Milestone_3

Moliehi Mokete and Bongekile Nkosi

2022-11-06

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))


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"))
```

```
##     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
```

```
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",
  " 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             age     character    age groups of the population
## 7   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",
  "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             cas     character    economical status of the population
## 7   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",
  "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             edu     character    the categories of education level
## 7   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",
  " 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             sex     character           sex of the population
## 7   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 sub regions in Europe",
  "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 sub regions in Europe
## 6          res_pop     character        categories of population density
## 7 total_pop_respop       integer total population per population density
```

One or more tables with descriptive statistics for 4 data element

```
summary_stas<-joined_df%>%
  group_by(month_rep)%>%
  summarise(mean= mean(total_conf_case), sd=sd(total_conf_case),
            max=max(total_conf_case),min=min(total_conf_case))
summary_stas
```

```
## # A tibble: 4 x 5
##   month_rep  mean    sd   max   min
##   <chr>     <dbl> <dbl> <dbl> <dbl>
## 1 August    141.  317.   1429     0
## 2 July      286.  695.   3244     0
## 3 June      142.  330.   1419     0
## 4 May        19.8  50.3   199     0
```

```
summary_stas<-joined_df%>%
  group_by(sub_region)%>%
  summarise(mean_cases= mean(total_conf_case), sd_cases=sd(total_conf_case),
            max_cases=max(total_conf_case),min_cases=min(total_conf_case))
summary_stas
```

```
## # A tibble: 7 x 5
##   sub_region          mean_cases sd_cases max_cases min_cases
##   <chr>                    <dbl>    <dbl>     <dbl>     <dbl>
## 1 Eastern Europe            11.5     15.2        58         0
## 2 Northern Europe           16       21.9        71         0
## 3 Northwestern Europe       11.8     11.4        25         0
## 4 Southeast Europe          13       11.5        26         1
## 5 Southern Europe          329.     733.       3244         0
## 6 Western Asia               1        2           4         0
## 7 Western Europe           410.     494.       1619         2
```

```
summary_stas<-joined_df_edu%>%
  group_by(sub_region)%>%
  summarise(mean_edu= mean(total_pop_edu), sd_edu=sd(total_pop_edu),
            max_edu=max(total_pop_edu),min_edu=min(total_pop_edu))
summary_stas
```

```
## # A tibble: 7 x 5
##   sub_region          mean_edu sd_edu max_edu min_edu
##   <chr>                  <dbl>  <dbl>   <int>   <int>
## 1 Eastern Europe          518.   53.1     566     336
## 2 Northern Europe         422.   92.2     571     336
## 3 Northwestern Europe     485.   64.3     525     336
## 4 Southeast Europe        542.   10.3     551     520
## 5 Southern Europe         447.   91.0     564     336
## 6 Western Asia            528    23.0     551     477
## 7 Western Europe          449.  105.      564     336
```

```
summary_stas<-joined_df_cas%>%
  group_by(sub_region)%>%
  summarise(mean_cas= mean(total_pop_cas), sd_cas=sd(total_pop_cas),
            max_cas=max(total_pop_cas),min_cas=min(total_pop_cas))
summary_stas
```

```
## # A tibble: 7 x 5
##   sub_region          mean_cas sd_cas max_cas min_cas
##   <chr>                  <dbl>  <dbl>   <int>   <int>
## 1 Eastern Europe          907.   68.0    1001     765
## 2 Northern Europe         739.  155.     1002     588
## 3 Northwestern Europe     848    52.6     908     767
## 4 Southeast Europe        948.   41.3    1003     890
## 5 Southern Europe         783.  148.     1002     588
## 6 Western Asia            924    51.7     978     842
## 7 Western Europe          785.  168.     1000     588
```

```
summary_stas<-joined_df_age%>%
  group_by(sub_region)%>%
  summarise(mean_age= mean(total_pop_age), sd_age=sd(total_pop_age),
            max_age=max(total_pop_age),min_age=min(total_pop_age))
summary_stas
```

```
## # A tibble: 7 x 5
##   sub_region          mean_age sd_age max_age min_age
##   <chr>                  <dbl>  <dbl>   <int>   <int>
## 1 Eastern Europe          605.   116.     725     392
## 2 Northern Europe         492.   127.     728     392
## 3 Northwestern Europe     565.   105.     663     392
## 4 Southeast Europe        632.   131.     727     392
## 5 Southern Europe         522.   127.     728     392
## 6 Western Asia            616    124.     714     392
## 7 Western Europe          523.   137.     721     392
```