

Case Study 2

AKSTA Statistical Computing

Tatzberger Jonas, Rasser Thomas, Grübling Alexander

03.05.2024

Exercises

1. Data Import

a.

Load in R the following data sets which you can find in TUWEL. For each data set, ensure that missing values are read in properly, that column names are unambiguous. Each data set should contain at the end only two columns: country and the variable.

Answer

```
check_number_of_rows <- function(expected_rowcount, expected_colcount, given_tibble) {
  real_rowcount = nrow(given_tibble)
  real_colcount = ncol(given_tibble)

  result <- assert_that(real_rowcount == expected_rowcount,
    msg = paste0("There should be ",
      expected_rowcount,
      " rows instead of ",
      real_rowcount))

  result <- assert_that(real_colcount == expected_colcount,
    msg = paste0("There should be ",
      real_colcount,
      " columns instead of ",
      expected_colcount))
}
```

```
# Import "rawdata_347.txt" for "net migration rate"

# Read the data file
file_path <- paste0(working_directory_path, "/data/rawdata_347.txt")
lines <- readLines(file_path)

# Convert lines to a tibble
migration_rate <- map_dfr(lines, function(line) {
  parts <- strsplit(trimws(line), "\\s{2,}")[[1]]
  tibble(
```

```

    Country = parts[2],
    Net_Migration_Rate = as.numeric(parts[3])
  )
}, .id = NULL) %>%
  filter(!is.na(Country), Country != "")

# Make sure all rows have been read
check_number_of_rows(227, 2, migration_rate)
head(migration_rate)

```

```

## # A tibble: 6 x 2
##   Country                Net_Migration_Rate
##   <chr>                  <dbl>
## 1 Syria                  27.1
## 2 British Virgin Islands 15.5
## 3 Luxembourg             13.3
## 4 Cayman Islands         13
## 5 Singapore              11.8
## 6 Anguilla                11.1

```

```

# Import "rawdata_343.txt" for "median age"

# Read the data file
file_path <- paste0(working_directory_path, "/data/rawdata_343.txt")
lines <- readLines(file_path)

# Convert lines to a tibble
median_age <- map_dfr(lines, function(line) {
  parts <- strsplit(trimws(line), "\\s{2,}")[[1]]
  tibble(
    Country = parts[2],
    Median_Age = as.numeric(parts[3])
  )
}, .id = NULL) %>%
  filter(!is.na(Country), Country != "")

# Make sure all rows have been read
check_number_of_rows(227, 2, median_age)
head(median_age)

```

```

## # A tibble: 6 x 2
##   Country                Median_Age
##   <chr>                  <dbl>
## 1 Monaco                 55.4
## 2 Japan                  48.6
## 3 Saint Pierre and Miquelon 48.5
## 4 Germany                47.8
## 5 Italy                   46.5
## 6 Andorra                 46.2

```

```

# Importing rawdata_373.csv (youth unemployment rate per country)
file_path <- paste0(working_directory_path, "/data/rawdata_373.csv")

```

```
youth_unemployment <- read_csv(file_path,
                                skip = 1, # Skip the predefined column names
                                col_names = c("Country", "Youth_Unemployment_Rate"),
                                col_types = c("c", "d")) %>%
  filter(!is.na(Country), Country != "")

# Make sure all rows have been read
check_number_of_rows(181, 2, youth_unemployment)
head(youth_unemployment)
```

```
## # A tibble: 6 x 2
##   Country      Youth_Unemployment_Rate
##   <chr>                <dbl>
## 1 French Polynesia      56.7
## 2 Kosovo                55.4
## 3 South Africa          53.4
## 4 Libya                 48.7
## 5 Eswatini              47.1
## 6 Saint Lucia           46.2
```

b.

Merge the data sets containing raw data using dplyr function on the unique keys. Keep the union of all observations in the tables. What key are you using for merging? Return the dimension of the merged data set.

Answer

```
# Merge the tibbles on the "Country" key
merged_country_data <- migration_rate %>%
  full_join(median_age, by = "Country") %>%
  full_join(youth_unemployment, by = "Country")

print(paste0("Dimensions: "))
```

```
## [1] "Dimensions: "
```

```
print(dim(merged_country_data))
```

```
## [1] 227  4
```

```
# Make sure the merge is correct
check_number_of_rows(227, 4, merged_country_data)
head(merged_country_data)
```

```
## # A tibble: 6 x 4
##   Country      Net_Migration_Rate Median_Age Youth_Unemployment_Rate
##   <chr>                <dbl>      <dbl>                <dbl>
## 1 Syria                27.1        23.5                 35.8
## 2 British Virgin Islands 15.5        37.2                  NA
```

```
## 3 Luxembourg          13.3      39.5      14.2
## 4 Cayman Islands      13       40.5      13.8
## 5 Singapore           11.8      35.6       9.1
## 6 Anguilla            11.1      35.7       NA
```

```
# empty values
na_value_countries <- merged_country_data %>%
  filter(apply(., 1, anyNA))

print(nrow(na_value_countries))
```

```
## [1] 46
```

```
head(na_value_countries)
```

```
## # A tibble: 6 x 4
##   Country                Net_Migration_Rate Median_Age Youth_Unemployment_Rate
##   <chr>                  <dbl>      <dbl>          <dbl>
## 1 British Virgin Islands    15.5      37.2            NA
## 2 Anguilla                  11.1      35.7            NA
## 3 Turks and Caicos Islands   8.9       34.6            NA
## 4 Aruba                     8.4       39.9            NA
## 5 Sint Maarten              6        41.1            NA
## 6 Djibouti                   5.1      24.9            NA
```

As expected, there are 46 rows with NA as value in the youth unemployment rate column, since the given rawdata file, has fewer countries listed

c.

You will acquire more country level information such as the classification of the country based on income. Such an information can be found at <https://datahelpdesk.worldbank.org/knowledgebase/articles/906519>. From there extract the classification for 2020 into low/lower-middle/upper-middle/high income countries.

Answer

```
# Importing rawdata from historical data file
file_path <- paste0(working_directory_path, "/data/OGHIST.xlsx")
full_excel_file <- read_excel(file_path,
  sheet = "Country Analytical History",
  range = cell_cols("A:AL"))
```

```
head(full_excel_file)
```

```
## # A tibble: 6 x 38
##   ...1 World Bank Analytical ~1 ...3 ...4 ...5 ...6 ...7 ...8 ...9 ...10
##   <chr> <chr>                  <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
## 1 <NA> (presented in World Dev~ <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
## 2 <NA> GNI per capita in US$ (~ <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
## 3 <NA> <NA>                  <NA> <NA> <NA> <NA> <NA> <NA> <NA>
## 4 <NA> Bank's fiscal year:    FY89 FY90 FY91 FY92 FY93 FY94 FY95 FY96
```

```
## 5 <NA> Data for calendar year : 1987 1988 1989 1990 1991 1992 1993 1994
## 6 <NA> Low income (L) <= 4~ <= 5~ <= 5~ <= 6~ <= 6~ <= 6~ <= 6~ <= 7~
## # i abbreviated name: 1: 'World Bank Analytical Classifications'
## # i 28 more variables: ...11 <chr>, ...12 <chr>, ...13 <chr>, ...14 <chr>,
## # ...15 <chr>, ...16 <chr>, ...17 <chr>, ...18 <chr>, ...19 <chr>,
## # ...20 <chr>, ...21 <chr>, ...22 <chr>, ...23 <chr>, ...24 <chr>,
## # ...25 <chr>, ...26 <chr>, ...27 <chr>, ...28 <chr>, ...29 <chr>,
## # ...30 <chr>, ...31 <chr>, ...32 <chr>, ...33 <chr>, ...34 <chr>,
## # ...35 <chr>, ...36 <chr>, ...37 <chr>, ...38 <chr>
```

```
# Get the range for the year columns
year_col_range <- 3:ncol(full_excel_file)

# Extract the years and change column names
classification <- full_excel_file[11:nrow(full_excel_file), ]
colnames(classification) <-
  c("ISO", "Country", full_excel_file[5, year_col_range])

# Convert classification columns to factors, replacing "." with NA
classification[, year_col_range] <-
  lapply(classification[, year_col_range],
    function(x) as.factor(replace(x, x == ".", NA)))

# Filter out rows where ISO is NA
classification <- classification[!is.na(classification$ISO), ]

head(classification)
```

```
## # A tibble: 6 x 38
## ISO Country '1987' '1988' '1989' '1990' '1991' '1992' '1993' '1994' '1995'
## <chr> <chr> <fct> <fct> <fct> <fct> <fct> <fct> <fct> <fct>
## 1 AFG Afghanis~ L L L L L L L L L
## 2 ALB Albania <NA> <NA> <NA> LM LM LM L L L
## 3 DZA Algeria UM UM LM LM LM LM LM LM LM LM
## 4 ASM American~ H H H UM UM UM UM UM UM UM
## 5 AND Andorra <NA> <NA> <NA> H H H H H H
## 6 AGO Angola <NA> LM LM LM LM LM LM LM LM L
## # i 27 more variables: '1996' <fct>, '1997' <fct>, '1998' <fct>, '1999' <fct>,
## # '2000' <fct>, '2001' <fct>, '2002' <fct>, '2003' <fct>, '2004' <fct>,
## # '2005' <fct>, '2006' <fct>, '2007' <fct>, '2008' <fct>, '2009' <fct>,
## # '2010' <fct>, '2011' <fct>, '2012' <fct>, '2013' <fct>, '2014' <fct>,
## # '2015' <fct>, '2016' <fct>, '2017' <fct>, '2018' <fct>, '2019' <fct>,
## # '2020' <fct>, '2021' <fct>, '2022' <fct>
```

```
# Get classification for 2020
classification_2020 <- classification[, c("ISO", "Country", "2020")]
colnames(classification_2020) <- c("ISO", "Country", "Classification_2020")
print(classification_2020[order(classification_2020$Country), ])
```

```
## # A tibble: 224 x 3
## ISO Country Classification_2020
## <chr> <chr> <fct>
## 1 AFG Afghanistan L
```

```
## 2 ALB Albania UM
## 3 DZA Algeria LM
## 4 ASM American Samoa UM
## 5 AND Andorra H
## 6 AGO Angola LM
## 7 ATG Antigua and Barbuda H
## 8 ARG Argentina UM
## 9 ARM Armenia UM
## 10 ABW Aruba H
## # i 214 more rows
```

d.

Merge this information to the data set in b.

1. What are the common variables? Can you merge using them? Why or why not?
2. A reliable merging for countries are ISO codes as they are standardized across data sources. Download the mapping of ISO codes to countries from <https://www.cia.gov/the-world-factbook/references/countrydata-codes/> and load it
3. Merge the data sets using the ISO codes.

Answer

```
# Check for countries which are in my merged list,
# but not in the classification list,
# if just merged by country name
missing_countries <- setdiff(merged_country_data$Country, classification_2020$Country)
print(sort(missing_countries))
```

```
## [1] "Anguilla"
## [2] "Brunei"
## [3] "Burma"
## [4] "Congo, Democratic Republic of the"
## [5] "Congo, Republic of the"
## [6] "Cook Islands"
## [7] "Cote d'Ivoire"
## [8] "Curacao"
## [9] "Czechia"
## [10] "Egypt"
## [11] "Faroe Islands"
## [12] "Gaza Strip"
## [13] "Guernsey"
## [14] "Hong Kong"
## [15] "Iran"
## [16] "Jersey"
## [17] "Korea, North"
## [18] "Korea, South"
## [19] "Kyrgyzstan"
## [20] "Laos"
## [21] "Macau"
## [22] "Macedonia"
```

```
## [23] "Micronesia, Federated States of"
## [24] "Montserrat"
## [25] "Russia"
## [26] "Saint Barthelemy"
## [27] "Saint Helena, Ascension, and Tristan da Cunha"
## [28] "Saint Kitts and Nevis"
## [29] "Saint Lucia"
## [30] "Saint Martin"
## [31] "Saint Pierre and Miquelon"
## [32] "Saint Vincent and the Grenadines"
## [33] "Sao Tome and Principe"
## [34] "Sint Maarten"
## [35] "Slovakia"
## [36] "Syria"
## [37] "Taiwan"
## [38] "Turkey"
## [39] "Venezuela"
## [40] "Virgin Islands"
## [41] "Wallis and Futuna"
## [42] "West Bank"
## [43] "Yemen"
```

If we would just merge by the country name, there would be over 40 countries missing, which are in my merged_data list. The reason could be e.g. different spelling, different order (Korea, South) or the countries are just not included. In summary, a lack of standardization hinders us in linking the data

```
# Importing country data codes from "Country Data Codes"
file_path <- paste0(working_directory_path, "/data/Country Data Codes.csv")
country_data_codes <- read_csv(file_path, show_col_types = FALSE)

# Get subset and rename columns
iso <- country_data_codes[, c("Name", "GENC")]
colnames(iso) <- c("Country", "ISO")
iso[iso == "-"] <- NA

# Merge iso into existing data set
merged_country_data_with_iso <- merged_country_data %>%
  full_join(iso, by = "Country")
merged_country_data <- merged_country_data_with_iso

head(merged_country_data)
```

```
## # A tibble: 6 x 5
##   Country          Net_Migration_Rate Median_Age Youth_Unemployment_R~1 ISO
##   <chr>              <dbl>         <dbl>          <dbl> <chr>
## 1 Syria              27.1           23.5           35.8 SYR
## 2 British Virgin Isl~ 15.5           37.2           NA   VGB
## 3 Luxembourg         13.3           39.5           14.2 LUX
## 4 Cayman Islands      13             40.5           13.8 CYM
## 5 Singapore          11.8           35.6            9.1 SGP
## 6 Anguilla            11.1           35.7           NA   AIA
## # i abbreviated name: 1: Youth_Unemployment_Rate
```

Even though we added most codes, there are still a few missing, which have to be added manually since the matching is not perfect

```
# List countries without iso
print(merged_country_data[is.na(merged_country_data$ISO), ])

## # A tibble: 10 x 5
##   Country          Net_Migration_Rate Median_Age Youth_Unemployment_R~1 ISO
##   <chr>              <dbl>         <dbl>         <dbl> <chr>
## 1 Macedonia          0.4           39          45.4 <NA>
## 2 Turkey             -4.3           32.2        20.2 <NA>
## 3 France, Metropoli~      NA           NA           NA <NA>
## 4 Myanmar            NA           NA           NA <NA>
## 5 United States Min~      NA           NA           NA <NA>
## 6 Virgin Islands (U~      NA           NA           NA <NA>
## 7 Virgin Islands (U~      NA           NA           NA <NA>
## 8 Western Samoa         NA           NA           NA <NA>
## 9 World               NA           NA           NA <NA>
## 10 Zaire              NA           NA           NA <NA>
## # i abbreviated name: 1: Youth_Unemployment_Rate
```

```
merged_country_data$ISO[merged_country_data$Country == "Turkey"] <- "TUR"
merged_country_data$ISO[merged_country_data$Country == "Macedonia"] <- "MKD"

# List countries without ISO
print(merged_country_data[is.na(merged_country_data$ISO), ])
```

```
## # A tibble: 8 x 5
##   Country          Net_Migration_Rate Median_Age Youth_Unemployment_R~1 ISO
##   <chr>              <dbl>         <dbl>         <dbl> <chr>
## 1 France, Metropolit~      NA           NA           NA <NA>
## 2 Myanmar            NA           NA           NA <NA>
## 3 United States Mino~      NA           NA           NA <NA>
## 4 Virgin Islands (UK)      NA           NA           NA <NA>
## 5 Virgin Islands (US)      NA           NA           NA <NA>
## 6 Western Samoa         NA           NA           NA <NA>
## 7 World               NA           NA           NA <NA>
## 8 Zaire              NA           NA           NA <NA>
## # i abbreviated name: 1: Youth_Unemployment_Rate
```

For the countries that are still without ISO, there are special political and regional reasons, which is why they cannot be added.

```
# Merge classification into existing data set
merged_country_data_with_class_2020 <- merged_country_data %>%
  full_join(classification_2020[, c("ISO", "Classification_2020")], by = "ISO")
merged_country_data <- merged_country_data_with_class_2020

head(merged_country_data)
```

```
## # A tibble: 6 x 6
##   Country          Net_Migration_Rate Median_Age Youth_Unemployment_R~1 ISO
```



```
##   <chr>                <dbl>      <dbl>                <dbl> <chr>
## 1 Syria                27.1       23.5                35.8 SYR
## 2 British Virgin Isl~  15.5       37.2                NA   VGB
## 3 Luxembourg           13.3       39.5                14.2 LUX
## 4 Cayman Islands       13         40.5                13.8 CYM
## 5 Singapore            11.8       35.6                 9.1 SGP
## 6 Anguilla             11.1       35.7                NA   AIA
## # i abbreviated name: 1: Youth_Unemployment_Rate
## # i 1 more variable: Classification_2020 <fct>
```

e.

Introduce into the data set information on continent for each country and subcontinent (region). You should find a way to gather this data. You can find an appropriate online resource, download the data and merge the information with the existing data set. Name the merged data set `df_vars`.

Answer

To add the requested region data, the following dataset has been used: <https://statisticstimes.com/geography/countries-by-continents.php>

```
# Importing continent and region data
file_path <- paste0(working_directory_path, "/data/continent_region_data.csv")
continent_region_data <- read_delim(file_path,
                                   delim = ";",
                                   locale = locale(encoding = "UTF-8"),
                                   show_col_types = FALSE)

# Get subset and rename columns
continent_region_data_subset <-
  continent_region_data[, c("ISO-alpha3", "Region 1", "Continent")]
colnames(continent_region_data_subset) <-
  c("ISO", "Region", "Continent")

# Merge into existing data set
merged_country_data_with_region <- merged_country_data %>%
  full_join(continent_region_data_subset, by = "ISO")

# Create new dataset
df_vars <- merged_country_data_with_region
head(df_vars)
```

```
## # A tibble: 6 x 8
##   Country                Net_Migration_Rate Median_Age Youth_Unemployment_R~1 ISO
##   <chr>                <dbl>      <dbl>                <dbl> <chr>
## 1 Syria                27.1       23.5                35.8 SYR
## 2 British Virgin Isl~  15.5       37.2                NA   VGB
## 3 Luxembourg           13.3       39.5                14.2 LUX
## 4 Cayman Islands       13         40.5                13.8 CYM
## 5 Singapore            11.8       35.6                 9.1 SGP
## 6 Anguilla             11.1       35.7                NA   AIA
## # i abbreviated name: 1: Youth_Unemployment_Rate
## # i 3 more variables: Classification_2020 <fct>, Region <chr>, Continent <chr>
```

f.

Discuss on the tidyness of the data set `df_vars`. What are the observational units, what are the variables? What can be considered fixed vs measured variables? Tidy the data if needed.

Answer

```
str(df_vars)
```

```
## tibble [293 x 8] (S3: tbl_df/tbl/data.frame)
##   $ Country           : chr [1:293] "Syria" "British Virgin Islands" "Luxembourg" "Cayman Islands" ...
##   $ Net_Migration_Rate : num [1:293] 27.1 15.5 13.3 13 11.8 11.1 10.6 8.9 8.4 8.3 ...
##   $ Median_Age         : num [1:293] 23.5 37.2 39.5 40.5 35.6 35.7 32.9 34.6 39.9 55.4 ...
##   $ Youth_Unemployment_Rate: num [1:293] 35.8 NA 14.2 13.8 9.1 NA 5.3 NA NA 26.6 ...
##   $ ISO                : chr [1:293] "SYR" "VGB" "LUX" "CYM" ...
##   $ Classification_2020 : Factor w/ 4 levels "H","L","LM","UM": 2 1 1 1 1 NA 1 1 1 1 ...
##   $ Region             : chr [1:293] "Western Asia" "Caribbean" "Western Europe" "Caribbean" ...
##   $ Continent          : chr [1:293] "Asia" "North America" "Europe" "North America" ...
```

```
# save the data for further usage
write_csv(df_vars, path = "../data/df_vars.csv")
```

```
## Warning: The 'path' argument of 'write_csv()' is deprecated as of readr 1.4.0.
## i Please use the 'file' argument instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```

Observational units vs variables*

In the given data frame (`df_vars`), the observational units are the countries. Each row represents a different country.

The variables in the data frame are:

1. **Country**: The name of the country (character/string variable).
2. **Net_Migration_Rate**: The net migration rate for the country (numeric variable).
3. **Median_Age**: The median age of the population in the country (numeric variable).
4. **Youth_Unemployment_Rate**: The youth unemployment rate in the country (numeric variable).
5. **ISO**: The ISO code for the country (character/string variable).
6. **Classification_2020**: The classification of the country, which is a factor with four levels: "H", "L", "LM", and "UM" (factor variable).
7. **Region**: The region to which the country belongs (character/string variable).
8. **Continent**: The continent where the country is located (character/string variable).

as can be seen via `str(df_vars)`

Fixed vs measured variables

Fixed variables are those that do not vary within the dataset. In this case, the fixed variables are: - **Country**
- **ISO** - **Region** - **Continent**

These variables are essentially identifiers or labels for the observational units (countries) and do not change within the dataset.

Measured variables are those that are measured for each observational unit. In this case, the measured variables are: - Net_Migration_Rate - Median_Age - Youth_Unemployment_Rate - Classification_2020

These variables represent quantitative or categorical measurements or characteristics of the countries.

Tidiness

A tidy dataset has the following characteristics:

1. Each variable forms a column.
2. Each observation forms a row.
3. Each type of observational unit forms a table.

Analyzing df_vars:

1. Each variable forms a column:
 - Each variable (Country, Net_Migration_Rate, ...) forms a separate column, which satisfies this principle.
2. Each observation forms a row:
 - In this case, each row represents an observation, which is a country. Therefore, this principle is also satisfied.
3. Each type of observational unit forms a table:
 - The dataset contains only one type of observational unit, which is countries. All the observations (rows) and variables (columns) pertain to countries, so this principle is satisfied as well.

Based on this analysis, the dataset appears to be in a tidy format.

2. Data analysis - Part 1

g.

Make a frequency table for the status variable in the merged data set. Briefly comment on the results.

```
df_vars %>% mutate(status = fct_infreq(Classification_2020)) %>% count(status)
```

```
##   status  n
## 1      H 80
## 2     UM 57
## 3     LM 55
## 4      L 27
## 5   <NA> 74
```

ADD COMMENT!

h.

What is the distribution of income status in the different continents? Compute the absolute frequencies as well as the relative frequency of status within each continent. Briefly comment on the results.

```
by_cont <- df_vars %>% group_by(Continent)
income_status <- by_cont %>% count(Classification_2020) %>% mutate(abs = n, freq = n/sum(n))
income_status$n <- NULL
head(income_status)
```

```
## # A tibble: 6 x 4
## # Groups:   Continent [2]
##   Continent Classification_2020 abs   freq
##   <chr>      <chr>          <int> <dbl>
## 1 Africa    H              1 0.0167
## 2 Africa    L             23 0.383
## 3 Africa    LM            23 0.383
## 4 Africa    UM              7 0.117
## 5 Africa    <NA>           6 0.1
## 6 Antarctica <NA>           1 1
```

Answer

i.

From h. identify the countries which are the only ones in their respective group. Explain in few words the output.

```
abs_1 <- income_status %>% filter(abs==1) %>% select(Continent, Classification_2020)
abs_1 <- left_join(abs_1, df_vars)
```

```
## Joining with 'by = join_by(Continent, Classification_2020)'
```

```
abs_1
```

```
## # A tibble: 5 x 8
## # Groups:   Continent [5]
##   Continent Classification_2020 Country Net_Migration_Rate Median_Age
##   <chr>      <chr>          <chr>          <dbl>      <dbl>
## 1 Africa    H              Seychelles          1        36.8
## 2 Antarctica <NA>          Antarctica         NA         NA
## 3 Europe    LM            Ukraine            2.3        41.2
## 4 South America LM        Bolivia           -0.3        25.3
## 5 <NA>      UM            <NA>              NA         NA
## # i 3 more variables: Youth_Unemployment_Rate <dbl>, ISO <chr>, Region <chr>
```

Answer

j.

For each continent count the number of sub-regions in the data set. How granular are the subcontinents that you employ in the analysis?

```
by_cont %>% summarise(sub_regions = n_distinct(Region))
```

```
## # A tibble: 8 x 2
##   Continent      sub_regions
##   <chr>          <int>
## 1 Africa          5
## 2 Antarctica      1
## 3 Asia            5
## 4 Europe          4
## 5 North America    3
## 6 Oceania          4
## 7 South America    1
## 8 <NA>            1
```

Answer

k.

Look at the frequency distribution of income status in the subregions of North- and South-Americas. Comment on the results.

```
na <- df_vars %>% filter(Continent=="North America")
sa <- df_vars %>% filter(Continent=="South America")
americas <- full_join(na, sa)
```

```
## Joining with 'by = join_by(Country, Net_Migration_Rate, Median_Age,
## Youth_Unemployment_Rate, ISO, Classification_2020, Region, Continent)'
```

```
by_region <- americas %>% group_by(Region)
income_status_am <- by_region %>% count(Classification_2020) %>% mutate(abs = n, freq = n/sum(n))
income_status_am$n <- NULL
head(income_status_am)
```

```
## # A tibble: 6 x 4
## # Groups:   Region [2]
##   Region      Classification_2020    abs    freq
##   <chr>      <chr>                <int>  <dbl>
## 1 Caribbean    H                14 0.5
## 2 Caribbean    LM                 1 0.0357
## 3 Caribbean    UM                 7 0.25
## 4 Caribbean    <NA>              6 0.214
## 5 Central America LM                 4 0.5
## 6 Central America UM                 4 0.5
```

Answer

l.

Dig deeper into the low-middle income countries of the Americas. Which ones are they? Are they primarily small island states in the Caribbean? Comment.

```
americas_LM <- income_status_am %>% filter(Classification_2020=="LM")
americas_LM <- left_join(americas_LM, df_vars)
```

```
## Joining with 'by = join_by(Region, Classification_2020)'
```

```
americas_LM
```

```
## # A tibble: 6 x 10
## # Groups:   Region [3]
##   Region Classification_2020 abs   freq Country Net_Migration_Rate Median_Age
##   <chr>      <chr>          <int>  <dbl> <chr>          <dbl>      <dbl>
## 1 Caribb~ LM              1 0.0357 Haiti          -1.9       24.1
## 2 Centra~ LM              4 0.5     Belize          -1         23.9
## 3 Centra~ LM              4 0.5     Hondur~        -1.4       24.4
## 4 Centra~ LM              4 0.5     Nicara~        -2.4       27.3
## 5 Centra~ LM              4 0.5     El Sal~        -4.8       27.7
## 6 South ~ LM              1 0.0625 Bolivia        -0.3       25.3
## # i 3 more variables: Youth_Unemployment_Rate <dbl>, ISO <chr>, Continent <chr>
```

Answer

3. Data analysis - Part 2

m.

Create a table of average values for median age, youth unemployment rate and net migration rate separated into income status. Make sure that in the output, the ordering of the income classes is proper (i.e., L, LM, UM, H or the other way around). Briefly comment the results.

Answer

n.

Look also at the standard deviation instead of the mean in m. Do you gain additional insights? Briefly comment the results.

Answer

o.

Repeat the analysis in m. for each income status and continent combination. Discuss the results.

Answer

p.

Identify countries which are doing well in terms of both youth unemployment and net migration rate (in the top 25% of their respective continent in terms of net migration rate and in the bottom 25% of their respective continent in terms of youth unemployment).

Answer

r.

Export the final data set to a csv with “;” separator and “.” as a symbol for missing values; no rownames should be included in the csv. Upload the .csv to TUWEL together with your .Rmd and .html (or .pdf).

Answer