BIN381 — Milestone 1: Data Understanding (Full QA, Clean Headings)

2025-09-09

Table of Contents

[1 1. Load All Datasets 2](#_Toc208344060)

[2 2. Dataset-Level Summary 4](#_Toc208344061)

[3 3. Data Quality Assessment 6](#_Toc208344062)

[3.1 3.1 Missing Values (Per Column, All Datasets) 6](#_Toc208344063)

[3.2 3.2 Duplicates (Row-Level) 12](#_Toc208344064)

[3.3 3.3 Outliers (Numeric Columns, |z| > 3) 14](#_Toc208344065)

[3.4 3.4 Consolidated Data Quality Issues Log 16](#_Toc208344066)

[4 4. Preliminary Visualizations 19](#_Toc208344067)

[4.1 4.1 Highest Variance Numeric Columns Summary 19](#_Toc208344068)

[4.2 4.2 Most Frequent Categories Summary 20](#_Toc208344069)

[5 5. Average Correlation Heatmap (Across All Datasets) 22](#_Toc208344070)

[6 Appendix. Session Info 23](#_Toc208344071)

# 1 1. Load All Datasets

base <- "../../Data/01\_Raw"  
# Optional: set to TRUE to write CSV exports alongside this HTML  
write\_exports <- FALSE  
# Optional: print full duplicate rows if count <= this threshold (to avoid huge output)  
dup\_print\_threshold <- 200L  
  
# Debug: check working directory and if base path exists  
cat("Working directory:", getwd(), "\n")

## Working directory: C:/Users/edcul/OneDrive/Documents/Work/Modules/Year 3/BIN381/data-analysis-dashboard/02\_Project/Milestone\_1/BIN381\_M1\_R

cat("Base path exists:", dir.exists(base), "\n")

## Base path exists: TRUE

cat("Base path contents:", length(list.files(base)), "files\n")

## Base path contents: 13 files

# Detect files  
files\_csv <- list.files(base, pattern = "\\.(csv)$", ignore.case = TRUE, full.names = TRUE)  
files\_xlsx <- list.files(base, pattern = "\\.(xlsx)$", ignore.case = TRUE, full.names = TRUE)  
  
# Read helpers  
read\_csv\_clean <- function(path){  
 # Read first line as headers, skip the comment line  
 headers <- readr::read\_lines(path, n\_max = 1)  
 readr::read\_csv(path, show\_col\_types = FALSE, skip = 2, col\_names = strsplit(headers, ",")[[1]]) |> janitor::clean\_names()  
}  
read\_xlsx\_all <- function(path){  
 sh <- readxl::excel\_sheets(path)  
 setNames(  
 purrr::map(sh, ~ readxl::read\_excel(path, sheet = .x) |> janitor::clean\_names() |> as\_tibble() ),  
 paste0(tools::file\_path\_sans\_ext(basename(path)), "\_\_", sh)  
 )  
}  
  
# Load data  
dfs\_csv <- purrr::map(files\_csv, read\_csv\_clean); names(dfs\_csv) <- tools::file\_path\_sans\_ext(basename(files\_csv))  
dfs\_xlsx <- purrr::map(files\_xlsx, read\_xlsx\_all); dfs\_xlsx <- if(length(dfs\_xlsx)) purrr::list\_flatten(dfs\_xlsx) else list()  
dfs <- c(dfs\_csv, dfs\_xlsx)  
  
# Ensure unique names  
if(length(dfs)){  
 names(dfs) <- make.unique(names(dfs), sep = "\_")  
}  
  
# Inventory  
inventory <- tibble(  
 dataset = names(dfs),  
 rows = purrr::map\_int(dfs, nrow),  
 cols = purrr::map\_int(dfs, ncol)  
) |> arrange(dataset)  
  
if(nrow(inventory) == 0){  
 stop("No datasets found. Place this .Rmd in the folder with your CSV/XLSX files and Knit again.")  
}  
  
gt::gt(inventory)

| dataset | rows | cols |
| --- | --- | --- |
| access-to-health-care\_national\_zaf | 275 | 29 |
| anthropometry\_national\_zaf | 37 | 29 |
| child-mortality-rates\_national\_zaf | 40 | 29 |
| covid-19-prevention\_national\_zaf | 34 | 29 |
| dhs-quickstats\_national\_zaf | 52 | 29 |
| hiv-behavior\_national\_zaf | 118 | 29 |
| immunization\_national\_zaf | 116 | 29 |
| iycf\_national\_zaf | 22 | 29 |
| literacy\_national\_zaf | 20 | 29 |
| maternal-mortality\_national\_zaf | 21 | 29 |
| symptoms-of-acute-respiratory-infection-ari\_national\_zaf | 26 | 29 |
| toilet-facilities\_national\_zaf | 46 | 29 |
| water\_national\_zaf | 100 | 29 |

# 2 2. Dataset-Level Summary

dataset\_summary <- purrr::imap\_dfr(dfs, function(df, nm){  
 n\_rows <- nrow(df); n\_cols <- ncol(df)  
 dup\_rows <- sum(duplicated(df))  
 total\_cells <- n\_rows \* n\_cols  
 miss\_cells <- sum(is.na(df))  
 miss\_pct <- if (total\_cells > 0) round(100 \* miss\_cells / total\_cells, 2) else 0  
 num\_cols <- df |> dplyr::select(where(is.numeric)) |> ncol()  
 tibble(  
 dataset = nm,  
 rows = n\_rows,  
 cols = n\_cols,  
 duplicate\_rows = dup\_rows,  
 missing\_cells = miss\_cells,  
 missing\_pct = miss\_pct,  
 numeric\_cols = num\_cols,  
 categorical\_cols = n\_cols - num\_cols  
 )  
}) |> arrange(dataset)  
  
gt::gt(dataset\_summary)

| dataset | rows | cols | duplicate\_rows | missing\_cells | missing\_pct | numeric\_cols | categorical\_cols |
| --- | --- | --- | --- | --- | --- | --- | --- |
| access-to-health-care\_national\_zaf | 275 | 29 | 0 | 1181 | 14.81 | 13 | 16 |
| anthropometry\_national\_zaf | 37 | 29 | 0 | 193 | 17.99 | 13 | 16 |
| child-mortality-rates\_national\_zaf | 40 | 29 | 0 | 192 | 16.55 | 15 | 14 |
| covid-19-prevention\_national\_zaf | 34 | 29 | 0 | 174 | 17.65 | 13 | 16 |
| dhs-quickstats\_national\_zaf | 52 | 29 | 0 | 249 | 16.51 | 15 | 14 |
| hiv-behavior\_national\_zaf | 118 | 29 | 0 | 667 | 19.49 | 13 | 16 |
| immunization\_national\_zaf | 116 | 29 | 0 | 536 | 15.93 | 13 | 16 |
| iycf\_national\_zaf | 22 | 29 | 0 | 114 | 17.87 | 13 | 16 |
| literacy\_national\_zaf | 20 | 29 | 0 | 104 | 17.93 | 13 | 16 |
| maternal-mortality\_national\_zaf | 21 | 29 | 0 | 133 | 21.84 | 15 | 14 |
| symptoms-of-acute-respiratory-infection-ari\_national\_zaf | 26 | 29 | 0 | 120 | 15.92 | 13 | 16 |
| toilet-facilities\_national\_zaf | 46 | 29 | 0 | 238 | 17.84 | 13 | 16 |
| water\_national\_zaf | 100 | 29 | 0 | 508 | 17.52 | 13 | 16 |

# 3 3. Data Quality Assessment

## 3.1 3.1 Missing Values (Per Column, All Datasets)

# Get all unique column names across datasets  
all\_columns <- unique(unlist(lapply(dfs, names)))  
  
# Create 2D table: rows = datasets, columns = fields, values = missing counts  
missingness\_2d <- purrr::imap\_dfr(dfs, function(df, nm){  
 total\_rows <- nrow(df)  
   
 # Create row for this dataset with all possible columns  
 row\_data <- tibble(dataset = nm)  
 for(col in all\_columns) {  
 if(col %in% names(df)) {  
 missing\_count <- sum(is.na(df[[col]]))  
 # If all entries are missing, the field effectively doesn't exist  
 if(missing\_count == total\_rows) {  
 row\_data[[col]] <- "N/A"  
 } else {  
 row\_data[[col]] <- as.character(missing\_count)  
 }  
 } else {  
 # Column doesn't exist in this dataset  
 row\_data[[col]] <- "N/A"  
 }  
 }  
 row\_data  
})  
  
# Display the 2D table with heatmap background colors  
numeric\_cols <- names(missingness\_2d)[-1] # exclude 'dataset' column  
  
# Create numeric version for gt color scaling  
missingdata <- missingness\_2d  
for(col in numeric\_cols) {  
 missingdata[[col]] <- ifelse(missingness\_2d[[col]] == "N/A", NA, as.numeric(missingness\_2d[[col]]))  
}  
  
# Get the actual range of values for proper scaling  
all\_values <- unlist(missingdata[numeric\_cols])  
all\_values <- all\_values[!is.na(all\_values)]  
max\_val <- if(length(all\_values) > 0) max(all\_values) else 1  
# Display a simplified missing values table that fits on A4  
print("Creating missing values table...")

## [1] "Creating missing values table..."

gt::gt(missingdata) |>  
 gt::tab\_header(title = "Missing Values (Count) - 2D View") |>  
 gt::data\_color(  
 columns = all\_of(numeric\_cols),  
 palette = c("white", "darkred"),  
 domain = c(0, max\_val),  
 na\_color = "lightgray"  
 ) |>  
 gt::fmt\_missing(columns = all\_of(numeric\_cols), missing\_text = "N/A") |>  
 gt::tab\_options(  
 table.font.size = px(8),  
 column\_labels.font.size = px(8),  
 data\_row.padding = px(2)  
 )

Table : Missing Values (Count) - 2D View

| dataset | iso3 | data\_id | indicator | value | precision | dhs\_country\_code | country\_name | survey\_year | survey\_id | indicator\_id | indicator\_order | indicator\_type | characteristic\_id | characteristic\_order | characteristic\_category | characteristic\_label | by\_variable\_id | by\_variable\_label | is\_total | is\_preferred | sdrid | region\_id | survey\_year\_label | survey\_type | denominator\_weighted | denominator\_unweighted | ci\_low | ci\_high | level\_rank |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| access-to-health-care\_national\_zaf | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 0 | 0 | 0 | N/A | 0 | 0 | 34 | 34 | N/A | N/A | N/A |
| anthropometry\_national\_zaf | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | N/A | 0 | 0 | 0 | N/A | 0 | 0 | 4 | 4 | N/A | N/A | N/A |
| child-mortality-rates\_national\_zaf | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 0 | 0 | 0 | N/A | 0 | 0 | 36 | 36 | 10 | 10 | N/A |
| covid-19-prevention\_national\_zaf | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | N/A | 0 | 0 | 0 | N/A | 0 | 0 | 2 | 2 | N/A | N/A | N/A |
| dhs-quickstats\_national\_zaf | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 33 | 0 | 0 | 0 | N/A | 0 | 0 | 18 | 18 | 38 | 38 | N/A |
| hiv-behavior\_national\_zaf | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | N/A | 0 | 0 | 0 | N/A | 0 | 0 | 39 | 38 | N/A | N/A | N/A |
| immunization\_national\_zaf | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 56 | 0 | 0 | 0 | N/A | 0 | 0 | 8 | 8 | N/A | N/A | N/A |
| iycf\_national\_zaf | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | N/A | 0 | 0 | 0 | N/A | 0 | 0 | 2 | 2 | N/A | N/A | N/A |
| literacy\_national\_zaf | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | N/A | 0 | 0 | 0 | N/A | 0 | 0 | 2 | 2 | N/A | N/A | N/A |
| maternal-mortality\_national\_zaf | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | N/A | 0 | 0 | 0 | N/A | 0 | 0 | 19 | 15 | 18 | 18 | N/A |
| symptoms-of-acute-respiratory-infection-ari\_national\_zaf | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | N/A | 0 | 0 | 8 | 8 | N/A | N/A | N/A |
| toilet-facilities\_national\_zaf | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | N/A | 0 | 0 | 0 | N/A | 0 | 0 | 4 | 4 | N/A | N/A | N/A |
| water\_national\_zaf | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | N/A | 0 | 0 | 0 | N/A | 0 | 0 | 4 | 4 | N/A | N/A | N/A |

## 3.2 3.2 Duplicates (Row-Level)

# Count duplicates per dataset (across all columns)  
dup\_summary <- purrr::imap\_dfr(dfs, function(df, nm){  
 tibble(dataset = nm, duplicate\_rows = sum(duplicated(df)))  
}) |> arrange(desc(duplicate\_rows))  
  
# Create 2D table: datasets as rows, duplicate\_rows as column  
dup\_2d <- dup\_summary |>  
 select(dataset, duplicate\_rows)  
  
# Apply heatmap styling to duplicates table  
max\_dup\_val <- max(dup\_2d$duplicate\_rows, na.rm = TRUE)  
  
gt::gt(dup\_2d) |>  
 gt::tab\_header(title = "Duplicate Rows Count - 2D View") |>  
 gt::data\_color(  
 columns = duplicate\_rows,  
 palette = c("white", "darkred"),  
 domain = c(0, max\_dup\_val),  
 na\_color = "lightgray"  
 )

Table : Duplicate Rows Count - 2D View

| dataset | duplicate\_rows |
| --- | --- |
| access-to-health-care\_national\_zaf | 0 |
| anthropometry\_national\_zaf | 0 |
| child-mortality-rates\_national\_zaf | 0 |
| covid-19-prevention\_national\_zaf | 0 |
| dhs-quickstats\_national\_zaf | 0 |
| hiv-behavior\_national\_zaf | 0 |
| immunization\_national\_zaf | 0 |
| iycf\_national\_zaf | 0 |
| literacy\_national\_zaf | 0 |
| maternal-mortality\_national\_zaf | 0 |
| symptoms-of-acute-respiratory-infection-ari\_national\_zaf | 0 |
| toilet-facilities\_national\_zaf | 0 |
| water\_national\_zaf | 0 |

## 3.3 3.3 Outliers (Numeric Columns, |z| > 3)

outlier\_counts <- function(df){  
 nums <- df |> dplyr::select(where(is.numeric))  
 if(ncol(nums) == 0) return(tibble(column=character(), outliers\_abs\_z\_gt\_3=integer()))  
 purrr::map\_dfr(names(nums), function(col){  
 v <- nums[[col]]  
 v <- v[!is.na(v)]  
 if(length(v) < 5 || sd(v) == 0) return(tibble(column = col, outliers\_abs\_z\_gt\_3 = 0L))  
 z <- (v - mean(v)) / sd(v)  
 tibble(column = col, outliers\_abs\_z\_gt\_3 = as.integer(sum(abs(z) > 3)))  
 }) |> arrange(desc(outliers\_abs\_z\_gt\_3))  
}  
  
outliers\_by\_dataset <- purrr::imap(dfs, function(df, nm){  
 oc <- outlier\_counts(df) |> mutate(dataset = nm, .before = 1)  
 oc  
})  
outliers\_all <- bind\_rows(outliers\_by\_dataset)  
  
if(nrow(outliers\_all) > 0){  
 # Get all unique numeric column names across datasets  
 all\_numeric\_columns <- unique(outliers\_all$column)  
   
 # Create 2D table: rows = datasets, columns = numeric fields, values = outlier counts  
 outliers\_2d <- outliers\_all |>  
 pivot\_wider(names\_from = column, values\_from = outliers\_abs\_z\_gt\_3, values\_fill = 0)  
   
 # Apply heatmap styling to outliers table  
 outlier\_cols <- names(outliers\_2d)[-1] # exclude 'dataset' column  
 max\_outlier\_val <- max(unlist(outliers\_2d[outlier\_cols]), na.rm = TRUE)  
   
 gt::gt(outliers\_2d) |>  
 gt::tab\_header(title = "Outlier Counts (|z|>3) - 2D View") |>  
 gt::data\_color(  
 columns = all\_of(outlier\_cols),  
 palette = c("white", "darkred"),  
 domain = c(0, max\_outlier\_val),  
 na\_color = "lightgray"  
 )  
   
} else {  
 cat("No numeric columns suitable for outlier analysis were found.")  
}

Table : Outlier Counts (|z|>3) - 2D View

| dataset | by\_variable\_id | value | data\_id | precision | survey\_year | indicator\_order | characteristic\_id | characteristic\_order | is\_total | is\_preferred | survey\_year\_label | denominator\_weighted | denominator\_unweighted | ci\_low | ci\_high |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| access-to-health-care\_national\_zaf | 13 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| anthropometry\_national\_zaf | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| child-mortality-rates\_national\_zaf | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| covid-19-prevention\_national\_zaf | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| dhs-quickstats\_national\_zaf | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| hiv-behavior\_national\_zaf | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| immunization\_national\_zaf | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| iycf\_national\_zaf | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| literacy\_national\_zaf | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| maternal-mortality\_national\_zaf | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| symptoms-of-acute-respiratory-infection-ari\_national\_zaf | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| toilet-facilities\_national\_zaf | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| water\_national\_zaf | 0 | 4 | 0 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## 3.4 3.4 Consolidated Data Quality Issues Log

# Build a tidy issues log: one row per issue instance  
issues\_missing <- missingness\_all |>  
 filter(missing\_count > 0 & missing\_pct < 100) |> # Exclude 100% missing (field doesn't exist)  
 transmute(dataset, issue\_type = "missing", column, detail = paste0(missing\_pct, "% (", missing\_count, " cells)"))  
  
issues\_dup <- dup\_summary |>  
 filter(duplicate\_rows > 0) |>  
 transmute(dataset, issue\_type = "duplicates", column = NA\_character\_, detail = paste0(duplicate\_rows, " duplicate rows"))  
  
issues\_outliers <- outliers\_all |>  
 filter(outliers\_abs\_z\_gt\_3 > 0) |>  
 transmute(dataset, issue\_type = "outliers", column, detail = paste0(outliers\_abs\_z\_gt\_3, " outliers (|z|>3)"))  
  
issues\_log <- bind\_rows(issues\_missing, issues\_dup, issues\_outliers) |>  
 arrange(dataset, issue\_type, desc(detail))  
  
gt::gt(issues\_log)

| dataset | issue\_type | column | detail |
| --- | --- | --- | --- |
| access-to-health-care\_national\_zaf | missing | by\_variable\_label | 4.73% (13 cells) |
| access-to-health-care\_national\_zaf | missing | denominator\_unweighted | 12.36% (34 cells) |
| access-to-health-care\_national\_zaf | missing | denominator\_weighted | 12.36% (34 cells) |
| access-to-health-care\_national\_zaf | outliers | value | 8 outliers (|z|>3) |
| access-to-health-care\_national\_zaf | outliers | by\_variable\_id | 13 outliers (|z|>3) |
| anthropometry\_national\_zaf | missing | denominator\_unweighted | 10.81% (4 cells) |
| anthropometry\_national\_zaf | missing | denominator\_weighted | 10.81% (4 cells) |
| child-mortality-rates\_national\_zaf | missing | denominator\_unweighted | 90% (36 cells) |
| child-mortality-rates\_national\_zaf | missing | denominator\_weighted | 90% (36 cells) |
| child-mortality-rates\_national\_zaf | missing | by\_variable\_label | 50% (20 cells) |
| child-mortality-rates\_national\_zaf | missing | ci\_high | 25% (10 cells) |
| child-mortality-rates\_national\_zaf | missing | ci\_low | 25% (10 cells) |
| child-mortality-rates\_national\_zaf | outliers | value | 2 outliers (|z|>3) |
| covid-19-prevention\_national\_zaf | missing | denominator\_unweighted | 5.88% (2 cells) |
| covid-19-prevention\_national\_zaf | missing | denominator\_weighted | 5.88% (2 cells) |
| covid-19-prevention\_national\_zaf | outliers | precision | 1 outliers (|z|>3) |
| dhs-quickstats\_national\_zaf | missing | ci\_high | 73.08% (38 cells) |
| dhs-quickstats\_national\_zaf | missing | ci\_low | 73.08% (38 cells) |
| dhs-quickstats\_national\_zaf | missing | by\_variable\_label | 63.46% (33 cells) |
| dhs-quickstats\_national\_zaf | missing | denominator\_unweighted | 34.62% (18 cells) |
| dhs-quickstats\_national\_zaf | missing | denominator\_weighted | 34.62% (18 cells) |
| dhs-quickstats\_national\_zaf | outliers | value | 2 outliers (|z|>3) |
| dhs-quickstats\_national\_zaf | outliers | by\_variable\_id | 1 outliers (|z|>3) |
| hiv-behavior\_national\_zaf | missing | denominator\_weighted | 33.05% (39 cells) |
| hiv-behavior\_national\_zaf | missing | denominator\_unweighted | 32.2% (38 cells) |
| hiv-behavior\_national\_zaf | outliers | value | 4 outliers (|z|>3) |
| immunization\_national\_zaf | missing | denominator\_unweighted | 6.9% (8 cells) |
| immunization\_national\_zaf | missing | denominator\_weighted | 6.9% (8 cells) |
| immunization\_national\_zaf | missing | by\_variable\_label | 48.28% (56 cells) |
| immunization\_national\_zaf | outliers | value | 2 outliers (|z|>3) |
| iycf\_national\_zaf | missing | denominator\_unweighted | 9.09% (2 cells) |
| iycf\_national\_zaf | missing | denominator\_weighted | 9.09% (2 cells) |
| literacy\_national\_zaf | missing | denominator\_unweighted | 10% (2 cells) |
| literacy\_national\_zaf | missing | denominator\_weighted | 10% (2 cells) |
| literacy\_national\_zaf | outliers | value | 1 outliers (|z|>3) |
| maternal-mortality\_national\_zaf | missing | denominator\_weighted | 90.48% (19 cells) |
| maternal-mortality\_national\_zaf | missing | ci\_high | 85.71% (18 cells) |
| maternal-mortality\_national\_zaf | missing | ci\_low | 85.71% (18 cells) |
| maternal-mortality\_national\_zaf | missing | denominator\_unweighted | 71.43% (15 cells) |
| symptoms-of-acute-respiratory-infection-ari\_national\_zaf | missing | denominator\_unweighted | 30.77% (8 cells) |
| symptoms-of-acute-respiratory-infection-ari\_national\_zaf | missing | denominator\_weighted | 30.77% (8 cells) |
| toilet-facilities\_national\_zaf | missing | denominator\_unweighted | 8.7% (4 cells) |
| toilet-facilities\_national\_zaf | missing | denominator\_weighted | 8.7% (4 cells) |
| toilet-facilities\_national\_zaf | outliers | value | 2 outliers (|z|>3) |
| water\_national\_zaf | missing | denominator\_unweighted | 4% (4 cells) |
| water\_national\_zaf | missing | denominator\_weighted | 4% (4 cells) |
| water\_national\_zaf | outliers | precision | 8 outliers (|z|>3) |
| water\_national\_zaf | outliers | value | 4 outliers (|z|>3) |

# 4 4. Preliminary Visualizations

## 4.1 4.1 Highest Variance Numeric Columns Summary

# Get highest variance column and its variance for each dataset  
variance\_summary <- purrr::imap\_dfr(dfs, function(df, nm){  
 nums <- df |> dplyr::select(where(is.numeric))  
 if(ncol(nums) == 0) return(tibble(dataset = nm, highest\_var\_column = "N/A", variance = "N/A"))  
   
 var\_tbl <- summarize(nums, across(everything(), function(y) var(y, na.rm = TRUE)))  
 var\_results <- var\_tbl |> pivot\_longer(everything(), names\_to="col", values\_to="v") |>  
 arrange(desc(v)) |> slice(1)  
   
 tibble(  
 dataset = nm,  
 highest\_var\_column = var\_results$col,  
 variance = as.character(round(var\_results$v, 2))  
 )  
})  
  
# Apply heatmap styling to variance values  
variance\_for\_gt <- variance\_summary  
variance\_for\_gt$variance\_numeric <- ifelse(variance\_summary$variance == "N/A", NA, as.numeric(variance\_summary$variance))  
  
max\_var <- max(variance\_for\_gt$variance\_numeric, na.rm = TRUE)  
  
gt::gt(variance\_for\_gt |> select(-variance\_numeric)) |>  
 gt::tab\_header(title = "Highest Variance Numeric Columns by Dataset") |>  
 gt::data\_color(  
 columns = variance,  
 palette = c("white", "darkblue"),  
 domain = c(0, max\_var),  
 na\_color = "lightgray"  
 )

Table : Highest Variance Numeric Columns by Dataset

| dataset | highest\_var\_column | variance |
| --- | --- | --- |
| access-to-health-care\_national\_zaf | indicator\_order | 25022211281567.1 |
| anthropometry\_national\_zaf | indicator\_order | 22901215200981.2 |
| child-mortality-rates\_national\_zaf | data\_id | 97344762672.13 |
| covid-19-prevention\_national\_zaf | indicator\_order | 16089366442421.4 |
| dhs-quickstats\_national\_zaf | indicator\_order | 3738559151946484 |
| hiv-behavior\_national\_zaf | data\_id | 49066328853.83 |
| immunization\_national\_zaf | data\_id | 61479147528.61 |
| iycf\_national\_zaf | data\_id | 49203252603.6 |
| literacy\_national\_zaf | data\_id | 4569355769.73 |
| maternal-mortality\_national\_zaf | data\_id | 91243653644.16 |
| symptoms-of-acute-respiratory-infection-ari\_national\_zaf | data\_id | 45078540239.26 |
| toilet-facilities\_national\_zaf | data\_id | 108611672796.4 |
| water\_national\_zaf | data\_id | 49926220181.53 |

## 

## 4.2 4.2 Most Frequent Categories Summary

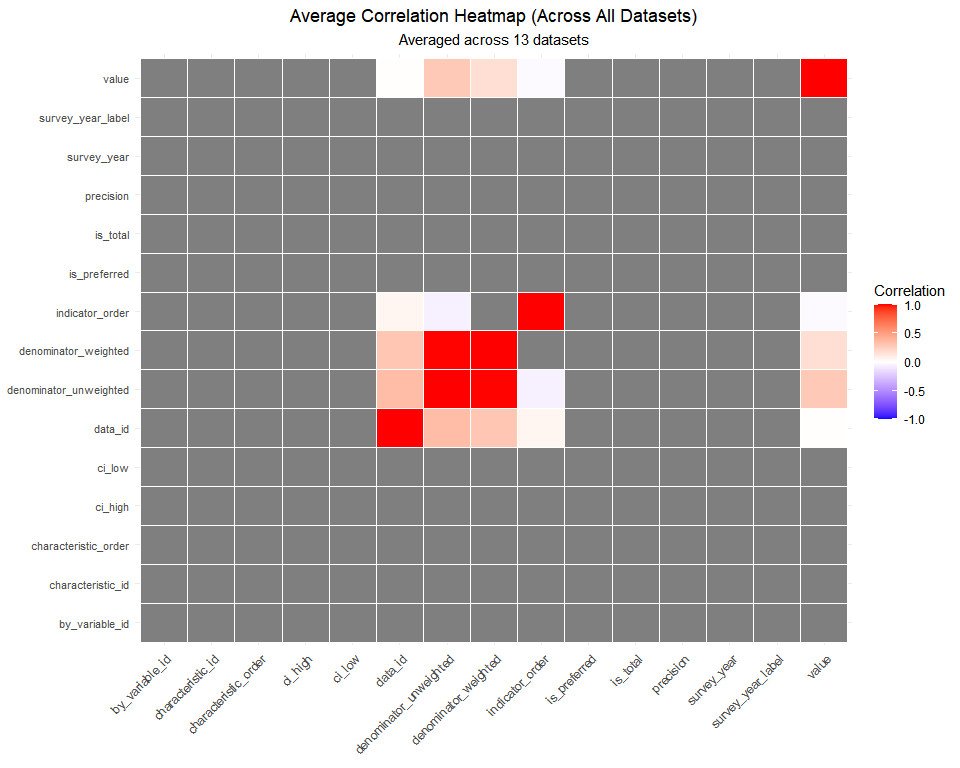
# Get most frequent category from first categorical column for each dataset  
category\_summary <- purrr::imap\_dfr(dfs, function(df, nm){  
 cats <- df |> dplyr::select(where(negate(is.numeric)))  
 if(ncol(cats) == 0) return(tibble(dataset = nm, categorical\_column = "N/A", most\_frequent\_value = "N/A", frequency = "N/A"))  
   
 col1 <- names(cats)[1]  
 top\_category <- df |> mutate(across(all\_of(col1), as.character)) |>  
 count(.data[[col1]], sort = TRUE) |> slice\_head(n = 1)  
   
 tibble(  
 dataset = nm,  
 categorical\_column = col1,  
 most\_frequent\_value = top\_category[[col1]][1],  
 frequency = as.character(top\_category$n[1])  
 )  
})  
  
# Apply heatmap styling to frequency values  
category\_for\_gt <- category\_summary  
category\_for\_gt$frequency\_numeric <- ifelse(category\_summary$frequency == "N/A", NA, as.numeric(category\_summary$frequency))  
  
max\_freq <- max(category\_for\_gt$frequency\_numeric, na.rm = TRUE)  
  
gt::gt(category\_for\_gt |> select(-frequency\_numeric)) |>  
 gt::tab\_header(title = "Most Frequent Categories by Dataset") |>  
 gt::data\_color(  
 columns = frequency,  
 palette = c("white", "darkgreen"),  
 domain = c(0, max\_freq),  
 na\_color = "lightgray"  
 )

Table : Most Frequent Categories by Dataset

| dataset | categorical\_column | most\_frequent\_value | frequency |
| --- | --- | --- | --- |
| access-to-health-care\_national\_zaf | iso3 | ZAF | 275 |
| anthropometry\_national\_zaf | iso3 | ZAF | 37 |
| child-mortality-rates\_national\_zaf | iso3 | ZAF | 40 |
| covid-19-prevention\_national\_zaf | iso3 | ZAF | 34 |
| dhs-quickstats\_national\_zaf | iso3 | ZAF | 52 |
| hiv-behavior\_national\_zaf | iso3 | ZAF | 118 |
| immunization\_national\_zaf | iso3 | ZAF | 116 |
| iycf\_national\_zaf | iso3 | ZAF | 22 |
| literacy\_national\_zaf | iso3 | ZAF | 20 |
| maternal-mortality\_national\_zaf | iso3 | ZAF | 21 |
| symptoms-of-acute-respiratory-infection-ari\_national\_zaf | iso3 | ZAF | 26 |
| toilet-facilities\_national\_zaf | iso3 | ZAF | 46 |
| water\_national\_zaf | iso3 | ZAF | 100 |

# 5 5. Average Correlation Heatmap (Across All Datasets)

# Get all unique numeric column names across datasets  
all\_numeric\_columns <- unique(unlist(lapply(dfs, function(df) names(df |> dplyr::select(where(is.numeric))))))  
  
if(length(all\_numeric\_columns) >= 2) {  
 # Calculate correlation matrices for each dataset and average them  
 correlation\_matrices <- purrr::map(dfs, function(df){  
 nums <- df |> dplyr::select(where(is.numeric))  
 if(ncol(nums) < 2) return(NULL)  
   
 # Ensure we have all columns (fill missing with NA)  
 for(col in all\_numeric\_columns) {  
 if(!col %in% names(nums)) {  
 nums[[col]] <- NA  
 }  
 }  
   
 # Reorder columns to match all\_numeric\_columns  
 nums <- nums |> select(all\_of(all\_numeric\_columns))  
   
 # Calculate correlation matrix  
 cor(nums, use = "pairwise.complete.obs")  
 })  
   
 # Remove NULL matrices (datasets with < 2 numeric columns)  
 correlation\_matrices <- correlation\_matrices[!sapply(correlation\_matrices, is.null)]  
   
 if(length(correlation\_matrices) > 0) {  
 # Average the correlation matrices  
 avg\_corr\_matrix <- Reduce("+", correlation\_matrices) / length(correlation\_matrices)  
   
 # Convert to tidy format for ggplot  
 tidy\_corr <- as\_tibble(avg\_corr\_matrix, rownames = "row") |>  
 pivot\_longer(-row, names\_to = "col", values\_to = "corr")  
   
 # Create heatmap  
 ggplot(tidy\_corr, aes(x = row, y = col, fill = corr)) +  
 geom\_tile(color = "white", size = 0.5) +  
 scale\_fill\_gradient2(low = "blue", mid = "white", high = "red",   
 midpoint = 0, limits = c(-1, 1), name = "Correlation") +  
 labs(title = "Average Correlation Heatmap (Across All Datasets)",  
 subtitle = paste("Averaged across", length(correlation\_matrices), "datasets"),  
 x = NULL, y = NULL) +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1, vjust = 1),  
 axis.text.y = element\_text(size = 8),  
 plot.title = element\_text(hjust = 0.5),  
 plot.subtitle = element\_text(hjust = 0.5))  
 } else {  
 cat("No datasets have sufficient numeric columns for correlation analysis.")  
 }  
} else {  
 cat("Insufficient numeric columns across all datasets for correlation analysis.")  
}



# 6 Appendix. Session Info

sessionInfo()

## R version 4.5.1 (2025-06-13 ucrt)  
## Platform: x86\_64-w64-mingw32/x64  
## Running under: Windows 11 x64 (build 26100)  
##   
## Matrix products: default  
## LAPACK version 3.12.1  
##   
## locale:  
## [1] LC\_COLLATE=English\_United Kingdom.utf8   
## [2] LC\_CTYPE=English\_United Kingdom.utf8   
## [3] LC\_MONETARY=English\_United Kingdom.utf8  
## [4] LC\_NUMERIC=C   
## [5] LC\_TIME=English\_United Kingdom.utf8   
##   
## time zone: Africa/Johannesburg  
## tzcode source: internal  
##   
## attached base packages:  
## [1] stats graphics grDevices utils datasets methods base   
##   
## other attached packages:  
## [1] gt\_1.0.0 janitor\_2.2.1 readxl\_1.4.5 lubridate\_1.9.4  
## [5] forcats\_1.0.0 stringr\_1.5.1 dplyr\_1.1.4 purrr\_1.1.0   
## [9] readr\_2.1.5 tidyr\_1.3.1 tibble\_3.3.0 ggplot2\_3.5.2   
## [13] tidyverse\_2.0.0  
##   
## loaded via a namespace (and not attached):  
## [1] bit\_4.6.0 gtable\_0.3.6 crayon\_1.5.3 compiler\_4.5.1   
## [5] tidyselect\_1.2.1 xml2\_1.4.0 parallel\_4.5.1 snakecase\_0.11.1   
## [9] scales\_1.4.0 yaml\_2.3.10 fastmap\_1.2.0 R6\_2.6.1   
## [13] labeling\_0.4.3 generics\_0.1.4 knitr\_1.50 pillar\_1.11.0   
## [17] RColorBrewer\_1.1-3 tzdb\_0.5.0 rlang\_1.1.6 stringi\_1.8.7   
## [21] xfun\_0.52 bit64\_4.6.0-1 timechange\_0.3.0 cli\_3.6.5   
## [25] withr\_3.0.2 magrittr\_2.0.3 digest\_0.6.37 grid\_4.5.1   
## [29] vroom\_1.6.5 rstudioapi\_0.17.1 hms\_1.1.3 lifecycle\_1.0.4   
## [33] vctrs\_0.6.5 evaluate\_1.0.5 glue\_1.8.0 cellranger\_1.1.0   
## [37] farver\_2.1.2 rmarkdown\_2.29 tools\_4.5.1 pkgconfig\_2.0.3   
## [41] htmltools\_0.5.8.1