HDPSA Milestone 1

BIN381 Group A

9/9/2025

Members:

Llewellyn Jakobus Fourie – 601314

Karl Christiaan Schmutz – 577511

Juan Oosthuizen – 600161

Erin David Cullen – 600531

Qhamaninande Ndlume – 601436

Table of Contents

Background

Dataset Context

This project analyses South African health datasets from the Health Data Platform South Africa (HDPSA), comprising 13 national-level datasets following DHS format covering:

- Healthcare access and mortality rates
- Child and maternal health indicators
- Nutrition and immunization data
- Disease prevention and management
- Social determinants (literacy, water, sanitation)

CRISP-DM Methodology

This project follows the CRISP-DM (Cross-Industry Standard Process for Data Mining) methodology, a structured approach for data science projects consisting of six phases:

1. **Business Understanding** - Define objectives and requirements.
2. **Data Understanding** - Collect, describe, and explore data.
3. **Data Preparation** - Clean and transform data for analysis.
4. **Modelling** - Apply analytical techniques.
5. **Evaluation** - Assess model performance and business value.
6. **Deployment** - Plan implementation and monitoring

CRISP-DM provides a proven framework for systematic data analysis, ensuring comprehensive coverage of all project aspects from business objectives through to deployment, making it ideal for this health data analysis project.

Business Objectives and Success Criteria Definition

Business Relevance

Understanding patterns across these interconnected health domains is crucial for:

- Government policy formulation and resource allocation
- Non-profit organizations targeting health interventions.
- International development agencies supporting health programs.
- Healthcare providers planning service delivery.
- Researchers investigating social determinants of health.

The comprehensive nature of this data collection allows for multifaceted analysis revealing relationships between healthcare access, socioeconomic factors, and health outcomes, providing actionable insights for improving population health in South Africa.

Business Problem and Objectives

Business Problem:
Despite numerous past interventions, vulnerable populations in South Africa still experience poor healthcare outcomes due to systematic issues like sanitation, lack of clean water and healthcare access. Stakeholders require reliable, data-driven insights to understand the relationships between living conditions, healthcare access, and healthcare risks.

Business Objectives:

- Provide actionable insights that stakeholders can use to target interventions more effectively.
- Discover the key factors that are currently contributing to poor health outcomes, for example clean water, sanitation, literacy etc.
- Develop interactive dashboards and reports that make complex data easy to understand.
- Identify high-risk regions and communities based on demographic and health indicators.

Stakeholders and Their Requirements

Stakeholder	Requirements	Example KPI
Government (Department of Health)	Identify provinces with the high health risks and resource gaps.	Reduction in child mortality rates (%).
NGOs/NPOs	Prioritize regions for campaigns (HIV prevention, sanitation, immunization)	Increase in immunization coverage (%).
Healthcare Providers and Clinics	Understand the population needs and healthcare access issues.	Clinic to population ratio.
Researchers/Academics	Access to structured datasets and insights for policy studies.	Number of published findings.

Success Criteria

Analytical Success:

- Build models (classification, clustering, association rules) that achieve acceptable performance (for example, prediction accuracy > 70%).
- Identify meaningful clusters with similar healthcare challenges.
- Extract meaningful rules (for example, low literacy + no sanitation = high ARI risk).

Business Success:

- Insights led to practical recommendations for healthcare and community interventions.
- Dashboards highlight priority areas for intervention (geographic or demographic).
- Stakeholders confirm results are useful for guiding decision-making.

Inventory of Resources Assessment

Data Resources

Primary Datasets

13 HDPSA (Health Data Platform South Africa) datasets in DHS (Demographic and Health Surveys) format:

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

Data Processing Pipeline

- **01_Raw/**: Original datasets
- **02_Cleaned/**: Processed data with empty records removed
- **03_Processed/**: Feature engineered data
- **04_Scaled/**: Scaled and encoded data ready for analysis

Documentation

- **Data_Dictionary.md**: Field definitions and structure
- **Data_Pipeline.md**: Processing methodology

Technical Tools

Analysis Software

- **R**: Statistical analysis and data processing
- **R Markdown**: Reproducible reporting and documentation
- **Power BI**: Interactive dashboards and visualizations
- **Web Technologies**: Interactive components for stakeholder review

Development Environment

- Integrated development environment for R and Python
- Version control and collaboration tools
- Data visualization libraries and packages

Risks, Assumptions and Constraints

Risks

A primary risk that we could face is having missing or incomplete data within our dataset. This compromises the accuracy and integrity of the data analysed. Such gaps can lead to a skewed and inaccurate understanding of South African healthcare. This issue extends not only from simple data gaps, such as rural population or specific age brackets.

Relying on national-level data may overshadow regional results, which could lead to misconceptions, causing generalized insights that fail to address specific situations faced by each province and/or community.

Finally, time limitations for modelling and cleaning data could force teams assigned to the project to take shortcuts and cut corners. This leads to sub-par model selections and fills the dataset with errors ranging from insignificant to major, reducing the reliability of the recorded datasets.

Assumptions

There are two fundamental assumptions made for this project. Firstly, all datasets provided will be accurate and a true representation of the population. This assumes that the Demographic and Health Surveys (DHS) are accurate and provide a proper representation of the nation's varied demographic and health environment. It further assumes that sampling (collection, cleaning, storage, etc.) will be bias-free.

Secondly, we assume that stakeholders will use the data collected as actionable insights to enhance healthcare policies. This is a critical component of the project's impact since it assumes organizational buy-in and a commitment to convert data-driven conclusions into concrete policy changes.

Constraints

Since there is no provision for primary data collection, the project’s reliance on the provided datasets represents its biggest limitation. As a result, any research questions not covered by the 13 available datasets cannot be addressed, as the analysis is tightly constrained by existing information. The team is therefore unable to confirm findings using fresh data, making the integrity of the HDPSA datasets crucial to the project’s overall conclusions.

Additionally, most of the datasets are cross-sectional, which makes it difficult to examine patterns or the long-term effects of health-related factors over time. This limitation significantly reduces the possibility of conducting longitudinal research.

Data Mining Goals and Success Criteria

Data-Mining Goals

1. **Predictive Modelling of Health Risks:**
   Develop classification models to predict high-risk populations and regions based on socioeconomic and healthcare access variables, aiming for prediction accuracy above 70%.
2. **Clustering of Communities by Health Challenges:**
   Identify meaningful clusters of communities with similar healthcare access issues and health outcomes to enable targeted interventions.
3. **Association Rule Mining for Key Risk Factors:**
   Extract actionable association rules that reveal combinations of factors (e.g., low literacy and poor sanitation) strongly linked to adverse health outcomes.
4. **Geospatial Analysis for Priority Area Identification:**
   Analyse geographic patterns to pinpoint provinces and districts with the greatest healthcare resource gaps and health risks.
5. **Interactive Dashboard Development:**
   Create user-friendly dashboards that visualize complex data and model results, facilitating stakeholder understanding and decision-making.
6. **Data Structuring and Integration for Research Use:**
   Prepare and structure datasets to support researchers and policymakers in conducting further studies on social determinants of health.

Data-Mining Success Criteria

1. **Model Performance Benchmarks:**
   Classification models must achieve a minimum prediction accuracy of 70% or higher on validation datasets to be considered successful.
2. **Meaningful Pattern Discovery:**
   Clustering and association rule mining should identify interpretable and actionable patterns, such as clusters of communities with similar health risks and rules linking key factors (e.g., sanitation, literacy) to health outcomes, validated by domain experts.
3. **Stakeholder Validation and Usefulness:**
   Insights and visualizations must be confirmed by stakeholders (government, NGOs, healthcare providers) as relevant and useful for guiding targeted health interventions and resource allocation.
4. **Deployment and Accessibility:**
   Interactive dashboards and reports must be successfully deployed and accessible to all key stakeholders, enabling real-time decision-making and ongoing monitoring of health indicators.

Detailed Descriptions of HDPSA Datasets

This document profiles 12 aggregated datasets compiled for the HDPSA project. Each dataset summarises key indicators from nationally representative surveys conducted in South Africa. The profiling includes variable definitions, data types, ranges, distributions, structural characteristics, and methodological notes.

1. Access to Health Care (national, ZAF)

Focus: Accessibility of primary healthcare services, including facility distance, travel time, barriers to access, and antenatal/postnatal coverage.

- Variables: Distance to facility (categorical, <30 mins, 30–59 mins, ≥1 hour); medical aid coverage (binary, yes/no); ANC visits (numeric, 0–10).
- Data types: Categorical, binary, and numeric.
- Value ranges: Proportions 0–100%.
- Distributions: Typically, right skewed for distance/time, normalized percentages for service coverage.
- Structure: Aggregated proportions, stratified by sex, age, and urban/rural.

2. Anthropometry (national, ZAF)

Focus: Nutritional status of children under five years.

- Variables: Stunting (height-for-age z-score < −2), wasting (weight-for-height z-score < −2), underweight (weight-for-age z-score < −2), overweight.
- Data types: Numeric (z-scores) and categorical (prevalence categories).
- Value ranges: Z-scores typically between −6 and +6; prevalence reported as %.
- Distributions: Bell-shaped for z-scores, prevalence clustered around WHO thresholds.
- Structure: National proportions disaggregated by age group and sex.

3. Child Mortality Rates (national, ZAF)

Focus: Neonatal, infant, and under-five mortality.

- Variables: Neonatal mortality (deaths <28 days), infant mortality (<1 year), under-five mortality (<5 years).
- Data types: Numeric (per 1,000 live births).
- Value ranges: Typically, 10–60 per 1,000.
- Distributions: Declining trend over years; higher in rural/poorer quintiles.
- Structure: Time-series rates derived from retrospective birth histories.

4. COVID-19 Prevention (national, ZAF)

Focus: Behavioural and preventive practices during COVID-19.

- Variables: Mask usage (binary), handwashing frequency (ordinal), vaccination awareness (binary).
- Data types: Binary, categorical.
- Value ranges: 0–100% prevalence.
- Distributions: Skewed towards high uptake of basic preventive measures.
- Structure: Cross-sectional, age/sex disaggregation.

5. HIV Behaviour (national, ZAF)

Focus: Risk behaviours and HIV-related knowledge.

- Variables: Condom use at last sex (binary), multiple partners (numeric), HIV testing history (binary), knowledge of PMTCT (binary).
- Data types: Binary, numeric.
- Value ranges: Binary (0/1), proportions 0–100%.
- Distributions: Urban/rural differences; higher condom use among youth.
- Structure: Individual-level survey items aggregated nationally.

6. Immunisation (national, ZAF)

Focus: Child vaccination coverage.

- Variables: BCG, DPT, Polio, Measles, PCV, Rotavirus, fully immunised (binary/coverage %).
- Data types: Binary and percentage.
- Value ranges: 0–100%.
- Distributions: Skewed towards high coverage for BCG, lower for measles.
- Structure: Aggregated for children 12–23 months.

7. Infant and Young Child Feeding (IYCF) (national, ZAF)

Focus: Breastfeeding and complementary feeding.

- Variables: Early initiation, exclusive breastfeeding (0–5 months), minimum dietary diversity (6–23 months).
- Data types: Binary, categorical.
- Value ranges: 0–100% prevalence.
- Distributions: Exclusive breastfeeding low, dietary diversity uneven across wealth quintiles.
- Structure: Aggregated by child age bands.

8. Literacy (national, ZAF)

Focus: Household literacy and education proxies.

- Variables: Adult literacy (binary: can read a sentence), education attainment (categorical levels).
- Data types: Binary, categorical.
- Value ranges: 0–100%.
- Distributions: Urban–rural disparities; female literacy lower in certain regions.
- Structure: Aggregated national proportions.

9. Maternal Mortality (national, ZAF)

Focus: Maternal health outcomes.

- Variables: Maternal mortality ratio (numeric, per 100,000 live births), skilled birth attendance (binary), facility delivery (binary).
- Data types: Numeric, binary.
- Value ranges: Ratios typically 100–500.
- Distributions: Higher ratios in rural provinces.
- Structure: Modelled estimates with survey indicators.

10. Symptoms of Acute Respiratory Infection (ARI) (national, ZAF)

Focus: Prevalence of ARI symptoms among under-fives.

- Variables: Cough, rapid breathing, care-seeking for ARI.
- Data types: Binary, categorical.
- Value ranges: Prevalence typically 5–20%.
- Distributions: Higher among poorer households.
- Structure: Two-week recall aggregated nationally.

11. Toilet Facilities (national, ZAF)

Focus: Sanitation access.

- Variables: Toilet type (improved/unimproved), shared facility (binary).
- Data types: Categorical, binary.
- Value ranges: 0–100% proportions.
- Distributions: Skewed by urban–rural divide.
- Structure: Aggregated at national level.

12. Water (national, ZAF)

Focus: Drinking water access.

- Variables: Source type (improved/unimproved), on-premises availability, time to fetch.
- Data types: Categorical, binary.
- Value ranges: 0–100% prevalence.
- Distributions: Improved sources dominant in urban areas, longer collection times in rural.
- Structure: Aggregated proportions nationally.

Data Sources and Methodology

All datasets are based on the South Africa Demographic and Health Survey (SADHS) 2016 and related national health surveys. The SADHS employed a two-stage stratified cluster sampling design: in the first stage, 750 enumeration areas (EAs) were selected from the national sampling frame, stratified by province and urban/rural status. In the second stage, households were systematically sampled within EAs. A total of 11,083 households were selected, with interviews conducted for 8,514 women (15–49 years) and 3,618 men (15–59 years) (Statistics South Africa et al., 2017).

Indicators such as anthropometry and immunisation were collected through biomarker measurements and vaccination card/recall, while literacy and access to services were captured via household and individual questionnaires. Mortality estimates were derived from retrospective birth histories.

Data Understanding

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

Dataset-Level Summary

```r
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

Data Quality Assessment

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 1: 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

Duplicates (Row-Level)

```r
# 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 2: 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

Outliers (Numeric Columns, |z| > 3)

```r
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 3: 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

Consolidated Data Quality Issues Log

```r
# 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)

Preliminary Visualizations

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 4: 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 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 5: 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

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.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.")
}
```

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
```

References

- Statistics South Africa, South African Medical Research Council (SAMRC) & ICF. (2017). *South Africa Demographic and Health Survey 2016: Key Indicators Report*. Pretoria, South Africa, and Rockville, Maryland, USA: Stats SA, SAMRC, and ICF.
- The DHS Program. (2015). *Demographic and Health Survey Sampling and Household Listing Manual*. Calverton, Maryland: ICF International.
- The DHS Program. (2018). *Guide to DHS Statistics, DHS-7*. Rockville, Maryland: ICF.
- United Nations Children's Fund (UNICEF). (2021). *Global databases on child health, nutrition, and WASH*. New York: UNICEF.
- World Health Organization (WHO). (2019). *World Health Statistics 2019: Monitoring health for the SDGs*. Geneva: WHO.