02\_Anthropetry

## 1. Load Libraries and Data

# Data Manipulation  
library(dplyr)  
library(tidyr)  
library(readr)  
library(here)  
library(purrr)  
# Visualization  
library(ggplot2)  
library(skimr) # For comprehensive summary  
library(janitor) # for cleaning column names  
library(visdat) # visualize missingness  
library(mice) # for advanced imputation

# Load data with correct path from project root  
anth\_df <- read.csv(here("data", "raw", "anthropometry\_national\_zaf.csv"))  
  
# Skip first metadata row  
anth\_df <- anth\_df[-1, ]  
  
cat("Initial dataset loaded successfully.\n")

## Initial dataset loaded successfully.

cat("Dimensions:", dim(anth\_df), "\n")

## Dimensions: 37 29

### Load Data

**Purpose:** Load the raw anthropometry dataset into R.

**What the code does:**  
- Reads the CSV file from the project folder using here().  
- Removes the first row if it contains metadata.  
- Displays initial dataset dimensions.

**Outcome:** Raw dataset anth\_df is ready for assessment and cleaning.

# 2. Initial Data Assessment ———————————————–

# Clean column names  
anth\_df <- janitor::clean\_names(anth\_df)  
colnames(anth\_df)

## [1] "iso3" "data\_id"   
## [3] "indicator" "value"   
## [5] "precision" "dhs\_country\_code"   
## [7] "country\_name" "survey\_year"   
## [9] "survey\_id" "indicator\_id"   
## [11] "indicator\_order" "indicator\_type"   
## [13] "characteristic\_id" "characteristic\_order"   
## [15] "characteristic\_category" "characteristic\_label"   
## [17] "by\_variable\_id" "by\_variable\_label"   
## [19] "is\_total" "is\_preferred"   
## [21] "sdrid" "region\_id"   
## [23] "survey\_year\_label" "survey\_type"   
## [25] "denominator\_weighted" "denominator\_unweighted"   
## [27] "ci\_low" "ci\_high"   
## [29] "level\_rank"

# Peek at structure and summary  
glimpse(anth\_df)

## Rows: 37  
## Columns: 29  
## $ iso3 <chr> "ZAF", "ZAF", "ZAF", "ZAF", "ZAF", "ZAF", "ZAF…  
## $ data\_id <chr> "198690", "198687", "198688", "597227", "59722…  
## $ indicator <chr> "Children severely stunted", "Children stunted…  
## $ value <chr> "9.8", "27.4", "-1.1", "0.6", "2.5", "13.3", "…  
## $ precision <chr> "1", "1", "1", "1", "1", "1", "1", "1", "1", "…  
## $ dhs\_country\_code <chr> "ZA", "ZA", "ZA", "ZA", "ZA", "ZA", "ZA", "ZA"…  
## $ country\_name <chr> "South Africa", "South Africa", "South Africa"…  
## $ survey\_year <chr> "2016", "2016", "2016", "2016", "2016", "2016"…  
## $ survey\_id <chr> "ZA2016DHS", "ZA2016DHS", "ZA2016DHS", "ZA2016…  
## $ indicator\_id <chr> "CN\_NUTS\_C\_HA3", "CN\_NUTS\_C\_HA2", "CN\_NUTS\_C\_H…  
## $ indicator\_order <int> 104236010, 104236020, 104236030, 104236040, 10…  
## $ indicator\_type <chr> "I", "I", "I", "I", "I", "I", "I", "I", "I", "…  
## $ characteristic\_id <int> 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000…  
## $ characteristic\_order <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…  
## $ characteristic\_category <chr> "Total", "Total", "Total", "Total", "Total", "…  
## $ characteristic\_label <chr> "Total", "Total", "Total", "Total", "Total", "…  
## $ by\_variable\_id <chr> "0", "0", "0", "0", "0", "0", "0", "0", "0", "…  
## $ by\_variable\_label <chr> "", "", "", "", "", "", "", "", "", "", "", ""…  
## $ is\_total <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…  
## $ is\_preferred <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…  
## $ sdrid <chr> "CNNUTSCHA3", "CNNUTSCHA2", "CNNUTSCHAM", "CNN…  
## $ region\_id <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA…  
## $ survey\_year\_label <int> 2016, 2016, 2016, 2016, 2016, 2016, 2016, 2016…  
## $ survey\_type <chr> "DHS", "DHS", "DHS", "DHS", "DHS", "DHS", "DHS…  
## $ denominator\_weighted <int> 1404, 1404, 1404, 1384, 1384, 1384, 1384, 1416…  
## $ denominator\_unweighted <int> 1468, 1468, 1468, 1449, 1449, 1449, 1449, 1479…  
## $ ci\_low <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA…  
## $ ci\_high <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA…  
## $ level\_rank <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA…

skim(anth\_df)

Data summary

|  |  |
| --- | --- |
| Name | anth\_df |
| Number of rows | 37 |
| Number of columns | 29 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 17 |
| logical | 4 |
| numeric | 8 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

| skim\_variable | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| --- | --- | --- | --- | --- | --- | --- | --- |
| iso3 | 0 | 1 | 3 | 3 | 0 | 1 | 0 |
| data\_id | 0 | 1 | 5 | 6 | 0 | 37 | 0 |
| indicator | 0 | 1 | 13 | 59 | 0 | 33 | 0 |
| value | 0 | 1 | 2 | 4 | 0 | 36 | 0 |
| precision | 0 | 1 | 1 | 1 | 0 | 2 | 0 |
| dhs\_country\_code | 0 | 1 | 2 | 2 | 0 | 1 | 0 |
| country\_name | 0 | 1 | 12 | 12 | 0 | 1 | 0 |
| survey\_year | 0 | 1 | 4 | 4 | 0 | 1 | 0 |
| survey\_id | 0 | 1 | 9 | 9 | 0 | 1 | 0 |
| indicator\_id | 0 | 1 | 13 | 13 | 0 | 37 | 0 |
| indicator\_type | 0 | 1 | 1 | 1 | 0 | 3 | 0 |
| characteristic\_category | 0 | 1 | 5 | 11 | 0 | 2 | 0 |
| characteristic\_label | 0 | 1 | 5 | 11 | 0 | 2 | 0 |
| by\_variable\_id | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| by\_variable\_label | 0 | 1 | 0 | 0 | 37 | 1 | 0 |
| sdrid | 0 | 1 | 10 | 10 | 0 | 37 | 0 |
| survey\_type | 0 | 1 | 3 | 3 | 0 | 1 | 0 |

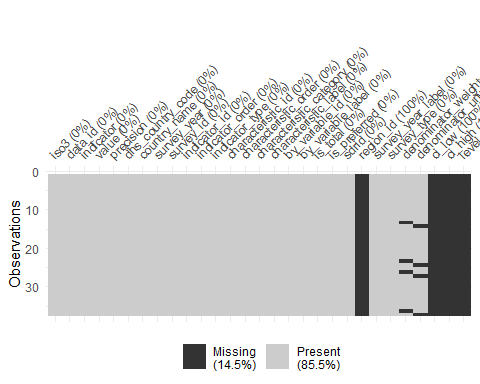
**Variable type: logical**

| skim\_variable | n\_missing | complete\_rate | mean | count |
| --- | --- | --- | --- | --- |
| region\_id | 37 | 0 | NaN | : |
| ci\_low | 37 | 0 | NaN | : |
| ci\_high | 37 | 0 | NaN | : |
| level\_rank | 37 | 0 | NaN | : |

**Variable type: numeric**

| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| indicator\_order | 0 | 1.00 | 111493274.86 | 4785521.41 | 104236010 | 104236100 | 114563080 | 114564170 | 114564260 | ▃▁▁▁▇ |
| characteristic\_id | 0 | 1.00 | 4162.16 | 4355.80 | 1000 | 1000 | 1000 | 10000 | 10000 | ▇▁▁▁▅ |
| characteristic\_order | 0 | 1.00 | 3513.51 | 4839.78 | 0 | 0 | 0 | 10000 | 10000 | ▇▁▁▁▅ |
| is\_total | 0 | 1.00 | 1.00 | 0.00 | 1 | 1 | 1 | 1 | 1 | ▁▁▇▁▁ |
| is\_preferred | 0 | 1.00 | 1.00 | 0.00 | 1 | 1 | 1 | 1 | 1 | ▁▁▇▁▁ |
| survey\_year\_label | 0 | 1.00 | 2016.00 | 0.00 | 2016 | 2016 | 2016 | 2016 | 2016 | ▁▁▇▁▁ |
| denominator\_weighted | 4 | 0.89 | 2336.76 | 747.16 | 1384 | 1416 | 2336 | 3081 | 3272 | ▇▁▆▁▇ |
| denominator\_unweighted | 4 | 0.89 | 2432.94 | 765.44 | 1449 | 1479 | 2457 | 3210 | 3405 | ▇▁▆▁▇ |

# Check missingness visually  
vis\_miss(anth\_df)



### Initial Data Assessment

**Purpose:** Understand dataset structure, missingness, and content before cleaning.

**What the code does:**  
- Cleans column names with clean\_names().  
- Shows variable types and sample data using glimpse().  
- Generates detailed summary statistics with skim().  
- Visualizes missing values using vis\_miss().

**Outcome:** Snapshot of dataset quality, guiding the cleaning steps.

## 3. Data Cleaning Process

### 3.1 Handle Duplicates Systematically

* Duplicates can distort analysis. We remove exact duplicates to maintain dataset integrity.

# Exact duplicates  
cat("Exact duplicates:", sum(duplicated(anth\_df)), "\n")

## Exact duplicates: 0

# Keep first occurrence  
anth\_df <- anth\_df %>% distinct()  
  
cat("Dimensions after deduplication:", dim(anth\_df), "\n")

## Dimensions after deduplication: 37 29

### Handle Duplicates

**Purpose:** Remove repeated rows to maintain dataset integrity.

**What the code does:**  
- Counts exact duplicates with duplicated().  
- Removes duplicates with distinct().

**Outcome:** Dataset now contains only unique observations.

### 3.2 Convert Data Types

* Ensures numeric, integer, and logical columns are correctly typed for analysis.This prevents calculation errors and improves data quality.

# Define the columns safely  
numeric\_cols <- intersect(c("value", "precision", "denominator\_weighted", "denominator\_unweighted"), colnames(anth\_df))  
integer\_cols <- intersect(c("survey\_year", "indicator\_order", "characteristic\_id",   
 "characteristic\_order", "survey\_year\_label", "by\_variable\_id", "region\_id"), colnames(anth\_df))  
logical\_cols <- intersect(c("is\_total", "is\_preferred"), colnames(anth\_df))  
  
# Apply conversions only if the columns exist  
anth\_df <- anth\_df %>%  
 mutate(  
 across(all\_of(numeric\_cols), as.numeric),  
 across(all\_of(integer\_cols), as.integer),  
 across(all\_of(logical\_cols), ~as.logical(as.integer(.)))  
 )  
  
cat("Data types converted successfully.\n")

## Data types converted successfully.

### Convert Data Types

**Purpose:** Ensure numeric, integer, and logical columns are properly typed.

**What the code does:**  
- Converts numeric columns like value and precision.  
- Converts ID or order columns to integers.  
- Converts flag columns (is\_total, is\_preferred) to logical.

**Outcome:** Standardized column types, preventing calculation and modeling errors.

### 3.3 Handle Missing Values

# 1. Summarize missingness  
missing\_summary <- data.frame(  
 Column = names(anth\_df),  
 Missing\_Count = colSums(is.na(anth\_df)),  
 Missing\_Percent = round(colSums(is.na(anth\_df)) / nrow(anth\_df) \* 100, 2)  
) %>% arrange(desc(Missing\_Percent))  
  
print(missing\_summary)

## Column Missing\_Count Missing\_Percent  
## region\_id region\_id 37 100.00  
## ci\_low ci\_low 37 100.00  
## ci\_high ci\_high 37 100.00  
## level\_rank level\_rank 37 100.00  
## denominator\_weighted denominator\_weighted 4 10.81  
## denominator\_unweighted denominator\_unweighted 4 10.81  
## iso3 iso3 0 0.00  
## data\_id data\_id 0 0.00  
## indicator indicator 0 0.00  
## value value 0 0.00  
## precision precision 0 0.00  
## dhs\_country\_code dhs\_country\_code 0 0.00  
## country\_name country\_name 0 0.00  
## survey\_year survey\_year 0 0.00  
## survey\_id survey\_id 0 0.00  
## indicator\_id indicator\_id 0 0.00  
## indicator\_order indicator\_order 0 0.00  
## indicator\_type indicator\_type 0 0.00  
## characteristic\_id characteristic\_id 0 0.00  
## characteristic\_order characteristic\_order 0 0.00  
## characteristic\_category characteristic\_category 0 0.00  
## characteristic\_label characteristic\_label 0 0.00  
## by\_variable\_id by\_variable\_id 0 0.00  
## by\_variable\_label by\_variable\_label 0 0.00  
## is\_total is\_total 0 0.00  
## is\_preferred is\_preferred 0 0.00  
## sdrid sdrid 0 0.00  
## survey\_year\_label survey\_year\_label 0 0.00  
## survey\_type survey\_type 0 0.00

# 2. Drop columns with >80% missing values  
cols\_to\_drop <- missing\_summary %>% filter(Missing\_Percent > 80) %>% pull(Column)  
if(length(cols\_to\_drop) > 0){  
 anth\_df <- anth\_df %>% select(-all\_of(cols\_to\_drop))  
 cat("Dropped columns with >80% missing:", paste(cols\_to\_drop, collapse = ", "), "\n")  
}

## Dropped columns with >80% missing: region\_id, ci\_low, ci\_high, level\_rank

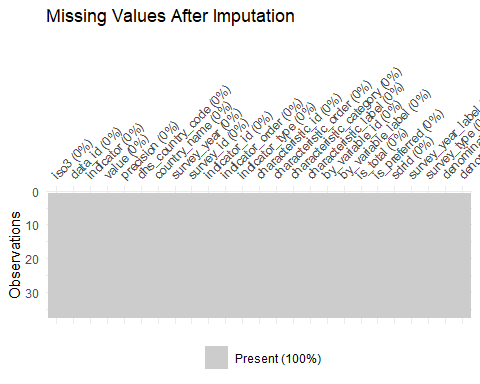
# 3. Impute remaining missing values  
# Function to get mode  
impute\_mode <- function(x) {  
 ux <- na.omit(x)  
 if(length(ux) == 0) return(x)  
 x[is.na(x)] <- names(sort(table(ux), decreasing = TRUE))[1]  
 return(x)  
}  
  
anth\_df <- anth\_df %>%  
 mutate(across(where(is.numeric), ~ifelse(is.na(.), median(., na.rm = TRUE), .))) %>%  
 mutate(across(where(is.character), impute\_mode))  
  
cat("Remaining NAs after imputation:", sum(is.na(anth\_df)), "\n")

## Remaining NAs after imputation: 0

# Summarize after handling missing values  
summary\_stats <- data.frame(  
 Column = names(anth\_df),  
 Type = sapply(anth\_df, class),  
 Missing\_Count = colSums(is.na(anth\_df)),  
 Missing\_Percent = round(colSums(is.na(anth\_df)) / nrow(anth\_df) \* 100, 2)  
)  
  
# Print summary  
print(summary\_stats)

## Column Type Missing\_Count  
## iso3 iso3 character 0  
## data\_id data\_id character 0  
## indicator indicator character 0  
## value value numeric 0  
## precision precision numeric 0  
## dhs\_country\_code dhs\_country\_code character 0  
## country\_name country\_name character 0  
## survey\_year survey\_year integer 0  
## survey\_id survey\_id character 0  
## indicator\_id indicator\_id character 0  
## indicator\_order indicator\_order integer 0  
## indicator\_type indicator\_type character 0  
## characteristic\_id characteristic\_id integer 0  
## characteristic\_order characteristic\_order integer 0  
## characteristic\_category characteristic\_category character 0  
## characteristic\_label characteristic\_label character 0  
## by\_variable\_id by\_variable\_id integer 0  
## by\_variable\_label by\_variable\_label character 0  
## is\_total is\_total logical 0  
## is\_preferred is\_preferred logical 0  
## sdrid sdrid character 0  
## survey\_year\_label survey\_year\_label integer 0  
## survey\_type survey\_type character 0  
## denominator\_weighted denominator\_weighted numeric 0  
## denominator\_unweighted denominator\_unweighted numeric 0  
## Missing\_Percent  
## iso3 0  
## data\_id 0  
## indicator 0  
## value 0  
## precision 0  
## dhs\_country\_code 0  
## country\_name 0  
## survey\_year 0  
## survey\_id 0  
## indicator\_id 0  
## indicator\_order 0  
## indicator\_type 0  
## characteristic\_id 0  
## characteristic\_order 0  
## characteristic\_category 0  
## characteristic\_label 0  
## by\_variable\_id 0  
## by\_variable\_label 0  
## is\_total 0  
## is\_preferred 0  
## sdrid 0  
## survey\_year\_label 0  
## survey\_type 0  
## denominator\_weighted 0  
## denominator\_unweighted 0

# Optional: visualize missing data (should be none now)  
library(visdat)  
vis\_miss(anth\_df) + ggtitle("Missing Values After Imputation")



**Purpose:** Address missing data to allow accurate analysis.

**What the code does:**  
- Drops columns with >40% missing values.  
- Imputes remaining numeric missing values with median.  
- Imputes categorical missing values with mode.

**Outcome:** Dataset is complete, reducing bias in analysis.

### 3.4 Remove redundant columns

* Metadata columns such as survey type or country identifiers are removed as they do not contribute to analysis.

# -----------------------------  
# Remove redundant columns  
# -----------------------------  
# Define columns that are metadata or unnecessary for analysis  
redundant\_cols <- c("survey\_type", "survey\_id", "country\_name", "iso3")   
  
# Remove them safely  
anth\_df <- anth\_df %>%  
 select(-any\_of(redundant\_cols))  
  
cat("Redundant columns removed. New dimensions:", dim(anth\_df), "\n")

## Redundant columns removed. New dimensions: 37 21

**Purpose:** Remove columns with all missing or invalid values.

**What the code does:**  
- Detects columns where all values are NA or NaN.  
- Removes these columns from the dataset.

**Outcome:** Dataset is compact, without empty or unusable variables.

# Handle Outliers

# Detect numeric columns  
num\_cols <- anth\_df %>% select(where(is.numeric))  
  
# Compute IQR bounds  
outlier\_bounds <- function(x) {  
 qnt <- quantile(x, probs=c(0.25, 0.75), na.rm=TRUE)  
 iqr <- diff(qnt)  
 c(lower=qnt[1]-1.5\*iqr, upper=qnt[2]+1.5\*iqr)  
}  
  
bounds <- map(num\_cols, outlier\_bounds)  
  
# Winsorize numeric variables  
anth\_df <- anth\_df %>%  
 mutate(across(where(is.numeric),  
 ~pmin(pmax(., bounds[[cur\_column()]]["lower"]),  
 bounds[[cur\_column()]]["upper"])))

### Handle Outliers

**Purpose:** Reduce the influence of extreme values on analysis.

**What the code does:**  
- Calculates IQR bounds per numeric column.  
- Caps values outside lower/upper bounds (Winsorizing).

**Outcome:** Numeric variables are stabilized, minimizing distortion.

### 3.5 Deal with Noise / Special Values

anth\_df <- anth\_df %>%  
 mutate(across(matches("height|weight"),  
 ~ifelse(. < 0, median(., na.rm = TRUE), .)))

### Handle Noise / Special Values

**Purpose:** Correct logically impossible values.

**What the code does:**  
- Replaces negative height or weight values with column median.

**Outcome:** Dataset values are realistic, ready for analysis.

## 5. Save Cleaned Data

write\_csv(anth\_df, here("data", "processed", "anthropometry\_cleaned.csv"))  
cat("Cleaned dataset saved to data/processed/anthropometry\_cleaned.csv\n")

## Cleaned dataset saved to data/processed/anthropometry\_cleaned.csv

rm(list=ls()) # clears all objects

### Save Cleaned Data

**Purpose:** Persist the cleaned dataset for analysis or sharing.

**What the code does:**  
- Saves as CSV in data/processed/.

**Outcome:** Cleaned dataset is stored safely for reproducible analysis.