

# Explore FMD data

2024-04-10

**Description:** This script uses some popular R libraries to explore and visualise a Foot and Mouth disease dataset

Begin by importing the relevant packages

```
library(readxl)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

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

```
library(ggplot2)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##   combine
```

```
library(scales)
```

Load and explore dataset

```
#Load data
df <- read_excel("C:/Users/column/Desktop/For me/Data projects/WOAH/clean_data.xlsx")

## New names:
## * ' ' -> '...1'
```

```
head(df)
```

```
## # A tibble: 6 x 16
##   ...1 Year Semester   'World region' Country 'Administrative Division'
##   <dbl> <dbl> <chr>         <chr>         <chr>      <chr>
## 1     0  2005 Jul-Dec 2005 Africa          Benin    Abomey
## 2     1  2005 Jul-Dec 2005 Africa          Benin    Banikoara
## 3     2  2005 Jul-Dec 2005 Africa          Benin    Come
## 4     3  2005 Jul-Dec 2005 Africa          Benin    Djougou
## 5     4  2005 Jul-Dec 2005 Africa          Benin    Gogounou
## 6     5  2005 Jul-Dec 2005 Africa          Benin    Kandi
## # i 10 more variables: 'Animal Category' <chr>, Species <chr>,
## #   'New outbreaks' <dbl>, Susceptible <dbl>, 'Measuring units' <chr>,
## #   Cases <dbl>, 'Killed and disposed of' <dbl>, Slaughtered <dbl>,
## #   Deaths <dbl>, Vaccinated <dbl>
```

Group data by region for descriptive figures

```
df_groupby_region<-group_by(df, `World region`)

df_region_descriptives <- summarise(df_groupby_region, Cases = sum(Cases),
                                   `New outbreaks` = sum(`New outbreaks`),
                                   `Countries` = length(unique(Country)))
```

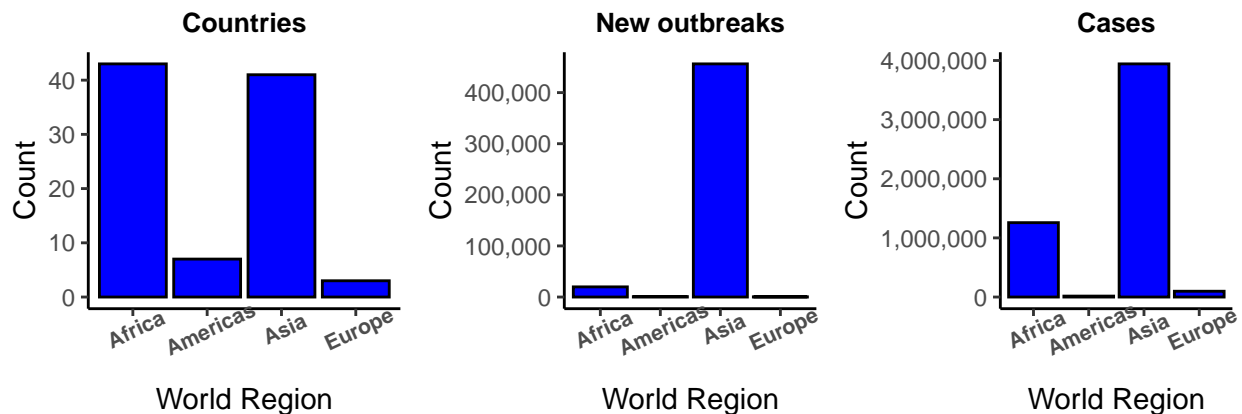
Create Barchart subplots

```
bar_vars <- c('Countries','New outbreaks','Cases')
plt_cnt <-1
subplots <- list()
for (ivar in bar_vars){
  plt <- ggplot(df_region_descriptives, aes(`World region`, .data[[ivar]]),
               height = 0.5, width = 1) +
    geom_bar(stat = "identity", colour = "black", fill = "Blue") +
    labs(x = "World Region", y = "Count", title = ivar)+
    scale_y_continuous(labels = comma)+
    theme_classic()

  plt <- plt + theme(plot.title = element_text(hjust = 0.5, size = 10, face = "bold"),
                    axis.text.x = element_text(face = "bold", size = 8,angle = 25))

  subplots[[plt_cnt]] <- plt
  plt_cnt<-plt_cnt+1
}

# Arrange plots into a grid. Note that the extra row is simply for height aesthetics
grid.arrange(grobs = subplots, nrow =2, ncol = 3)
```



Group by variables of interest and year for time-series plots

```
df_groupby_yr <- group_by(df, `World region`, `Year`)
VOI <- c('New outbreaks', 'Susceptible', 'Cases', 'Killed and disposed of',
         'Slaughtered', 'Deaths', 'Vaccinated')

df_yr_region_descriptives <- summarise_at(df_groupby_yr,
                                         VOI,
                                         sum)
```

Create time-series plots

```
subplots <- list()
ivar <- c()
plt <- c()
plt_cnt <- 1
for (ivar in VOI){
  plt <- ggplot(df_yr_region_descriptives, aes(.data[["Year"]],
                                              .data[[ivar]],
                                              color = .data[["World region"]])) +

  geom_line() +
  labs(x = 'Year', y = 'Count', title = ivar) +
```

```

scale_color_manual(values = c("Africa" = "Blue", "Americas" = "Orange",
                              "Asia" = "Green", "Europe" = "Red"))+
scale_y_continuous(labels = comma,
                   n.breaks = 4)+
theme_classic()

plt <- plt + theme(plot.title = element_text(hjust = 0.5, size = 10, face = "bold"),
                  legend.title = element_blank(),
                  legend.text = element_text(size = 6, face = "bold"),
                  legend.key.height = unit(0.5, "mm"),
                  legend.key.width = unit(2, "mm"),
                  legend.key.spacing.y = unit(0.2, "mm"))

subplots[[plt_cnt]] <- plt

plt_cnt <- plt_cnt + 1
}

grid.arrange(grobs = subplots, nrow = 4, ncol = 2)

```

