e+00 0.0000000 0.0000000 5.790000e+02  
## min NA 1.828300e+04 -104.0000000 -4.0000000 0.000000e+00  
## max NA 1.842900e+04 323.0000000 259.0000000 1.000000e+02  
## range NA 1.460000e+02 427.0000000 263.0000000 1.000000e+02  
## sum NA 2.698332e+07 18557.0000000 7083.0000000 2.424250e+04  
## median NA 1.835600e+04 0.0000000 0.0000000 2.000000e+01  
## mean NA 1.835600e+04 12.6238095 4.8183673 2.720819e+01  
## SE.mean NA 1.107148e+00 0.9267192 0.4922918 8.853239e-01  
## CI.mean NA 2.171760e+00 1.8178341 0.9656699 1.737566e+00  
## var NA 1.801892e+03 1262.4485818 356.2562996 6.983644e+02  
## std.dev NA 4.244870e+01 35.5309524 18.8747530 2.642659e+01  
## coef.var NA 2.312525e-03 2.8145983 3.9172507 9.712731e-01

# Plot a bar chart

EA\_countries %>%   
 filter(country %in% c("Burundi","Djibouti","Eritrea","Ethiopia","Kenya",  
 "Rwanda","Somalia", "South Sudan", "Tanzania", "Uganda"), 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")

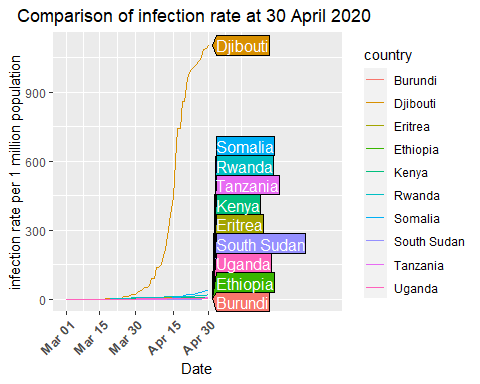


ggsave("confirmed\_EA1.png",  
 width =30, height = 15, units = "cm", dpi = 70)

# Plot a line graph as at 30 April

confirmed\_cases\_EA %>%  
 filter(country %in% c("Burundi","Djibouti","Eritrea","Ethiopia","Kenya",  
 "Rwanda","Somalia", "South Sudan", "Tanzania", "Uganda"),  
 date<"2020-05-01") %>%  
 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\_fill\_brewer(palette = "Paired") +  
 ggtitle("Comparison of infection rate at 30 April 2020 ") +  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 ylab("infection rate per 1 million population")+  
 xlab("Date") +  
 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 390 row(s) containing missing values (geom\_path).



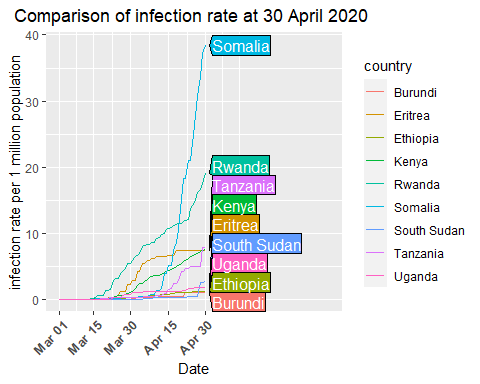
ggsave("confirmed\_EA2.png", #other supported files;- "eps", "ps", "tex" # (pictex),"pdf","jpeg", "tiff","bmp", "svg" or "wmf".  
width =30, height = 15,  
units = "cm", # other options c("in", "cm", "mm")  
dpi = 70)

## Warning: Removed 390 row(s) containing missing values (geom\_path).

# Plot a line graph as at 30 April

confirmed\_cases\_EA %>%  
 filter(country %in% c("Burundi","Eritrea","Ethiopia","Kenya",  
 "Rwanda","Somalia", "South Sudan", "Tanzania", "Uganda"),  
 date<"2020-05-01") %>%  
 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\_fill\_brewer(palette = "Paired") +  
 ggtitle("Comparison of infection rate at 30 April 2020 ") +  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 ylab("infection rate per 1 million population")+  
 xlab("Date") +  
 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 351 row(s) containing missing values (geom\_path).



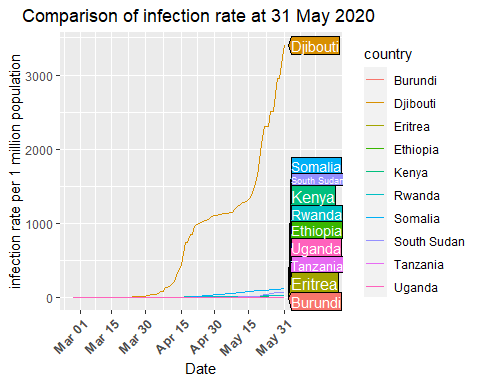
ggsave("confirmed\_EA3.png", #other supported files;- "eps", "ps", "tex" # (pictex),"pdf","jpeg", "tiff","bmp", "svg" or "wmf".  
width =30, height = 15,  
units = "cm", # other options c("in", "cm", "mm")  
dpi = 70)

## Warning: Removed 351 row(s) containing missing values (geom\_path).

# Plot a line graph as at 31 May

confirmed\_cases\_EA %>%  
 filter(country %in% c("Burundi","Djibouti","Eritrea","Ethiopia","Kenya",  
 "Rwanda","Somalia", "South Sudan", "Tanzania", "Uganda"),  
 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\_fill\_brewer(palette = "Paired") +  
 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\_x\_date(limits = as.Date(c("2020-02-27","2020-06-20")),  
 date\_labels = ("%b %d"),  
 breaks = as.Date(c("2020-02-15","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 360 row(s) containing missing values (geom\_path).



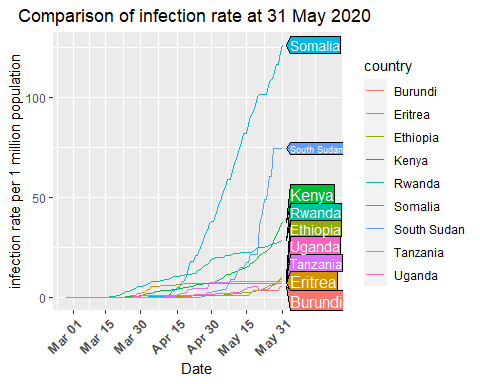
ggsave("confirmed\_EA4.png", #other supported files;- "eps", "ps", "tex" # (pictex),"pdf","jpeg", "tiff","bmp", "svg" or "wmf".  
width =30, height = 15,  
units = "cm", # other options c("in", "cm", "mm")  
dpi = 70)

## Warning: Removed 360 row(s) containing missing values (geom\_path).

# Plot a line graph as at 31 May

confirmed\_cases\_EA %>%  
 filter(country %in% c("Burundi","Eritrea","Ethiopia","Kenya",  
 "Rwanda","Somalia", "South Sudan", "Tanzania", "Uganda"),  
 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\_fill\_brewer(palette = "Paired") +  
 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\_x\_date(limits = as.Date(c("2020-02-27","2020-06-20")),  
 date\_labels = ("%b %d"),  
 breaks = as.Date(c("2020-02-15","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 324 row(s) containing missing values (geom\_path).



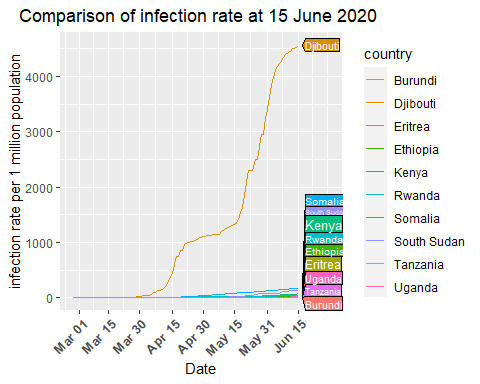
ggsave("confirmed\_EA5.png", #other supported files;- "eps", "ps", "tex" # (pictex),"pdf","jpeg", "tiff","bmp", "svg" or "wmf".  
width =30, height = 15,  
units = "cm", # other options c("in", "cm", "mm")  
dpi = 70)

## Warning: Removed 324 row(s) containing missing values (geom\_path).

# Plot a line graph as at 15 June

confirmed\_cases\_EA %>%  
 filter(country %in% c("Burundi","Djibouti","Eritrea","Ethiopia","Kenya",  
 "Rwanda","Somalia", "South Sudan", "Tanzania", "Uganda"),  
 date<"2020-06-16") %>%  
 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\_fill\_brewer(palette = "Paired") +  
 ggtitle("Comparison of infection rate at 15 June 2020 ") +  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 ylab("infection rate per 1 million population")+  
 xlab("Date") +  
 scale\_x\_date(limits = as.Date(c("2020-02-27","2020-06-30")),  
 date\_labels = ("%b %d"),  
 breaks = as.Date(c("2020-02-15","2020-03-01","2020-03-15","2020-03-30",  
 "2020-04-15","2020-04-30","2020-05-15",  
 "2020-05-31","2020-06-15"))) +  
 theme(axis.text.x = element\_text(angle = 45, vjust = 1, hjust = 1,   
 size = 10, face = "bold"))

## Warning: Removed 360 row(s) containing missing values (geom\_path).



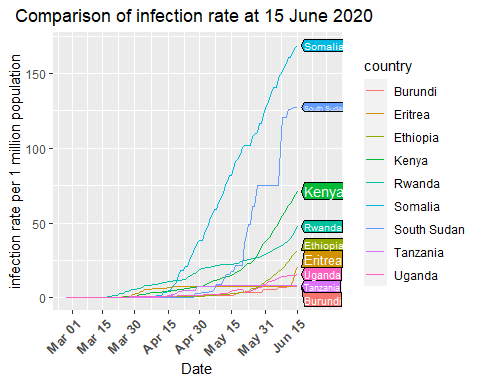
ggsave("confirmed\_EA6.png", #other supported files;- "eps", "ps", "tex" # (pictex),"pdf","jpeg", "tiff","bmp", "svg" or "wmf".  
width =30, height = 15,  
units = "cm", # other options c("in", "cm", "mm")  
dpi = 70)

## Warning: Removed 360 row(s) containing missing values (geom\_path).

# Plot a line graph as at 15 June

confirmed\_cases\_EA %>%  
 filter(country %in% c("Burundi","Eritrea","Ethiopia","Kenya",  
 "Rwanda","Somalia", "South Sudan", "Tanzania", "Uganda"),  
 date<"2020-06-16") %>%  
 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\_fill\_brewer(palette = "Paired") +  
 ggtitle("Comparison of infection rate at 15 June 2020 ") +  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 ylab("infection rate per 1 million population")+  
 xlab("Date") +  
 scale\_x\_date(limits = as.Date(c("2020-02-27","2020-06-30")),  
 date\_labels = ("%b %d"),  
 breaks = as.Date(c("2020-02-15","2020-03-01","2020-03-15","2020-03-30",  
 "2020-04-15","2020-04-30","2020-05-15",  
 "2020-05-31","2020-06-15"))) +  
 theme(axis.text.x = element\_text(angle = 45, vjust = 1, hjust = 1,   
 size = 10, face = "bold"))

## Warning: Removed 324 row(s) containing missing values (geom\_path).

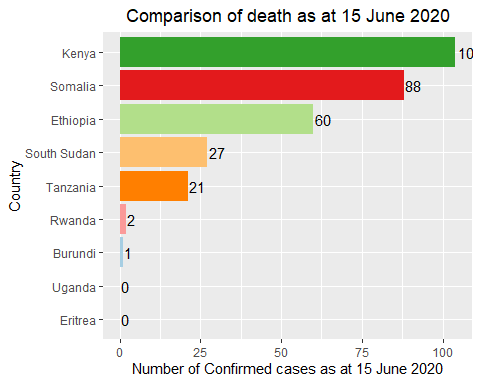


ggsave("confirmed\_EA7.png", #other supported files;- "eps", "ps", "tex" # (pictex),"pdf","jpeg", "tiff","bmp", "svg" or "wmf".  
width =30, height = 15,  
units = "cm", # other options c("in", "cm", "mm")  
dpi = 70)

## Warning: Removed 324 row(s) containing missing values (geom\_path).

# Plot a bar graph for the deaths

EA\_countries %>%   
 filter(country %in% c("Burundi","Eritrea","Ethiopia","Kenya",  
 "Rwanda","Somalia", "South Sudan", "Tanzania","Uganda"), 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 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")

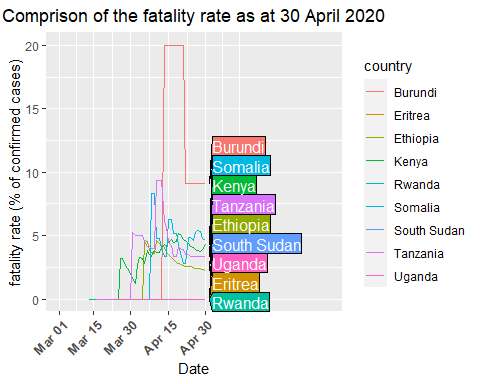


ggsave("death\_EA1.png",  
 width =30, height = 15, units = "cm", dpi = 70)

# Plot line graph for fatality rate as at 30 April

death\_cases\_EA %>%  
 filter(country %in% c("Burundi","Eritrea","Ethiopia","Kenya",  
 "Rwanda","Somalia", "South Sudan","Tanzania","Uganda"),  
 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\_fill\_brewer(palette = "Paired") +  
 ggtitle("Comprison 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\_EA$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 523 row(s) containing missing values (geom\_path).



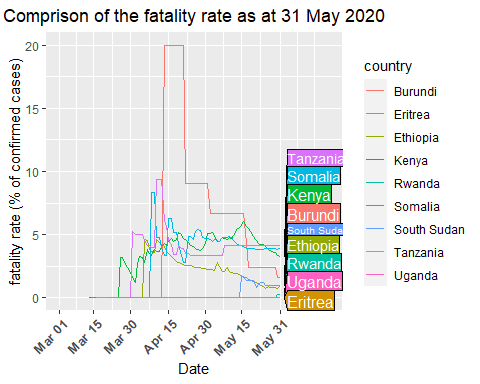
ggsave("death\_EA2.png",  
 width =30, height = 15,units = "cm",dpi = 70)

## Warning: Removed 523 row(s) containing missing values (geom\_path).

# Plot line graph for fatality rate as at 31 May

death\_cases\_EA %>%  
 filter(country %in% c("Burundi","Eritrea","Ethiopia","Kenya",  
 "Rwanda","Somalia", "South Sudan","Tanzania","Uganda"),  
 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\_fill\_brewer(palette = "Paired") +  
 ggtitle("Comprison 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\_EA$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 523 row(s) containing missing values (geom\_path).



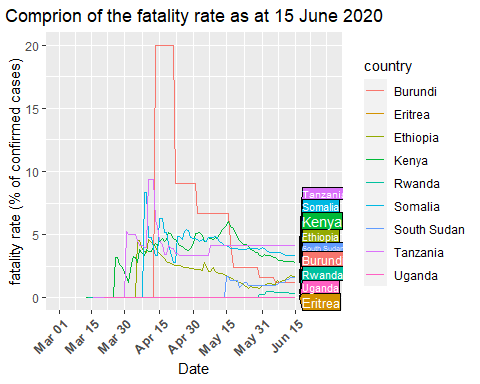
ggsave("death\_EA3.png",  
 width =30, height = 15,units = "cm",dpi = 70)

## Warning: Removed 523 row(s) containing missing values (geom\_path).

# Plot line graph for fatality rate as at 15 June

death\_cases\_EA %>%  
 filter(country %in% c("Burundi","Eritrea","Ethiopia","Kenya",  
 "Rwanda","Somalia", "South Sudan","Tanzania","Uganda"),  
 date<"2020-06-16") %>%  
 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\_fill\_brewer(palette = "Paired") +  
 ggtitle("Comprion of the fatality rate as at 15 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\_EA$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","2020-06-15"))) +  
 theme(axis.text.x = element\_text(angle = 45, vjust = 1,   
 hjust = 1, size = 10, face = "bold"))

## Warning: Removed 523 row(s) containing missing values (geom\_path).

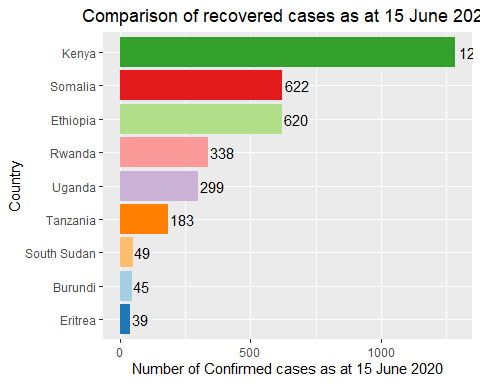


ggsave("death\_EA4.png",  
 width =30, height = 15,units = "cm",dpi = 70)

## Warning: Removed 523 row(s) containing missing values (geom\_path).

# Plot the bar chart for recoveries

EA\_countries %>%   
 filter(country %in% c("Burundi","Eritrea","Ethiopia","Kenya",  
 "Rwanda","Somalia", "South Sudan","Tanzania","Uganda"), 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=-0.1, colour="black") +  
 scale\_fill\_brewer(palette = "Paired") +  
 coord\_flip() +  
 ggtitle("Comparison of recovered 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")

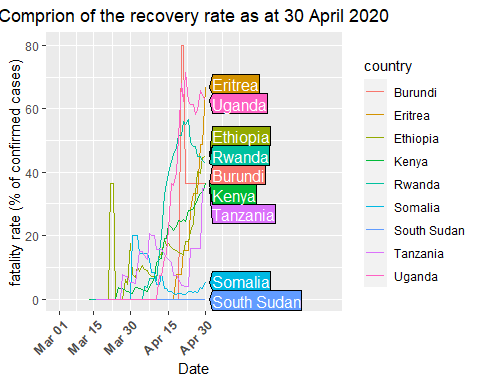


ggsave("recovered\_EA1.png",  
 width =30, height = 15, units = "cm", dpi = 70)

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

recovered\_cases\_EA %>%  
 filter(country %in% c("Burundi","Eritrea","Ethiopia","Kenya",  
 "Rwanda","Somalia", "South Sudan","Tanzania","Uganda"),  
 date<"2020-05-01") %>%  
 ggplot( aes(x=date, y=recovery\_rate, group=country, color=country)) +  
 geom\_line() +  
 geom\_dl(aes(label = country),  
 method = list(dl.trans(x = x + 0.1), "last.polygons")) +  
 scale\_fill\_brewer(palette = "Paired") +  
 ggtitle("Comprion of the recovery 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(recovered\_cases\_EA$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"))) +  
 theme(axis.text.x = element\_text(angle = 45, vjust = 1,   
 hjust = 1, size = 10, face = "bold"))

## Warning: Removed 523 row(s) containing missing values (geom\_path).



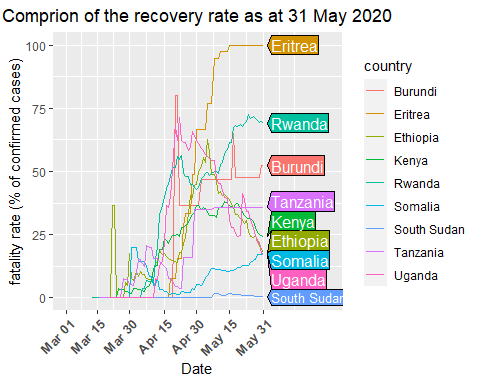
ggsave("recovered\_EA2.png",  
 width =30, height = 15,units = "cm",dpi = 70)

## Warning: Removed 523 row(s) containing missing values (geom\_path).

# Plot a line graph for recovery rate as at 31 May

recovered\_cases\_EA %>%  
 filter(country %in% c("Burundi","Eritrea","Ethiopia","Kenya",  
 "Rwanda","Somalia", "South Sudan","Tanzania","Uganda"),  
 date<"2020-06-01") %>%  
 ggplot( aes(x=date, y=recovery\_rate, group=country, color=country)) +  
 geom\_line() +  
 geom\_dl(aes(label = country),  
 method = list(dl.trans(x = x + 0.1), "last.polygons")) +  
 scale\_fill\_brewer(palette = "Paired") +  
 ggtitle("Comprion of the recovery 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(recovered\_cases\_EA$recovery\_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"))) +  
 theme(axis.text.x = element\_text(angle = 45, vjust = 1,   
 hjust = 1, size = 10, face = "bold"))

## Warning: Removed 523 row(s) containing missing values (geom\_path).



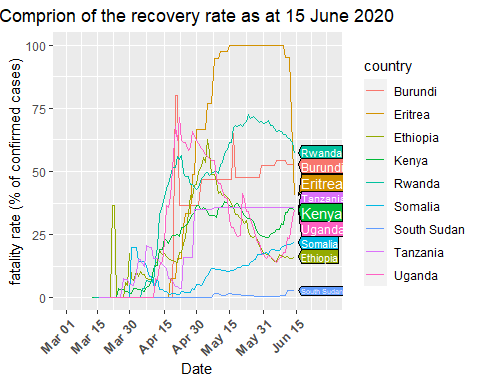
ggsave("recovered\_EA3.png",  
 width =30, height = 15,units = "cm",dpi = 70)

## Warning: Removed 523 row(s) containing missing values (geom\_path).

# Plot a line graph for recovery rate as at 15 June

recovered\_cases\_EA %>%  
 filter(country %in% c("Burundi","Eritrea","Ethiopia","Kenya",  
 "Rwanda","Somalia", "South Sudan","Tanzania","Uganda"),  
 date<"2020-06-15") %>%  
 ggplot( aes(x=date, y=recovery\_rate, group=country, color=country)) +  
 geom\_line() +  
 geom\_dl(aes(label = country),  
 method = list(dl.trans(x = x + 0.1), "last.polygons")) +  
 scale\_fill\_brewer(palette = "Paired") +  
 ggtitle("Comprion of the recovery rate as at 15 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(recovered\_cases\_EA$recovery\_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","2020-06-15"))) +  
 theme(axis.text.x = element\_text(angle = 45, vjust = 1,   
 hjust = 1, size = 10, face = "bold"))

## Warning: Removed 523 row(s) containing missing values (geom\_path).



ggsave("recovered\_EA4.png",  
 width =30, height = 15,units = "cm",dpi = 70)

## Warning: Removed 523 row(s) containing missing values (geom\_path).