04_covid_prevention

Covid 19 Prevention

1. Load Libraries & data

```
# Data manipulation
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(tidyr)
library(readr)
library(here)

## here() starts at C:/Users/morul/School/3rd
Year/BIN381/BIN381_PROJECT/BIN381_PROJECT

library(purrr)

# Visualization and summaries
library(ggplot2)
library(skimr)
library(visdat)

# Load the COVID-19 prevention dataset
# Load the COVID-19 prevention dataset, skipping first row if it contains
metadata
covid_df <- read_csv(here("data", "raw", "covid-19-
prevention_national_zaf.csv"))

## Rows: 35 Columns: 29

## ── Column specification
────────────────────────────────────────────────────────

## Delimiter: ","
## chr (17): ISO3, DataId, Indicator, Value, Precision, DHS_CountryCode,
Countr...
## dbl  (8): IndicatorOrder, CharacteristicId, CharacteristicOrder, IsTotal,
Is...
```

```
## lgl  (4): RegionId, CILow, CIHigh, LevelRank
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.
```

```r
# Step 2: Remove first row (metadata)
covid_df <- covid_df[-1, ]

# Step 3: Reset row names
rownames(covid_df) <- NULL

# Step 4: Optional: in
cat("COVID-19 prevention dataset loaded successfully.\n")
```

```
## COVID-19 prevention dataset loaded successfully.
```

```r
cat("Dimensions:", dim(covid_df), "\n")
```

```
## Dimensions: 34 29
```

2. Initial Data Assessment and Column Renaming

- Column names are standardized to lowercase with underscores for readability. The dataset structure, summary statistics, and missingness are explored to identify potential quality issues.

```r
# Quick glimpse of dataset
glimpse(covid_df)
```

```
## Rows: 34
## Columns: 29
## $ ISO3                    <chr> "ZAF", "ZAF", "ZAF", "ZAF", "ZAF", "ZAF",
"ZAF"…
## $ DataId                  <chr> "795844", "795750", "795755", "795740",
"795744…
## $ Indicator               <chr> "Population using an improved water
source", "P…
## $ Value                   <chr> "83.5", "36", "23.1", "19.3", "60.3",
"80.2", "…
## $ Precision               <chr> "1", "1", "1", "1", "1", "1", "1", "1",
"1", "1…
## $ DHS_CountryCode         <chr> "ZA", "ZA", "ZA", "ZA", "ZA", "ZA", "ZA",
"ZA",…
## $ CountryName             <chr> "South Africa", "South Africa", "South
Africa",…
## $ SurveyYear              <chr> "1998", "1998", "1998", "1998", "1998",
"1998",…
## $ SurveyId                <chr> "ZA1998DHS", "ZA1998DHS", "ZA1998DHS",
"ZA1998D…
## $ IndicatorId             <chr> "WS_SRCE_P_IMP", "WS_SRCE_P_PIP",
```

```
"WS_SRCE_P_PY…
## $ IndicatorOrder         <dbl> 250162010, 250162020, 250162025, 250162030,
250…
## $ IndicatorType          <chr> "I", "I", "I", "I", "I", "I", "I", "I",
"I", "I…
## $ CharacteristicId       <dbl> 1000, 1000, 1000, 1000, 1000, 1000, 1000,
1000,…
## $ CharacteristicOrder    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0,…
## $ CharacteristicCategory <chr> "Total", "Total", "Total", "Total",
"Total", "T…
## $ CharacteristicLabel    <chr> "Total", "Total", "Total", "Total",
"Total", "T…
## $ ByVariableId           <chr> "0", "0", "0", "0", "0", "0", "0", "0",
"0", "0…
## $ ByVariableLabel        <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA,…
## $ IsTotal                <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1,…
## $ IsPreferred            <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1,…
## $ SDRID                  <chr> "WSSRCEPIMP", "WSSRCEPPIP", "WSSRCEPPYD",
"WSSR…
## $ RegionId               <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA,…
## $ SurveyYearLabel        <dbl> 1998, 1998, 1998, 1998, 1998, 1998, 1998,
1998,…
## $ SurveyType             <chr> "DHS", "DHS", "DHS", "DHS", "DHS", "DHS",
"DHS"…
## $ DenominatorWeighted    <dbl> 52007, 52007, 52007, 52007, 52007, 52007,
52007…
## $ DenominatorUnweighted  <dbl> 52465, 52465, 52465, 52465, 52465, 52465,
52465…
## $ CILow                  <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA,…
## $ CIHigh                 <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA,…
## $ LevelRank              <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA,…
```

```r
# Summary of missingness
skim(covid_df)
```

Data summary

Name	covid_df
Number of rows	34
Number of columns	29

Column type frequency:	
character	17
logical	4
numeric	8
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ISO3	0	1	3	3	0	1	0
DataId	0	1	5	6	0	34	0
Indicator	0	1	32	75	0	20	0
Value	0	1	2	4	0	34	0
Precision	0	1	1	1	0	2	0
DHS_CountryCode	0	1	2	2	0	1	0
CountryName	0	1	12	12	0	1	0
SurveyYear	0	1	4	4	0	2	0
SurveyId	0	1	9	9	0	2	0
IndicatorId	0	1	13	13	0	20	0
IndicatorType	0	1	1	1	0	1	0
CharacteristicCategory	0	1	5	5	0	1	0
CharacteristicLabel	0	1	5	5	0	1	0
ByVariableId	0	1	1	1	0	1	0
ByVariableLabel	34	0	NA	NA	0	0	0
SDRID	0	1	10	10	0	20	0
SurveyType	0	1	3	3	0	1	0

Variable type: logical

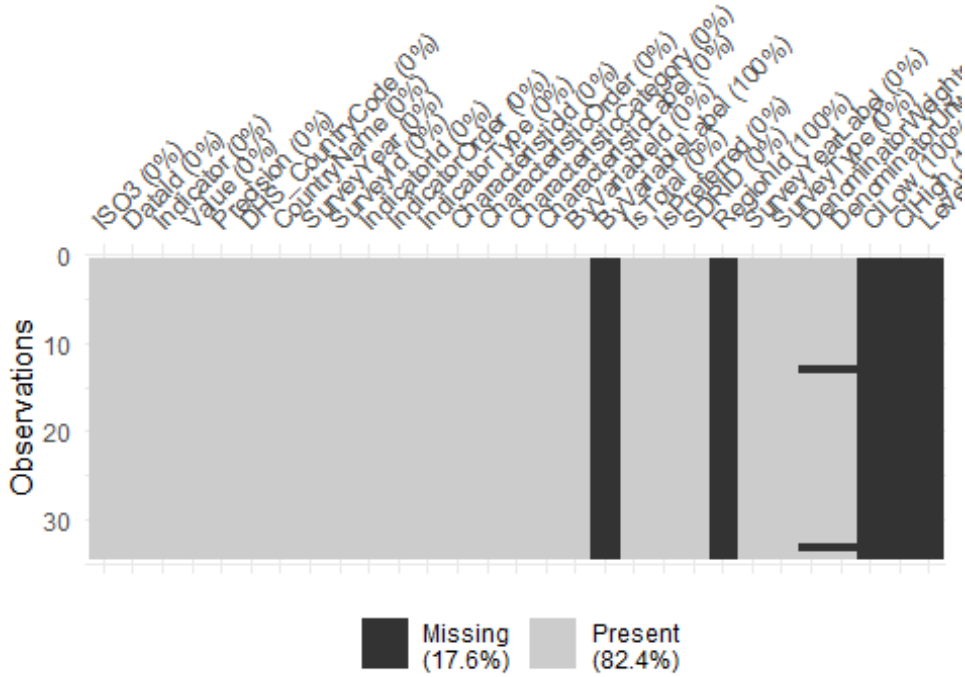
skim_variable	n_missing	complete_rate	mean	count
RegionId	34	0	NaN	:
CILow	34	0	NaN	:
CIHigh	34	0	NaN	:
LevelRank	34	0	NaN	:

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
IndicatorOrder	0	1.00	252040162.06	40111155.25	250162010	250162190	250252010	250292085	260831120	
CharacteristicId	0	1.00	1000.00	0.00	1000	1000	1000	1000	1000	
CharacteristicOrder	0	1.00	0.00	0.00	0	0	0	0	0	
IsTotal	0	1.00	1.00	0.00	1	1	1	1	1	
IsPreferred	0	1.00	1.00	0.00	1	1	1	1	1	
SurveyYearLabel	0	1.00	2008.59	8.99	1998	1998	2016	2016	2016	
DenominatorWeighted	2	0.94	38815.16	12658.90	11066	37205	37205	52007	52007	
DenominatorUnweighted	2	0.94	39352.88	12765.80	11066	37925	37925	52465	52465	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
										▅▇▇

```r
# visualize missing values
vis_miss(covid_df)
```




```r
# Clean column names to lowercase with underscores
covid_df <- covid_df %>% janitor::clean_names()

# Check new names
colnames(covid_df)

##  [1] "iso3"                    "data_id"
##  [3] "indicator"               "value"
##  [5] "precision"               "dhs_country_code"
##  [7] "country_name"            "survey_year"
##  [9] "survey_id"               "indicator_id"
## [11] "indicator_order"         "indicator_type"
## [13] "characteristic_id"       "characteristic_order"
## [15] "characteristic_category" "characteristic_label"
## [17] "by_variable_id"          "by_variable_label"
## [19] "is_total"                "is_preferred"
## [21] "sdrid"                   "region_id"
```

```
## [23] "survey_year_label"       "survey_type"
## [25] "denominator_weighted"    "denominator_unweighted"
## [27] "ci_low"                  "ci_high"
## [29] "level_rank"
```

Handle Duplicates

```
# Check for exact duplicates
exact_dups <- sum(duplicated(covid_df))
cat("Exact duplicate rows:", exact_dups, "\n")

## Exact duplicate rows: 0

# Remove all duplicates, keeping first occurrence
covid_df <- covid_df %>%
  distinct(indicator, survey_year, characteristic_id, value, .keep_all =
TRUE)

cat("Dimensions after duplicate removal:", dim(covid_df), "\n")

## Dimensions after duplicate removal: 34 29
```

Convert Data Types

- Ensures all numeric, integer, and logical columns have correct types for downstream analysis. Prevents calculation errors and improves consistency.

```
# Convert numeric columns
covid_df <- covid_df %>%
  mutate(
    across(c(value, precision, denominator_weighted, denominator_unweighted,
ci_low, ci_high), as.numeric),
    across(c(survey_year, indicator_order, characteristic_id,
characteristic_order, survey_year_label, by_variable_id), as.integer),
    across(c(is_total, is_preferred), ~as.logical(as.integer(.)))
  )
```

Drop Redundant Columns

```
redundant_cols <- c("iso3", "data_id", "dhs_country_code", "country_name",
                    "survey_id", "indicator_id", "sdrid", "region_id",
                    "survey_type", "level_rank", "denominator_weighted",
"denominator_unweighted")

covid_df <- covid_df %>% select(-any_of(redundant_cols))

# Remove columns that are entirely NA
covid_df <- covid_df %>% select(where(~!all(is.na(.))))

cat("Redundant and empty columns removed.\n")

## Redundant and empty columns removed.
```

```r
cat("New dimensions:", dim(covid_df), "\n")
```

```
## New dimensions: 34 14
```

Handle Missing Values

- Numeric columns: filled with the median
- Character columns: filled with the most frequent value
- Logical columns: missing values set to FALSE
- Key metadata (survey_year_label, survey_type) imputed explicitly for clarity

```r
covid_df <- covid_df %>%
  select(where(~!all(is.na(.))))


impute_mode <- function(x) {
  ux <- na.omit(x)
  if(length(ux) == 0) return(x)
  rep(names(sort(table(ux), decreasing = TRUE))[1], length(x))
}


covid_df <- covid_df %>%
  mutate(
    # Numeric columns → median
    across(where(is.numeric), ~ifelse(is.na(.), median(., na.rm = TRUE), .)),

    # Character columns → mode
    across(where(is.character), ~ifelse(is.na(.), impute_mode(.), .)),

    # Logical columns → set missing to FALSE (or TRUE if appropriate)
    across(where(is.logical), ~ifelse(is.na(.), FALSE, .))
  )


missing_summary <- data.frame(
  Column = colnames(covid_df),
  n_missing = colSums(is.na(covid_df)),
  total_rows = nrow(covid_df),
  missing_percent = round(colSums(is.na(covid_df)) / nrow(covid_df) * 100, 2)
)

missing_summary %>% arrange(desc(missing_percent))
```

```
##                                          Column n_missing total_rows
## indicator                             indicator         0         34
## value                                     value         0         34
```

```
## precision                              precision         0          34
## survey_year                          survey_year         0          34
## indicator_order                  indicator_order         0          34
## indicator_type                    indicator_type         0          34
## characteristic_id              characteristic_id         0          34
## characteristic_order        characteristic_order         0          34
## characteristic_category characteristic_category         0          34
## characteristic_label        characteristic_label         0          34
## by_variable_id                    by_variable_id         0          34
## is_total                                 is_total         0          34
## is_preferred                         is_preferred         0          34
## survey_year_label              survey_year_label         0          34
##                          missing_percent
## indicator                              0
## value                                  0
## precision                              0
## survey_year                            0
## indicator_order                        0
## indicator_type                         0
## characteristic_id                      0
## characteristic_order                   0
## characteristic_category                0
## characteristic_label                   0
## by_variable_id                         0
## is_total                               0
## is_preferred                           0
## survey_year_label                      0
```

Handle Outliers

- Extreme values in value are capped to the IQR boundaries (Winsorization), which reduces their influence while keeping most data intact.

```
# Quick check for extreme values in 'value' and denominators
summary(covid_df$value)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.50    6.45   31.20   36.74   56.85   96.00

summary(covid_df$denominator_weighted)

## Warning: Unknown or uninitialised column: `denominator_weighted`.

## Length  Class   Mode
##      0   NULL   NULL

summary(covid_df$denominator_unweighted)

## Warning: Unknown or uninitialised column: `denominator_unweighted`.

## Length  Class   Mode
##      0   NULL   NULL
```

```r
# Identify potential outliers using IQR method
outliers <- covid_df %>%
  filter(value > quantile(value, 0.75, na.rm = TRUE) + 1.5 * IQR(value, na.rm
= TRUE) |
         value < quantile(value, 0.25, na.rm = TRUE) - 1.5 * IQR(value, na.rm
= TRUE))

cat("Potential outlier rows in 'value':", nrow(outliers), "\n")

## Potential outlier rows in 'value': 0
```

Final Validation

```r
## Final Dataset Check Before Saving (existing columns only)

# Check dataset dimensions and structure
cat("Final dataset dimensions:", dim(covid_df), "\n")

## Final dataset dimensions: 34 14

str(covid_df)

## tibble [34 × 14] (S3: tbl_df/tbl/data.frame)
##  $ indicator              : chr [1:34] "Population using an improved water
source" "Population using water piped into dwelling" "Population using water
piped into yard/plot" "Population using a public tap/standpipe" ...
##  $ value                  : num [1:34] 83.5 36 23.1 19.3 60.3 80.2 3.3 8.4
46.4 40.8 ...
##  $ precision              : num [1:34] 1 1 1 1 1 1 1 1 1 1 ...
##  $ survey_year            : int [1:34] 1998 1998 1998 1998 1998 1998 1998
1998 1998 1998 ...
##  $ indicator_order        : int [1:34] 250162010 250162020 250162025
250162030 250162190 250162200 250162210 250202030 250262010 250262150 ...
##  $ indicator_type         : chr [1:34] "I" "I" "I" "I" ...
##  $ characteristic_id      : int [1:34] 1000 1000 1000 1000 1000 1000 1000
1000 1000 1000 ...
##  $ characteristic_order   : int [1:34] 0 0 0 0 0 0 0 0 0 0 ...
##  $ characteristic_category: chr [1:34] "Total" "Total" "Total" "Total" ...
##  $ characteristic_label   : chr [1:34] "Total" "Total" "Total" "Total" ...
##  $ by_variable_id         : int [1:34] 0 0 0 0 0 0 0 0 0 0 ...
##  $ is_total               : logi [1:34] TRUE TRUE TRUE TRUE TRUE TRUE ...
##  $ is_preferred           : logi [1:34] TRUE TRUE TRUE TRUE TRUE TRUE ...
##  $ survey_year_label      : int [1:34] 1998 1998 1998 1998 1998 1998 1998
1998 1998 1998 ...

# Identify numeric columns that exist
numeric_cols <- covid_df %>% select(where(is.numeric)) %>% colnames()

# Summarize numeric columns for final inspection
summary(select(covid_df, all_of(numeric_cols)))
```

```
##      value          precision       survey_year   indicator_order
##  Min.   : 1.50   Min.   :0.0000   Min.   :1998   Min.   :250162010
##  1st Qu.: 6.45   1st Qu.:1.0000   1st Qu.:1998   1st Qu.:250162190
##  Median :31.20   Median :1.0000   Median :2016   Median :250252010
##  Mean   :36.74   Mean   :0.9706   Mean   :2009   Mean   :252040162
##  3rd Qu.:56.85   3rd Qu.:1.0000   3rd Qu.:2016   3rd Qu.:250292085
##  Max.   :96.00   Max.   :1.0000   Max.   :2016   Max.   :260831120
##  characteristic_id characteristic_order by_variable_id survey_year_label
##  Min.   :1000      Min.   :0            Min.   :0      Min.   :1998
##  1st Qu.:1000      1st Qu.:0            1st Qu.:0      1st Qu.:1998
##  Median :1000      Median :0            Median :0      Median :2016
##  Mean   :1000      Mean   :0            Mean   :0      Mean   :2009
##  3rd Qu.:1000      3rd Qu.:0            3rd Qu.:0      3rd Qu.:2016
##  Max.   :1000      Max.   :0            Max.   :0      Max.   :2016
```

```r
# Confirm no remaining missing values in all columns
missing_summary <- covid_df %>%
  summarise(across(everything(), ~sum(is.na(.)))) %>%
  pivot_longer(cols = everything(), names_to = "column", values_to =
"n_missing") %>%
  mutate(
    total_rows = nrow(covid_df),
    missing_percent = round(n_missing / total_rows * 100, 2)
  )

missing_summary %>% arrange(desc(missing_percent))
```

```
## # A tibble: 14 × 4
##    column                  n_missing total_rows missing_percent
##    <chr>                       <int>      <int>           <dbl>
##  1 indicator                       0         34               0
##  2 value                           0         34               0
##  3 precision                       0         34               0
##  4 survey_year                     0         34               0
##  5 indicator_order                 0         34               0
##  6 indicator_type                  0         34               0
##  7 characteristic_id               0         34               0
##  8 characteristic_order            0         34               0
##  9 characteristic_category         0         34               0
## 10 characteristic_label            0         34               0
## 11 by_variable_id                  0         34               0
## 12 is_total                        0         34               0
## 13 is_preferred                    0         34               0
## 14 survey_year_label               0         34               0
```

Save Dataset

```r
# Define path to save cleaned CSV
clean_path <- here("data", "processed", "covid_prevention_cleaned_zaf.csv")
```

```r
# Create folder if it doesn't exist
if(!dir.exists(dirname(clean_path))) dir.create(dirname(clean_path),
recursive = TRUE)

# Write cleaned dataset
write_csv(covid_df, clean_path)

cat("Cleaned COVID-19 prevention dataset saved successfully at:\n",
clean_path, "\n")
```

```
## Cleaned COVID-19 prevention dataset saved successfully at:
##  C:/Users/morul/School/3rd
Year/BIN381/BIN381_PROJECT/BIN381_PROJECT/data/processed/covid_prevention_cle
aned_zaf.csv
```