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

2025-09-09

Contents

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-dashboa
```

```
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]]) |> ja
}
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(basenam
dfs_xlsx <- purrr::map(files_xlsx, read_xlsx_all); dfs_xlsx <- if(length(dfs_xlsx)) purrr::list_flatten
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)
```

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

```
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	duplicate_rows	missing_cells
access-to-health-care_national_zaf	275	29	0	1181
anthropometry_national_zaf	37	29	0	193
child-mortality-rates_national_zaf	40	29	0	192
covid-19-prevention_national_zaf	34	29	0	174
dhs-quickstats_national_zaf	52	29	0	249
hiv-behavior_national_zaf	118	29	0	667
immunization_national_zaf	116	29	0	536
iycf_national_zaf	22	29	0	114
literacy_national_zaf	20	29	0	104
maternal-mortality_national_zaf	21	29	0	133
symptoms-of-acute-respiratory-infection-ari_national_zaf	26	29	0	120
toilet-facilities_national_zaf	46	29	0	238
water_national_zaf	100	29	0	508

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)
```

```
if (write_exports) {
  readr::write_csv(dataset_summary, "M1_dataset_summary.csv")
}
```

3. Data Quality Assessment

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"
```

Missing Values (Count) - 2D View

dataset	iso3	data_id	indicator	value	precision	dhs_country_code	country_name	survey_year
access-to-health-care_national_zaf	0	0	0	0	0	0	0	0
anthropometry_national_zaf	0	0	0	0	0	0	0	0
child-mortality-rates_national_zaf	0	0	0	0	0	0	0	0
covid-19-prevention_national_zaf	0	0	0	0	0	0	0	0
dhs-quickstats_national_zaf	0	0	0	0	0	0	0	0
hiv-behavior_national_zaf	0	0	0	0	0	0	0	0
immunization_national_zaf	0	0	0	0	0	0	0	0
iycf_national_zaf	0	0	0	0	0	0	0	0
literacy_national_zaf	0	0	0	0	0	0	0	0
maternal-mortality_national_zaf	0	0	0	0	0	0	0	0
symptoms-of-acute-respiratory-infection-ari_national_zaf	0	0	0	0	0	0	0	0
toilet-facilities_national_zaf	0	0	0	0	0	0	0	0
water_national_zaf	0	0	0	0	0	0	0	0

```
  ) |>
  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)
  )
```

```
# Also keep the flat format for exports if needed
missingness_all <- purrr::imap(dfs, function(df, nm){
  tibble(
    dataset = nm,
    column = names(df),
    missing_pct = round(colMeans(is.na(df))*100, 2),
    missing_count = colSums(is.na(df))
  )
}) |> bind_rows() |> arrange(desc(missing_pct), dataset, column)

if (write_exports) {
  readr::write_csv(missingness_2d, "M1_missingness_2d_table.csv")
  readr::write_csv(missingness_all, "M1_missingness_all_columns.csv")
}
```

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

3.2 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"
  )
```

```
if (write_exports) {
  readr::write_csv(dup_2d, "M1_duplicates_2d_table.csv")
  readr::write_csv(dup_summary, "M1_duplicates_by_dataset.csv")
}

# Optionally show duplicate rows when counts are manageable
purrr::iwalk(dfs, function(df, nm){
  dup_ct <- sum(duplicated(df))
  if (dup_ct > 0 && dup_ct <= dup_print_threshold) {
    cat("\n\n### Duplicate Rows -", nm, "(showing all duplicates because count <=", dup_print_threshold
```

```
    dups <- df[duplicated(df) | duplicated(df, fromLast = TRUE), , drop = FALSE]
    print(dups)
    if (write_exports) readr::write_csv(dups, paste0("M1_duplicates__", nm, ".csv"))
  } else if (dup_ct > dup_print_threshold) {
    cat("\n\n### Duplicate Rows -", nm, "(too many to print; exporting if write_exports=TRUE)\n\n")
    if (write_exports) {
      dups <- df[duplicated(df) | duplicated(df, fromLast = TRUE), , drop = FALSE]
      readr::write_csv(dups, paste0("M1_duplicates__", nm, ".csv"))
    }
  }
})
```

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.")
}
```

```
if (write_exports) {
  if(nrow(outliers_all) > 0) {
    readr::write_csv(outliers_2d, "M1_outliers_2d_table.csv")
  }
  readr::write_csv(outliers_all, "M1_outliers_all_numeric_columns.csv")
}
```

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

dataset	by_variable_id	value	data_id	precision
access-to-health-care_national_zaf	13	8	0	0
anthropometry_national_zaf	0	0	0	0
child-mortality-rates_national_zaf	0	2	0	0
covid-19-prevention_national_zaf	0	0	0	1
dhs-quickstats_national_zaf	1	2	0	0
hiv-behavior_national_zaf	0	4	0	0
immunization_national_zaf	0	2	0	0
iycf_national_zaf	0	0	0	0
literacy_national_zaf	0	1	0	0
maternal-mortality_national_zaf	0	0	0	0
symptoms-of-acute-respiratory-infection-ari_national_zaf	0	0	0	0
toilet-facilities_national_zaf	0	2	0	0
water_national_zaf	0	4	0	8

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,

issues_dup <- dup_summary |>
  filter(duplicate_rows > 0) |>
  transmute(dataset, issue_type = "duplicates", column = NA_character_, detail = paste0(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 (

issues_log <- bind_rows(issues_missing, issues_dup, issues_outliers) |>
  arrange(dataset, issue_type, desc(detail))

gt::gt(issues_log)
```

```
if (write_exports) {
  readr::write_csv(issues_log, "M1_data_quality_issues_log.csv")
}
```

4. Preliminary Visualizations

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_s

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"
  )
```

4.2 Most Frequent Categories Summary

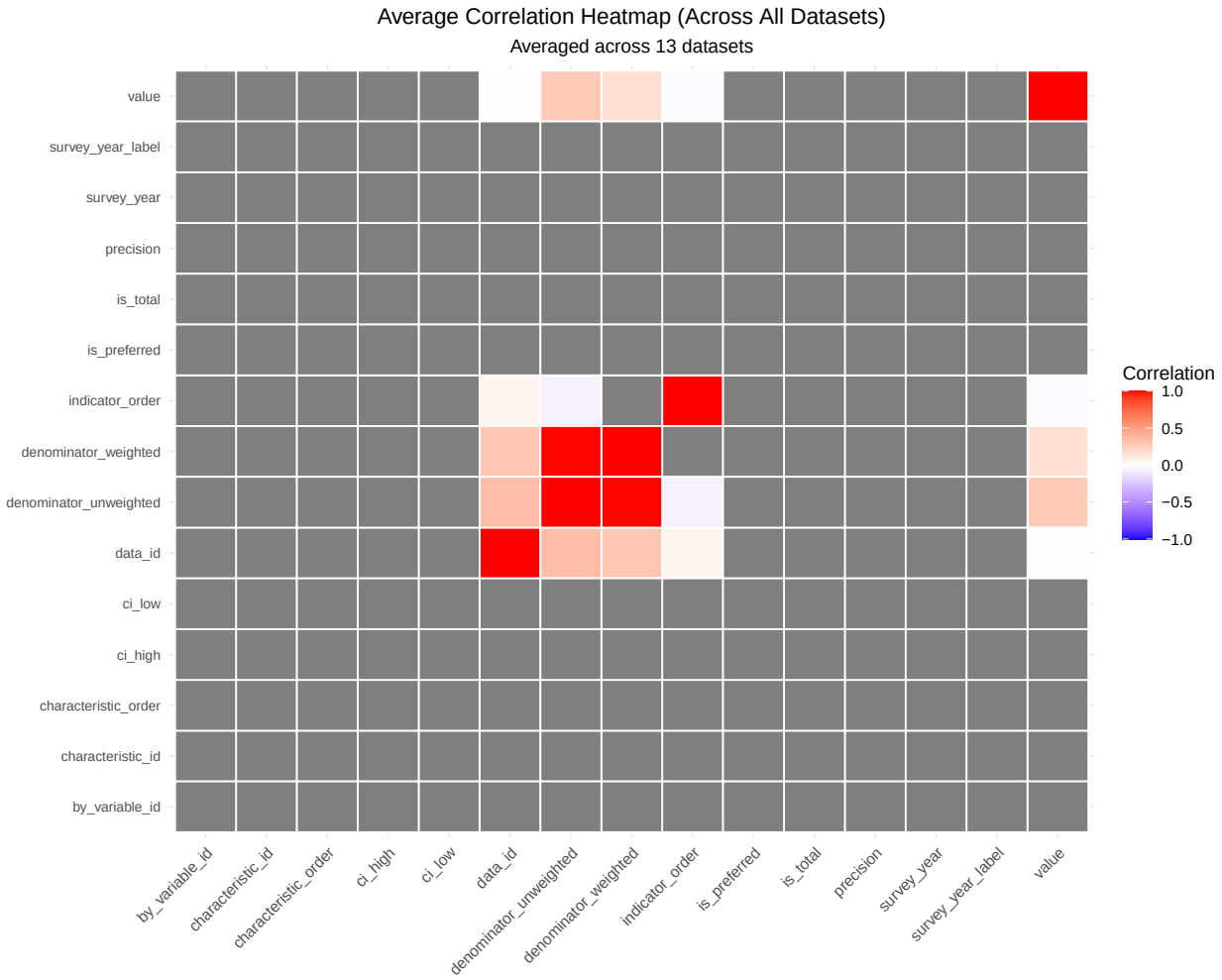
```
# Get most frequent category from first categorical column for each dataset
category_summary <- purrr::imap_dfr(dfs, function(df, nm){
  cats <- df |> dplyr::select(where(negate(is.numeric)))
  if(ncol(cats) == 0) return(tibble(dataset = nm, categorical_column = "N/A", most_frequent_value = "N/

  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])
  )
})
```

```r
# 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(categor

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"
  )
```

5. Average Correlation Heatmap (Across All Datasets)

```r
# Get all unique numeric column names across datasets
all_numeric_columns <- unique(unlist(lapply(dfs, function(df) names(df |> dplyr::select(where(is.numeri

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.")
}
```



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
```

dataset	issue_type	column	deta
access-to-health-care_national_zaf	missing	by_variable_label	4.73%
access-to-health-care_national_zaf	missing	denominator_unweighted	12.36
access-to-health-care_national_zaf	missing	denominator_weighted	12.36
access-to-health-care_national_zaf	outliers	value	8 ou
access-to-health-care_national_zaf	outliers	by_variable_id	13 o
anthropometry_national_zaf	missing	denominator_unweighted	10.81
anthropometry_national_zaf	missing	denominator_weighted	10.81
child-mortality-rates_national_zaf	missing	denominator_unweighted	90%
child-mortality-rates_national_zaf	missing	denominator_weighted	90%
child-mortality-rates_national_zaf	missing	by_variable_label	50%
child-mortality-rates_national_zaf	missing	ci_high	25%
child-mortality-rates_national_zaf	missing	ci_low	25%
child-mortality-rates_national_zaf	outliers	value	2 ou
covid-19-prevention_national_zaf	missing	denominator_unweighted	5.88%
covid-19-prevention_national_zaf	missing	denominator_weighted	5.88%
covid-19-prevention_national_zaf	outliers	precision	1 ou
dhs-quickstats_national_zaf	missing	ci_high	73.08
dhs-quickstats_national_zaf	missing	ci_low	73.08
dhs-quickstats_national_zaf	missing	by_variable_label	63.46
dhs-quickstats_national_zaf	missing	denominator_unweighted	34.62
dhs-quickstats_national_zaf	missing	denominator_weighted	34.62
dhs-quickstats_national_zaf	outliers	value	2 ou
dhs-quickstats_national_zaf	outliers	by_variable_id	1 ou
hiv-behavior_national_zaf	missing	denominator_weighted	33.05
hiv-behavior_national_zaf	missing	denominator_unweighted	32.2%
hiv-behavior_national_zaf	outliers	value	4 ou
immunization_national_zaf	missing	denominator_unweighted	6.9%
immunization_national_zaf	missing	denominator_weighted	6.9%
immunization_national_zaf	missing	by_variable_label	48.28
immunization_national_zaf	outliers	value	2 ou
iycf_national_zaf	missing	denominator_unweighted	9.09%
iycf_national_zaf	missing	denominator_weighted	9.09%
literacy_national_zaf	missing	denominator_unweighted	10%
literacy_national_zaf	missing	denominator_weighted	10%
literacy_national_zaf	outliers	value	1 ou
maternal-mortality_national_zaf	missing	denominator_weighted	90.48
maternal-mortality_national_zaf	missing	ci_high	85.71
maternal-mortality_national_zaf	missing	ci_low	85.71
maternal-mortality_national_zaf	missing	denominator_unweighted	71.43
symptoms-of-acute-respiratory-infection-ari_national_zaf	missing	denominator_unweighted	30.77
symptoms-of-acute-respiratory-infection-ari_national_zaf	missing	denominator_weighted	30.77
toilet-facilities_national_zaf	missing	denominator_unweighted	8.7%
toilet-facilities_national_zaf	missing	denominator_weighted	8.7%
toilet-facilities_national_zaf	outliers	value	2 ou
water_national_zaf	missing	denominator_unweighted	4% (
water_national_zaf	missing	denominator_weighted	4% (
water_national_zaf	outliers	precision	8 ou
water_national_zaf	outliers	value	4 ou

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

Most Frequent Categories by Dataset

dataset	categorical_column	most_frequent_value
access-to-health-care_national_zaf	iso3	ZAF
anthropometry_national_zaf	iso3	ZAF
child-mortality-rates_national_zaf	iso3	ZAF
covid-19-prevention_national_zaf	iso3	ZAF
dhs-quickstats_national_zaf	iso3	ZAF
hiv-behavior_national_zaf	iso3	ZAF
immunization_national_zaf	iso3	ZAF
iycf_national_zaf	iso3	ZAF
literacy_national_zaf	iso3	ZAF
maternal-mortality_national_zaf	iso3	ZAF
symptoms-of-acute-respiratory-infection-ari_national_zaf	iso3	ZAF
toilet-facilities_national_zaf	iso3	ZAF
water_national_zaf	iso3	ZAF