

## 2\_cross-continent\_analysis

Lucas Salamuni - 7429674

2025-08-21

### Packages

```
# I. Load required packages
packages <- c("dplyr", "knitr", "tinytex", "readxl", "tidyr", "fastDummies",
             "sandwich", "lmtest", "estimatr", "purrr", "tibble", "writexl",
             "readr", "stringr", "sf", "rnatruralearth", "dplyr", "units",
             "igraph", "countrycode", "geosphere", "haven", "glmnet",
             "gravity", "modelsummary", "sessioninfo")

# II. Install packages if not already installed
if(sum(as.numeric(!packages %in% installed.packages())) != 0){
  instalador <- packages[!packages %in% installed.packages()]
  for(i in 1:length(instalador)) {
    install.packages(instalador, dependencies = T)
    break()}
  sapply(packages, require, character = T)
} else {
  sapply(packages, require, character = T)
}
```

##	dplyr	knitr	tinytex	readxl	tidyr
##	TRUE	TRUE	TRUE	TRUE	TRUE
##	fastDummies	sandwich	lmtest	estimatr	purrr
##	TRUE	TRUE	TRUE	TRUE	TRUE
##	tibble	writexl	readr	stringr	sf
##	TRUE	TRUE	TRUE	TRUE	TRUE
##	rnatruralearth	dplyr	units	igraph	countrycode
##	TRUE	TRUE	TRUE	TRUE	TRUE
##	geosphere	haven	glmnet	gravity	modelsummary
##	TRUE	TRUE	TRUE	TRUE	TRUE
##	sessioninfo				
##	TRUE				

---

### Session info

```
session_info()
```

```
## - Session info -----
## setting value
## version R version 4.4.1 (2024-06-14 ucrt)
## os Windows 11 x64 (build 26100)
## system x86_64, mingw32
## ui RTerm
## language (EN)
## collate Portuguese_Brazil.utf8
## ctype Portuguese_Brazil.utf8
## tz Europe/Berlin
## date 2025-08-22
## pandoc 3.4 @ C:/Program Files/RStudio/resources/app/bin/quarto/bin/tools/ (via rmarkdown)
##
## - Packages -----
## package * version date (UTC) lib source
## bdsmatrix 1.3-7 2024-03-02 [1] CRAN (R 4.4.0)
## boot 1.3-30 2024-02-26 [2] CRAN (R 4.4.1)
## cellranger 1.1.0 2016-07-27 [1] CRAN (R 4.4.1)
## censReg 0.5-38 2024-05-20 [1] CRAN (R 4.4.3)
## class 7.3-22 2023-05-03 [2] CRAN (R 4.4.1)
## classInt 0.4-10 2023-09-05 [1] CRAN (R 4.4.1)
## cli 3.6.3 2024-06-21 [1] CRAN (R 4.4.1)
## coda 0.19-4.1 2024-01-31 [1] CRAN (R 4.4.1)
## codetools 0.2-20 2024-03-31 [2] CRAN (R 4.4.1)
## collapse 2.0.15 2024-07-08 [1] CRAN (R 4.4.1)
## countrycode * 1.6.1 2025-03-31 [1] CRAN (R 4.4.3)
## data.table 1.15.4 2024-03-30 [1] CRAN (R 4.4.1)
## DBI 1.2.3 2024-06-02 [1] CRAN (R 4.4.1)
## digest 0.6.36 2024-06-23 [1] CRAN (R 4.4.1)
## dplyr * 1.1.4 2023-11-17 [1] CRAN (R 4.4.1)
## e1071 1.7-16 2024-09-16 [1] CRAN (R 4.4.1)
## emmeans 1.10.3 2024-07-01 [1] CRAN (R 4.4.1)
## estimability 1.5.1 2024-05-12 [1] CRAN (R 4.4.1)
## estimatr * 1.0.4 2024-03-31 [1] CRAN (R 4.4.2)
## evaluate 0.24.0 2024-06-10 [1] CRAN (R 4.4.1)
## fansi 1.0.6 2023-12-08 [1] CRAN (R 4.4.1)
## fastDummies * 1.7.5 2025-01-20 [1] CRAN (R 4.4.3)
## fastmap 1.2.0 2024-05-15 [1] CRAN (R 4.4.1)
## forcats 1.0.0 2023-01-29 [1] CRAN (R 4.4.1)
## foreach 1.5.2 2022-02-02 [1] CRAN (R 4.4.1)
## Formula 1.2-5 2023-02-24 [1] CRAN (R 4.4.0)
## generics 0.1.3 2022-07-05 [1] CRAN (R 4.4.1)
## geosphere * 1.5-20 2024-10-04 [1] CRAN (R 4.4.3)
## glmML 1.1.7 2024-09-20 [1] CRAN (R 4.4.3)
## glmnet * 4.1-8 2023-08-22 [1] CRAN (R 4.4.1)
## glue 1.7.0 2024-01-09 [1] CRAN (R 4.4.1)
## gravity * 1.1 2023-05-02 [1] CRAN (R 4.4.3)
## haven * 2.5.4 2023-11-30 [1] CRAN (R 4.4.1)
## hms 1.1.3 2023-03-21 [1] CRAN (R 4.4.1)
## htmltools 0.5.8.1 2024-04-04 [1] CRAN (R 4.4.1)
## httr 1.4.7 2023-08-15 [1] CRAN (R 4.4.1)
```

##	igraph	*	2.0.3	2024-03-13	[1]	CRAN	(R 4.4.1)
##	insight		1.0.0	2024-11-26	[1]	CRAN	(R 4.4.2)
##	iterators		1.0.14	2022-02-05	[1]	CRAN	(R 4.4.1)
##	KernSmooth		2.23-24	2024-05-17	[2]	CRAN	(R 4.4.1)
##	knitr	*	1.48	2024-07-07	[1]	CRAN	(R 4.4.1)
##	lattice		0.22-6	2024-03-20	[2]	CRAN	(R 4.4.1)
##	lifecycle		1.0.4	2023-11-07	[1]	CRAN	(R 4.4.1)
##	lmtest	*	0.9-40	2022-03-21	[1]	CRAN	(R 4.4.1)
##	magrittr		2.0.3	2022-03-30	[1]	CRAN	(R 4.4.1)
##	MASS		7.3-60.2	2024-04-26	[2]	CRAN	(R 4.4.1)
##	Matrix	*	1.7-0	2024-04-26	[2]	CRAN	(R 4.4.1)
##	maxLik		1.5-2.1	2024-03-24	[1]	CRAN	(R 4.4.1)
##	miscTools		0.6-28	2023-05-03	[1]	CRAN	(R 4.4.1)
##	modelsummary	*	2.2.0	2024-09-02	[1]	CRAN	(R 4.4.2)
##	multcomp		1.4-26	2024-07-18	[1]	CRAN	(R 4.4.1)
##	multiwayvcov		1.2.3	2016-05-05	[1]	CRAN	(R 4.4.1)
##	mvtnorm		1.2-5	2024-05-21	[1]	CRAN	(R 4.4.1)
##	nlme		3.1-164	2023-11-27	[2]	CRAN	(R 4.4.1)
##	pillar		1.9.0	2023-03-22	[1]	CRAN	(R 4.4.1)
##	pkgconfig		2.0.3	2019-09-22	[1]	CRAN	(R 4.4.1)
##	plm		2.6-4	2024-04-01	[1]	CRAN	(R 4.4.1)
##	proxy		0.4-27	2022-06-09	[1]	CRAN	(R 4.4.1)
##	purrr	*	1.0.2	2023-08-10	[1]	CRAN	(R 4.4.1)
##	R6		2.5.1	2021-08-19	[1]	CRAN	(R 4.4.1)
##	rbibutils		2.2.16	2023-10-25	[1]	CRAN	(R 4.4.1)
##	Rcpp		1.0.13	2024-07-17	[1]	CRAN	(R 4.4.1)
##	Rdpack		2.6	2023-11-08	[1]	CRAN	(R 4.4.1)
##	readr	*	2.1.5	2024-01-10	[1]	CRAN	(R 4.4.1)
##	readxl	*	1.4.3	2023-07-06	[1]	CRAN	(R 4.4.1)
##	rlang		1.1.4	2024-06-04	[1]	CRAN	(R 4.4.1)
##	rmarkdown		2.27	2024-05-17	[1]	CRAN	(R 4.4.1)
##	rnaturalearth	*	1.1.0	2025-07-28	[1]	CRAN	(R 4.4.3)
##	rstudioapi		0.16.0	2024-03-24	[1]	CRAN	(R 4.4.1)
##	sandwich	*	3.1-0	2023-12-11	[1]	CRAN	(R 4.4.1)
##	sessioninfo	*	1.2.2	2021-12-06	[1]	CRAN	(R 4.4.2)
##	sf	*	1.0-17	2024-09-06	[1]	CRAN	(R 4.4.1)
##	shape		1.4.6.1	2024-02-23	[1]	CRAN	(R 4.4.0)
##	sp		2.1-4	2024-04-30	[1]	CRAN	(R 4.4.1)
##	stringi		1.8.4	2024-05-06	[1]	CRAN	(R 4.4.0)
##	stringr	*	1.5.1	2023-11-14	[1]	CRAN	(R 4.4.1)
##	survival		3.6-4	2024-04-24	[2]	CRAN	(R 4.4.1)
##	tables		0.9.31	2024-08-29	[1]	CRAN	(R 4.4.2)
##	texreg		1.39.4	2024-07-24	[1]	CRAN	(R 4.4.1)
##	TH.data		1.1-2	2023-04-17	[1]	CRAN	(R 4.4.1)
##	tibble	*	3.2.1	2023-03-20	[1]	CRAN	(R 4.4.1)
##	tidyr	*	1.3.1	2024-01-24	[1]	CRAN	(R 4.4.1)
##	tidyselect		1.2.1	2024-03-11	[1]	CRAN	(R 4.4.1)
##	tinytex	*	0.52	2024-07-18	[1]	CRAN	(R 4.4.1)
##	tzdb		0.4.0	2023-05-12	[1]	CRAN	(R 4.4.1)
##	units	*	0.8-5	2023-11-28	[1]	CRAN	(R 4.4.1)
##	utf8		1.2.4	2023-10-22	[1]	CRAN	(R 4.4.1)
##	vctrs		0.6.5	2023-12-01	[1]	CRAN	(R 4.4.1)
##	writexl	*	1.5.0	2024-02-09	[1]	CRAN	(R 4.4.1)
##	xfun		0.46	2024-07-18	[1]	CRAN	(R 4.4.1)

```
## xtable          1.8-4    2019-04-21 [1] CRAN (R 4.4.1)
## yaml            2.3.10   2024-07-26 [1] CRAN (R 4.4.1)
## zoo             * 1.8-12  2023-04-13 [1] CRAN (R 4.4.1)
##
## [1] C:/Users/Lucas/AppData/Local/R/win-library/4.4
## [2] C:/Program Files/R/R-4.4.1/library
##
## -----
```

---

## Part 2. Cross-continent Analysis

This part of the script adds country names and regions to the WYD dataset.

### 2.1. Load data (just a backup, making it possible to run the script from this point on without running everything else before)

```
# I. Load the original RData file
load("Datasets/final08_1.RData")

# II. Convert to tibble for easier manipulation
df <- x %>%
  as_tibble()

# III. Remove original object from memory
rm(x)

# IV. Sort data by country code and group
data <- df %>%
  arrange(contcod, group)

# V. Display summary statistics
summary(df)
```

```
##   contcod          year    year_survey    DummyY
## Length:11737      Min.   :2008      Min.   :2004      Min.   :0.0000
## Class :AsIs        1st Qu.:2008      1st Qu.:2008      1st Qu.:0.0000
## Mode  :character   Median :2008      Median :2008      Median :1.0000
##                               Mean  :2008      Mean   :2008      Mean   :0.5144
##                               3rd Qu.:2008      3rd Qu.:2008      3rd Qu.:1.0000
##                               Max.   :2008      Max.   :2011      Max.   :1.0000
##
##   group          maxgroup      inc      lninc
## Min.   : 1.00      Min.   : 54.00      Min.   :    16.72      Min.   : 2.817
## 1st Qu.: 25.00      1st Qu.:100.00      1st Qu.:   911.70      1st Qu.: 6.815
## Median : 50.00      Median :100.00      Median :  2460.07      Median : 7.808
## Mean   : 50.44      Mean   : 99.79      Mean   :  6316.11      Mean   : 7.864
## 3rd Qu.: 75.00      3rd Qu.:100.00      3rd Qu.:  7802.75      3rd Qu.: 8.962
## Max.   :100.00      Max.   :100.00      Max.   :211296.72      Max.   :12.261
```

```
##
##      pop      gdpppp      lngdpppp      gini
## Min.   : 0.00310   Min.    : 303.2   Min.    : 5.714   Min.    :0.2307
## 1st Qu.: 0.04526   1st Qu.: 2576.0   1st Qu.: 7.854   1st Qu.:0.3074
## Median : 0.13479   Median : 7560.0   Median : 8.931   Median :0.3597
## Mean   : 0.52334   Mean    :12886.0   Mean    : 8.823   Mean    :0.3785
## 3rd Qu.: 0.38534   3rd Qu.:18773.0   3rd Qu.: 9.840   3rd Qu.:0.4375
## Max.   :13.25640   Max.    :73127.0   Max.    :11.200   Max.    :0.6721
##      NA's      :200      NA's      :200
##      ayos
## Min.   : 1.239
## 1st Qu.: 6.474
## Median : 8.713
## Mean   : 8.132
## 3rd Qu.:10.075
## Max.   :12.749
## NA's   :2654
```

```
# VI. Show first few rows
head(df, 10)
```

```
## # A tibble: 10 x 13
##   contcod  year year_survey DummyY group maxgroup  inc lninc  pop gdpppp
##   <I<chr>> <int>      <dbl>  <dbl> <int>      <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 ALB      2008      2008    0     1      100  729.  6.59 0.0314 7297
## 2 ALB      2008      2008    0     2      100  917.  6.82 0.0314 7297
## 3 ALB      2008      2008    0     3      100 1011.  6.92 0.0314 7297
## 4 ALB      2008      2008    0     4      100 1087.  6.99 0.0314 7297
## 5 ALB      2008      2008    0     5      100 1133.  7.03 0.0314 7297
## 6 ALB      2008      2008    0     6      100 1171.  7.07 0.0314 7297
## 7 ALB      2008      2008    0     7      100 1201.  7.09 0.0314 7297
## 8 ALB      2008      2008    0     8      100 1241.  7.12 0.0314 7297
## 9 ALB      2008      2008    0     9      100 1286.  7.16 0.0314 7297
## 10 ALB     2008      2008    0    10      100 1325.  7.19 0.0314 7297
## # i 3 more variables: lngdpppp <dbl>, gini <dbl>, ayos <dbl>
```

## 2.2. Check unique country codes

```
# I. Get unique country codes
unique_codes <- unique(df$contcod)
print(paste("Number of unique country codes:", length(unique_codes)))
```

```
## [1] "Number of unique country codes: 118"
```

```
# II. Display all unique codes
print(sort(unique_codes))
```

```
## [1] "ALB" "ARG" "ARM" "AUT" "AZE" "BEL" "BFA" "BGD" "BGR" "BIH" "BLR" "BOL"
## [13] "BRA" "BTN" "CAF" "CAN" "CHE" "CHL" "CHN" "CIV" "CMR" "COL" "CRI" "CYP"
## [25] "CZE" "DEU" "DNK" "DOM" "ECU" "EGY" "ESP" "EST" "FIN" "FJI" "FRA" "GBR"
```

```
## [37] "GEO" "GHA" "GIN" "GRC" "GTM" "HND" "HRV" "HUN" "IDN" "IND" "IRL" "IRN"
## [49] "IRQ" "ISL" "ISR" "ITA" "JOR" "JPN" "KAZ" "KEN" "KGZ" "KHM" "KOR" "KOS"
## [61] "LAO" "LBR" "LKA" "LTU" "LUX" "LVA" "MAR" "MDA" "MDG" "MEX" "MKD" "MLI"
## [73] "MNE" "MNG" "MOZ" "MRT" "MWI" "MYS" "NER" "NGA" "NIC" "NLD" "NOR" "NPL"
## [85] "PAK" "PAN" "PER" "PHL" "POL" "PRT" "PRY" "ROM" "RUS" "SDN" "SGP" "SLV"
## [97] "SRB" "SVK" "SVN" "SWE" "SWZ" "SYR" "THA" "TJK" "TLS" "TUR" "TWN" "TZA"
## [109] "UGA" "UKR" "URY" "USA" "VEN" "VNM" "WBG" "YEM" "ZAF" "ZAR"
```

### 2.3. Create country mapping

```
# I. Create comprehensive country mapping based on ISO codes
# Note: Handling special cases:
# - ROM = Romania (instead of ROU)
# - KOS = Kosovo
# - WBG will be renamed to PSE = Palestine
# - ZAR will be renamed to COD = Dem. Rep. Congo
```

```
country_mapping <- tibble(contcod = c("ALB", "DZA", "AGO", "ARG", "ARM", "AUS", "AUT", "AZE", "BGD", "BI",
  "BEL", "BEN", "BTN", "BOL", "BIH", "BWA", "BRA", "BGR", "BFA", "BI",
  "KHM", "CMR", "CAN", "CPV", "CAF", "TCD", "CHL", "CHN", "COL", "CO",
  "COG", "CRI", "CIV", "HRV", "CZE", "COD", "DNK", "DJI", "DOM", "EC",
  "EGY", "SLV", "EST", "ETH", "FJI", "FIN", "FRA", "GAB", "GMB", "GH",
  "DEU", "GHA", "GRC", "GTM", "GIN", "GNB", "GUY", "HTI", "HND", "HN",
  "HUN", "ISL", "IND", "IDN", "IRN", "IRQ", "IRL", "ISR", "ITA", "JP",
  "JPN", "JOR", "KAZ", "KEN", "KGZ", "LAO", "LVA", "LBN", "LSO", "LI",
  "LTU", "LUX", "MKD", "MDG", "MWI", "MYS", "MDV", "MLI", "MLT", "MM",
  "MUS", "MEX", "MDA", "MNG", "MNE", "MAR", "MOZ", "MMR", "NAM", "NE",
  "NLD", "NIC", "NER", "NGA", "NOR", "PAK", "PSE", "PAN", "PNG", "PR",
  "PER", "PHL", "POL", "PRT", "ROM", "RUS", "RWA", "STP", "SEN", "SI",
  "SLE", "SGP", "SVK", "SVN", "ZAF", "KOR", "ESP", "LKA", "SDN", "SV",
  "SWE", "CHE", "SYR", "TWN", "TJK", "TZA", "THA", "TLS", "TGO", "TG",
  "TUN", "TUR", "TKM", "UGA", "UKR", "GBR", "USA", "URY", "UZB", "VI",
  "VNM", "YEM", "ZMB", "ZWE", "ARE", "AFG", "ATG", "AND", "BHS", "BH",
  "BRB", "BLZ", "BMU", "BRN", "CYP", "DMA", "ERI", "GRD", "GNQ", "IS",
  "KWT", "LIE", "MAC", "MCO", "OMN", "PLW", "QAT", "KNA", "LCA", "VC",
  "WSM", "SMR", "SAU", "SYC", "SOM", "SSD", "TON", "VUT", "VAT", "KV",
  "cont = c("Albania", "Algeria", "Angola", "Argentina", "Armenia", "Australia", "Austria", "Azerbaijan",
  "Belgium", "Benin", "Bhutan", "Bolivia", "Bosnia and Herzegovina", "Botswana", "Brazil", "Bu",
  "Cambodia", "Cameroon", "Canada", "Cape Verde", "Central African Republic", "Chad", "Chile",
  "Congo", "Costa Rica", "Côte d'Ivoire", "Croatia", "Czech Republic", "Dem. Rep. Congo", "Denm",
  "Ecuador", "Egypt", "El Salvador", "Estonia", "Ethiopia", "Fiji", "Finland", "France", "Gabor",
  "Germany", "Ghana", "Greece", "Guatemala", "Guinea", "Guinea-Bissau", "Guyana", "Haiti", "Hon",
  "Hungary", "Iceland", "India", "Indonesia", "Iran", "Iraq", "Ireland", "Israel", "Italy", "J",
  "Japan", "Jordan", "Kazakhstan", "Kenya", "Kyrgyzstan", "Laos", "Latvia", "Lebanon", "Lesotho",
  "Lithuania", "Luxembourg", "Macedonia", "Madagascar", "Malawi", "Malaysia", "Maldives", "Mal",
  "Mauritius", "Mexico", "Moldova", "Mongolia", "Montenegro", "Morocco", "Mozambique", "Myanmar",
  "Netherlands", "Nicaragua", "Niger", "Nigeria", "Norway", "Pakistan", "Palestine", "Panama",
  "Peru", "Philippines", "Poland", "Portugal", "Romania", "Russia", "Rwanda", "São Tomé and Pr",
  "Sierra Leone", "Singapore", "Slovakia", "Slovenia", "South Africa", "South Korea", "Spain",
  "Sweden", "Switzerland", "Syria", "Taiwan", "Tajikistan", "Tanzania", "Thailand", "Timor-Les",
  "Tunisia", "Turkey", "Turkmenistan", "Uganda", "Ukraine", "United Kingdom", "United States",
  "Vietnam", "Yemen", "Zambia", "Zimbabwe", "United Arab Emirates", "Afghanistan", "Antigua an
```

```

    "Barbados", "Belize", "Bermuda", "Brunei", "Cyprus", "Dominica", "Eritrea", "Grenada", "Equa
    "Liechtenstein", "Macao", "Monaco", "Oman", "Palau", "Qatar", "Saint Kitts and Nevis", "Sain
    "Samoa", "San Marino", "Saudi Arabia", "Seychelles", "Somalia", "South Sudan", "Tonga", "Van
reg = c("Europe", "Africa", "Africa", "South America", "Asia", "Oceania", "Europe", "Asia", "Asia", "I
    "Europe", "Africa", "Asia", "South America", "Europe", "Africa", "South America", "Europe", "I
    "Asia", "Africa", "North America", "Africa", "Africa", "Africa", "South America", "Asia", "So
    "Africa", "Central America", "Africa", "Europe", "Europe", "Africa", "Europe", "Africa", "Cent
    "Africa", "Central America", "Europe", "Africa", "Oceania", "Europe", "Europe", "Africa", "Af
    "Europe", "Africa", "Europe", "Central America", "Africa", "Africa", "South America", "Centra
    "Europe", "Europe", "Asia", "Asia", "Asia", "Asia", "Europe", "Asia", "Europe", "Central Amer
    "Asia", "Asia", "Asia", "Africa", "Asia", "Asia", "Europe", "Asia", "Africa", "Africa",
    "Europe", "Europe", "Europe", "Africa", "Africa", "Asia", "Asia", "Africa", "Europe", "Africa
    "Africa", "North America", "Europe", "Asia", "Europe", "Africa", "Africa", "Asia", "Africa",
    "Europe", "Central America", "Africa", "Africa", "Europe", "Asia", "Asia", "Central America",
    "South America", "Asia", "Europe", "Europe", "Europe", "Europe", "Africa", "Africa", "Africa"
    "Africa", "Asia", "Europe", "Europe", "Africa", "Asia", "Europe", "Asia", "Africa", "Africa",
    "Europe", "Europe", "Asia", "Asia", "Asia", "Africa", "Asia", "Asia", "Africa", "Central Amer
    "Africa", "Asia", "Asia", "Africa", "Europe", "Europe", "North America", "South America", "As
    "Asia", "Asia", "Africa", "Africa", "Asia", "Asia", "Central America", "Europe", "Central Amer
    "North America", "Asia", "Europe", "Central America", "Africa", "Central America", "Africa",
    "Asia", "Europe", "Asia", "Europe", "Asia", "Oceania", "Asia", "Central America", "Central Am
    "Central America", "Oceania", "Europe", "Asia", "Africa", "Africa", "Africa", "Oceania", "Oce

```

```

save(country_mapping,
      file = "Auxiliary/country_mapping.RData")

# II. Handle country code renames before joining
# Rename WBG to PSE (Palestine) and ZAR to COD (Dem. Rep. Congo)
df <- df %>%
  mutate(contcod = case_when(contcod == "WBG" ~ "PSE", # Palestine
                             contcod == "ZAR" ~ "COD", # Democratic Republic of Congo
                             TRUE ~ contcod))

# III. Re-check unique codes after renaming
unique_codes <- unique(df$contcod)

# IV. Check which country codes in the data don't have mappings
missing_codes <- setdiff(unique_codes, country_mapping$contcod)
if(length(missing_codes) > 0) {
  print("Country codes in data but not in mapping:")
  print(missing_codes)
}

```

## 2.4. Add country information to dataset

```

# I. Join the mapping to the original data
df_updated <- df %>%
  left_join(country_mapping, by = "contcod")

# II. Check if join was successful
print("Sample of joined data:")

```

```
## [1] "Sample of joined data:"
```

```
df_updated %>%  
  select(contcod, cont, reg) %>%  
  distinct() %>%  
  head(10)
```

```
## # A tibble: 10 x 3  
##   contcod cont                reg  
##   <I<chr>> <chr>             <chr>  
## 1 ALB     Albania          Europe  
## 2 ARG     Argentina        South America  
## 3 ARM     Armenia            Asia  
## 4 AUT     Austria            Europe  
## 5 AZE     Azerbaijan          Asia  
## 6 BEL     Belgium            Europe  
## 7 BFA     Burkina Faso          Africa  
## 8 BGD     Bangladesh          Asia  
## 9 BGR     Bulgaria              Europe  
## 10 BIH    Bosnia and Herzegovina Europe
```

## 2.5. Reorder columns to put cont and reg as columns B and C

```
# I. Get all column names  
col_names <- names(df_updated)  
print("Original column order:")
```

```
## [1] "Original column order:"
```

```
print(col_names)
```

```
## [1] "contcod"      "year"          "year_survey"  "DummyY"        "group"  
## [6] "maxgroup"    "inc"           "lninc"        "pop"           "gdppppp"  
## [11] "lngdppppp"   "gini"          "ayos"         "cont"          "reg"
```

```
# II. Reorder: first column (contcod), then cont, then reg, then rest  
# Remove cont and reg from their current positions  
other_cols <- col_names[!col_names %in% c("cont", "reg")]
```

```
# III. Create new order  
df_final <- df_updated %>%  
  select(all_of(c(other_cols[1], "cont", "reg", other_cols[-1])))
```

```
# IV. Verify new column order  
print("New column order (first 10 columns):")
```

```
## [1] "New column order (first 10 columns):"
```



```
print(names(df_final)[1:10])
```

```
## [1] "contcod"      "cont"         "reg"          "year"         "year_survey"
## [6] "DummyY"      "group"        "maxgroup"     "inc"          "lninc"
```

## 2.6. Check for missing values

```
# I. Check for any missing country names or regions
missing_info <- df_final %>%
  filter(is.na(cont) | is.na(reg)) %>%
  select(contcod) %>%
  distinct()

# II. Display information about missing values
if(nrow(missing_info) > 0) {
  print("Country codes with missing names or regions:")
  print(missing_info)

  # III. Count how many rows are affected
  missing_rows <- df_final %>%
    filter(is.na(cont) | is.na(reg)) %>%
    nrow()

  # IV. Calculate percentage of data affected
  print(paste("Total rows with missing country info:", missing_rows))
  print(paste("Percentage of data affected:", round(missing_rows/nrow(df_final)*100, 2), "%"))
}
```

## 2.7. Create summary table

```
# I. Create a summary table of countries by region
df_reg <- df_final

country_summary <- df_reg %>%
  select(contcod, cont, reg) %>%
  distinct() %>%
  arrange(reg, cont)

# II. Display countries grouped by region
country_summary %>%
  group_by(reg) %>%
  summarise(Countries = paste(cont, collapse = ", ")) %>%
  print()
```

```
## # A tibble: 7 x 2
##   reg      Countries
##   <chr>    <chr>
## 1 Africa  Burkina Faso, Cameroon, Central African Republic, Côte d'Ivoi-
## 2 Asia    Armenia, Azerbaijan, Bangladesh, Bhutan, Cambodia, China, Geo-
```

```
## 3 Central America Costa Rica, Dominican Republic, El Salvador, Guatemala, Hondu-
## 4 Europe Albania, Austria, Belarus, Belgium, Bosnia and Herzegovina, B-
## 5 North America Canada, Mexico, United States
## 6 Oceania Fiji
## 7 South America Argentina, Bolivia, Brazil, Chile, Colombia, Ecuador, Paragua-
```

## 2.8. Create continental aggregates

```
# I. Filter for complete data and aggregate by region
df_cont <- df_reg %>%
  filter(maxgroup == 100) %>%
  group_by(reg, group) %>%
  summarise(inc = mean(inc),
            pop = sum(pop),
            gdpppp = mean(gdpppp, na.rm = TRUE),
            gini = mean(gini, na.rm = TRUE)) %>%
  mutate(lninc = log(inc),
         lngdpppp = log(gdpppp)) %>%
  relocate(lninc, .after = "inc") %>%
  relocate(lngdpppp, .after = "gdpppp")
```

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

```
# II. Display first few rows
head(df_cont)
```

```
## # A tibble: 6 x 8
## # Groups:   reg [1]
##   reg   group   inc lninc   pop gdpppp lngdpppp  gini
##   <chr> <int> <dbl> <dbl> <dbl> <dbl>   <dbl> <dbl>
## 1 Africa     1  125.  4.83  7.08  1996.    7.60  0.420
## 2 Africa     2  173.  5.15  7.08  1996.    7.60  0.420
## 3 Africa     3  200.  5.30  7.08  1996.    7.60  0.420
## 4 Africa     4  220.  5.40  7.08  1996.    7.60  0.420
## 5 Africa     5  238.  5.47  7.08  1996.    7.60  0.420
## 6 Africa     6  254.  5.54  7.08  1996.    7.60  0.420
```

## 2.9. Save the updated dataset

```
# I. Rename dataset
rm(df_final, df_updated, missing_info)

# II. Save as Excel file
write_xlsx(df_reg, "Datasets/WYD_reg.xlsx")
write_xlsx(df_cont, "Datasets/WYD_cont.xlsx")

# III. Save multiple objects together
save(df_reg,
```

```

df_cont,
file = "Datasets/replication_results_part2.RData")

# IV. Show distribution by region
df_reg %>%
  group_by(reg) %>%
  summarise(n_countries = n_distinct(contcod),
            n_observations = n()) %>%
  arrange(desc(n_countries))

```

```

## # A tibble: 7 x 3
##   reg          n_countries n_observations
##   <chr>          <int>          <int>
## 1 Europe           40           3983
## 2 Asia             34           3354
## 3 Africa           23           2300
## 4 South America    10           1000
## 5 Central America   7             700
## 6 North America     3             300
## 7 Oceania           1             100

```