milestone3

Moliehi Mokete

2022-11-02

```
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 ...
```

Subset rows or columns, as needed

```
joined_df <- merge(monkey_pox, pop_denominator, by.x = "country_code",
             by.y = "country_code", all.x = TRUE, all.y = FALSE)

sub_region_categories<-unique(joined_df$country_code)
sub_region_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_))
joined_df <- merge(joined_df, world_country_region, by.x = "country_code",
             by.y = "country_code", all.x = TRUE, all.y = FALSE)
#joined_df1 <- merge(joined_df, census_stats, by.x = "country_code",
            # by.y = "country_code", all.x = TRUE, all.y = FALSE)
```

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

```
monkey_pox<- monkey_pox%>% mutate(month_rep= months(date_rep))
```

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