

Monkeypox

2022-10-01

Description of dataset What is the data source? (1-2 sentences on where the data is coming from, dates included, etc.) The data is about the Monkey case data is from European Centre for Disease Prevention and Control The population denominator and census data is from Eurostat

How does the dataset relate to the group problem statement and question? Is there a difference between monkeypox cases by region in the EU We will be using the monkeypox case rate, population denominator and region data set to help us find the difference between monkeypox cases by region in the EU. Is there a difference between country level monkeypox case rates by certain demographic factors? We will use the monkeypox case, population denominator and census data to see if there is a difference between country level case rates based on certain demographic factors.

Import statement Use appropriate import function and package based on the type of file Utilize function arguments to control relevant components (i.e. change column types, column names, missing values, etc.) Document the import process 1, We navigated to the PHW290/phw251_projectdata git repo. We clicked on Code and downloaded the Zip file which we then extracted the csv files from. We imported the relevant datasets into our RStudio files for this project.

```
library(readr)
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
```

```
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 stringr 1.4.0
## v tidyr 1.2.0        v forcats 0.5.1
```

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

```
mp_cases <- read_csv("~/r_for_ph_monkeypox/euro_mpx_cases.csv")
```

```
## 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.
```

```
pop_den <- read_csv("~/r_for_ph_monkeypox/euro_pop_denominators.csv")
```

```
## Rows: 603 Columns: 8
## -- Column specification -----
## Delimiter: ","
## chr (6): DATAFLOW, LAST UPDATE, freq, indic_de, geo, OBS_FLAG
## dbl (2): TIME_PERIOD, OBS_VALUE
##
## 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.
```

```
eu_census <- read_csv("euro_census_stats.csv")
```

```
## Rows: 152534 Columns: 10
## -- Column specification -----
## Delimiter: ","
## chr (8): COUNTRY_CODE, SEX, AGE, CAS, EDU, FLAGS, FOOTNOTES, RES_POP
## dbl (2): TIME, pop
##
## 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.
```

```
wrld_region <- read_csv("~/r_for_ph_monkeypox/world_country_regions.csv")
```

```
## Rows: 247 Columns: 10
## -- Column specification -----
## Delimiter: ","
## chr (7): name, alpha-2, alpha-3, iso_3166-2, region, sub-region, intermediat...
## dbl (3): country-code, region-code, sub-region-code
##
## 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.
```

Identify data types for 5+ data elements/columns/variables 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. Utilize functions or resources in RStudio to determine the types of each data element (i.e. character, numeric, factor) Identify the desired type/format for each variable—will you need to convert any columns to numeric or another type? Provide a basic description of the 5+ data elements Numeric: mean, median, range Character: unique values/categories Or any other descriptives that will be useful to the analysis

```
clean_eu_census <- rename_with(eu_census, ~ tolower((gsub(" ", "_",.x,fixed=TRUE)))) %>%
  select(c(-"flags",- "footnotes"))
```

```
clean_mp_cases <- rename_with(mp_cases, ~ tolower((gsub(" ", "_",.x,fixed=TRUE)))) %>%
  select(c(-"source"))
```

```
clean_pop_den <- rename_with(pop_den, ~ tolower((gsub(" ", "_",.x,fixed=TRUE)))) %>%
  select(c(-"dataflow", -"last_update", -"freq", -"obs_flag"))
```

```
clean_wrld_region <- filter(wrld_region, region == "Europe") %>%
  select(c(-"alpha-3", -"iso_3166-2", -"intermediate-region"))
```

```
str(clean_pop_den)
```

```
## tibble [603 x 4] (S3: tbl_df/tbl/data.frame)
## $ indic_de : chr [1:603] "JAN" "JAN" "JAN" "JAN" ...
## $ geo      : chr [1:603] "AD" "AD" "AD" "AD" ...
## $ time_period: num [1:603] 2011 2012 2013 2016 2018 ...
## $ obs_value : num [1:603] 78115 78115 76246 71732 74794 ...
```

```
str(clean_eu_census)
```

```
## tibble [152,534 x 8] (S3: tbl_df/tbl/data.frame)
## $ country_code: chr [1:152534] "AT" "AT" "AT" "AT" ...
## $ sex         : chr [1:152534] "F" "F" "F" "F" ...
## $ age         : chr [1:152534] "Y_GE85" "Y_GE85" "Y_GE85" "Y_GE85" ...
## $ cas         : chr [1:152534] "ACT" "ACT" "ACT" "ACT" ...
## $ edu         : chr [1:152534] "ED1" "ED1" "ED1" "ED1" ...
## $ time        : num [1:152534] 2011 2011 2011 2011 2011 ...
## $ res_pop     : chr [1:152534] "500000-999999" "10000-99999" "200000-499999" "100000-199999" ...
## $ pop         : num [1:152534] 0 4 5 6 6 8 18 19 21 25 ...
```

```
str(clean_mp_cases)
```

```
## tibble [2,987 x 4] (S3: tbl_df/tbl/data.frame)
## $ daterep     : Date[1:2987], format: "2022-05-09" "2022-05-09" ...
## $ countryexp  : chr [1:2987] "Austria" "Belgium" "Bulgaria" "Croatia" ...
## $ countrycode: chr [1:2987] "AT" "BE" "BG" "HR" ...
## $ confcases   : num [1:2987] 0 0 0 0 0 0 0 0 0 0 ...
```

```
str(clean_wrld_region)
```

```
## tibble [51 x 7] (S3: tbl_df/tbl/data.frame)
## $ name      : chr [1:51] "ALAND ISLANDS" "ALBANIA" "ANDORRA" "AUSTRIA" ...
## $ alpha-2   : chr [1:51] "ax-248" "al-8" "ad-20" "at-40" ...
## $ country-code : num [1:51] 248 8 20 40 112 56 70 100 191 203 ...
## $ region    : chr [1:51] "Europe" "Europe" "Europe" "Europe" ...
## $ sub-region : chr [1:51] "Northern Europe" "Southern Europe" "Southern Europe" "Western Europe" ...
## $ region-code : num [1:51] 150 150 150 150 150 150 150 150 150 150 ...
## $ sub-region-code: num [1:51] 154 39 39 155 151 155 39 151 39 151 ...
```