Monkeypox

Susan Awor and Nakita Nicholls

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 monkeypox case data is from European Centre for Disease Prevention and Control. The population denominator, world region, and census data are from Eurostat.

How does the dataset relate to the group problem statement and question?

Problem statement 1: Is there a difference between monkeypox cases by region in the EU? We will be using the monkeypox data, population denominator and region data sets to help us find the difference between monkeypox cases by region in the EU.

Problem statement 2: 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

```r
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
```

```r
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()
```

```r
library(stringr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

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

```r
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.
```

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

```
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"))
```

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.

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.

daterep = date confcases = numeric sub_region =character age = character edu = character countrycode = character county_code = character geo = character

Identify the desired type/format for each variable—will you need to convert any columns to numeric or another type? We would have desired for the age variable to be numeric.

Utilize functions or resources in RStudio to determine the types of each data element (i.e. character, numeric, factor)

```
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 ...
```

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

We will use the daterep column from the clean_mp_cases data set to establish the number of confirmed cases per week or per month. daterep lists the date that the cases were confirmed on.

We used the summary function to give us a summary of monkeypox cases in the entire dataset

```
summary(clean_mp_cases$confcases)
```

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

We will the sub-region column to categorize the different EU countries by region.

We will use the eu census dataset to look at the age and education demographic(columns "age" and "edu") and their relationship to monkeypox cases.

We will use the countrycode/geo/country_code columns which are listed in all datasets to join our datasets as needed in the future milestone(s).

Please turn in a PDF created from an Rmd with the following components -

Subset rows or columns, as needed

```
unique(clean_mp_cases$countrycode)
```

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

```
unique(clean_pop_den$geo)
```

```
##  [1] "AD"         "AL"         "AM"         "AT"         "AZ"         "BA"
##  [7] "BE"         "BG"         "BY"         "CH"         "CY"         "CZ"
## [13] "DE"         "DK"         "EA18"       "EA19"       "EE"         "EL"
## [19] "ES"         "EU27_2020" "EU28"       "FI"         "FR"         "FX"
## [25] "GE"         "HR"         "HU"         "IE"         "IS"         "IT"
## [31] "LI"         "LT"         "LU"         "LV"         "MC"         "MD"
## [37] "ME"         "MK"         "MT"         "NL"         "NO"         "PL"
## [43] "PT"         "RO"         "RS"         "RU"         "SE"         "SI"
## [49] "SK"         "SM"         "TR"         "UA"         "UK"         "XK"
```

```
pop_den_geo <- filter(clean_pop_den,geo %in% c("AT", "BE", "BG", "HR", "CY", "CZ", "DK", "EE",
                                               "FI", "FR","DE", "EL", "HU","IS", "IE" ,"IT", "LV",
                                               "LT","LU", "MT", "NL", "NO","PL", "PT", "RO", "SK", "SI"
                        select(-indic_de)
```

```
date_mp_cases <- clean_mp_cases %>%
   group_by(countrycode, countryexp, month = floor_date(daterep, 'month')) %>%
    summarize(monthly_total_cases = sum(confcases))
```

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

```
join_df <- left_join(pop_den_geo, date_mp_cases, by = c("geo" = "countrycode"))
```

```
join_df$countryexp <- str_to_upper(join_df$countryexp)
```

```
regions_df <-clean_wrld_region %>%
  select(name, `sub-region`)
```

```
join_region_mp <- inner_join(regions_df, join_df, by = c("name" = "countryexp"))%>%
  mutate(rate = round(monthly_total_cases/obs_value * 100000,3))
```

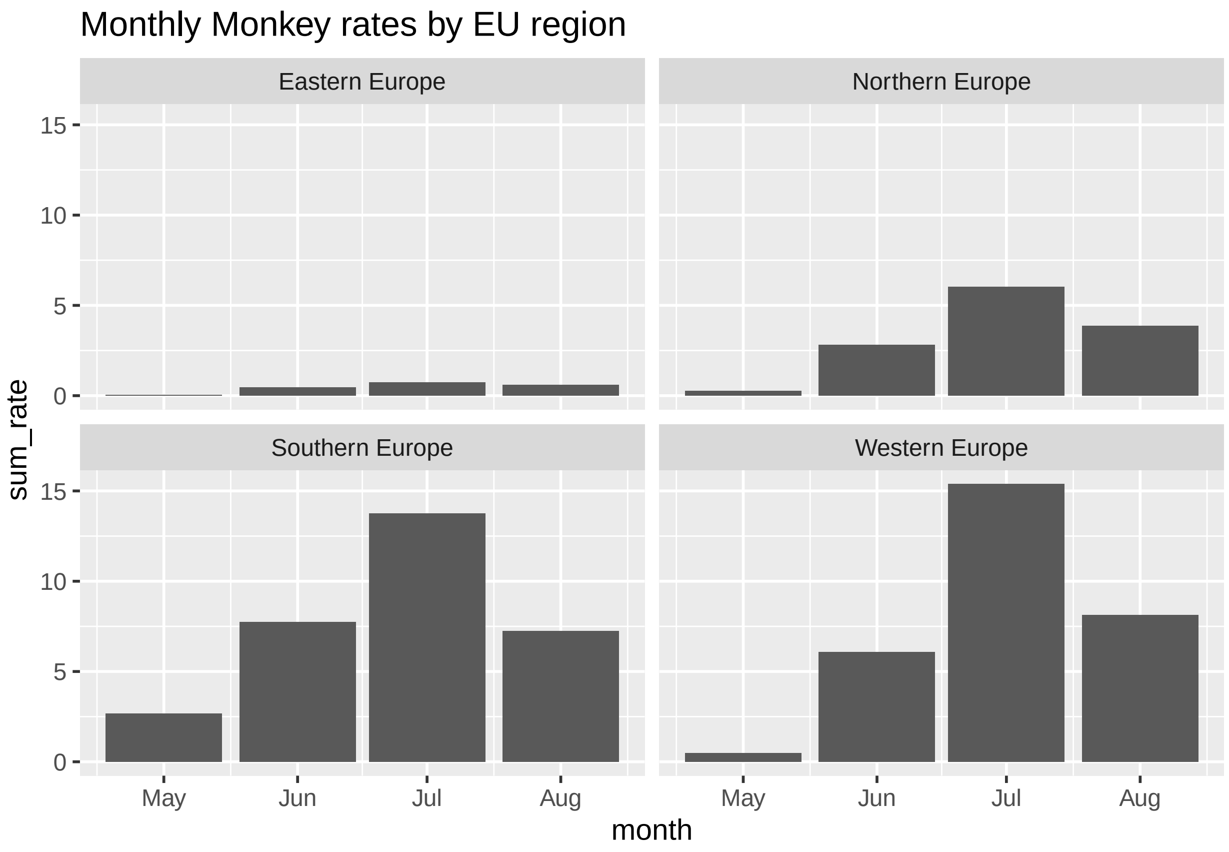
```
eu_cens_demo <- clean_eu_census %>%
  group_by(country_code, time, sex)%>%
  summarize(sum_pop = sum(pop))
```

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

```r
rate_region_mp <- join_region_mp %>%
  group_by(`sub-region`,month)%>%
  summarize(sum_rate = sum(rate))%>%
  ungroup()
```

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

```r
ggplot(rate_region_mp, aes(x=month, y=sum_rate)) +
  geom_bar(stat = "identity")+
  facet_wrap(~ `sub-region`) +
  labs(title = "Monthly Monkey rates by EU region ")
```



Data dictionary based on clean dataset (minimum 4 data elements), including: join_region_mp dataset Name :month Data type: DATE Description: A grouping of the of all the dates in the month and is listed as for example “2022-05-01”. It includes all the days in that respective month.

Name: monthly_total_cases Data type: numeric Description: The sum of monthly monkeypox cases per country.

Name: rate Data type: numeric Description: The rate is the number of monthly total cases divided by the obs_value per 100000, rounded to 3 decimal places.

rate_region_mp dataset

Name: sum_rate Data type: numeric Description: The sum monthly rate of monkeypox cases in each sub-region.

```r
str(join_region_mp)
```

```
## tibble [112 x 8] (S3: tbl_df/tbl/data.frame)
##  $ name               : chr [1:112] "AUSTRIA" "AUSTRIA" "AUSTRIA" "AUSTRIA" ...
##  $ sub-region         : chr [1:112] "Western Europe" "Western Europe" "Western Europe" "Western Europ
##  $ geo                : chr [1:112] "AT" "AT" "AT" "AT" ...
##  $ time_period        : num [1:112] 2022 2022 2022 2022 2022 ...
##  $ obs_value          : num [1:112] 8978929 8978929 8978929 8978929 11631136 ...
##  $ month              : Date[1:112], format: "2022-05-01" "2022-06-01" ...
##  $ monthly_total_cases: num [1:112] 2 36 96 97 16 127 351 177 0 3 ...
##  $ rate               : num [1:112] 0.022 0.401 1.069 1.08 0.138 ...
```

```r
# eu_census_demo_clean <- eu_cens_demo %>%
#   mutate(age_ord = recode(age, "Y_LT15" = "1",
#                              "Y15-29" = "2",
#                              "Y30-49" = "3",
#                              "Y50-64" = "4",
#                              "Y65-84" = "5",
#                              "Y_GE85" = "6"))

# eu_census_demo_clean <- eu_cens_demo %>%
#    mutate(age_cat = case_when(age == "Y_LT15" ~ "Younger than 15",
#                               age == "Y15-29" ~ "Age 15-29",
#                               age == "Y30-49" ~ "Age 30-49",
#                               age == "Y50-64" ~ "Age 50-64",
#                               age == "Y65-84" ~ "Age 65-85",
#                               age == "Y_GE85" ~ "85 and Older"))

# eu_census_demo_clean$age_cat <- factor(eu_census_demo_clean$age_cat,
# levels = c("Younger than 15","Age 15-29","Age 30-49","Age 50-64","Age 65-85","85 and Older"), #"2","3
# #labels = c("Staten Island","Manhattan",
# #"Bronx", "Queens", "Brooklyn"),
# ordered=TRUE)
```

```r
# join_cens_mp <- left_join(eu_cens_demo, join_region_mp, by = c("country_code" = "geo"))%>%
#   group_by(sex, monthly_total_cases)
```

Create new variables needed for analysis (minimum 2) New variables should be created based on existing columns; for example Calculating a rate Combining character strings If no new values are needed for final tables/graphs, please create 2 new variables anyway Clean variables needed for analysis (minimum 2) Examples Recode invalid values Handle missing fields Recode categories If not needed for final analysis, please create at least 2 new variables anyway Data dictionary based on clean dataset (minimum 4 data elements), including: Variable name Data type Description One or more tables with descriptive statistics for 4 data element PDF that is professionally prepared for presentation Each part of the milestone is clearly on one page (use to push to a new page) Only the necessary information is outputted (you should suppress, for example, entire data frame outputs) Use of headers and sub headers to create an organized document