GlobalMpoxDataAnalysis&Insights

Elvira Khwatenge

2024-09-30

```
# Load the dataset
mpox_data <- read.csv("C:\\Users\\elvir\\OneDrive\\Desktop\\RProjects\\mpox\\owid-monkeypox-data.csv\\o
```

```
# Explore the dataset by viewing the first few rows
str(mpox_data)
```

```
## 'data.frame':    33666 obs. of  15 variables:
##  $ location                      : chr  "Africa" "Africa" "Africa" "Africa" ...
##  $ iso_code                      : chr  "OWID_AFR" "OWID_AFR" "OWID_AFR" "OWID_AFR" ...
##  $ date                          : chr  "2022-05-01" "2022-05-02" "2022-05-03" "2022-05-04" ...
##  $ total_cases                   : num  27 27 27 27 27 27 27 27 27 27 ...
##  $ total_deaths                  : num  2 2 2 2 2 2 2 2 2 2 ...
##  $ new_cases                     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ new_deaths                    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ new_cases_smoothed            : num  0.29 0.29 0.29 0.29 0.29 0 0 0 0 0 ...
##  $ new_deaths_smoothed           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ new_cases_per_million         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ total_cases_per_million       : num  0.019 0.019 0.019 0.019 0.019 0.019 0.019 0.019 0.019 0.019
##  $ new_cases_smoothed_per_million : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ new_deaths_per_million        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ total_deaths_per_million      : num  0.0014 0.0014 0.0014 0.0014 0.0014 0.0014 0.0014 0.0014 0.0
##  $ new_deaths_smoothed_per_million: num  0 0 0 0 0 0 0 0 0 0 ...
```

```
# Set the CRAN mirror
options(repos = c(CRAN = "https://cran.rstudio.com"))

# Now install the package
install.packages("dplyr")
```

```
## Installing package into 'C:/Users/elvir/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'dplyr' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\elvir\AppData\Local\Temp\RtmpKoeg6e\downloaded_packages
```

```
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
```

```
# Filter and select relevant data (location, date, total_cases, total_deaths, new_cases, new_deaths)
selected_data <- mpox_data %>%
  select(location, date, total_cases, total_deaths, new_cases, new_deaths)
```

```
# Aggregate data by country and calculate summary statistics
aggregated_data <- selected_data %>%
  group_by(location) %>%
  summarise(
    total_cases = sum(total_cases, na.rm = TRUE),
    total_deaths = sum(total_deaths, na.rm = TRUE),
    total_new_cases = sum(new_cases, na.rm = TRUE),
    total_new_deaths = sum(new_deaths, na.rm = TRUE)
  )
```

```
# Calculate overall summary statistics
summary_statistics <- summarise(
  aggregated_data,
  mean_cases = mean(total_cases, na.rm = TRUE),
  mean_deaths = mean(total_deaths, na.rm = TRUE),
  mean_new_cases = mean(total_new_cases, na.rm = TRUE),
  mean_new_deaths = mean(total_new_deaths, na.rm = TRUE)
)
```

```
# Identify top 10 countries by total cases
top_countries <- aggregated_data %>%
  arrange(desc(total_cases)) %>%
  head(10)
```

```
# Store key findings in vectors and matrices
total_cases_vector <- aggregated_data$total_cases
total_deaths_vector <- aggregated_data$total_deaths
```

```
# Create a matrix combining total cases and total deaths
cases_matrix <- cbind(total_cases_vector, total_deaths_vector)
```

```
# Compile results into a comprehensive list structure
results_list <- list(
  aggregated_data = aggregated_data,
  summary_statistics = summary_statistics,
  top_countries = top_countries,
  cases_matrix = cases_matrix
)
```

```r
# Print results
print(results_list)
```

```
## $aggregated_data
## # A tibble: 118 x 5
##    location  total_cases total_deaths total_new_cases total_new_deaths
##    <chr>           <dbl>        <dbl>           <dbl>            <dbl>
##  1 Africa         329060         4734            1585               17
##  2 Andorra          1140            0               4                0
##  3 Argentina      221979          285            1129                2
##  4 Aruba            751            0               3                0
##  5 Asia           120679          280             673                1
##  6 Australia       38597            0             145                0
##  7 Austria         89324            0             328                0
##  8 Bahamas           580            0               2                0
##  9 Bahrain           214            0               2                0
## 10 Barbados          295            0               1                0
## # i 108 more rows
##
## $summary_statistics
## # A tibble: 1 x 4
##   mean_cases mean_deaths mean_new_cases mean_new_deaths
##        <dbl>       <dbl>          <dbl>           <dbl>
## 1    553169.        487.          2221.            3.51
##
## $top_countries
## # A tibble: 10 x 5
##    location       total_cases total_deaths total_new_cases total_new_deaths
##    <chr>                <dbl>        <dbl>           <dbl>            <dbl>
##  1 World             21786907        19242           87349              138
##  2 North America      8987007         6656           36958               71
##  3 United States      7565490         5326           30154               42
##  4 Europe             7236587         1287           25609                6
##  5 South America      5032614         6202           22336               43
##  6 Brazil             2561789         2973           10920               16
##  7 Spain              2110058          735            7551                3
##  8 France             1141622            0            4146                0
##  9 United Kingdom     1095032            0            3741                0
## 10 Germany            1080948            0            3691                0
##
## $cases_matrix
##       total_cases_vector total_deaths_vector
##   [1,]             329060                4734
##   [2,]               1140                   0
##   [3,]             221979                 285
##   [4,]                751                   0
##   [5,]             120679                 280
##   [6,]              38597                   0
##   [7,]              89324                   0
##   [8,]                580                   0
##   [9,]                214                   0
##  [10,]                295                   0
##  [11,]             223210                 321
```

```
##  [12,]              948                0
##  [13,]              291                0
##  [14,]            61126                0
##  [15,]             2123                0
##  [16,]          2561789             2973
##  [17,]             1724                0
##  [18,]             4742              918
##  [19,]           415381                0
##  [20,]             5220               85
##  [21,]           319436              336
##  [22,]             2874                0
##  [23,]           884578                0
##  [24,]             1485                0
##  [25,]            24319                0
##  [26,]             8385                0
##  [27,]             1681              259
##  [28,]              724                0
##  [29,]             1344                0
##  [30,]            19050              231
##  [31,]            91969                0
##  [32,]            53961                0
##  [33,]            12217                0
##  [34,]            93836              386
##  [35,]               82                0
##  [36,]            14016                0
##  [37,]             3111                0
##  [38,]          7236587             1287
##  [39,]            11324                0
##  [40,]          1141622                0
##  [41,]              593                0
##  [42,]          1080948                0
##  [43,]            32061             1061
##  [44,]             1814                0
##  [45,]            22805                0
##  [46,]              544                0
##  [47,]              278                0
##  [48,]                1                0
##  [49,]            60516               35
##  [50,]              512                0
##  [51,]             4260                0
##  [52,]            22567                0
##  [53,]             4502                0
##  [54,]             4699              268
##  [55,]              248                0
##  [56,]                1                0
##  [57,]            59510                0
##  [58,]            73561                0
##  [59,]           261439                0
##  [60,]             4413                0
##  [61,]             6572                0
##  [62,]                1                0
##  [63,]             1678                0
##  [64,]             4083                0
##  [65,]             1597                0
```

```
##  [66,]             1374                0
##  [67,]            15693                0
##  [68,]             9556                0
##  [69,]              996                0
##  [70,]           803478             1010
##  [71,]              538                0
##  [72,]              810                0
##  [73,]              542                0
##  [74,]               91                0
##  [75,]              211              197
##  [76,]           364821                0
##  [77,]                1                0
##  [78,]             4124                0
##  [79,]           183953             2240
##  [80,]          8987007             6656
##  [81,]            26629                0
##  [82,]            47478                0
##  [83,]               13                0
##  [84,]            24667               26
##  [85,]            14574                0
##  [86,]           868182             2222
##  [87,]               36                0
##  [88,]            55518                0
##  [89,]           283098                0
##  [90,]            51829                0
##  [91,]              171                0
##  [92,]            12684                0
##  [93,]              541                0
##  [94,]              279                0
##  [95,]              211                0
##  [96,]              209                0
##  [97,]            10701                0
##  [98,]             5585                0
##  [99,]             3848                0
## [100,]            13864                0
## [101,]             1398                0
## [102,]          5032614             6202
## [103,]             1585                0
## [104,]          2110058              735
## [105,]              337                0
## [106,]             3554              203
## [107,]            64881                0
## [108,]           153483                0
## [109,]             3232                0
## [110,]             2689                0
## [111,]             1124                0
## [112,]              698                0
## [113,]          1095032                0
## [114,]          7565490             5326
## [115,]             3952                0
## [116,]             2650                0
## [117,]               18                0
## [118,]         21786907            19242
```

```r
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

```r
install.packages("tinytex")
```

```
## Installing package into 'C:/Users/elvir/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## package 'tinytex' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\elvir\AppData\Local\Temp\RtmpKoeg6e\downloaded_packages
```

```r
tinytex::install_tinytex()
```

```
## tlmgr --repository http://www.preining.info/tlgpg/ install tlgpg
```

```
## tlmgr option repository "https://ftp.fau.de/ctan/systems/texlive/tlnet"
```

```
## tlmgr update --list
```