Case Study 2

AKSTA Statistical Computing

2025-04-02

The .Rmd **and** *.html (or .pdf) should be uploaded in TUWEL by the deadline. Refrain from using explanatory comments in the R code chunks but write them as text instead. Points will be deducted if the submitted file is not in a decent form.*

DISCLAIMER: In case students did not contribute equally, include a disclaimer stating what each student's contribution was.

The CIA World Factbook provides intelligence on various aspects of 266 world entities, including history, people, government, economy, energy, geography, environment, communications, transportation, military, terrorism, and transnational issues. This case study involves analyzing world data from 2020, focusing on:

- **Education Expenditure (% of GDP)**
- **Youth Unemployment Rate (15-24 years)**
- **Net Migration Rate** (difference between the number of people entering and leaving a country per 1,000 persons)

The data was sourced from the CIA World Factbook Archives. You are required to use `dplyr` for data manipulation, while any package can be used for importing data.

Tasks:

a. Data Import and Cleaning

Load the following datasets from TUWEL and ensure that missing values are handled correctly and column names are clear. Each dataset should ultimately contain only two columns: **country** and the respective variable. Note that some data sets also contain information on the year when the value was last updated.

```r
# Load the tidyverse package collection
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(dbplyr)
```

```
##
## Attache Paket: 'dbplyr'
##
```

```
## Die folgenden Objekte sind maskiert von 'package:dplyr':
##
##     ident, sql
```

```
# Set path to data folder
data_path <- "data"
# data_path <- "src/Case_Study_2/data" anscheinend für knit
```

- `rawdata_369.txt` which contains the (estimated) public expenditure on education as a percent of GDP. *Pay attention! The delimiter is 2 or more white spaces (one space would not work as it would separate country names which contain a space); you have to skip the first two lines.*

```
# Read lines from file, skip first two lines (title and separator)
edu_lines <- readLines(file.path(data_path, "rawdata_369.txt"))[-c(1, 2)]

# Convert to tibble with fixed-width fields
edu_exp <- tibble(raw = edu_lines) %>%
  separate(raw, into = c("index", "country", "education_expenditure", "year"),
           sep = "\\s{2,}", extra = "merge", fill = "right") %>%
  select(country, education_expenditure, year) %>%
  mutate(
    country = str_to_lower(country),
    country = str_remove_all(country, "-|-|-"),       # remove various dashes
    country = str_squish(country),                    # normalize whitespace
    education_expenditure = as.numeric(education_expenditure),
    year = as.integer(year)
  )
```

- `rawdata_373.csv` which contains the (estimated) youth unemployment rate (15-24) per country

```
# Read youth unemployment data from CSV

youth_unemp <- read_csv(
  file.path(data_path, "rawdata_373.csv"),
  show_col_types = FALSE
) %>%
  select(
    country = 1,
    youth_unemployment = 2
  ) %>%
  mutate(
    country = str_to_lower(country),  # make country names lowercase
    country = str_replace_all(country, "timor\\-?leste", "timorleste")  # unify country name, because o
  )
```

- `rawdata_347.txt` which contains (estimated) net migration rate per country.

```
# Read lines from file, skip first two lines (header and separator)
mig_lines <- readLines(file.path(data_path, "rawdata_347.txt"))[-c(1, 2)]

# Convert to tibble and split into components
net_mig <- tibble(raw = mig_lines) %>%
  separate(raw, into = c("index", "country", "net_migration", "year_raw"),
           sep = "\\s{2,}", extra = "merge", fill = "right") %>%
  select(country, net_migration, year_raw) %>%
  mutate(
    country = str_to_lower(country),                # make country names lowercase
    country = str_remove_all(country, "-|-|-"),      # remove dashes
    country = str_squish(country),                # normalize whitespace
    net_migration = as.numeric(net_migration),      # convert to numeric
    year = str_extract(year_raw, "\\d{4}"),     # extract the 4-digit year
    year = as.integer(year)
  ) %>%
  select(-year_raw)
```

b. Merging Raw Data

Merge the datasets using `dplyr` on a unique key and retain the union of all observations.

```
# Rename year columns before joining
edu_exp_clean <- edu_exp %>%
  rename(edu_year = year)

net_mig_clean <- net_mig %>%
  rename(mig_year = year)

# Merge all datasets on 'country'
merged_data <- youth_unemp %>%
  full_join(edu_exp_clean, by = "country") %>%
  full_join(net_mig_clean, by = "country") %>%
  arrange(country)
```

- What key are you using for merging? -> Country
- Return the dimensions of the merged dataset.

```
# Check structure and dimensions of the merged dataset
dim(merged_data)
```

```
## [1] 227   6
```

```
merged_data
```

```
## # A tibble: 227 x 6
##    country       youth_unemployment education_expenditure edu_year net_migration
##    <chr>                      <dbl>                 <dbl>    <int>         <dbl>
##  1 afghanistan                 17.6                   4.1     2017          -0.1
##  2 albania                     31.9                   3.6     2017          -3.3
##  3 algeria                     39.3                  NA         NA          -0.9
##  4 american sam~               NA                    NA         NA         -26.1
##  5 andorra                     NA                     3.2     2019           0
##  6 angola                      39.4                   3.4     2010          -0.2
##  7 anguilla                    NA                    NA         NA          11.1
##  8 antigua and ~               NA                    NA         NA           2.1
##  9 argentina                   23.7                   5.5     2017          -0.1
## 10 armenia                     36.3                   2.7     2017          -5.5
## # i 217 more rows
## # i 1 more variable: mig_year <int>
```

c. Enriching Data with Income Classification

Obtain country income classification (low, lower-middle, upper-middle, high) from the World Bank and merge it with the dataset.

- Identify common variables between datasets. Can they be used for merging? Why or why not? The only common variable between our existing merged_data and the data from the world bank, is the country name. Although the latter doesn't use lowercase names for the countries.

```
library(readxl)
world_bank_income <- read_excel(file.path(data_path, "CLASS.xlsx"),
                    sheet = "List of economies",
                    col_names = TRUE) %>%
                    select(Code, Economy, `Income group`)

world_bank_income <- world_bank_income %>%
  mutate(Economy = tolower(Economy),
         Economy = str_trim(Economy))

merged_data_income <- merged_data %>%
  left_join(world_bank_income, by = c("country" = "Economy"))

data_without_income <- merged_data_income %>%
  filter(is.na(`Income group`))

nrow(data_without_income)
```

```
## [1] 43
```

As seen above merging World Bank data with our existing data, results in 43 countries not receiving the corresponding data. That doesn't mean those countries don't exist in the World Bank data, but that they carry a different name.

- Since ISO codes are standardized, download and use the CIA country data codes for merging. Make sure you are not losing any of the countries in your original data set when merging.

```
country_codes <- read.csv(file.path(data_path, "Country_Data_Codes.csv"), stringsAsFactors = FALSE) %>%
  select(GENC, Name)

clean_income_data <- world_bank_income %>%
  full_join(country_codes, by = c("Code" = "GENC")) %>%
  mutate(Economy = ifelse(!is.na(Name), Name, Economy)) %>%
  select(-Name)

clean_income_data <- clean_income_data %>%
  mutate(Economy = tolower(Economy),
         Economy = str_trim(Economy),
         Economy = ifelse(Economy == "north macedonia", "macedonia", Economy),
         Economy = ifelse(Economy == "timor-leste", "timorleste", Economy),
         Economy = ifelse(Economy == "turkey (turkiye)", "turkey", Economy),
         Economy = ifelse(Economy == "guinea-bissau", "guineabissau", Economy)
         )

merged_data_income_clean <- merged_data %>%
  left_join(clean_income_data, by = c("country" = "Economy")) %>%
  filter(is.na(Code) | Code != "XKS")

data_without_income_clean <- merged_data_income_clean %>%
  filter(is.na(`Income group`))

nrow(data_without_income_clean)
```

```
## [1] 12
```

This time we receive a much lower result of countries, which don't have a corresponding value.

d. Adding Geographical Information

Introduce continent and subcontinent (or region) data for each country.

- Find and download an appropriate online resource.

  The United Nations Statistics Division offers a data set fitting the criteria

```
un_stat <- read.csv(file.path(data_path, "UNSD-Methodology.csv"), sep = ";", stringsAsFactors = FALSE) %
  rename(
    Continent = `Region.Name`,
    Subcontinent = `Sub.region.Name`,
    iso3 = `ISO.alpha3.Code`
  ) %>%
  select(Continent, Subcontinent, iso3)
```

- Merge this information into the dataset, naming the final dataset `df_vars`. Make sure you are not losing any of the countries in your original data set when merging.

```
df_vars <- merged_data_income_clean %>%
  left_join(un_stat, by = c("Code" = "iso3"))

summary(df_vars)
```

```
##    country          youth_unemployment education_expenditure    edu_year   
##  Length:227         Min.   : 0.40      Min.   : 1.200        Min.   :2010  
##  Class :character   1st Qu.: 8.70      1st Qu.: 3.200        1st Qu.:2017  
##  Mode  :character   Median :14.90      Median : 4.300        Median :2017  
##                     Mean   :18.15      Mean   : 4.451        Mean   :2017  
##                     3rd Qu.:25.90      3rd Qu.: 5.375        3rd Qu.:2018  
##                     Max.   :56.70      Max.   :12.800        Max.   :2019  
##                     NA's   :46         NA's   :57            NA's   :57    
##  net_migration        mig_year        Code           Income group      
##  Min.   :-88.700   Min.   :2020   Length:227         Length:227        
##  1st Qu.: -2.650   1st Qu.:2020   Class :character   Class :character  
##  Median : -0.300   Median :2020   Mode  :character   Mode  :character  
##  Mean   : -1.083   Mean   :2020                                        
##  3rd Qu.:  1.150   3rd Qu.:2020                                        
##  Max.   : 27.100   Max.   :2020                                        
##                                                                        
##   Continent         Subcontinent      
##  Length:227         Length:227        
##  Class :character   Class :character  
##  Mode  :character   Mode  :character  
##                                       
##                                       
##                                       
##                                       
```

e. Data Tidiness and Summary Statistics

- Evaluate the tidiness of `df_vars` • Column headers are values, not variable names. • Multiple types of observational units are stored in the same table. • A single observational unit is stored in multiple tables.

- Variables are stored in both rows and columns. (observational units, variables, fixed vs. measured variables). Make adjustments to tidy the data, if necessary.

```
library("tidyverse")
library(dplyr)
str(df_vars)
```

```
## tibble [227 x 10] (S3: tbl_df/tbl/data.frame)
##  $ country               : chr [1:227] "afghanistan" "albania" "algeria" "american samoa" ...
##  $ youth_unemployment    : num [1:227] 17.6 31.9 39.3 NA NA 39.4 NA NA 23.7 36.3 ...
##  $ education_expenditure: num [1:227] 4.1 3.6 NA NA 3.2 3.4 NA NA 5.5 2.7 ...
##  $ edu_year              : int [1:227] 2017 2017 NA NA 2019 2010 NA NA 2017 2017 ...
##  $ net_migration         : num [1:227] -0.1 -3.3 -0.9 -26.1 0 -0.2 11.1 2.1 -0.1 -5.5 ...
##  $ mig_year              : int [1:227] 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ...
##  $ Code                  : chr [1:227] "AFG" "ALB" "DZA" "ASM" ...
##  $ Income group          : chr [1:227] "Low income" "Upper middle income" "Upper middle income" "High
##  $ Continent             : chr [1:227] "Asia" "Europe" "Africa" "Oceania" ...
##  $ Subcontinent          : chr [1:227] "Southern Asia" "Southern Europe" "Northern Africa" "Polynesia
```

- Multiple variables are stored in one column.

```
for (col in names(df_vars)) {
  cat("Unique values in column", col, ":\n")
  print(unique(df_vars[[col]]))
  cat("\n")
}
```

```
## Unique values in column country :
##   [1] "afghanistan"
##   [2] "albania"
##   [3] "algeria"
##   [4] "american samoa"
##   [5] "andorra"
##   [6] "angola"
##   [7] "anguilla"
##   [8] "antigua and barbuda"
##   [9] "argentina"
##  [10] "armenia"
##  [11] "aruba"
##  [12] "australia"
##  [13] "austria"
##  [14] "azerbaijan"
##  [15] "bahamas, the"
##  [16] "bahrain"
##  [17] "bangladesh"
##  [18] "barbados"
##  [19] "belarus"
##  [20] "belgium"
##  [21] "belize"
##  [22] "benin"
##  [23] "bermuda"
##  [24] "bhutan"
##  [25] "bolivia"
##  [26] "bosnia and herzegovina"
##  [27] "botswana"
##  [28] "brazil"
##  [29] "british virgin islands"
##  [30] "brunei"
##  [31] "bulgaria"
##  [32] "burkina faso"
##  [33] "burma"
##  [34] "burundi"
##  [35] "cabo verde"
##  [36] "cambodia"
##  [37] "cameroon"
##  [38] "canada"
##  [39] "cayman islands"
##  [40] "central african republic"
##  [41] "chad"
##  [42] "chile"
##  [43] "china"
##  [44] "colombia"
##  [45] "comoros"
##  [46] "congo, democratic republic of the"
##  [47] "congo, republic of the"
##  [48] "cook islands"
##  [49] "costa rica"
##  [50] "cote d'ivoire"
##  [51] "croatia"
##  [52] "cuba"
##  [53] "curacao"
##  [54] "cyprus"
##  [55] "czechia"
##  [56] "denmark"
##  [57] "djibouti"
##  [58] "dominica"
##  [59] "dominican republic"
##  [60] "ecuador"
##  [61] "egypt"
##  [62] "el salvador"
##  [63] "equatorial guinea"
##  [64] "eritrea"
##  [65] "estonia"
##  [66] "eswatini"
##  [67] "ethiopia"
##  [68] "faroe islands"
##  [69] "fiji"
##  [70] "finland"
##  [71] "france"
##  [72] "french polynesia"
##  [73] "gabon"
##  [74] "gambia, the"
##  [75] "gaza strip"
##  [76] "georgia"
##  [77] "germany"
##  [78] "ghana"
##  [79] "gibraltar"
##  [80] "greece"
##  [81] "greenland"
##  [82] "grenada"
##  [83] "guam"
##  [84] "guatemala"
##  [85] "guernsey"
##  [86] "guinea"
##  [87] "guineabissau"
##  [88] "guyana"
##  [89] "haiti"
##  [90] "honduras"
##  [91] "hong kong"
##  [92] "hungary"
##  [93] "iceland"
##  [94] "india"
##  [95] "indonesia"
##  [96] "iran"
##  [97] "iraq"
##  [98] "ireland"
##  [99] "isle of man"
## [100] "israel"
## [101] "italy"
## [102] "jamaica"
## [103] "japan"
## [104] "jersey"
## [105] "jordan"
## [106] "kazakhstan"
## [107] "kenya"
## [108] "kiribati"
## [109] "korea, north"
## [110] "korea, south"
## [111] "kosovo"
## [112] "kuwait"
## [113] "kyrgyzstan"
## [114] "laos"
## [115] "latvia"
## [116] "lebanon"
## [117] "lesotho"
## [118] "liberia"
## [119] "libya"
## [120] "liechtenstein"
## [121] "lithuania"
## [122] "luxembourg"
## [123] "macau"
## [124] "macedonia"
## [125] "madagascar"
## [126] "malawi"
## [127] "malaysia"
## [128] "maldives"
## [129] "mali"
## [130] "malta"
## [131] "marshall islands"
## [132] "mauritania"
## [133] "mauritius"
## [134] "mexico"
## [135] "micronesia, federated states of"
## [136] "moldova"
## [137] "monaco"
## [138] "mongolia"
## [139] "montenegro"
## [140] "montserrat"
## [141] "morocco"
## [142] "mozambique"
## [143] "namibia"
## [144] "nauru"
## [145] "nepal"
## [146] "netherlands"
## [147] "new caledonia"
## [148] "new zealand"
## [149] "nicaragua"
## [150] "niger"
## [151] "nigeria"
## [152] "northern mariana islands"
## [153] "norway"
## [154] "oman"
## [155] "pakistan"
## [156] "palau"
## [157] "panama"
## [158] "papua new guinea"
## [159] "paraguay"
## [160] "peru"
## [161] "philippines"
## [162] "poland"
## [163] "portugal"
## [164] "puerto rico"
## [165] "qatar"
## [166] "romania"
## [167] "russia"
## [168] "rwanda"
## [169] "saint barthelemy"
## [170] "saint helena, ascension, and tristan da cunha"
## [171] "saint kitts and nevis"
## [172] "saint lucia"
## [173] "saint martin"
## [174] "saint pierre and miquelon"
## [175] "saint vincent and the grenadines"
## [176] "samoa"
## [177] "san marino"
## [178] "sao tome and principe"
## [179] "saudi arabia"
## [180] "senegal"
## [181] "serbia"
## [182] "seychelles"
## [183] "sierra leone"
## [184] "singapore"
## [185] "sint maarten"
## [186] "slovakia"
## [187] "slovenia"
## [188] "solomon islands"
## [189] "somalia"
## [190] "south africa"
## [191] "south sudan"
## [192] "spain"
## [193] "sri lanka"
## [194] "sudan"
## [195] "suriname"
## [196] "sweden"
## [197] "switzerland"
## [198] "syria"
## [199] "taiwan"
## [200] "tajikistan"
## [201] "tanzania"
## [202] "thailand"
## [203] "timorleste"
## [204] "togo"
## [205] "tonga"
## [206] "trinidad and tobago"
## [207] "tunisia"
## [208] "turkey"
## [209] "turkmenistan"
## [210] "turks and caicos islands"
## [211] "tuvalu"
## [212] "uganda"
## [213] "ukraine"
## [214] "united arab emirates"
## [215] "united kingdom"
## [216] "united states"
## [217] "uruguay"
## [218] "uzbekistan"
## [219] "vanuatu"
## [220] "venezuela"
## [221] "vietnam"
## [222] "virgin islands"
## [223] "wallis and futuna"
## [224] "west bank"
## [225] "yemen"
## [226] "zambia"
## [227] "zimbabwe"
##
## Unique values in column youth_unemployment :
##   [1] 17.6 31.9 39.3   NA 39.4 23.7 36.3 11.8  9.4 13.4 25.8  5.3 12.8 29.6 10.6
##  [16] 15.8 15.3  5.6 29.3 10.7  6.9 33.8 36.0 28.5 28.9 12.7  8.7  2.0  2.9 27.8
##  [31]  1.1  6.3 11.1 13.8 18.1 18.5 19.5 20.6  5.5  6.1 20.2  6.7 13.5  7.9  9.6
##  [46] 47.1 25.2 15.4 17.0 20.8 56.7 35.7 13.1 42.2 29.9  6.2  9.1 39.9 29.4  5.0
##  [61]  1.0 21.5 10.2 22.5 16.5 27.6 25.6 10.1  7.2 32.2 24.2  3.6 35.6  3.8  7.4
##  [76] 17.1 55.4 14.2 18.2 12.2 34.4  2.3 48.7 45.4 40.5 10.5 15.9 16.9 11.0 15.2
##  [91] 23.9 18.9 26.6 16.8 22.2 38.0 21.4 38.4 11.5  8.5  0.7  9.7 13.7  7.8 14.5
## [106] 14.7 11.7 20.3  0.4 16.2 46.2 27.4 28.8  8.1 29.7 11.6 14.9  8.8  1.3 53.4
## [121] 38.6 34.3 21.0 35.8  3.9  3.7 13.2 35.0 14.8 17.9 11.3  8.6 25.9 14.6 24.5
## [136] 24.0
##
## Unique values in column education_expenditure :
##  [1]  4.1  3.6   NA  3.2  3.4  5.5  2.7  5.1  5.4  2.5  2.3  1.3  4.4  4.8  6.4
## [16]  7.6  2.9  1.5  6.9  7.3  6.3  1.9  5.2  2.2  3.1  5.3  1.2  4.5  3.5  4.7
## [31]  7.0  3.3  3.9 12.8  4.9  5.8  7.8  5.6  5.0  7.1  2.4  4.0  2.1  2.8  6.1
## [46]  3.8  7.7  2.6  4.3  6.0  4.2 12.5  8.8  7.9  6.8  4.6  5.7  9.9  6.5  6.6
## [61]  5.9
##
## Unique values in column edu_year :
##  [1] 2017   NA 2019 2010 2016 2018 2014 2011 2015 2013 2012
##
## Unique values in column net_migration :
##   [1]  -0.1  -3.3  -0.9 -26.1   0.0  -0.2  11.1   2.1  -5.5   8.4   8.1   3.6
##  [13]  10.6  -3.0  -0.3   0.7   4.8  -1.0   0.3   1.6  -0.4   2.9  15.5   2.3
##  [25]  -0.6  -1.4  -0.8   5.6  13.0  -2.3 -29.9   0.8   1.2  -3.7  -1.3   7.6
##  [37]   2.8   5.1  -5.3  -2.7  -4.8 -11.6  -3.1  -6.8  -6.2   2.6   1.1  -0.7
##  [49]   3.9  -1.6  -4.7   0.1   1.5   0.9  -6.0  -2.6 -11.0  -1.7   1.9  -3.8
##  [61]  -1.9   1.7   1.3   3.3  -1.1  -0.5   5.2   3.2  -9.4 -11.3   0.4  -2.8
##  [73]  -1.8  -5.0  -5.9 -88.7  -6.1  -2.9   4.9  13.3 -12.7  -3.9   6.6  -4.5
##  [85] -20.9  -9.0   8.3  -4.9   3.8   8.0  -2.4 -15.4   4.0 -14.1   6.5  -7.7
##  [97]  -7.2  -8.1  -7.9   4.7   1.0  -1.2  11.8   6.0   0.2   7.0   0.5   4.6
## [109]  27.1 -17.9  -5.4  -4.3   8.9  -6.5  -3.5   2.5   3.0  -3.4  -7.5  -4.2
##
## Unique values in column mig_year :
## [1] 2020
##
## Unique values in column Code :
##   [1] "AFG" "ALB" "DZA" "ASM" "AND" "AGO" "AIA" "ATG" "ARG" "ARM" "ABW" "AUS"
##  [13] "AUT" "AZE" "BHS" "BHR" "BGD" "BRB" "BLR" "BEL" "BLZ" "BEN" "BMU" "BTN"
##  [25] "BOL" "BIH" "BWA" "BRA" "VGB" "BRN" "BGR" "BFA" "MMR" "BDI" "CPV" "KHM"
##  [37] "CMR" "CAN" "CYM" "CAF" "TCD" "CHL" "CHN" "COL" "COM" "COD" "COG" "COK"
##  [49] "CRI" "CIV" "HRV" "CUB" "CUW" "CYP" "CZE" "DNK" "DJI" "DMA" "DOM" "ECU"
##  [61] "EGY" "SLV" "GNQ" "ERI" "EST" "SWZ" "ETH" "FRO" "FJI" "FIN" "FRA" "PYF"
##  [73] "GAB" "GMB" "XGZ" "GEO" "DEU" "GHA" "GIB" "GRC" "GRL" "GRD" "GUM" "GTM"
##  [85] "GGY" "GIN" "GNB" "GUY" "HTI" "HND" "HKG" "HUN" "ISL" "IND" "IDN" "IRN"
##  [97] "IRQ" "IRL" "IMN" "ISR" "ITA" "JAM" "JPN" "JEY" "JOR" "KAZ" "KEN" "KIR"
## [109] "PRK" "KOR" "XKX" "KWT" "KGZ" "LAO" "LVA" "LBN" "LSO" "LBR" "LBY" "LIE"
## [121] "LTU" "LUX" "MAC" "MKD" "MDG" "MWI" "MYS" "MDV" "MLI" "MLT" "MHL" "MRT"
## [133] "MUS" "MEX" "FSM" "MDA" "MCO" "MNG" "MNE" "MSR" "MAR" "MOZ" "NAM" "NRU"
## [145] "NPL" "NLD" "NCL" "NZL" "NIC" "NER" "NGA" "MNP" "NOR" "OMN" "PAK" "PLW"
## [157] "PAN" "PNG" "PRY" "PER" "PHL" "POL" "PRT" "PRI" "QAT" "ROU" "RUS" "RWA"
## [169] "BLM" "SHN" "KNA" "LCA" "MAF" "SPM" "VCT" "WSM" "SMR" "STP" "SAU" "SEN"
## [181] "SRB" "SYC" "SLE" "SGP" "SXM" "SVK" "SVN" "SLB" "SOM" "ZAF" "SSD" "ESP"
## [193] "LKA" "SDN" "SUR" "SWE" "CHE" "SYR" "TWN" "TJK" "TZA" "THA" "TLS" "TGO"
## [205] "TON" "TTO" "TUN" "TUR" "TKM" "TCA" "TUV" "UGA" "UKR" "ARE" "GBR" "USA"
## [217] "URY" "UZB" "VUT" "VEN" "VNM" "VIR" "WLF" "XWB" "YEM" "ZMB" "ZWE"
##
## Unique values in column Income group :
## [1] "Low income"          "Upper middle income" "High income"
## [4] "Lower middle income" NA
##
## Unique values in column Continent :
## [1] "Asia"     "Europe"   "Africa"   "Oceania"  "Americas" NA
##
## Unique values in column Subcontinent :
##  [1] "Southern Asia"                   "Southern Europe"
##  [3] "Northern Africa"                 "Polynesia"
##  [5] "Sub-Saharan Africa"              "Latin America and the Caribbean"
##  [7] "Western Asia"                    "Australia and New Zealand"
##  [9] "Western Europe"                  "Eastern Europe"
## [11] "Northern America"                "South-eastern Asia"
## [13] "Eastern Asia"                    "Northern Europe"
## [15] "Melanesia"                       NA
## [17] "Micronesia"                      "Central Asia"
```

- Create a frequency table for the income status variable and briefly interpret the results.

```r
table(df_vars$`Income group`)
```

```
##
##         High income          Low income Lower middle income Upper middle income
##                  85                  26                  50                  54
```

- Analyze the distribution of income status across continents by computing absolute and relative frequencies. Comment on the findings.

```r
for (cont in unique(df_vars$Continent)) {
  cat("Absolute Income group counts for Continent:", cont, "\n")
  print(table(df_vars$`Income group`[df_vars$Continent == cont]))
  cat("\n")
}
```

```
## Absolute Income group counts for Continent: Asia
##
##         High income          Low income Lower middle income Upper middle income
##                  14                   4                  17                  14
##
## Absolute Income group counts for Continent: Europe
##
##         High income Upper middle income
##                  38                   8
##
## Absolute Income group counts for Continent: Africa
##
##         High income          Low income Lower middle income Upper middle income
##                   1                  22                  23                   8
##
## Absolute Income group counts for Continent: Oceania
##
##         High income Lower middle income Upper middle income
##                   9                   6                   4
##
## Absolute Income group counts for Continent: Americas
##
##         High income Lower middle income Upper middle income
##                  22                   4                  19
##
## Absolute Income group counts for Continent: NA
## < table of extent 0 >
```

```r
for (cont in unique(df_vars$Continent)) {
  cat("Relative Income group distribution for Continent:", cont, "\n")
  print(prop.table(table(df_vars$`Income group`[df_vars$Continent == cont])))
  cat("\n")
}
```

```
## Relative Income group distribution for Continent: Asia
##
##         High income          Low income Lower middle income Upper middle income
##          0.28571429          0.08163265          0.34693878          0.28571429
##
## Relative Income group distribution for Continent: Europe
##
##         High income Upper middle income
##            0.826087            0.173913
##
## Relative Income group distribution for Continent: Africa
##
##         High income          Low income Lower middle income Upper middle income
##          0.01851852          0.40740741          0.42592593          0.14814815
##
## Relative Income group distribution for Continent: Oceania
##
##         High income Lower middle income Upper middle income
##           0.4736842           0.3157895           0.2105263
##
## Relative Income group distribution for Continent: Americas
##
##         High income Lower middle income Upper middle income
##          0.48888889          0.08888889          0.42222222
##
## Relative Income group distribution for Continent: NA
## numeric(0)
```

Europe has the highest relative amount of high income countries, followed by Oceania and America. The lowest is Africa.

- Using the distribution of income status across continents, identify which countries are the only ones in their income group across the continent. Discuss briefly.

```
na.omit(df_vars$country[df_vars$Continent == 'Africa' & df_vars$`Income group` == 'High income'])
```

```
## [1] "seychelles"
## attr(,"na.action")
## [1] 1 2 4 5
## attr(,"class")
## [1] "omit"
```

Since in the absolute Table only Africa has a 1 in any position, we could easly filtet for the high income country in africa ## f. Further Summary Statistics and Insights

- Create a table of average (mean and median) values for expenditure, 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 and any differences between the mean and median.

```
df_vars %>%
  mutate(`Income group` = factor(`Income group`,
                                 levels = c("High income", "Upper middle income",
                                            "Lower middle income", "Low income"))) %>%
  group_by(`Income group`) %>%
  summarise(
    avg_youth_unemp = mean(youth_unemployment, na.rm = TRUE),
    avg_expend = mean(education_expenditure, na.rm = TRUE),
    avg_mig_rate = mean(net_migration, na.rm = TRUE),
    median_youth_unemp = median(youth_unemployment, na.rm = TRUE),
    median_expend = median(education_expenditure, na.rm = TRUE),
    median_mig_rate = median(net_migration, na.rm = TRUE)
  )
```

```
## # A tibble: 5 x 7
##   `Income group`      avg_youth_unemp avg_expend avg_mig_rate median_youth_unemp
##   <fct>                         <dbl>      <dbl>        <dbl>              <dbl>
## 1 High income                    16.5       4.56        1.46                12.7
## 2 Upper middle income            22.8       4.53       -2.16                20.2
## 3 Lower middle income            15.9       4.47       -3.95                13.8
## 4 Low income                     15.4       3.58       -0.538               13.1
## 5 <NA>                           33         6.02       -3.49                42.2
## # i 2 more variables: median_expend <dbl>, median_mig_rate <dbl>
```

- Look at the standard deviation and the interquartile range of the variables per income status instead of the location statistics above. Do you gain additional insights? Briefly comment the results.

```
df_vars %>%
  mutate(`Income group` = factor(`Income group`,
                                 levels = c("High income", "Upper middle income",
                                            "Lower middle income", "Low income"))) %>%
  group_by(`Income group`) %>%
  summarise(
    sd_youth_unemp = sd(youth_unemployment, na.rm = TRUE),
    sd_expend = sd(education_expenditure, na.rm = TRUE),
    sd_mig_rate = sd(net_migration, na.rm = TRUE),
    IQR_youth_unemp = IQR(youth_unemployment, na.rm = TRUE),
    IQR_expend = IQR(education_expenditure, na.rm = TRUE),
    IQR_mig_rate = IQR(net_migration, na.rm = TRUE)
  )
```

```
## # A tibble: 5 x 7
##   `Income group` sd_youth_unemp sd_expend sd_mig_rate IQR_youth_unemp IQR_expend
##   <fct>                   <dbl>     <dbl>       <dbl>           <dbl>      <dbl>
## 1 High income              10.3      1.47        6.28            14.6       1.8
## 2 Upper middle ~           13.6      1.84        3.81            17.5       1.85
## 3 Lower middle ~           11.5      2.21       12.8             14.9       2.4
## 4 Low income               12.6      1.71        6.15            16.9       2.48
## 5 <NA>                     15.9      1.87        9.62            13.8       1.03
## # i 1 more variable: IQR_mig_rate <dbl>
```

The results show that high-income countries have the most consistent patterns in youth unemployment and education expenditure, reflecting stable systems and lower variability. Lower-middle income countries display the highest volatility in net migration, suggesting greater economic or political instability. While low-income countries show less variability overall, this may reflect limited resources or structural constraints rather than true consistency. The NA group stands out with unusually high variation, indicating data quality issues or unclassified outliers that should be treated separately.

- Extend the analysis of the statistics median and IQR to **each income status and continent combination**. Play around with displaying the resulting table. Use `pivot_longer()` and/or `pivot_wider()` to generate different outputs. Discuss the results as well as the readability of the different tables.

```r
df_vars %>%
  mutate(`Income group` = factor(`Income group`,
                                 levels = c("High income", "Upper middle income",
                                            "Lower middle income", "Low income"))) %>%
  group_by(`Income group`,`Continent`) %>%
  summarise(
    sd_youth_unemp = sd(youth_unemployment, na.rm = TRUE),
    sd_expend = sd(education_expenditure, na.rm = TRUE),
    sd_mig_rate = sd(net_migration, na.rm = TRUE),
    IQR_youth_unemp = IQR(youth_unemployment, na.rm = TRUE),
    IQR_expend = IQR(education_expenditure, na.rm = TRUE),
    IQR_mig_rate = IQR(net_migration, na.rm = TRUE)
  )
```

```
## `summarise()` has grouped output by 'Income group'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 23 x 8
## # Groups:   Income group [5]
##    `Income group` Continent sd_youth_unemp sd_expend sd_mig_rate IQR_youth_unemp
##    <fct>          <chr>              <dbl>     <dbl>       <dbl>           <dbl>
##  1 High income    Africa             NA       NA           NA               0
##  2 High income    Americas            8.44     1.44         6.82           15.5
##  3 High income    Asia                8.83     1.54         4.30            9.28
##  4 High income    Europe              8.41     1.48         3.72            7.52
##  5 High income    Oceania            18.0      0.849       11.7            22.2
##  6 High income    <NA>               NA       NA           NA              NA
##  7 Upper middle ~ Africa              9.57     1.73         1.77            8.15
##  8 Upper middle ~ Americas           10.5      2.35         2.74           12.2
##  9 Upper middle ~ Asia               10.1      0.676        3.54           13.4
## 10 Upper middle ~ Europe             12.8      0.910        3.72           16.3
## # i 13 more rows
## # i 2 more variables: IQR_expend <dbl>, IQR_mig_rate <dbl>
```

```r
df_vars %>%
  mutate(`Income group` = factor(`Income group`,
                                 levels = c("High income", "Upper middle income",
                                            "Lower middle income", "Low income"))) %>%
  group_by(`Income group`, Continent) %>%
  summarise(
    sd_youth_unemp = sd(youth_unemployment, na.rm = TRUE),
    sd_expend = sd(education_expenditure, na.rm = TRUE),
    sd_mig_rate = sd(net_migration, na.rm = TRUE),
    IQR_youth_unemp = IQR(youth_unemployment, na.rm = TRUE),
    IQR_expend = IQR(education_expenditure, na.rm = TRUE),
    IQR_mig_rate = IQR(net_migration, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  pivot_wider(names_from = Continent, values_from = c(sd_youth_unemp, sd_expend, sd_mig_rate,
                                                      IQR_youth_unemp, IQR_expend, IQR_mig_rate))
```

```
## # A tibble: 5 x 37
##   `Income group`      sd_youth_unemp_Africa sd_youth_unemp_Americas
##   <fct>                               <dbl>                   <dbl>
## 1 High income                         NA                       8.44
## 2 Upper middle income                  9.57                   10.5
## 3 Lower middle income                 13.0                     1.91
## 4 Low income                          12.4                    NA
## 5 <NA>                                NA                      NA
## # i 34 more variables: sd_youth_unemp_Asia <dbl>, sd_youth_unemp_Europe <dbl>,
## #   sd_youth_unemp_Oceania <dbl>, sd_youth_unemp_NA <dbl>,
## #   sd_expend_Africa <dbl>, sd_expend_Americas <dbl>, sd_expend_Asia <dbl>,
## #   sd_expend_Europe <dbl>, sd_expend_Oceania <dbl>, sd_expend_NA <dbl>,
## #   sd_mig_rate_Africa <dbl>, sd_mig_rate_Americas <dbl>,
## #   sd_mig_rate_Asia <dbl>, sd_mig_rate_Europe <dbl>,
## #   sd_mig_rate_Oceania <dbl>, sd_mig_rate_NA <dbl>, ...
```

All in all, the table with the continents in the header is more readable for comparing within and between income groups and continents. The drwaback is that the table becomes somewhat to long to read clearly on one screen, without scroling. Otherwise it is apparent from the results that oceania has the highest youth unemployment rat, as well as the lowest expenditure.

- Identify countries performing well in terms of both **youth unemployment** and **net migration rate** (top 25% in net migration and bottom 25% in youth unemployment within their continent).

```r
# Create a list to store the results
best_countries <- list()

# Loop through each unique continent
for (cont in unique(df_vars$Continent)) {

  # Subset data for this continent
  sub_df <- df_vars[df_vars$Continent == cont, ]

  # Define thresholds: low youth unemployment and high net migration
  youth_unemp_threshold <- quantile(sub_df$youth_unemployment, 0.25, na.rm = TRUE)
  net_mig_threshold <- quantile(sub_df$net_migration, 0.75, na.rm = TRUE)

  # Filter for countries meeting both criteria
  good_performers <- na.omit(sub_df$country[
    sub_df$youth_unemployment <= youth_unemp_threshold &
    sub_df$net_migration >= net_mig_threshold
  ])

  # Store in the list
  best_countries[[cont]] <- good_performers
}

# View result
best_countries
```

```
## $Asia
## [1] "bahrain"              "kazakhstan"           "macau"
## [4] "qatar"                "united arab emirates"
## attr(,"na.action")
## [1] 2 4 7 9
## attr(,"class")
## [1] "omit"
##
## $Europe
## [1] "malta"       "norway"      "switzerland"
## attr(,"na.action")
## [1] 1 2 3 7 8
## attr(,"class")
## [1] "omit"
##
## $Africa
## [1] "benin"         "cote d'ivoire" "guinea"        "madagascar"
## [5] "togo"
## attr(,"na.action")
## [1]  2  4  5  6  8 10 11 13
## attr(,"class")
## [1] "omit"
##
## $Oceania
## [1] "palau"            "papua new guinea"
## attr(,"na.action")
## [1] 1 2 5 6
## attr(,"class")
## [1] "omit"
##
## $Americas
## [1] "united states"
## attr(,"na.action")
##  [1]  1  2  3  4  5  6  7  8  9 10 12
## attr(,"class")
## [1] "omit"
##
## $<NA>
## character(0)
## attr(,"na.action")
##   [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##  [19]  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##  [37]  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##  [55]  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##  [73]  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##  [91]  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
## [109] 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
## [127] 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
## [145] 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162
## [163] 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
## [181] 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198
## [199] 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216
## [217] 217 218 219 220 221 222 223 224 225 226 227
## attr(,"class")
## [1] "omit"
```

g. Conditional Probabilities

Estimate the following based on the observed frequencies in the data:

- What is the (posterior or conditional) probability that a European country belongs to the high income group? What is the prior probability that a country belongs to the high income group?

```r
# P(European country)
p_european_country <- sum(df_vars$Continent == "Europe", na.rm = TRUE) / nrow(df_vars)
```

```r
# P(High income and European)
p_high_and_europe <- sum(df_vars$Continent == "Europe" & df_vars$`Income group` == "High income", na.rm

# Posterior: P(High income | Europe)
p_high_income_given_europe <- p_high_and_europe / p_european_country

p_high_income_given_europe
```

```
## [1] 0.7916667
```

Given a country is european, the probability it belongs to a high income group is 84%

- Given a country has high youth unemployment (above %25), what is the probability that it also has negative net migration?

```r
# P(Youth unemployment > 25%)
p_high_youth_unemp <- sum(df_vars$youth_unemployment > 25, na.rm = TRUE) / nrow(df_vars)

# P(Youth unemployment > 25% AND net migration < 0)
p_youth_unemp_and_neg_mig <- sum(df_vars$youth_unemployment > 25 & df_vars$net_migration < 0, na.rm = TI

# P(Negative migration | High youth unemployment)
p_neg_mig_given_high_youth_unemp <- p_youth_unemp_and_neg_mig / p_high_youth_unemp

p_neg_mig_given_high_youth_unemp
```

```
## [1] 0.6122449
```

The probability of a high youth unemployment the probabilty is that it has negative net migration is 61% ## h. Simpson's Paradox Analysis

Investigate whether an overall trend in youth unemployment rate in the high and low income groups reverses when analyzed at the continent level. E.g., does the youth unemployment rate appear lower in low-income countries overall, but higher when controlling for continent? Explain the results and possible reasons behind this paradox.

To explore whether Simpson's Paradox occurs in this dataset, we analyze youth unemployment rates across income groups at two levels:

1. **Overall (global) level**
2. **Within each continent**

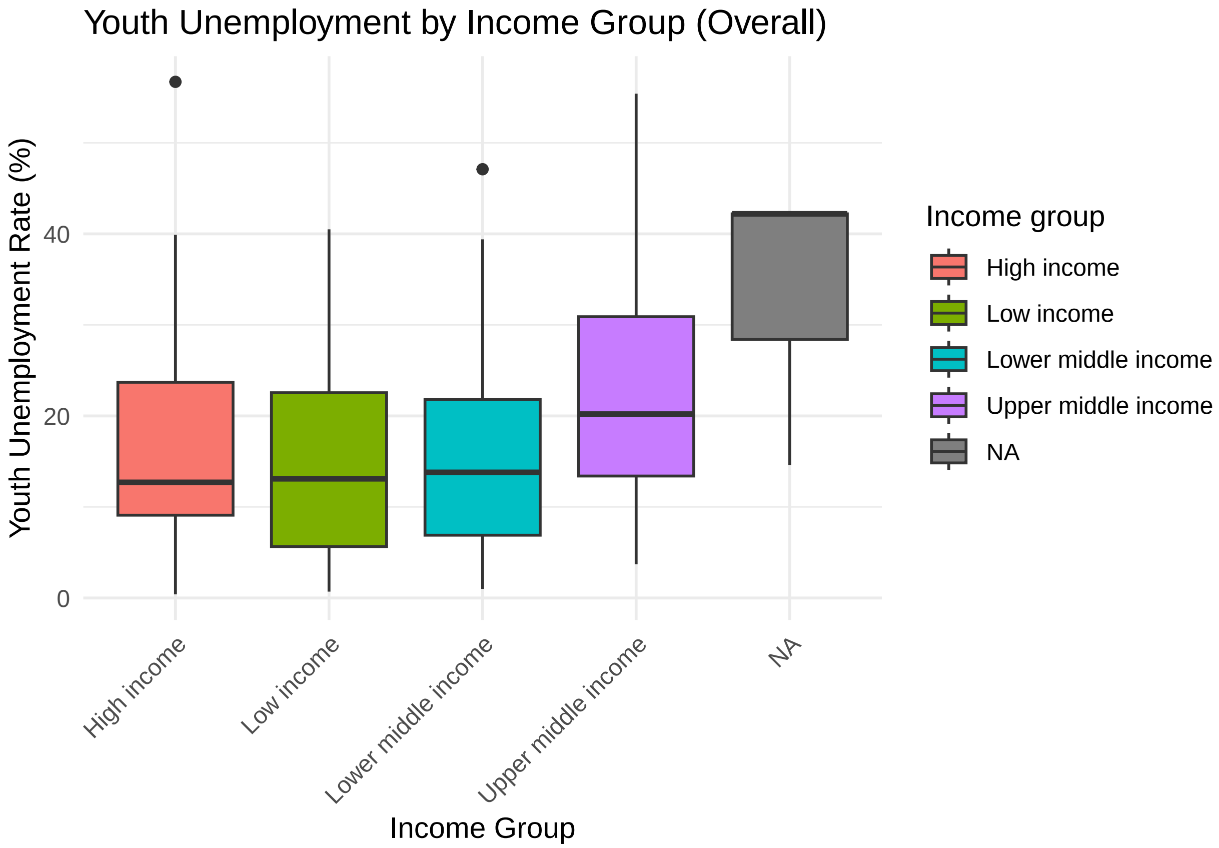
This allows us to check whether the observed global trend reverses when controlling for continent.

---

1. Overall Youth Unemployment by Income Group

```r
ggplot(df_vars, aes(x = `Income group`, y = youth_unemployment, fill = `Income group`)) +
  geom_boxplot() +
  theme_minimal() +
  labs(
    title = "Youth Unemployment by Income Group (Overall)",
    x = "Income Group",
    y = "Youth Unemployment Rate (%)"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: Removed 46 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```

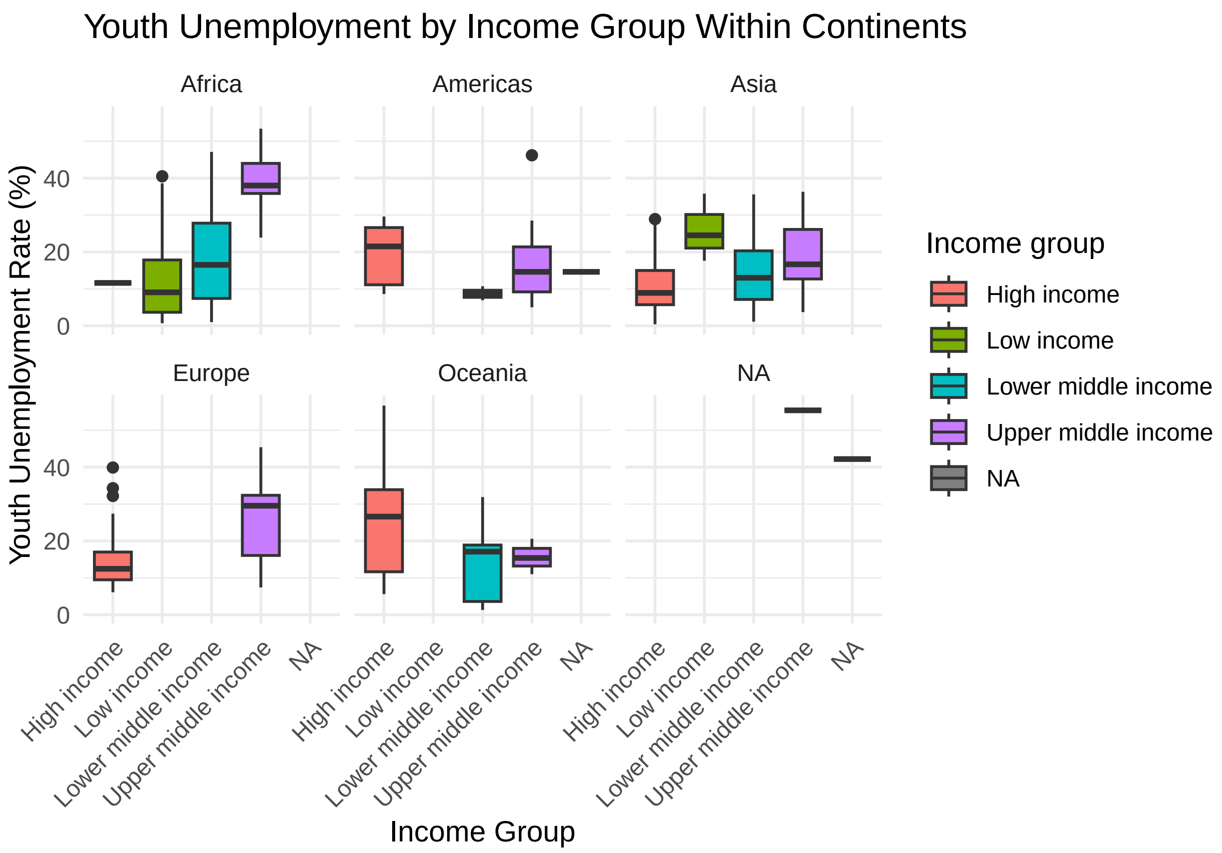


The overall boxplot suggests that, on average, youth unemployment appears lower in low-income countries compared to higher-income groups. The figure also shows that the median of youth unemployment rate is approximately the same in the categories Low-, Lower middle- and High income.

2. Youth Unemployment by continent

```
ggplot(df_vars, aes(x = `Income group`, y = youth_unemployment, fill = `Income group`)) +
  geom_boxplot() +
  facet_wrap(~ Continent) +
  theme_minimal() +
  labs(
    title = "Youth Unemployment by Income Group Within Continents",
    x = "Income Group",
    y = "Youth Unemployment Rate (%)"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: Removed 46 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



The within-continent trends differ from the global pattern. In regions such as Africa and Asia, low-income countries tend to show **higher youth unemployment** than high-income countries, **reversing** the global trend.

This indicates a potential case of **Simpson's Paradox**: while the aggregated data suggests that low-income countries have lower unemployment, this is due to their concentration in regions with generally lower overall unemployment. When controlling for continent, the underlying association between income level and unemployment becomes visible, revealing that lower income is often linked to higher youth unemployment within regions.

Thus, this analysis provides evidence for Simpson's Paradox in the relationship between income group and youth unemployment.

i. Data Export

Export the final tidy dataset from e. as a **CSV** with:`;` as a separator; `.` representing missing values; no row names included. Upload the `.csv` to TUWEL, together with the submission.

```
write.table(df_vars,
            file = "results/df_vars_output.csv",
            sep = ";",
            na = ".",
            row.names = FALSE,
            quote = TRUE)
```