# 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"
```

```r
# 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…
```

```r
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 | : |

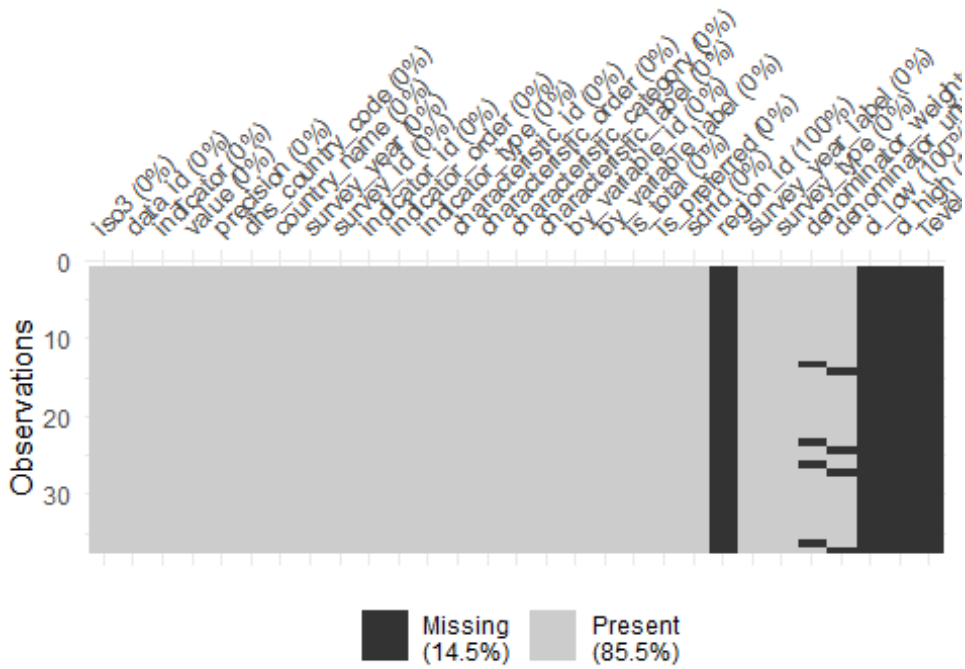
**Variable type: numeric**

| skim_variable | n_missing | complete_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
|---|---|---|---|---|---|---|---|---|---|---|

| skim_variable | n_missing | complete_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
|---|---|---|---|---|---|---|---|---|---|---|
| indicator_order | 0 | 1.00 | 111493274.86 | 4785521.41 | 10423601 0 | 10423610 0 | 11456308 0 | 11456417 0 | 11456426 0 | ▅▁▁▁▇ |
| 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_ | 4 | 0.89 | 2432.9 | 765.4 | 1449 | 1479 | 2457 | 3210 | 3405 | ▇ |

| skim_variable | n_missing | complete_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
|---|---|---|---|---|---|---|---|---|---|---|
| unweighted | | | 4 | 4 | | | | | | ▁▇▁▇ |

```
# 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

```r
# 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")
}
```

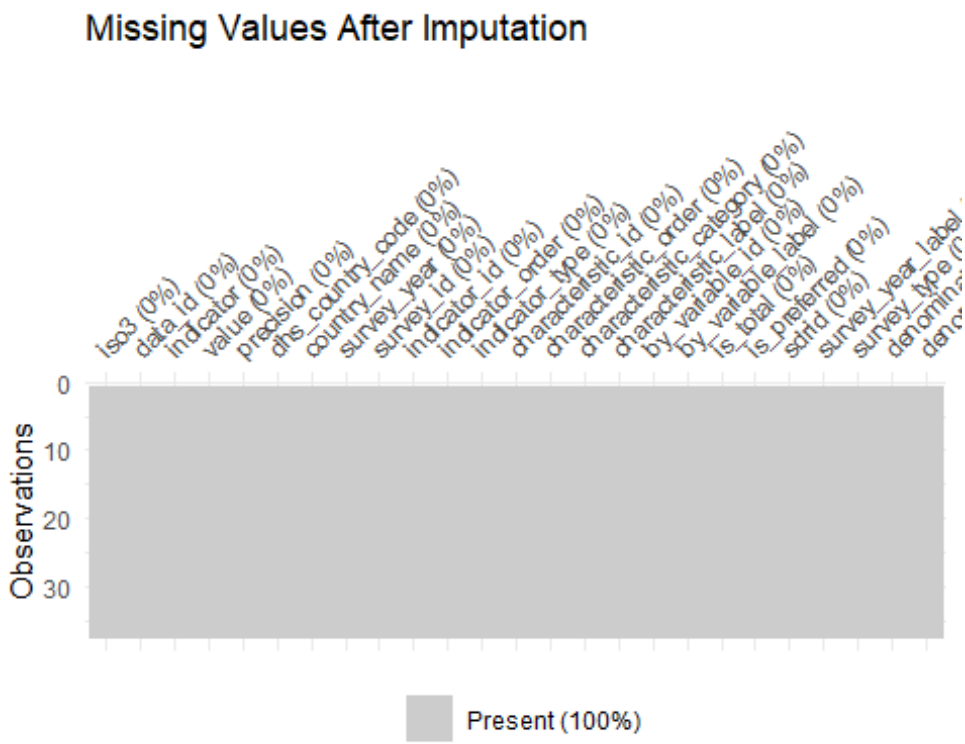
```
## Dropped columns with >80% missing: region_id, ci_low, ci_high, level_rank
```

```
# 3. Impute remaining missing values
# Function to get mode
impute_mode <- function(x) {
  ux <- na.omit(x)
  if(length(ux) == 0) return(x)
```

```r
  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
```

```r
# 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.