Milestone_3

Moliehi Mokete and Bongekile Nkosi

2022-11-07

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

```
file_path1<-"https://raw.githubusercontent.com/PHW290/phw251_projectdata/main/euro_mpx_cases.csv"
monkey_pox <-read_csv(file_path1,na=c("","NA","*", "n/a"),
                      show_col_types = FALSE)%>% clean_names()

monkey_pox<- monkey_pox%>% mutate(month_rep= months(date_rep))%>%
  group_by(country_code,month_rep)%>%
  mutate(total_conf_case = sum(conf_cases))

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

```
##          Variable VariableType MissingValues
## 1        date_rep         Date             0
## 2     country_exp    character             0
## 3    country_code    character             0
## 4          source    character             0
## 5      conf_cases      numeric             0
## 6       month_rep    character             0
## 7 total_conf_case      numeric             0
```

```
 monkey_pox <- monkey_pox %>%
arrange(country_code) %>%
group_by(month_rep) %>%
filter(!duplicated(country_code))

 monkey_pox<-monkey_pox %>%
   select(country_code,month_rep,total_conf_case)
```

Loading Population denominator dataset

```
file_path2<-"https://raw.githubusercontent.com/PHW290/phw251_projectdata/main/euro_pop_denominators.csv
pop_denominator<- read.csv(file_path2,na = c("", "NA", "*", "n/a")) %>%
  clean_names() %>%
  rename(country_code = geo)

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

```
##       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
```

```
 pop_denominator <- pop_denominator%>%
   filter(time_period== 2022)%>%
   select(country_code, time_period)
```

Loading world country region dataset

```
file_path4<-"https://raw.githubusercontent.com/PHW290/phw251_projectdata/main/world_country_regions.csv
world_country_region <- read.csv(file_path4,na = c("", "NA", "*", "n/a"))%>%
  clean_names()

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

```
##               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
```

```
country_code_categories<-unique(monkey_pox$country_code)
country_code_categories
```

```
##  [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"
```

```
 world_country_region<-world_country_region %>%
  mutate(country_code= case_when(str_detect(alpha_2,"at")~"AT",
                          str_detect(alpha_2,"be")~"BE",
                          str_detect(alpha_2,"bg")~"BG",
                          str_detect(alpha_2,"cy")~"CY",
                          str_detect(alpha_2,"cz")~"CZ",
                          str_detect(alpha_2,"de")~"DE",
                          str_detect(alpha_2,"dk")~"DK",
                          str_detect(alpha_2,"ee")~"EE",
                          str_detect(alpha_2,"el")~"EL",
                          str_detect(alpha_2,"es")~"ES",
                          str_detect(alpha_2,"fi")~"FI",
                          str_detect(alpha_2,"fr")~"FR",
                          str_detect(alpha_2,"hr")~"HR",
                          str_detect(alpha_2,"hu")~"HU",
                          str_detect(alpha_2,"ie")~"IE",
                          str_detect(alpha_2,"is")~"IS",
                          str_detect(alpha_2,"it")~"IT",
                          str_detect(alpha_2,"lt")~"LT",
                          str_detect(alpha_2,"lu")~"lu",
                          str_detect(alpha_2,"lv")~"LV",
                          str_detect(alpha_2,"mt")~"MT",
                          str_detect(alpha_2,"nl")~"NL",
                          str_detect(alpha_2,"no")~"NO",
                          str_detect(alpha_2,"pl")~"PL",
                          str_detect(alpha_2,"pt")~"PT",
                          str_detect(alpha_2,"ro")~"RO",
                          str_detect(alpha_2,"se")~"SE",
                          str_detect(alpha_2,"si")~"SI",
                          str_detect(alpha_2,"sk")~"SK",
                         TRUE~NA_character_))%>%
   drop_na(country_code)

 world_country_region<- world_country_region%>%
   select(country_code, sub_region)
```

Loading Census Data set

```
file_path3<-"https://raw.githubusercontent.com/PHW290/phw251_projectdata/main/euro_census_stats.csv"
census_stats <- read.csv(file_path3,na = c("", "NA", "*", "n/a"))%>%
  clean_names()

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

```
##        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
```

```
edu_categories<-unique(census_stats$edu)
edu_categories
```

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

```
cas_categories<-unique(census_stats$cas)
cas_categories
```

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

```
age_categories <- unique(census_stats$age)
age_categories
```

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

```
sex_categories<- unique(census_stats$sex)
sex_categories
```

```
## [1] "F" "M"
```

```
census_stats<-census_stats %>%
  mutate(edu= case_when(edu=="NONE" ~ "No formal education",
                      edu== "ED1" ~ "Primary education",
                      edu== "ED2" ~ "Lower secondary education",
                      edu== "ED3" ~ "Upper secondary education",
           edu== "ED4" ~ "Post secondary non-tertiary education (tradeschool)",
                  edu== "ED5" ~ "First stage of tertiary education (college)",
           edu== "ED6" ~ "Second stage of tertiary education (grad school)",
           TRUE~NA_character_))%>%
  drop_na(edu)%>%
  mutate(cas= case_when(cas== "ACT" ~ "Total economically active",
                        cas== "EMP" ~ "Employed (among economically active)",
                        cas== "UNE" ~ "Unemployed (among economically active)",
                        cas== "INAC" ~ "Total economically inactive",
                        TRUE~NA_character_))%>%
  drop_na(cas)%>%
  mutate(age=case_when(age== "Y_LT15" ~ " < 15",
                         age== "Y15-29" ~ "15-29",
                         age== "Y30-49" ~ "30-49",
                         age== "Y50-64" ~ "50-64",
                         age== "Y65-84" ~ "65-84",
                         TRUE ~ " 85+"))%>%
  mutate(sex=case_when(sex=="F"~ "Female",
                       TRUE~"Male"))

  census_stats_edu<- census_stats %>%
  group_by(country_code,edu) %>%
  summarise(total_pop_edu = n())
```

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

```
census_stats_cas <- census_stats %>%
  group_by(country_code, cas)%>%
summarise(total_pop_cas = n())
```

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

```
census_stats_sex <- census_stats %>%
  group_by(country_code, sex)%>%
summarise(total_pop_sex= n())
```

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

```
census_stats_age <- census_stats %>%
  group_by(country_code, age)%>%
summarise(total_pop_age= n())
```

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

```
census_stats_respop <- census_stats %>%
  group_by(country_code, res_pop)%>%
summarise(total_pop_respop= n())
```

```
## `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       month_rep    character             0
## 3 total_conf_case      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       month_rep    character             0
## 3 total_conf_case      numeric             0
## 4     time_period      integer             0
## 5      sub_region    character             8
```

```
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)
```

```r
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

```r
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         month_rep     character           months cases were reported
## 3   total_conf_case       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
```

```r
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 
## 2         month_rep     character          months cases were reported
## 3   total_conf_case       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         month_rep     character             months cases were reported
## 3   total_conf_case       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         month_rep     character           months cases were reported
## 3   total_conf_case       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         month_rep     character       months cases were reported
## 3   total_conf_case       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         month_rep     character              months cases were reported
## 3   total_conf_case       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. 
##    0.00    2.00   11.50  147.17   54.25 3244.00
```

```
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. 
##   0.0000   0.1762   1.5625   6.0345   5.7615 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   667.0   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