

Milestone_4

Moliehi Mokete and Bongekile Nkosi

2022-11-18

Loading the libraries

```
## Warning in system("timedatectl", intern = TRUE): running command 'timedatectl'
## had status 1

## -- Attaching packages ----- tidyverse 1.3.1 --

## v ggplot2 3.3.5      v purrr 0.3.4
## v tibble 3.1.6       v dplyr 1.0.8
## v tidyr 1.2.0        v stringr 1.4.0
## v readr 2.1.2        v forcats 0.5.1

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()

##
## Attaching package: 'janitor'

## The following objects are masked from 'package:stats':
##
##   chisq.test, fisher.test
```

loading the MPX data

```
##      Variable VariableType MissingValues
## 1  country_code    character            0
## 2 total_conf_case    numeric            0
## 3 overall_cases     numeric            0
## 4   cases_rate      numeric            0
```

Loading Population denominator dataset

##	Variable	VariableType	MissingValues
## 1	dataflow	character	0
## 2	last_update	character	0
## 3	freq	character	0
## 4	indic_de	character	0
## 5	country_code	character	0
## 6	time_period	integer	0
## 7	obs_value	integer	0
## 8	obs_flag	character	509

Loading world country region dataset

```
##          Variable VariableType MissingValues
## 1          name      character              0
## 2         alpha_2      character              0
## 3         alpha_3      character              0
## 4    country_code        integer              0
## 5      iso_3166_2      character              0
## 6         region      character              0
## 7      sub_region      character              0
## 8 intermediate_region      character          92
## 9      region_code        integer              0
## 10   sub_region_code        integer              0

## [1] "AT" "BE" "BG" "CY" "CZ" "DE" "DK" "EE" "EL" "ES" "FI" "FR" "HR" "HU" "IE"
## [16] "IS" "IT" "LT" "LU" "LV" "MT" "NL" "NO" "PL" "PT" "RO" "SE" "SI" "SK"
```

Loading Census Data set

```
##      Variable VariableType MissingValues
## 1  country_code    character            0
## 2      sex        character            0
## 3      age        character            0
## 4      cas        character            0
## 5      edu        character            0
## 6      time        integer             0
## 7      flags      character      135428
## 8  footnotes      character      147878
## 9    res_pop      character            0
## 10     pop        integer            0

## [1] "ED1" "ED2" "ED3" "ED4" "ED5" "ED6" "NAP" "NONE" "UNK"

## [1] "ACT" "EMP" "INAC" "UNE" "UNK"

## [1] "Y_GE85" "Y_LT15" "Y15-29" "Y30-49" "Y50-64" "Y65-84"

## [1] "F" "M"

## 'summarise()' has grouped output by 'country_code'. You can override using the
## '.groups' argument.
## 'summarise()' has grouped output by 'country_code'. You can override using the
## '.groups' argument.
## 'summarise()' has grouped output by 'country_code'. You can override using the
## '.groups' argument.
## 'summarise()' has grouped output by 'country_code'. You can override using the
## '.groups' argument.
## 'summarise()' has grouped output by 'country_code'. You can override using the
## '.groups' argument.
```

Joining all data sets

```
joined_df <- merge(monkey_pox, pop_denominator, by.x = "country_code",
                  by.y = "country_code", all.x = TRUE, all.y = FALSE)

var_info <- data.frame(Variable = names(joined_df),
                      VariableType = sapply(joined_df, class),
                      MissingValues = sapply(joined_df, function(y)
                                              sum(length(which(is.na(y))))),
                      row.names = NULL)

var_info
```

```
##      Variable VariableType MissingValues
## 1  country_code    character            0
## 2 total_conf_case    numeric            0
## 3   cases_rate    numeric            0
## 4   time_period    integer            0
```

```

joined_df <- merge(joined_df, world_country_region, by.x = "country_code",
  by.y = "country_code", all.x = TRUE, all.y = FALSE)

var_info <- data.frame(Variable = names(joined_df),
  VariableType = sapply(joined_df, class),
  MissingValues = sapply(joined_df, function(y)
    sum(length(which(is.na(y))))),
  row.names = NULL)

var_info

```

```

##      Variable VariableType MissingValues
## 1  country_code    character            0
## 2 total_conf_case    numeric            0
## 3   cases_rate    numeric            0
## 4  time_period    integer            0
## 5   sub_region    character            2

```

```

joined_df <- joined_df %>%
  mutate(sub_region = ifelse(country_code == "EL", "Southeast Europe", sub_region),
    sub_region = ifelse(country_code == "LU", "Northwestern Europe", sub_region)) %>%
  group_by(sub_region) %>%
  mutate(total_case_region = sum(total_conf_case),
    rate_per_region = total_conf_case / total_case_region * 100)

```

```

joined_df_edu <- merge(joined_df, census_stats_edu, by.x = "country_code",
  by.y = "country_code", all.x = TRUE, all.y = FALSE)

```

```

joined_df_cas <- merge(joined_df, census_stats_cas, by.x = "country_code",
  by.y = "country_code", all.x = TRUE, all.y = FALSE)

```

```

joined_df_sex <- merge(joined_df, census_stats_sex, by.x = "country_code",
  by.y = "country_code", all.x = TRUE, all.y = FALSE)

```

```

joined_df_age <- merge(joined_df, census_stats_age, by.x = "country_code",
  by.y = "country_code", all.x = TRUE, all.y = FALSE)

```

```

joined_df_popdensity <- merge(joined_df,
  census_stats_respop, by.x = "country_code",
  by.y = "country_code", all.x = TRUE, all.y = FALSE)

```

Data dictionary based on clean dataset (minimum 4 data elements), including: Variable name Data type Description

```
data_dict <- function(joined_df, desc = c()){
  data.frame(
    "Variable Name" = names(joined_df),
    "Variable Type" = sapply(joined_df,class),
    "Variable Description" = desc,
    check.names = FALSE, row.names = NULL
  )
}
```

```
data_dict(joined_df[], desc =c(
  "country code ",
  "months cases were reported",
  "total MPX cases recorded",
  " the recent time period",
  "countries sub regions in Europe",
  "total MPX cases per sub region",
  "rate of MPX per month per sub_region"))
```

##	Variable Name	Variable Type	Variable Description
## 1	country_code	character	country code
## 2	total_conf_case	numeric	months cases were reported
## 3	cases_rate	numeric	total MPX cases recorded
## 4	time_period	integer	the recent time period
## 5	sub_region	character	countries sub regions in Europe
## 6	total_case_region	numeric	total MPX cases per sub region
## 7	rate_per_region	numeric	rate of MPX per month per sub_region

```
data_dict <- function(joined_df_age, desc = c()){
  data.frame(
    "Variable Name" = names(joined_df_age),
    "Variable Type" = sapply(joined_df_age,class),
    "Variable Description" = desc,
    check.names = FALSE, row.names = NULL
  )
}
```

```
data_dict(joined_df_age[],desc=c(
  "country code ",
  "months cases were reported",
  "total MPX cases recorded",
  "the recent time period",
  "countries sub regions in Europe",
  "total MPX cases per sub region",
  "rate of MPX per month per sub_region",
  "age groups of the population",
  "total population per age group"
))
```

##	Variable Name	Variable Type	Variable Description
## 1	country_code	character	country code

##	Variable Name	Variable Type	Variable Description
## 2	total_conf_case	numeric	months cases were reported
## 3	cases_rate	numeric	total MPX cases recorded
## 4	time_period	integer	the recent time period
## 5	sub_region	character	countries sub regions in Europe
## 6	total_case_region	numeric	total MPX cases per sub region
## 7	rate_per_region	numeric	rate of MPX per month per sub_region
## 8	age	character	age groups of the population
## 9	total_pop_age	integer	total population per age group

```
data_dict <- function(joined_df_cas, desc = c()){
  data.frame(
    "Variable Name" = names(joined_df_cas),
    "Variable Type" = sapply(joined_df_cas,class),
    "Variable Description" = desc,
    check.names = FALSE, row.names = NULL
  )
}
```

```
data_dict(joined_df_cas[],desc=c(
  "country code ",
  "months cases were reported",
  "total MPX cases recorded",
  "the recent time period",
  "countries sub regions in Europe",
  "total MPX cases per sub region",
  "rate of MPX per month per sub_region",
  "economical status of the population",
  "total population per economical status"))
```

##	Variable Name	Variable Type	Variable Description
## 1	country_code	character	country code
## 2	total_conf_case	numeric	months cases were reported
## 3	cases_rate	numeric	total MPX cases recorded
## 4	time_period	integer	the recent time period
## 5	sub_region	character	countries sub regions in Europe
## 6	total_case_region	numeric	total MPX cases per sub region
## 7	rate_per_region	numeric	rate of MPX per month per sub_region
## 8	cas	character	economical status of the population
## 9	total_pop_cas	integer	total population per economical status

```
data_dict <- function(joined_df_edu, desc = c()){
  data.frame(
    "Variable Name" = names(joined_df_edu),
    "Variable Type" = sapply(joined_df_edu,class),
    "Variable Description" = desc,
    check.names = FALSE, row.names = NULL
  )
}
```

```
data_dict(joined_df_edu[],desc=c(
  "country code ",
  "months cases were reported",
  "total MPX cases recorded",
```

```
"the recent time period",
"countries sub regions in Europe",
"total MPX cases per sub region",
"rate of MPX per month per sub_region",
"the categories of education level",
"total population per education level"))
```

##	Variable Name	Variable Type	Variable Description
## 1	country_code	character	country code
## 2	total_conf_case	numeric	months cases were reported
## 3	cases_rate	numeric	total MPX cases recorded
## 4	time_period	integer	the recent time period
## 5	sub_region	character	countries sub regions in Europe
## 6	total_case_region	numeric	total MPX cases per sub region
## 7	rate_per_region	numeric	rate of MPX per month per sub_region
## 8	edu	character	the categories of education level
## 9	total_pop_edu	integer	total population per education level

```
data_dict <- function(joined_df_sex, desc = c()){
  data.frame(
    "Variable Name" = names(joined_df_sex),
    "Variable Type" = sapply(joined_df_sex,class),
    "Variable Description" = desc,
    check.names = FALSE, row.names = NULL
  )
}
```

```
data_dict(joined_df_sex[],desc=c(
  "country code ",
  "months cases were reported",
  "total MPX cases recorded",
  "the recent time period",
  "countries sub regions in Europe",
  "total MPX cases per region",
  "rate of MPX per month per region",
  "sex of the population",
  "total population per sex"))
```

##	Variable Name	Variable Type	Variable Description
## 1	country_code	character	country code
## 2	total_conf_case	numeric	months cases were reported
## 3	cases_rate	numeric	total MPX cases recorded
## 4	time_period	integer	the recent time period
## 5	sub_region	character	countries sub regions in Europe
## 6	total_case_region	numeric	total MPX cases per region
## 7	rate_per_region	numeric	rate of MPX per month per region
## 8	sex	character	sex of the population
## 9	total_pop_sex	integer	total population per sex

```
data_dict <- function(joined_df_popdensity, desc = c()){
  data.frame(
    "Variable Name" = names(joined_df_popdensity),
```



```

    "Variable Type" = sapply(joined_df_popdensity,class),
    "Variable Description" = desc,
    check.names = FALSE, row.names = NULL
  )
}

```

```

data_dict(joined_df_popdensity [],desc=c(
  "country code ",
  "months cases were reported",
  "total MPX cases recorded",
  "the recent time period",
  "countries regions in Europe",
  "total MPX cases per region",
  "rate of MPX per month per region",
  "categories of population density",
  "total population per population density"))

```

##	Variable Name	Variable Type	Variable Description
## 1	country_code	character	country code
## 2	total_conf_case	numeric	months cases were reported
## 3	cases_rate	numeric	total MPX cases recorded
## 4	time_period	integer	the recent time period
## 5	sub_region	character	countries regions in Europe
## 6	total_case_region	numeric	total MPX cases per region
## 7	rate_per_region	numeric	rate of MPX per month per region
## 8	res_pop	character	categories of population density
## 9	total_pop_respop	integer	total population per population density

One or more tables with descriptive statistics for 4 data element

```
summary(joined_df$total_conf_case)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       4.0    22.0    52.0   588.7   231.0  6283.0
```

```
summary(joined_df$total_case_region)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##        4      277     576    3291    7906    8210
```

```
summary(joined_df$rate_per_region)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.3162   2.0833  12.6354  24.1379  29.3403 100.0000
```

```
summary(joined_df_age$total_pop_age)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    392.0   392.0   549.0   538.7   666.2   728.0
```

```
library(kableExtra)
```

```
##
```

```
## Attaching package: 'kableExtra'
```

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

```
##
```

```
##      group_rows
```

```
descriptive_statistics_table<-data.frame(
  "Minimum"= c(0.00,4,0.0000,392.0),
  "First Quartile"= c(2.00,277,0.1762,392.0),
  "Median"= c(11.50,576,1.5625,549.0),
  "Mean" = c(147.17,3291,6.0345,538.7),
  "Third Quartile"=c (54.25, 7906, 5.7615, 667.0),
  "Maximum" = c(3244.00,8210,100.000,728.0),
  row.names = c("Monthly Total Cases","Total Cases per Region",
    "Rate per month per region", "total population per age group"))
```

```
kable(descriptive_statistics_table, booktabs=T, digits= c(1,1,1,0),
  caption= "Descriptive statistics for data elements")
```

Table 1: Descriptive statistics for data elements

	Minimum	First.Quartile	Median	Mean	Third.Quartile	Maximum
Monthly Total Cases	0	2.0	11.5	147	54.2	3244
Total Cases per Region	4	277.0	576.0	3291	7906.0	8210
Rate per month per region	0	0.2	1.6	6	5.8	100
total population per age group	392	392.0	549.0	539	667.0	728