Milestone # 2

Moliehi Mokete and Bongekile Nkosi

2022-10-01

This is a team assignment; each team should complete and turn in a PDF created from an Rmd via Github. Please include code and output for the following components:

Description of data set

1. What is the data source? (1-2 sentences on where the data is coming from, dates included, etc.)

Data sources

monkey_pox data source is from European Center for Disease Prevention and Control(ECDC)

pop_denominator data source is from European commission

census_stat data source is from European census statistics 2011

world_country_region data source is from European census

2. How does the data set relate to the group problem statement and question?

The data set is going to help us understand how monkey pox case rates may differ by region and various demographic factors,additionally, the data set will allow us to determine if there is a relationship between certain demographics and monkey pox case rates.

Import statement

NOTE: Please use data sets available in the PHW251 Project Data github repo Links to an external site. (this is important to make sure everyone is using the same data sets)

Use appropriate import function and package based on the type of file(done)

Loading the library to be used in this assignment

```
library(tidyverse)
```

```
## 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()
```

```
library(readr)
library(janitor)
```

```
##
## Attaching package: 'janitor'
```

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

```
library(purrr)
library(stringr)
```

Utilize function arguments to control relevant components (i.e. change column types, column names, missing values, etc.)

Importing data from git repositories

```
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"))%>% clean_names()
```

```
## Rows: 2987 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): CountryExp, CountryCode, Source
## dbl  (1): ConfCases
## date (1): DateRep
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
str(monkey_pox)
```

```
## spec_tbl_df [2,987 x 5] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ date_rep    : Date[1:2987], format: "2022-05-09" "2022-05-09" ...
##  $ country_exp : chr [1:2987] "Austria" "Belgium" "Bulgaria" "Croatia" ...
##  $ country_code: chr [1:2987] "AT" "BE" "BG" "HR" ...
##  $ source      : chr [1:2987] "TESSy" "TESSy" "TESSy" "TESSy" ...
##  $ conf_cases  : num [1:2987] 0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   DateRep = col_date(format = ""),
##   ..   CountryExp = col_character(),
##   ..   CountryCode = col_character(),
##   ..   Source = col_character(),
##   ..   ConfCases = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
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)
str(pop_denominator)
```

```
## 'data.frame':    603 obs. of  8 variables:
##  $ dataflow    : chr  "ESTAT:TPS00001(1.0)" "ESTAT:TPS00001(1.0)" "ESTAT:TPS00001(1.0)" "ESTAT:TPS000
##  $ last_update : chr  "11/07/22 11:00:00" "11/07/22 11:00:00" "11/07/22 11:00:00" "11/07/22 11:00:00
##  $ freq        : chr  "A" "A" "A" "A" ...
##  $ indic_de    : chr  "JAN" "JAN" "JAN" "JAN" ...
##  $ country_code: chr  "AD" "AD" "AD" "AD" ...
##  $ time_period : int  2011 2012 2013 2016 2018 2019 2020 2022 2011 2012 ...
##  $ obs_value   : int  78115 78115 76246 71732 74794 76177 77543 79535 2907361 2903008 ...
##  $ obs_flag    : chr  "b" NA NA NA ...
```

```
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()
str(census_stats)
```

```
## 'data.frame':    152534 obs. of  10 variables:
##  $ country_code: chr  "AT" "AT" "AT" "AT" ...
##  $ sex         : chr  "F" "F" "F" "F" ...
##  $ age         : chr  "Y_GE85" "Y_GE85" "Y_GE85" "Y_GE85" ...
##  $ cas         : chr  "ACT" "ACT" "ACT" "ACT" ...
##  $ edu         : chr  "ED1" "ED1" "ED1" "ED1" ...
##  $ time        : int  2011 2011 2011 2011 2011 2011 2011 2011 2011 2011 ...
##  $ flags       : chr  NA "d" "d" NA ...
##  $ footnotes   : chr  NA "For data privacy protection reasons, the statistical disclosure control me
##  $ res_pop     : chr  "500000-999999" "10000-99999" "200000-499999" "100000-199999" ...
##  $ pop         : int  0 4 5 6 6 8 18 19 21 25 ...
```

```
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()
str(world_country_region)
```

```
## 'data.frame':    247 obs. of  10 variables:
##  $ name               : chr  "AFGHANISTAN" "ALAND ISLANDS" "ALBANIA" "ALGERIA" ...
##  $ alpha_2            : chr  "af-4" "ax-248" "al-8" "dz-12" ...
##  $ alpha_3            : chr  "afg-4" "ala-248" "alb-8" "dza-12" ...
##  $ country_code       : int  4 248 8 12 16 20 24 660 28 32 ...
##  $ iso_3166_2         : chr  "ISO 3166-2:AF" "ISO 3166-2:AX" "ISO 3166-2:AL" "ISO 3166-2:DZ" ...
##  $ region             : chr  "Asia" "Europe" "Europe" "Africa" ...
##  $ sub_region         : chr  "Southern Asia" "Northern Europe" "Southern Europe" "Northern Africa" .
##  $ intermediate_region: chr  NA "Nordic" "Southeast Europe" NA ...
##  $ region_code        : int  142 150 150 2 9 150 2 19 19 19 ...
##  $ sub_region_code    : int  34 154 39 15 61 39 202 419 419 419 ...
```

Document the import process

Firstly the packages were loaded that will enable us to import data and clean the data

Then the file paths were extracted and renamed for each data set that we are going to use through out the project.

Each data set was imported separately

All data set were imported using read_csv() function because all file were in the text format

All the variable were cleaned through clean_name() function which makes converted the variable names into lower case font , and remove the space between the variable by replacing it with underscore symbol

For those with missing information we used N/A because it was not easy to compute the information due to the fact that it is not accessible

Finally, we used str() function to view the information regarding variable names, data type and few row data for each variable

Identify data types for 5+ data elements/columns/variables Utilize functions or resources in RStudio to determine the types of each data element (i.e. character, numeric, factor)

```
map(monkey_pox, class)
```

```
## $date_rep
## [1] "Date"
##
## $country_exp
## [1] "character"
##
## $country_code
## [1] "character"
##
## $source
## [1] "character"
##
## $conf_cases
## [1] "numeric"
```

```
map(pop_denominator,class)
```

```
## $dataflow
## [1] "character"
##
## $last_update
## [1] "character"
##
## $freq
## [1] "character"
##
## $indic_de
## [1] "character"
##
## $country_code
## [1] "character"
##
## $time_period
## [1] "integer"
##
## $obs_value
## [1] "integer"
##
## $obs_flag
## [1] "character"
```

```
map(world_country_region,class)
```

```
## $name
## [1] "character"
##
## $alpha_2
```

```
## [1] "character"
##
## $alpha_3
## [1] "character"
##
## $country_code
## [1] "integer"
##
## $iso_3166_2
## [1] "character"
##
## $region
## [1] "character"
##
## $sub_region
## [1] "character"
##
## $intermediate_region
## [1] "character"
##
## $region_code
## [1] "integer"
##
## $sub_region_code
## [1] "integer"
```

```
map(census_stats,class)
```

```
## $country_code
## [1] "character"
##
## $sex
## [1] "character"
##
## $age
## [1] "character"
##
## $cas
## [1] "character"
##
## $edu
## [1] "character"
##
## $time
## [1] "integer"
##
## $flags
## [1] "character"
##
## $footnotes
## [1] "character"
##
## $res_pop
## [1] "character"
```

```
##
## $pop
## [1] "integer"
```

Identify 5+ data elements required for your specified scenario. If <5 elements are required to complete the analysis, please choose additional variables of interest in the data set to explore in this milestone.

confirmed cases,date reported, sub_region, time period, age, education, sex, employment status

Identify the desired type/format for each variable—will you need to convert any columns to numeric or another type?

education from character to string, date reported from date to integer (month) then convert month data type into character or string

Provide a basic description of the 5+ data elements Numeric: mean, median, range

```
summary(monkey_pox$conf_cases)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   0.000   5.715   1.000 655.000
```

```
summary(pop_denominator$time_period)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2011    2013    2016    2016    2019    2022
```

Basic description

Looking at the confirmed cases from monkey pox data set: the minimum cases reported is zero while the mean is 5.715 followed by the maximum cases reported being 655 cases.

As for the time period from population denominator data set: the minimum year is 2011 , mean year is 2016 and the most recent time period is 2022

Character: unique values/categories

```
sub_region_categories<-unique(world_country_region$sub_region)
sub_region_categories
```

```
##  [1] "Southern Asia"                   "Northern Europe"
##  [3] "Southern Europe"                 "Northern Africa"
##  [5] "Polynesia"                       "Sub-Saharan Africa"
##  [7] "Latin America and the Caribbean" "Western Asia"
##  [9] "Australia and New Zealand"       "Western Europe"
## [11] "Eastern Europe"                  "Northern America"
## [13] "South-eastern Asia"              "Eastern Asia"
## [15] "Melanesia"                       "Micronesia"
## [17] "Central Asia"
```

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

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

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

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

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

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

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

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

```
date_reported_values <- unique(monkey_pox$date_rep)
date_reported_values
```

```
##   [1] "2022-05-09" "2022-05-13" "2022-05-15" "2022-05-16" "2022-05-17"
##   [6] "2022-05-18" "2022-05-19" "2022-05-20" "2022-05-21" "2022-05-22"
##  [11] "2022-05-23" "2022-05-24" "2022-05-25" "2022-05-26" "2022-05-27"
##  [16] "2022-05-28" "2022-05-29" "2022-05-30" "2022-05-31" "2022-06-01"
##  [21] "2022-06-02" "2022-06-03" "2022-06-04" "2022-06-05" "2022-06-06"
##  [26] "2022-06-07" "2022-06-08" "2022-06-09" "2022-06-10" "2022-06-11"
##  [31] "2022-06-12" "2022-06-13" "2022-06-14" "2022-06-15" "2022-06-16"
##  [36] "2022-06-17" "2022-06-18" "2022-06-19" "2022-06-20" "2022-06-21"
##  [41] "2022-06-22" "2022-06-23" "2022-06-24" "2022-06-25" "2022-06-26"
##  [46] "2022-06-27" "2022-06-28" "2022-06-29" "2022-06-30" "2022-07-01"
##  [51] "2022-07-02" "2022-07-03" "2022-07-04" "2022-07-05" "2022-07-06"
##  [56] "2022-07-07" "2022-07-08" "2022-07-09" "2022-07-10" "2022-07-11"
##  [61] "2022-07-12" "2022-07-13" "2022-07-14" "2022-07-15" "2022-07-16"
##  [66] "2022-07-17" "2022-07-18" "2022-07-19" "2022-07-20" "2022-07-21"
##  [71] "2022-07-22" "2022-07-23" "2022-07-24" "2022-07-25" "2022-07-26"
##  [76] "2022-07-27" "2022-07-28" "2022-07-29" "2022-07-30" "2022-07-31"
##  [81] "2022-08-01" "2022-08-02" "2022-08-03" "2022-08-04" "2022-08-05"
##  [86] "2022-08-06" "2022-08-07" "2022-08-08" "2022-08-09" "2022-08-10"
##  [91] "2022-08-11" "2022-08-12" "2022-08-13" "2022-08-14" "2022-08-15"
##  [96] "2022-08-16" "2022-08-17" "2022-08-18" "2022-08-19" "2022-08-20"
## [101] "2022-08-21" "2022-08-22" "2022-08-23"
```

Basic description

Sub_region has 17 categories Age variable consists of 6 categories Education variable consists of 9 categories Sex variable consists of 2 categories Employment_status variable consists of 5 categories

date reported variable start with 2022/05/09 and the most recent date reported is 2022/08/23 this means that the cases were reported from May 2022 to August 2022. Therefore the cases were reported for 4 months in total.

any other descriptive that will be useful to the analysis

Age, education, sex, and employment variable categories need to be renamed so that the will be very informative and to allow easy understanding for the reader because at the moment the data was captured using the code defined in the questionnaire

Date reported variable need to be manipulated : we are going to extract only month so that we can do the analysis comparing cases per month.