West Africa

Bbosa Robert

6/17/2020

Contents

1 Install the necessary packages

```r
library(tidyverse)
```

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

## v ggplot2 3.3.1     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()
```

```r
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
```

```r
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
```

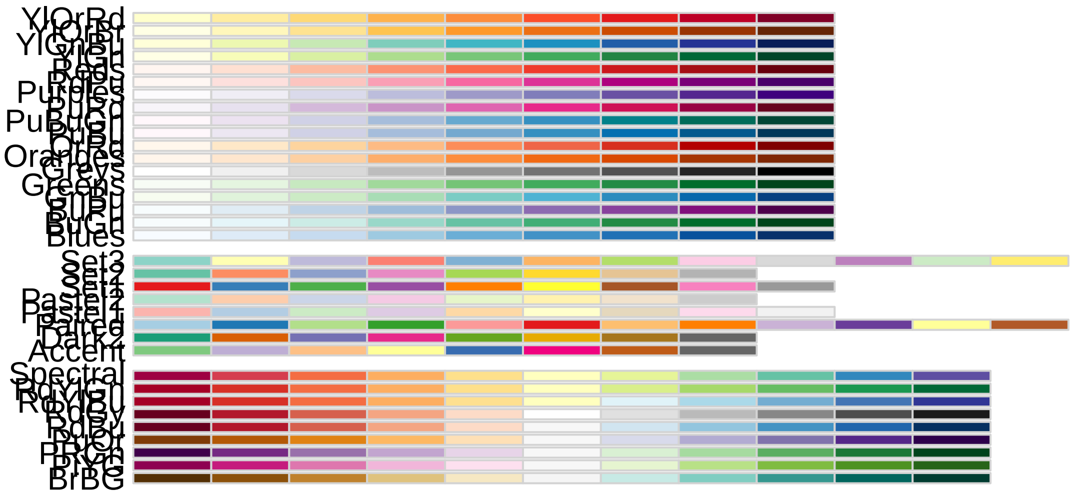
```
library(directlabels)
library(knitr)
library(pastecs)
```

```
##
## 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()
```



2 Read the file

```
WA_global <- read.csv("~/COVID-19/COVID-19_Data/coronavirus.csv",
                          header = TRUE, stringsAsFactors = FALSE)
attach(WA_global)
summary(WA_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(WA_global)
```

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

```
str(WA_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" ...
```

3 Eliminate the unnecessary columns

```
WA_global <- select(WA_global, -Province.State, -Lat, -Long)
class(WA_global)
```

```
## [1] "data.frame"
```

```
head(WA_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
```

4 Transfor date from "Character format to date format"

```
WA_global <- rename(WA_global, country = Country.Region)
myformat <- "%Y-%m-%d"
WA_global$date <- as.Date(WA_global$date, myformat)
str(WA_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" ...
```

5 Check if the date has changed

```
WA_global <- tbl_df(WA_global)
class(WA_global)
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

```
str(WA_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" ...
```

6 Include a column for totals

```
WA_global_total <- WA_global %>%
  group_by(country, type ) %>%
  mutate(total_cases = cumsum(cases))
head(WA_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(WA_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
```

7 Select Eastern Africa countries

```
WA_countries <- WA_global_total %>%
  filter(country %in% c("Ghana","Nigeria","Niger","Guinea-Bissau","Guinea",
                        "Cabo Verde","Burkina Faso", "Togo", "Benin", "Gambia", "Mauritania", "Senegal"

head(WA_countries)
```

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

```
tail(WA_countries)
```

```
## # A tibble: 6 x 5
## # Groups:   country, type [1]
##   country date       cases type      total_cases
##   <chr>   <date>     <int> <chr>           <int>
## 1 Togo    2020-06-12     8 recovered         279
## 2 Togo    2020-06-13    12 recovered         291
## 3 Togo    2020-06-14     0 recovered         291
## 4 Togo    2020-06-15     8 recovered         299
## 5 Togo    2020-06-16    45 recovered         344
## 6 Togo    2020-06-17     9 recovered         353
```

8 Compute the infection rate

```
confirmed_cases_WA <- WA_countries %>%
  filter(type=="confirmed") %>%
  group_by(country,type) %>%
  mutate(infection_rate=ifelse(country=="Ghana",
                               total_cases*(1000000/31072940),
                        ifelse(country=="Nigeria",
                               total_cases*(1000000/206139589),
                        ifelse(country=="Niger",
                               total_cases*(1000000/24206644),
                        ifelse(country=="Cabo Verde",
                               total_cases*(1000000/555987),
                        ifelse(country=="Guinea-Bissau",
                               total_cases*(1000000/1968001),
                        ifelse(country=="Guinea",
                               total_cases*(1000000/13132795),
                        ifelse(country=="Burkina Faso",
                               total_cases*(1000000/20903273),
                        ifelse(country=="Togo",
                               total_cases*(1000000/8278724),
                        ifelse(country=="Benin",
                               total_cases*(1000000/12123200),
                        ifelse(country=="Gambia",
                               total_cases*(1000000/2416668),
                        ifelse(country=="Mauritania",
                               total_cases*(1000000/4649658),
                        ifelse(country=="Senegal",
                               total_cases*(1000000/16743927),
                        ifelse(country=="Cote d’Ivoire",
                               total_cases*(1000000/26378274),
                        ifelse(country=="Mali",
                               total_cases*(1000000/20250833),
                        ifelse(country=="Sierra Leona",
                               total_cases*(1000000/7976983),
                               total_cases*(1000000/5057681))))))))))))))))
head(confirmed_cases_WA)
```

```
## # A tibble: 6 x 6
## # Groups:   country, type [1]
##   country date       cases type      total_cases infection_rate
##   <chr>   <date>     <int> <chr>           <int>          <dbl>
## 1 Benin   2020-01-22     0 confirmed           0              0
## 2 Benin   2020-01-23     0 confirmed           0              0
## 3 Benin   2020-01-24     0 confirmed           0              0
## 4 Benin   2020-01-25     0 confirmed           0              0
## 5 Benin   2020-01-26     0 confirmed           0              0
## 6 Benin   2020-01-27     0 confirmed           0              0
```

```
tail(confirmed_cases_WA)
```

```
## # A tibble: 6 x 6
## # Groups:   country, type [1]
##   country date       cases type      total_cases infection_rate
##   <chr>   <date>     <int> <chr>           <int>          <dbl>
## 1 Togo    2020-06-12     1 confirmed         525           63.4
## 2 Togo    2020-06-13     5 confirmed         530           64.0
## 3 Togo    2020-06-14     0 confirmed         530           64.0
## 4 Togo    2020-06-15     1 confirmed         531           64.1
## 5 Togo    2020-06-16     6 confirmed         537           64.9
## 6 Togo    2020-06-17     7 confirmed         544           65.7
```

```
str(confirmed_cases_WA)
```

```
## tibble [2,368 x 6] (S3: grouped_df/tbl_df/tbl/data.frame)
##  $ country       : chr [1:2368] "Benin" "Benin" "Benin" "Benin" ...
##  $ date          : Date[1:2368], format: "2020-01-22" "2020-01-23" ...
##  $ cases         : int [1:2368] 0 0 0 0 0 0 0 0 0 0 ...
##  $ type          : chr [1:2368] "confirmed" "confirmed" "confirmed" "confirmed" ...
##  $ total_cases   : int [1:2368] 0 0 0 0 0 0 0 0 0 0 ...
##  $ infection_rate: num [1:2368] 0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, "groups")= tibble [16 x 3] (S3: tbl_df/tbl/data.frame)
##   ..$ country: chr [1:16] "Benin" "Burkina Faso" "Cabo Verde" "Cote d'Ivoire" ...
##   ..$ type   : chr [1:16] "confirmed" "confirmed" "confirmed" "confirmed" ...
##   ..$ .rows  :List of 16
##   .. ..$ : 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 ...
##   .. ..$ : int [1:148] 1481 1482 1483 1484 1485 1486 1487 1488 1489 1490 ...
##   .. ..$ : int [1:148] 1629 1630 1631 1632 1633 1634 1635 1636 1637 1638 ...
##   .. ..$ : int [1:148] 1777 1778 1779 1780 1781 1782 1783 1784 1785 1786 ...
##   .. ..$ : int [1:148] 1925 1926 1927 1928 1929 1930 1931 1932 1933 1934 ...
##   .. ..$ : int [1:148] 2073 2074 2075 2076 2077 2078 2079 2080 2081 2082 ...
##   .. ..$ : int [1:148] 2221 2222 2223 2224 2225 2226 2227 2228 2229 2230 ...
##   ..- attr(*, ".drop")= logi TRUE
```

```
summary(confirmed_cases_WA$infection_rate)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
##    0.000    0.000    2.887   56.896   38.945 1424.494
```

```
stat.desc(confirmed_cases_WA)
```

```
##           country         date        cases type  total_cases infection_rate
## nbr.val       NA 2.368000e+03  2368.000000   NA 2.368000e+03   2.368000e+03
## nbr.null      NA 0.000000e+00  1303.000000   NA 8.470000e+02   8.470000e+02
## nbr.na        NA 0.000000e+00     0.000000   NA 0.000000e+00   0.000000e+00
## min           NA 1.828300e+04  -209.000000   NA 0.000000e+00   0.000000e+00
## max           NA 1.843000e+04   921.000000   NA 1.773500e+04   1.424494e+03
## range         NA 1.470000e+02  1130.000000   NA 1.773500e+04   1.424494e+03
## sum           NA 4.346819e+07 57656.000000   NA 1.645590e+06   1.347287e+05
## median        NA 1.835650e+04     0.000000   NA 2.250000e+01   2.887027e+00
## mean          NA 1.835650e+04    24.347973   NA 6.949282e+02   5.689556e+01
## SE.mean       NA 8.781365e-01     1.469730   NA 3.795369e+01   3.039254e+00
## CI.mean       NA 1.721997e+00     2.882091   NA 7.442593e+01   5.959876e+00
## var           NA 1.826021e+03  5115.128547   NA 3.411063e+06   2.187337e+04
## std.dev       NA 4.273197e+01    71.520127   NA 1.846906e+03   1.478965e+02
## coef.var      NA 2.327893e-03     2.937416   NA 2.657694e+00   2.599438e+00
```

9 Fatality rate

```
death_cases_WA <- WA_countries %>%
  filter(type %in% c("death", "confirmed")) %>%
  select(-total_cases) %>%
  spread(type, cases) %>%
  mutate(fatality_rate=ifelse(country=="Ghana",
                             round((cumsum(death)/cumsum(confirmed))*100,digits=2),
                       ifelse(country=="Nigeria",
                             round((cumsum(death)/cumsum(confirmed))*100,digits=2),
                       ifelse(country=="Niger",
                             round((cumsum(death)/cumsum(confirmed))*100,digits=2),
                       ifelse(country=="Cabo Verde",
                             round((cumsum(death)/cumsum(confirmed))*100,digits=2),
                       ifelse(country=="Guinea-Bissau",
                             round((cumsum(death)/cumsum(confirmed))*100,digits=2),
                       ifelse(country=="Guinea",
                             round((cumsum(death)/cumsum(confirmed))*100,digits=2),
                       ifelse(country=="Bukina Faso",
                             round((cumsum(death)/cumsum(confirmed))*100,digits=2),
                       ifelse(country=="Togo",
                             round((cumsum(death)/cumsum(confirmed))*100,digits=2),
                       ifelse(country=="Benin",
                              round((cumsum(death)/cumsum(confirmed))*100,digits=2),
                       ifelse(country=="Gambia",
                             round((cumsum(death)/cumsum(confirmed))*100,digits=2),
                       ifelse(country=="Mauritania",
                             round((cumsum(death)/cumsum(confirmed))*100,digits=2),
                       ifelse(country=="Senegal",
                             round((cumsum(death)/cumsum(confirmed))*100,digits=2),
                       ifelse(country=="Cote d’Ivoire",
                             round((cumsum(death)/cumsum(confirmed))*100,digits=2),
                       ifelse(country=="Mali",
                             round((cumsum(death)/cumsum(confirmed))*100,digits=2),
                       ifelse(country=="Sierra Leona",
                             round((cumsum(death)/cumsum(confirmed))*100,digits=2),
                             round((cumsum(death)/cumsum(confirmed))*100,
                                   digits=2)))))))))))))))))

head(death_cases_WA)
```

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

```
tail(death_cases_WA)
```

```
## # A tibble: 6 x 5
## # Groups:   country [1]
##   country date       confirmed death fatality_rate
##   <chr>   <date>         <int> <int>         <dbl>
## 1 Togo    2020-06-12         1     0          2.48
## 2 Togo    2020-06-13         5     0          2.45
## 3 Togo    2020-06-14         0     0          2.45
## 4 Togo    2020-06-15         1     0          2.45
## 5 Togo    2020-06-16         6     0          2.42
## 6 Togo    2020-06-17         7     0          2.39
```

```
str(death_cases_WA)
```

```
## tibble [2,368 x 5] (S3: grouped_df/tbl_df/tbl/data.frame)
##  $ country      : chr [1:2368] "Benin" "Benin" "Benin" "Benin" ...
##  $ date         : Date[1:2368], format: "2020-01-22" "2020-01-23" ...
##  $ confirmed    : int [1:2368] 0 0 0 0 0 0 0 0 0 0 ...
##  $ death        : int [1:2368] 0 0 0 0 0 0 0 0 0 0 ...
##  $ fatality_rate: num [1:2368] NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN ...
##  - attr(*, "groups")= tibble [16 x 2] (S3: tbl_df/tbl/data.frame)
##   ..$ country: chr [1:16] "Benin" "Burkina Faso" "Cabo Verde" "Cote d'Ivoire" ...
##   ..$ .rows  :List of 16
##   .. ..$ : 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 ...
##   .. ..$ : int [1:148] 1481 1482 1483 1484 1485 1486 1487 1488 1489 1490 ...
##   .. ..$ : int [1:148] 1629 1630 1631 1632 1633 1634 1635 1636 1637 1638 ...
##   .. ..$ : int [1:148] 1777 1778 1779 1780 1781 1782 1783 1784 1785 1786 ...
##   .. ..$ : int [1:148] 1925 1926 1927 1928 1929 1930 1931 1932 1933 1934 ...
##   .. ..$ : int [1:148] 2073 2074 2075 2076 2077 2078 2079 2080 2081 2082 ...
##   .. ..$ : int [1:148] 2221 2222 2223 2224 2225 2226 2227 2228 2229 2230 ...
##   ..- attr(*, ".drop")= logi TRUE
```

```
stat.desc(death_cases_WA)
```

```
##           country         date    confirmed        death fatality_rate
## nbr.val       NA 2.368000e+03  2368.000000 2.368000e+03  1521.0000000
## nbr.null      NA 0.000000e+00  1303.000000 1.943000e+03   271.0000000
## nbr.na        NA 0.000000e+00     0.000000 0.000000e+00   847.0000000
## min           NA 1.828300e+04  -209.000000 0.000000e+00     0.0000000
## max           NA 1.843000e+04   921.000000 3.100000e+01    50.0000000
## range         NA 1.470000e+02  1130.000000 3.100000e+01    50.0000000
## sum           NA 4.346819e+07 57656.000000 1.133000e+03  5882.4100000
## median        NA 1.835650e+04     0.000000 0.000000e+00     2.2700000
## mean          NA 1.835650e+04    24.347973 4.784628e-01     3.8674622
## SE.mean       NA 8.781365e-01     1.469730 3.407720e-02     0.1259503
## CI.mean       NA 1.721997e+00     2.882091 6.682425e-02     0.2470548
## var           NA 1.826021e+03  5115.128547 2.749853e+00    24.1283630
## std.dev       NA 4.273197e+01    71.520127 1.658268e+00     4.9120630
## coef.var      NA 2.327893e-03     2.937416 3.465824e+00     1.2700998
```

10 Recovery rate

```
recovered_cases_WA <- WA_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=="Nigeria",
                           round((cumsum(recovered)/cumsum(confirmed))*100,digits=2),
                       ifelse(country=="Niger",
                           round((cumsum(recovered)/cumsum(confirmed))*100,digits=2),
                       ifelse(country=="Cabo Verde",
                           round((cumsum(recovered)/cumsum(confirmed))*100,digits=2),
                       ifelse(country=="Guinea-Bissau",
                           round((cumsum(recovered)/cumsum(confirmed))*100,digits=2),
                       ifelse(country=="Guinea",
                             round((cumsum(recovered)/cumsum(confirmed))*100,digits=2),
                       ifelse(country=="Bukina Faso",
                             round((cumsum(recovered)/cumsum(confirmed))*100,digits=2),
                       ifelse(country=="Togo",
                             round((cumsum(recovered)/cumsum(confirmed))*100,digits=2),
                       ifelse(country=="Benin",
                              round((cumsum(recovered)/cumsum(confirmed))*100,digits=2),
                       ifelse(country=="Gambia",
                             round((cumsum(recovered)/cumsum(confirmed))*100,digits=2),
                       ifelse(country=="Mauritania",
                             round((cumsum(recovered)/cumsum(confirmed))*100,digits=2),
                       ifelse(country=="Senegal",
                             round((cumsum(recovered)/cumsum(confirmed))*100,digits=2),
                       ifelse(country=="Cote d’Ivoire",
                             round((cumsum(recovered)/cumsum(confirmed))*100,digits=2),
                       ifelse(country=="Mali",
                             round((cumsum(recovered)/cumsum(confirmed))*100,digits=2),
                       ifelse(country=="Sierra Leona",
                             round((cumsum(recovered)/cumsum(confirmed))*100,digits=2),
                             round((cumsum(recovered)/cumsum(confirmed))*100,
                                   digits=2)))))))))))))))))

head(recovered_cases_WA)
```

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

```r
tail(recovered_cases_WA)
```

```
## # A tibble: 6 x 5
## # Groups:   country [1]
##   country date       confirmed recovered recovery_rate
##   <chr>   <date>         <int>     <int>         <dbl>
## 1 Togo    2020-06-12         1         8          53.1
## 2 Togo    2020-06-13         5        12          54.9
## 3 Togo    2020-06-14         0         0          54.9
## 4 Togo    2020-06-15         1         8          56.3
## 5 Togo    2020-06-16         6        45          64.1
## 6 Togo    2020-06-17         7         9          64.9
```

```r
str(recovered_cases_WA)
```

```
## tibble [2,368 x 5] (S3: grouped_df/tbl_df/tbl/data.frame)
##  $ country      : chr [1:2368] "Benin" "Benin" "Benin" "Benin" ...
##  $ date         : Date[1:2368], format: "2020-01-22" "2020-01-23" ...
##  $ confirmed    : int [1:2368] 0 0 0 0 0 0 0 0 0 0 ...
##  $ recovered    : int [1:2368] 0 0 0 0 0 0 0 0 0 0 ...
##  $ recovery_rate: num [1:2368] NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN ...
##  - attr(*, "groups")= tibble [16 x 2] (S3: tbl_df/tbl/data.frame)
##   ..$ country: chr [1:16] "Benin" "Burkina Faso" "Cabo Verde" "Cote d’Ivoire" ...
##   ..$ .rows  :List of 16
##   .. ..$ : 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 ...
##   .. ..$ : int [1:148] 1481 1482 1483 1484 1485 1486 1487 1488 1489 1490 ...
##   .. ..$ : int [1:148] 1629 1630 1631 1632 1633 1634 1635 1636 1637 1638 ...
##   .. ..$ : int [1:148] 1777 1778 1779 1780 1781 1782 1783 1784 1785 1786 ...
##   .. ..$ : int [1:148] 1925 1926 1927 1928 1929 1930 1931 1932 1933 1934 ...
##   .. ..$ : int [1:148] 2073 2074 2075 2076 2077 2078 2079 2080 2081 2082 ...
##   .. ..$ : int [1:148] 2221 2222 2223 2224 2225 2226 2227 2228 2229 2230 ...
##   ..- attr(*, ".drop")= logi TRUE
```

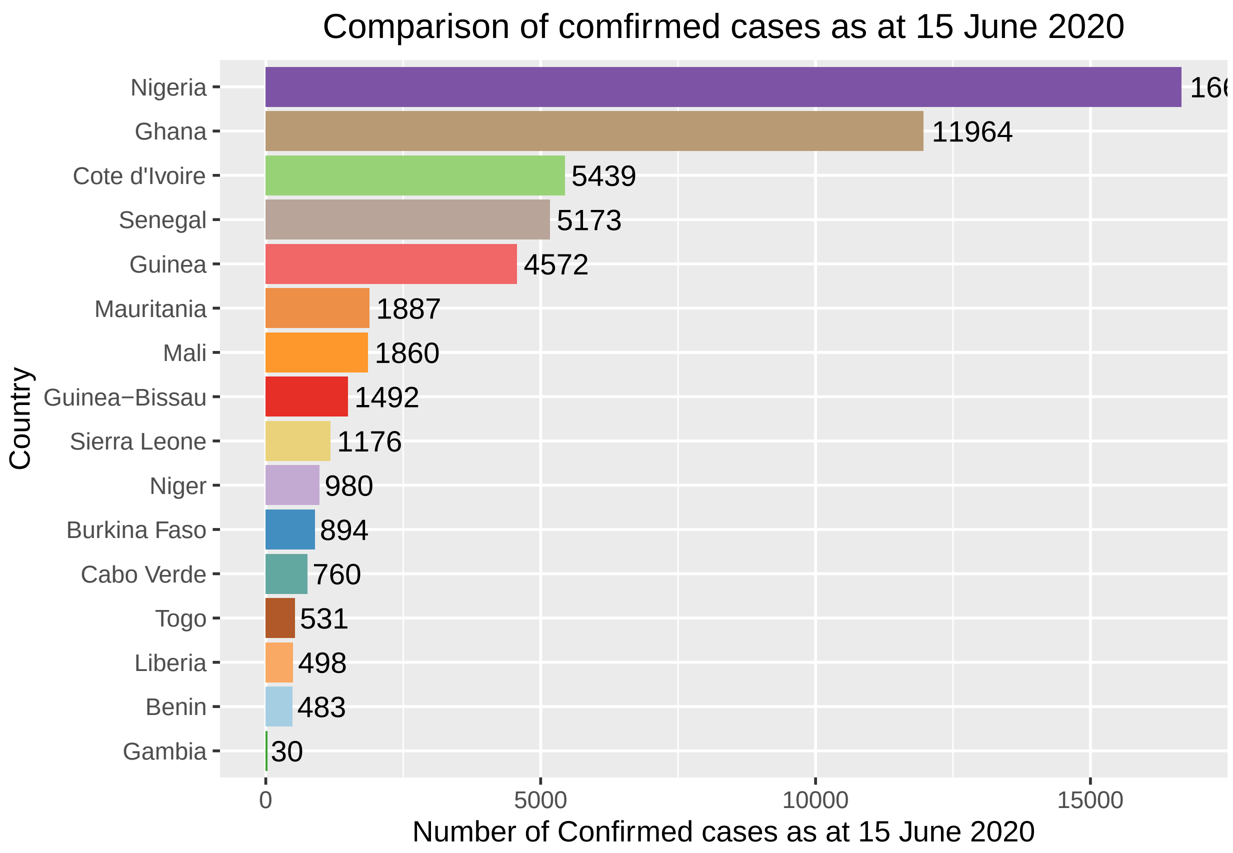
```r
stat.desc(recovered_cases_WA)
```

```
##           country         date     confirmed     recovered recovery_rate
## nbr.val       NA 2.368000e+03  2368.000000  2368.0000000  1.521000e+03
## nbr.null      NA 0.000000e+00  1303.000000  1572.0000000  2.400000e+02
## nbr.na        NA 0.000000e+00     0.000000     0.0000000  8.470000e+02
## min           NA 1.828300e+04  -209.000000   -31.0000000  0.000000e+00
## max           NA 1.843000e+04   921.000000   786.0000000  9.031000e+01
## range         NA 1.470000e+02  1130.000000   817.0000000  9.031000e+01
## sum           NA 4.346819e+07 57656.000000 25484.0000000  4.829217e+04
## median        NA 1.835650e+04     0.000000     0.0000000  2.879000e+01
## mean          NA 1.835650e+04    24.347973    10.7618243  3.175028e+01
## SE.mean       NA 8.781365e-01     1.469730     0.7331091  6.705149e-01
## CI.mean       NA 1.721997e+00     2.882091     1.4376026  1.315232e+00
## var           NA 1.826021e+03  5115.128547  1272.6792011  6.838267e+02
## std.dev       NA 4.273197e+01    71.520127    35.6746297  2.615008e+01
## coef.var      NA 2.327893e-03     2.937416     3.3149240  8.236174e-01
```

11 Plot a bar chart

```
nb.colors <- 16
mycolors <- colorRampPalette(brewer.pal(12, "Paired"))(nb.colors)

WA_countries %>%
  filter(country %in% c("Ghana","Nigeria","Niger","Guinea-Bissau","Guinea",
                        "Cabo Verde","Burkina Faso", "Togo", "Benin","Gambia",
                        "Mauritania","Senegal","Cote d'Ivoire",
                        "Sierra Leone","Mali","Liberia"), 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_manual(values = mycolors) +
  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")
```

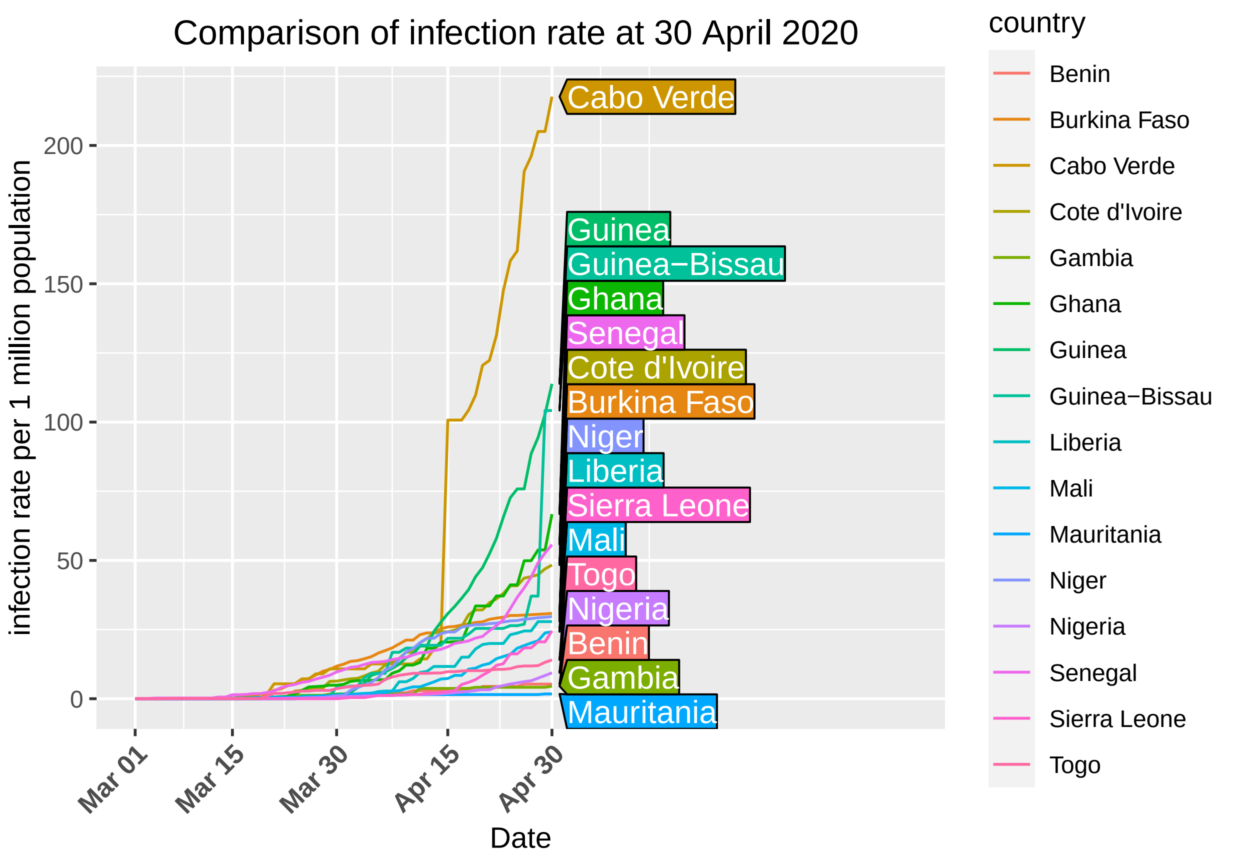


```
ggsave("confirmed_WA1.png",
      width =30, height = 15, units = "cm", dpi = 70)
```

12 Plot a line graph as at 30 April

```
confirmed_cases_WA %>%
  filter(country %in% c("Ghana","Nigeria","Niger","Guinea","Guinea-Bissau",
                        "Cabo Verde","Burkina Faso", "Togo", "Benin","Gambia",
                        "Mauritania","Senegal","Cote d'Ivoire",
                        "Sierra Leone","Mali","Liberia"),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_manual(values = mycolors) +
  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 624 row(s) containing missing values (geom_path).
```



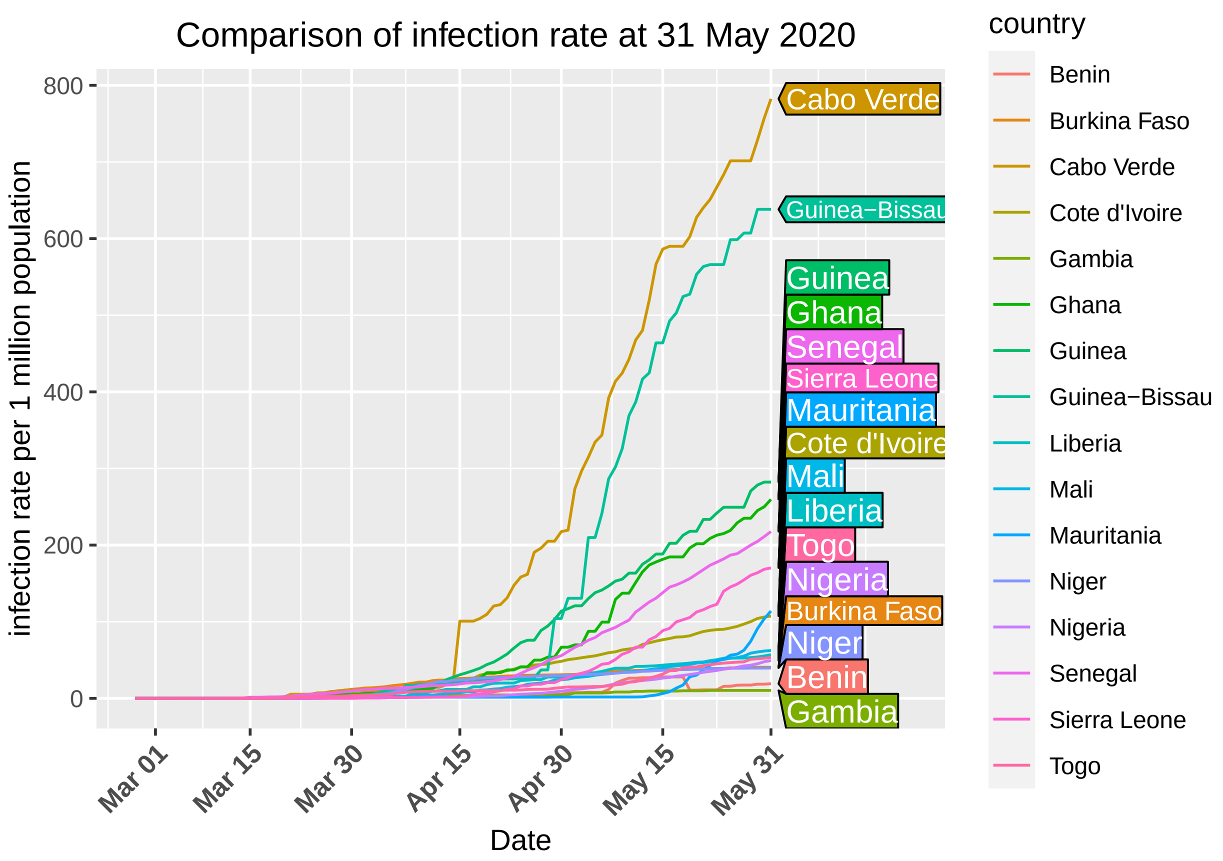
```
ggsave("confirmed_WA2.png", #other supported files;- "eps", "ps", "tex"
width =30, height = 15,
units = "cm", # other options c("in", "cm", "mm")
dpi = 70)
```

```
## Warning: Removed 624 row(s) containing missing values (geom_path).
```

13 Plot a line graph as at 31 May

```
confirmed_cases_WA %>%
  filter(country %in% c("Ghana","Nigeria","Niger","Guinea","Guinea-Bissau",
                        "Cabo Verde","Burkina Faso", "Togo", "Benin","Gambia",
                        "Mauritania","Senegal","Cote d'Ivoire",
                        "Sierra Leone","Mali","Liberia"),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_manual(values = mycolors) +
  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 576 row(s) containing missing values (geom_path).
```



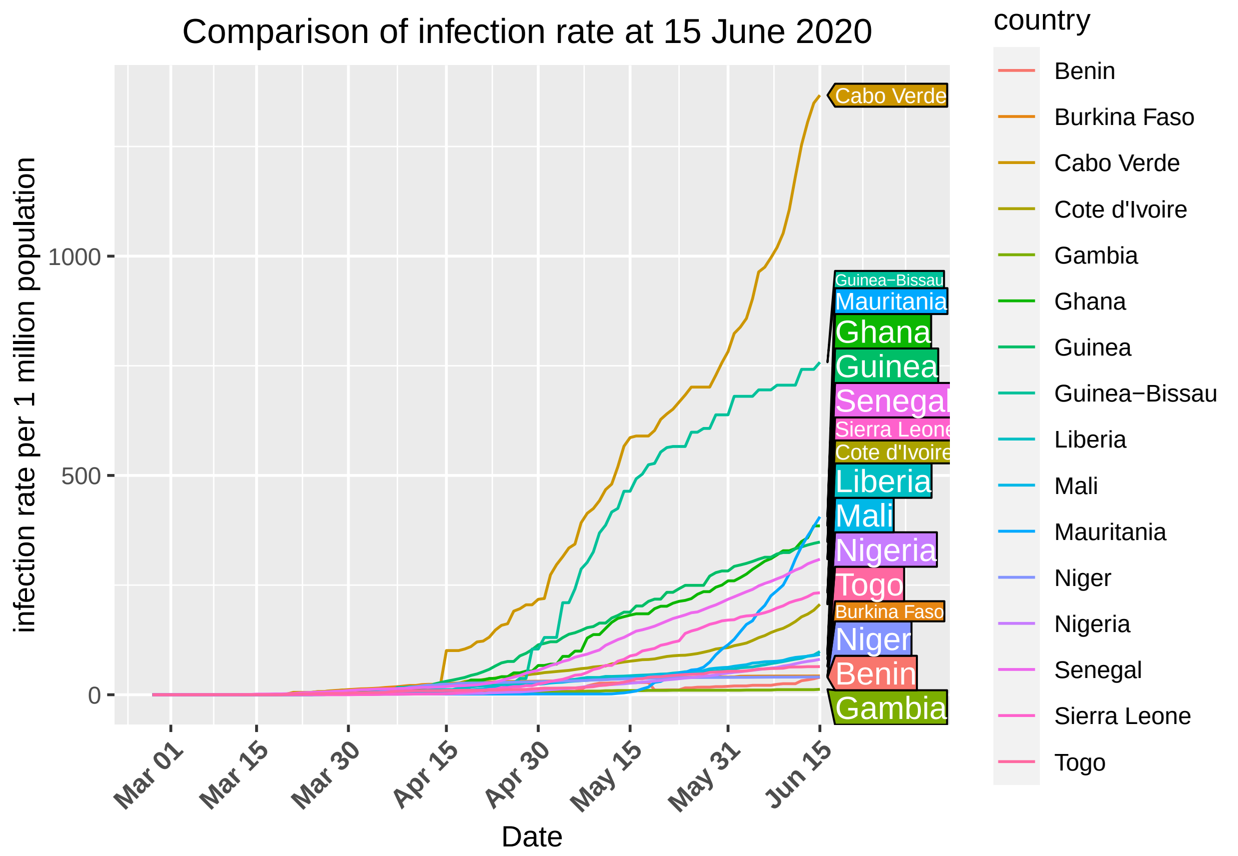
```
ggsave("confirmed_WA3.png", #other supported files;- "eps", "ps", "tex"
width =30, height = 15,
units = "cm", # other options c("in", "cm", "mm")
dpi = 70)
```

```
## Warning: Removed 576 row(s) containing missing values (geom_path).
```

14 Plot a line graph as at 15 June

```
confirmed_cases_WA %>%
  filter(country %in% c("Ghana","Nigeria","Niger","Guinea","Guinea-Bissau",
                        "Cabo Verde","Burkina Faso", "Togo", "Benin","Gambia",
                        "Mauritania","Senegal","Cote d'Ivoire",
                        "Sierra Leone","Mali","Liberia"),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_manual(values = mycolors) +
  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 576 row(s) containing missing values (geom_path).
```

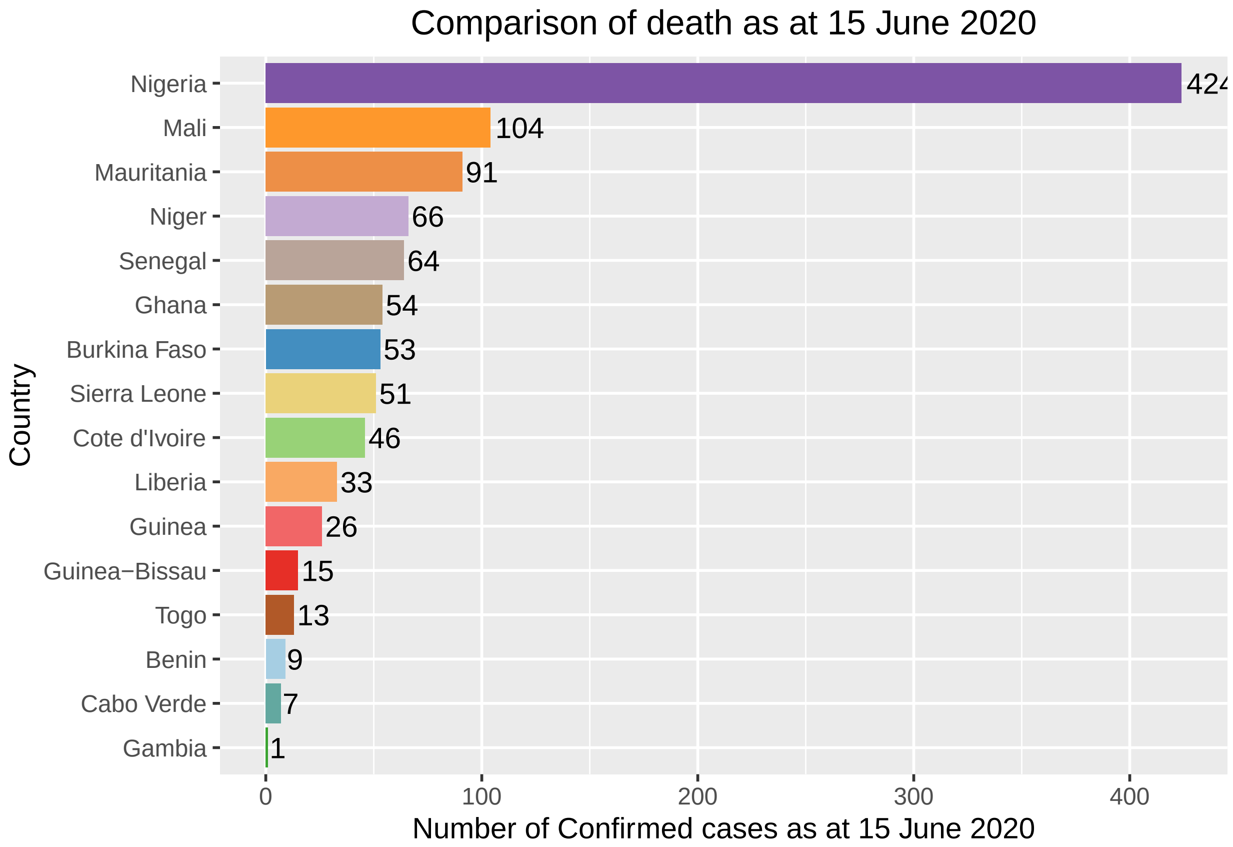


```
ggsave("confirmed_WA4.png", #other supported files;- "eps", "ps", "tex"
width =30, height = 15,
units = "cm", # other options c("in", "cm", "mm")
dpi = 70)
```

```
## Warning: Removed 576 row(s) containing missing values (geom_path).
```

15 Plot a bar graph for the deaths

```
WA_countries %>%
  filter(country %in% c("Ghana","Nigeria","Niger","Guinea-Bissau","Guinea",
                        "Cabo Verde","Burkina Faso", "Togo", "Benin","Gambia",
                        "Mauritania","Senegal","Cote d’Ivoire",
                        "Sierra Leone","Mali","Liberia"), 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_manual(values = mycolors) +
  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")
```

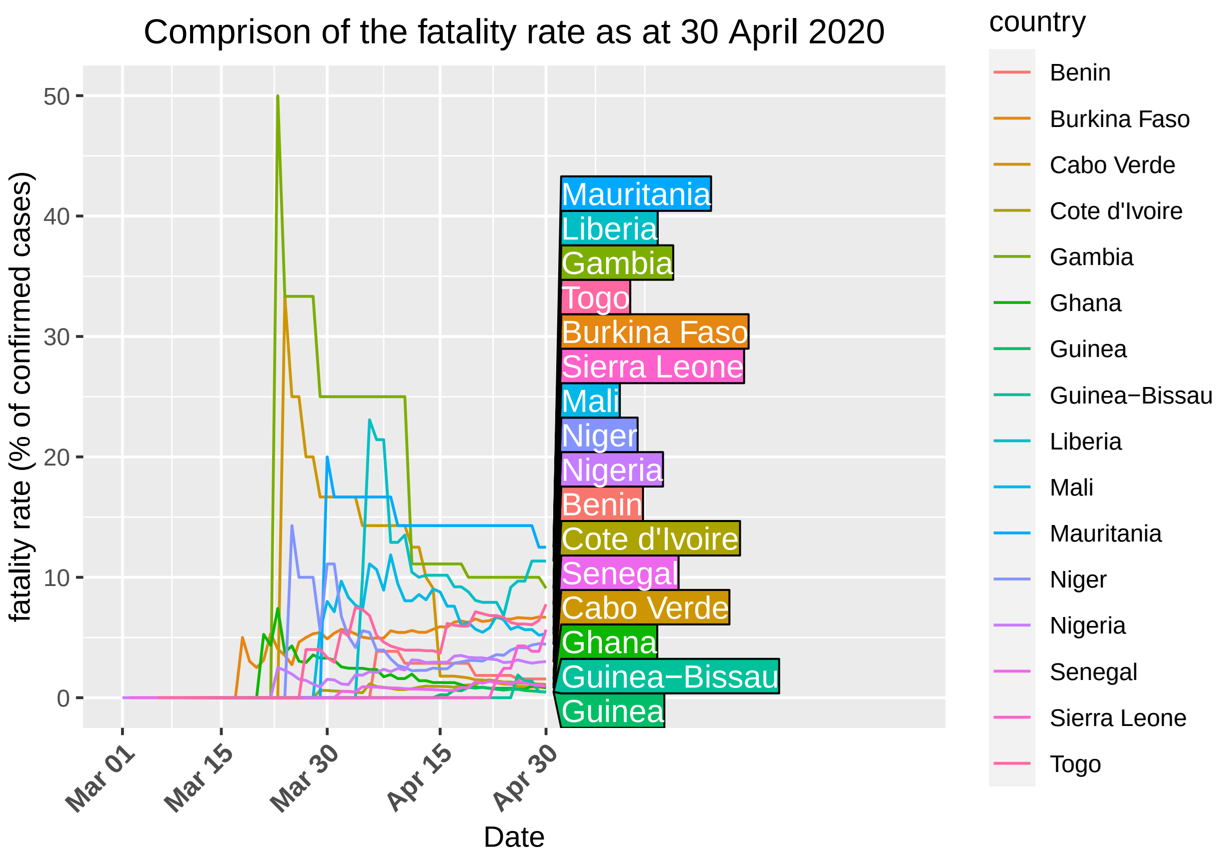


```
ggsave("death_WA1.png",
      width =30, height = 15, units = "cm", dpi = 70)
```

16 Plot line graph for fatality rate as at 30 April

```
death_cases_WA %>%
  filter(country %in% c("Ghana","Nigeria","Niger","Guinea-Bissau","Guinea",
                        "Cabo Verde","Burkina Faso", "Togo", "Benin","Gambia",
                        "Mauritania","Senegal","Cote d'Ivoire",
                        "Sierra Leone","Mali","Liberia"),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_manual(values = mycolors)  +
  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_WA$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 849 row(s) containing missing values (geom_path).
```



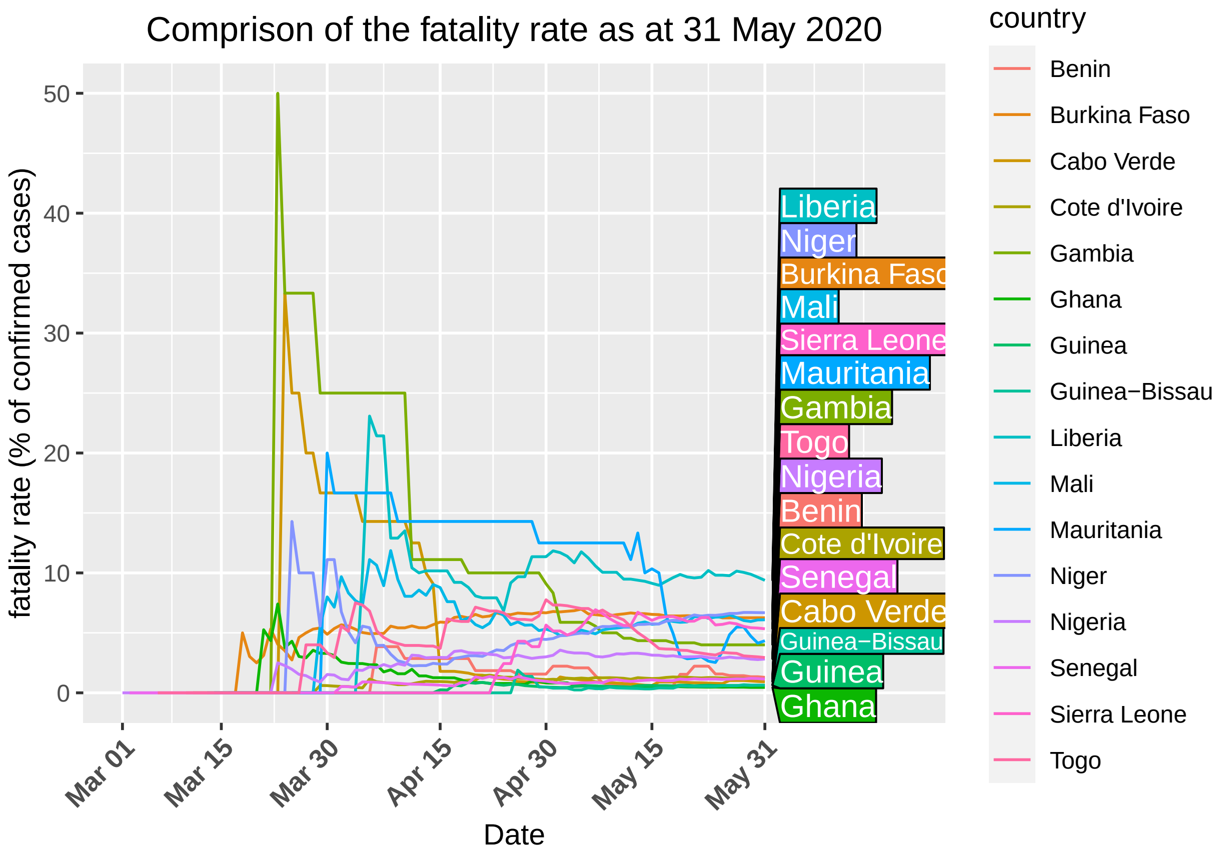
```
ggsave("death_WA2.png",
       width =30, height = 15,units = "cm",dpi = 70)
```

```
## Warning: Removed 849 row(s) containing missing values (geom_path).
```

17 Plot line graph for fatality rate as at 31 May

```
death_cases_WA %>%
  filter(country %in% c("Ghana","Nigeria","Niger","Guinea-Bissau","Guinea",
                        "Cabo Verde","Burkina Faso", "Togo", "Benin","Gambia",
                        "Mauritania","Senegal","Cote d’Ivoire",
                        "Sierra Leone","Mali","Liberia"),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_manual(values = mycolors)  +
  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_WA$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 849 row(s) containing missing values (geom_path).
```



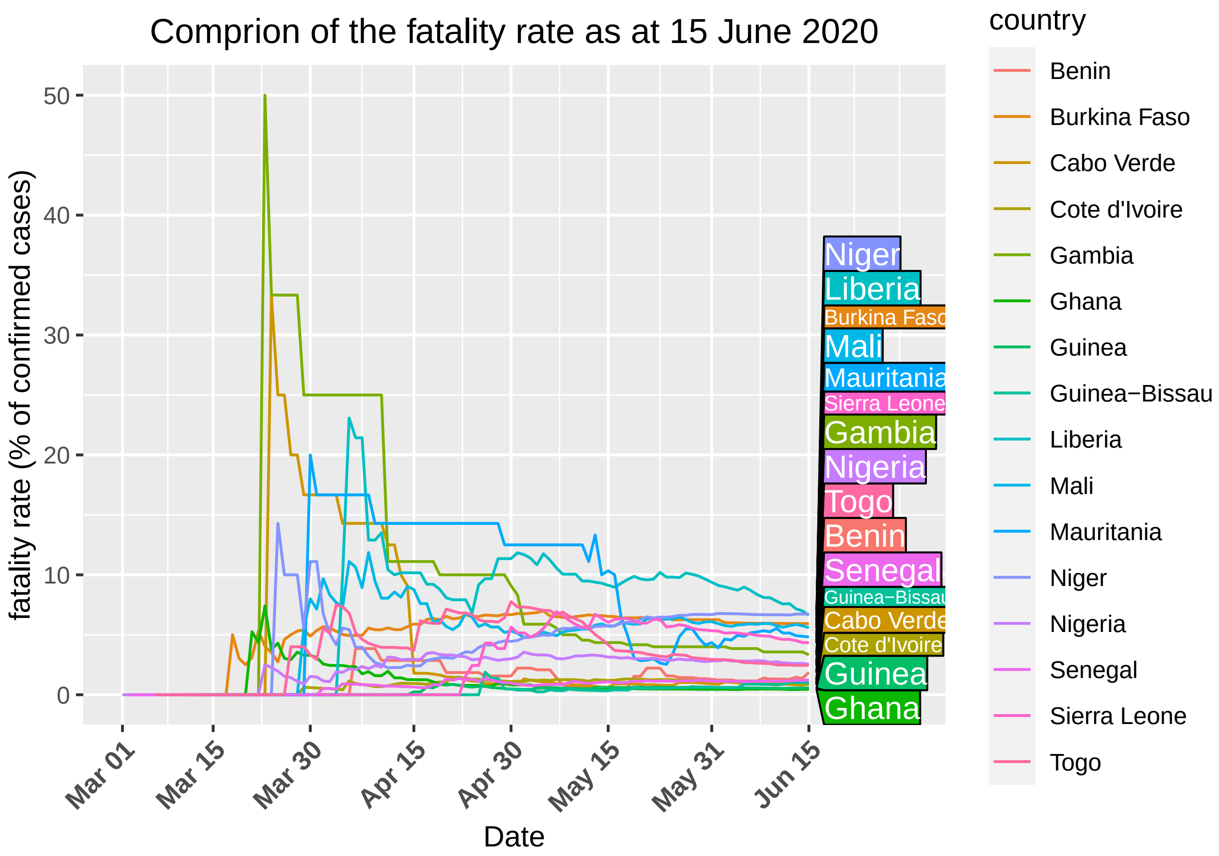
```
ggsave("death_WA3.png",
       width =30, height = 15,units = "cm",dpi = 70)
```

```
## Warning: Removed 849 row(s) containing missing values (geom_path).
```

18 Plot line graph for fatality rate as at 15 June

```
death_cases_WA %>%
  filter(country %in% c("Ghana","Nigeria","Niger","Guinea-Bissau","Guinea",
                        "Cabo Verde","Burkina Faso", "Togo", "Benin","Gambia",
                        "Mauritania","Senegal","Cote d'Ivoire",
                        "Sierra Leone","Mali","Liberia"),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_manual(values = mycolors)  +
  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_WA$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 849 row(s) containing missing values (geom_path).
```

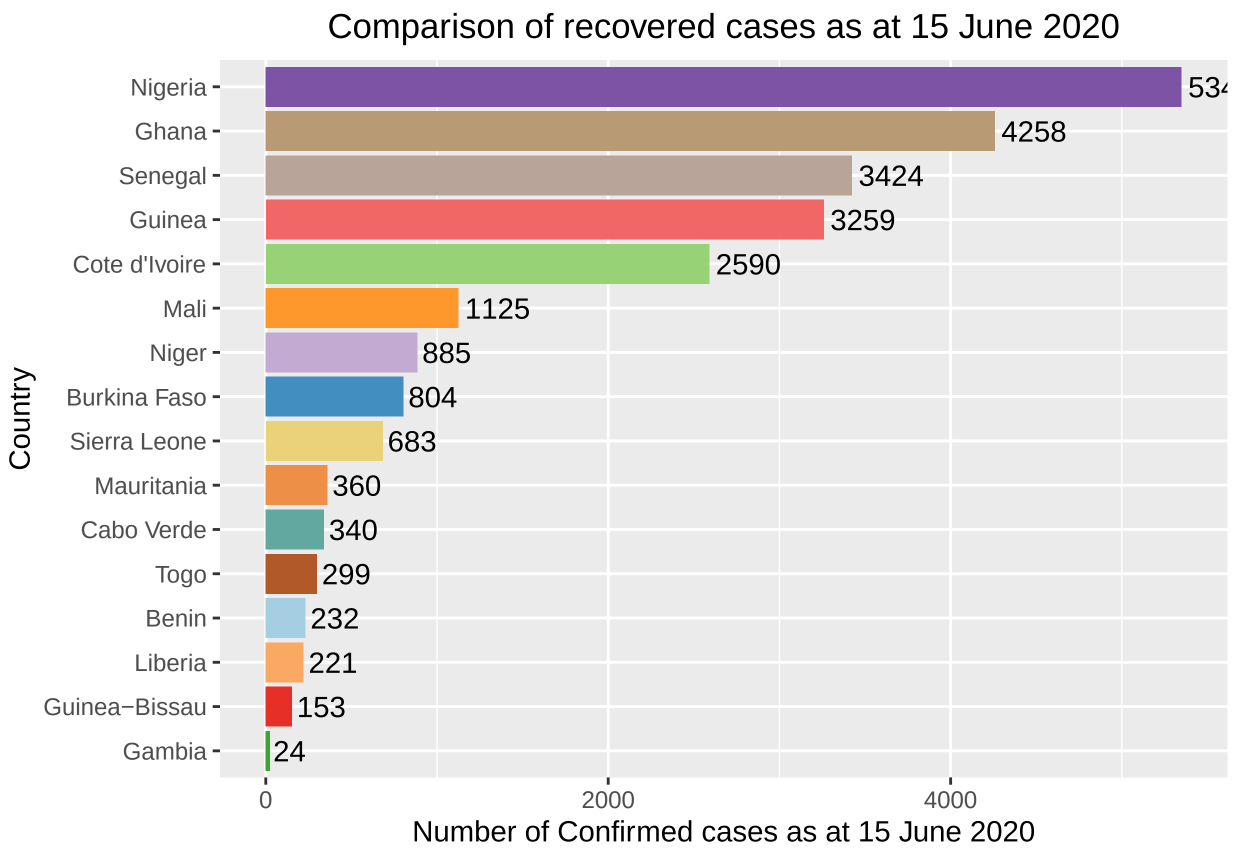


```
ggsave("death_WA4.png",
       width =30, height = 15,units = "cm",dpi = 70)
```

```
## Warning: Removed 849 row(s) containing missing values (geom_path).
```

19 Plot the bar chart for recoveries

```
WA_countries %>%
  filter(country %in% c("Ghana","Nigeria","Niger","Guinea-Bissau","Guinea",
                        "Cabo Verde","Burkina Faso", "Togo", "Benin","Gambia",
                        "Mauritania","Senegal","Cote d’Ivoire",
                        "Sierra Leone","Mali","Liberia"), 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_manual(values = mycolors) +
  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")
```

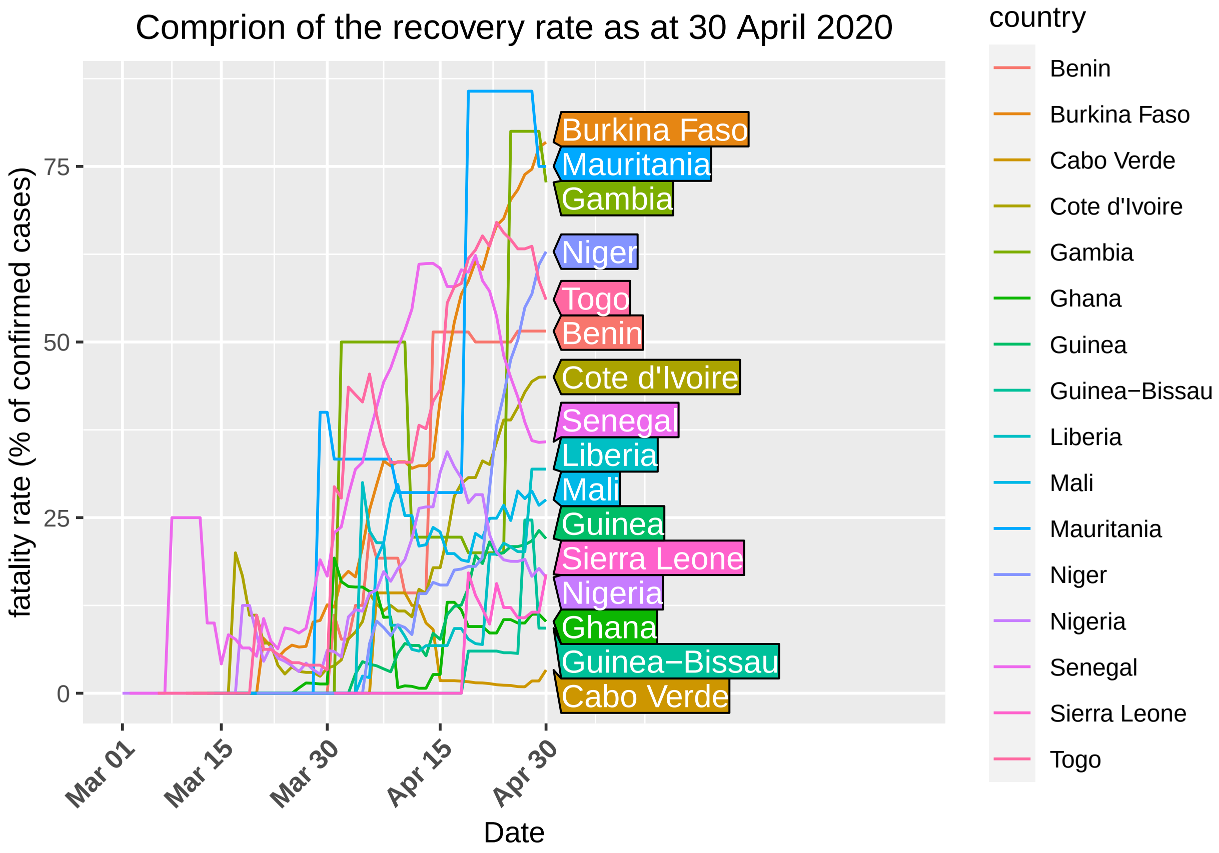


```
 ggsave("recovered_WA1.png",
       width =30, height = 15, units = "cm", dpi = 70)
```

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

```
recovered_cases_WA %>%
  filter(country %in% c("Ghana","Nigeria","Niger","Guinea-Bissau","Guinea",
                        "Cabo Verde","Burkina Faso", "Togo", "Benin","Gambia",
                        "Mauritania","Senegal","Cote d’Ivoire",
                        "Sierra Leone","Mali","Liberia"),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_manual(values = mycolors)  +
  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_WA$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 849 row(s) containing missing values (geom_path).
```



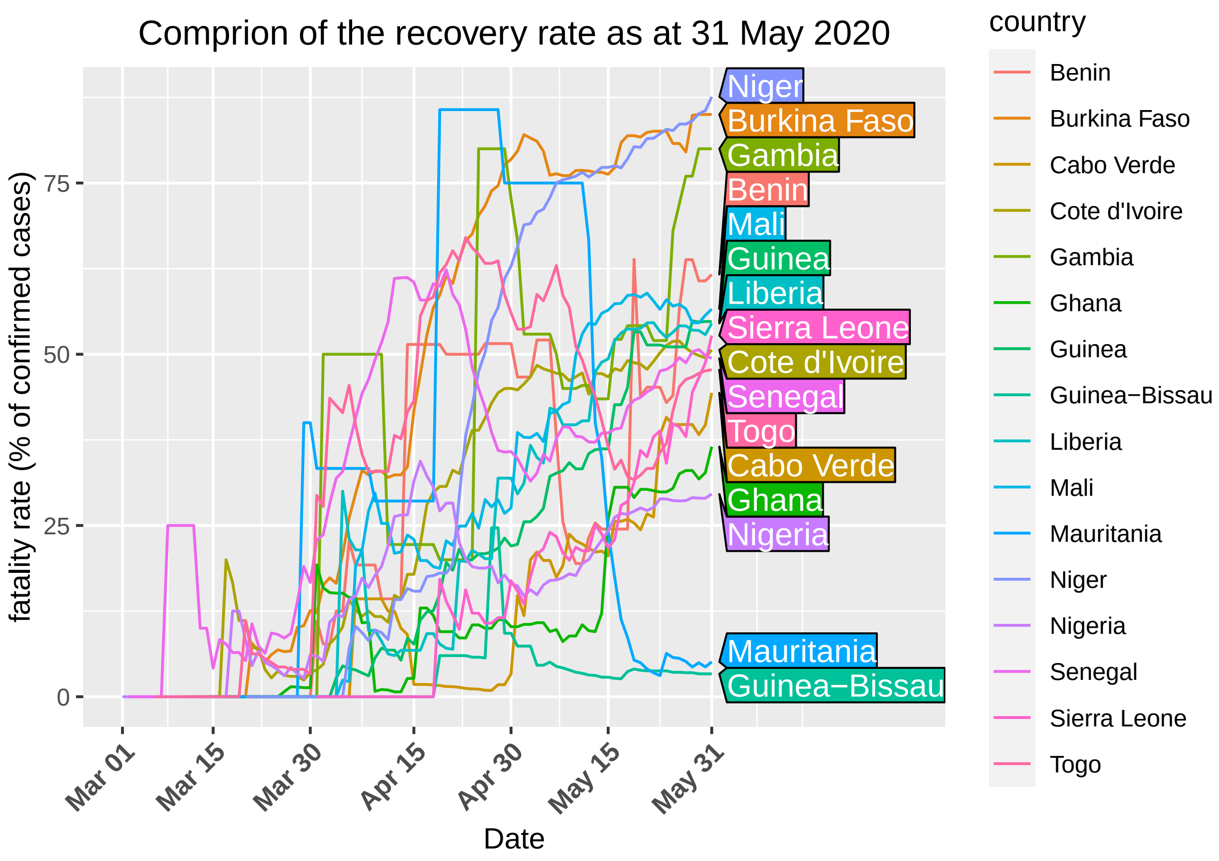
```
ggsave("recovered_WA2.png",
       width =30, height = 15,units = "cm",dpi = 70)
```

```
## Warning: Removed 849 row(s) containing missing values (geom_path).
```

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

```
recovered_cases_WA %>%
  filter(country %in% c("Ghana","Nigeria","Niger","Guinea-Bissau","Guinea",
                        "Cabo Verde","Burkina Faso", "Togo", "Benin","Gambia",
                        "Mauritania","Senegal","Cote d’Ivoire",
                        "Sierra Leone","Mali","Liberia"),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_manual(values = mycolors)  +
  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_WA$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 849 row(s) containing missing values (geom_path).
```



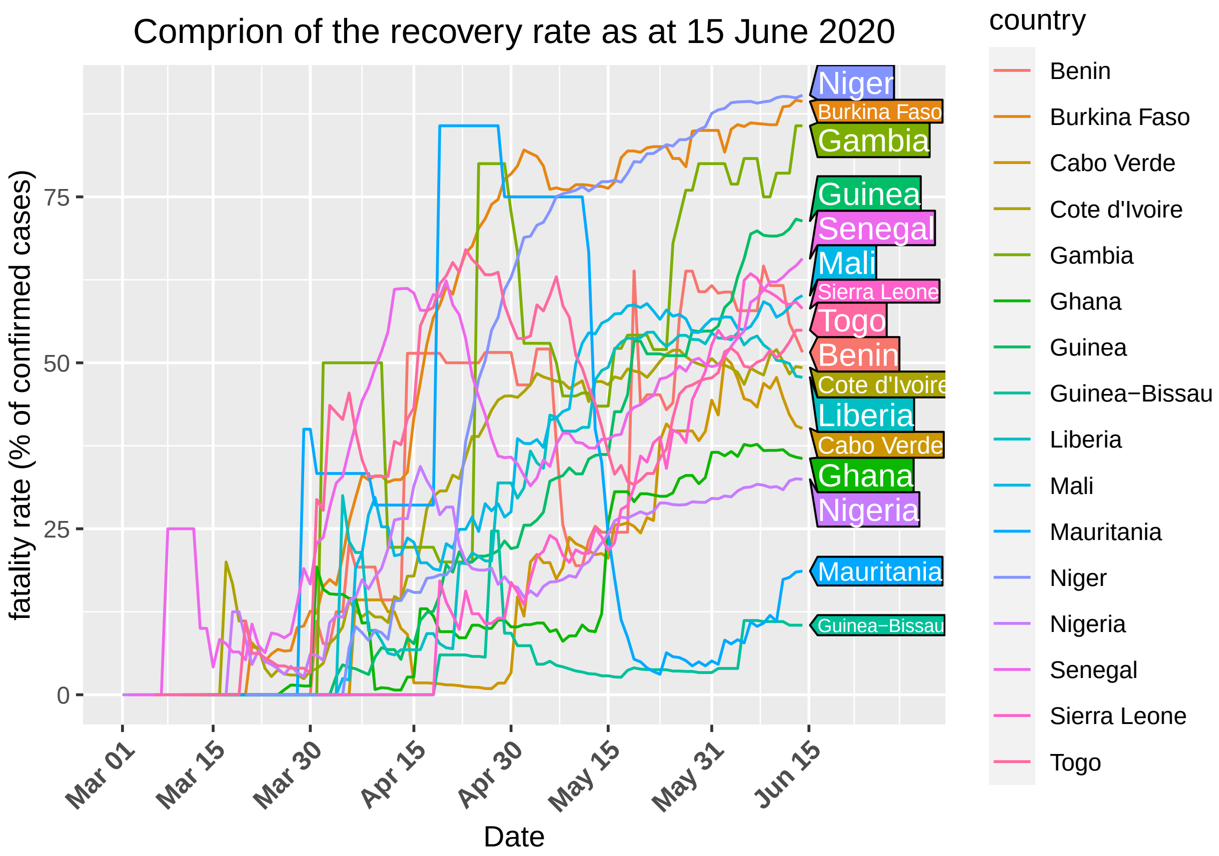
```
ggsave("recovered_WA3.png",
       width =30, height = 15,units = "cm",dpi = 70)
```

```
## Warning: Removed 849 row(s) containing missing values (geom_path).
```

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

```
recovered_cases_WA %>%
  filter(country %in% c("Ghana","Nigeria","Niger","Guinea-Bissau","Guinea",
                        "Cabo Verde","Burkina Faso", "Togo", "Benin","Gambia",
                        "Mauritania","Senegal","Cote d’Ivoire",
                        "Sierra Leone","Mali","Liberia"),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_manual(values = mycolors)  +
  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_WA$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 849 row(s) containing missing values (geom_path).
```



```
ggsave("recovered_WA4.png",
       width =30, height = 15,units = "cm",dpi = 70)
```

```
## Warning: Removed 849 row(s) containing missing values (geom_path).
```

```
writexl::write_xlsx(x = confirmed_cases_WA, path = "C:/Users/uganda/Documents/COVID-19/COVID-19_Data/co

write.csv(confirmed_cases_WA, "C:/Users/uganda/Documents/COVID-19/COVID-19_Data/confirmed_cases_WA.csv"

writexl::write_xlsx(x = recovered_cases_WA, path = "C:/Users/uganda/Documents/COVID-19/COVID-19_Data/re

write.csv(recovered_cases_WA, "C:/Users/uganda/Documents/COVID-19/COVID-19_Data/recovered_cases_WA.csv"


writexl::write_xlsx(x = death_cases_WA, path = "C:/Users/uganda/Documents/COVID-19/COVID-19_Data/death_
                    col_names = TRUE)

write.csv(death_cases_WA, "C:/Users/uganda/Documents/COVID-19/COVID-19_Data/death_cases_WA.csv",
          row.names = FALSE)
```