07\_immunization

#Loading Libraries

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(stringr)  
library(readr)  
library(here)

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

library(ggplot2)

# Load Dataset

imm\_df <- read\_csv(here("data", "raw", "immunization\_national\_zaf.csv"))

## Rows: 117 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.

# Display Dataset content

# Make sure the file exists  
  
  
head(imm\_df)

## # A tibble: 6 × 29  
## ISO3 DataId Indicator Value Precision DHS\_CountryCode CountryName SurveyYear  
## <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>   
## 1 #coun… #meta… #indicat… #ind… #indicat… <NA> #country+n… #date+year  
## 2 ZAF 330965 BCG vacc… 96.8 1 ZA South Afri… 1998   
## 3 ZAF 139796 BCG vacc… 94.9 1 ZA South Afri… 1998   
## 4 ZAF 330966 DPT 1 va… 93.3 1 ZA South Afri… 1998   
## 5 ZAF 139797 DPT 1 va… 93.1 1 ZA South Afri… 1998   
## 6 ZAF 330967 DPT 2 va… 86.2 1 ZA South Afri… 1998   
## # ℹ 21 more variables: SurveyId <chr>, IndicatorId <chr>, IndicatorOrder <dbl>,  
## # IndicatorType <chr>, CharacteristicId <dbl>, CharacteristicOrder <dbl>,  
## # CharacteristicCategory <chr>, CharacteristicLabel <chr>,  
## # ByVariableId <chr>, ByVariableLabel <chr>, IsTotal <dbl>,  
## # IsPreferred <dbl>, SDRID <chr>, RegionId <lgl>, SurveyYearLabel <dbl>,  
## # SurveyType <chr>, DenominatorWeighted <dbl>, DenominatorUnweighted <dbl>,  
## # CILow <lgl>, CIHigh <lgl>, LevelRank <lgl>

# Remove the first row(meta data)

imm\_df <- imm\_df[-1, ]

# dimensions

dim(imm\_df)

## [1] 116 29

# Inspect Duplicated rows

dup\_check <- imm\_df %>%  
 group\_by(Indicator, SurveyYear, CharacteristicId, Value) %>%  
 filter(n() > 1)  
  
dup\_check

## # A tibble: 0 × 29  
## # Groups: Indicator, SurveyYear, CharacteristicId, Value [0]  
## # ℹ 29 variables: ISO3 <chr>, DataId <chr>, Indicator <chr>, Value <chr>,  
## # Precision <chr>, DHS\_CountryCode <chr>, CountryName <chr>,  
## # SurveyYear <chr>, SurveyId <chr>, IndicatorId <chr>, IndicatorOrder <dbl>,  
## # IndicatorType <chr>, CharacteristicId <dbl>, CharacteristicOrder <dbl>,  
## # CharacteristicCategory <chr>, CharacteristicLabel <chr>,  
## # ByVariableId <chr>, ByVariableLabel <chr>, IsTotal <dbl>,  
## # IsPreferred <dbl>, SDRID <chr>, RegionId <lgl>, SurveyYearLabel <dbl>, …

# Missing Values

# 1. Remove completely empty columns  
imm\_df <- imm\_df %>% select(where(~!all(is.na(.))))  
  
# 2. Impute numeric columns with median  
num\_cols <- imm\_df %>% select(where(is.numeric)) %>% names()  
imm\_df <- imm\_df %>%  
 mutate(across(all\_of(num\_cols), ~ifelse(is.na(.), median(., na.rm = TRUE), .)))  
  
# 3. Impute categorical/character columns with mode  
cat\_cols <- imm\_df %>% select(where(is.character)) %>% names()  
get\_mode <- function(x) {  
 ux <- na.omit(x)  
 if(length(ux) == 0) return(NA\_character\_)  
 names(sort(table(ux), decreasing = TRUE))[1]  
}  
imm\_df <- imm\_df %>%  
 mutate(across(all\_of(cat\_cols), ~ifelse(is.na(.), get\_mode(.), .)))  
  
# 4. Summary after handling missing values  
missing\_summary <- data.frame(  
 Column = names(imm\_df),  
 Missing\_Count = colSums(is.na(imm\_df)),  
 Missing\_Percent = round(colMeans(is.na(imm\_df)) \* 100, 2)  
)  
  
cat("Total remaining NAs:", sum(is.na(imm\_df)), "\n")

## Total remaining NAs: 0

cat("Missing value summary per column:\n")

## Missing value summary per column:

print(missing\_summary)

## Column Missing\_Count Missing\_Percent  
## ISO3 ISO3 0 0  
## DataId DataId 0 0  
## Indicator Indicator 0 0  
## Value Value 0 0  
## Precision Precision 0 0  
## DHS\_CountryCode DHS\_CountryCode 0 0  
## CountryName CountryName 0 0  
## SurveyYear SurveyYear 0 0  
## SurveyId SurveyId 0 0  
## IndicatorId IndicatorId 0 0  
## IndicatorOrder IndicatorOrder 0 0  
## IndicatorType IndicatorType 0 0  
## CharacteristicId CharacteristicId 0 0  
## CharacteristicOrder CharacteristicOrder 0 0  
## CharacteristicCategory CharacteristicCategory 0 0  
## CharacteristicLabel CharacteristicLabel 0 0  
## ByVariableId ByVariableId 0 0  
## ByVariableLabel ByVariableLabel 0 0  
## IsTotal IsTotal 0 0  
## IsPreferred IsPreferred 0 0  
## SDRID SDRID 0 0  
## SurveyYearLabel SurveyYearLabel 0 0  
## SurveyType SurveyType 0 0  
## DenominatorWeighted DenominatorWeighted 0 0  
## DenominatorUnweighted DenominatorUnweighted 0 0

Handling Missing Values

Strategies applied:

1. Remove empty columns – columns entirely missing were removed.
2. Numeric columns – missing values imputed with the median of available values.
3. Categorical/character columns – missing values imputed with the mode (most frequent value).
4. Denominators – missing values in DenominatorWeighted and DenominatorUnweighted were forward-filled using fill().

Outcome: All columns have complete values, making calculations and analyses reliable. - Missing denominators are filled using the previous non-missing value (fill()).

* Ensures numeric calculations work correctly.

#Convert Data Types

# Define columns to convert, only if they exist  
cols\_to\_numeric <- c("Value", "Precision", "DenominatorWeighted", "DenominatorUnweighted")  
cols\_to\_integer <- c("SurveyYear", "IndicatorOrder", "CharacteristicId")  
cols\_to\_logical <- c("IsPreferred")  
  
cols\_to\_numeric <- cols\_to\_numeric[cols\_to\_numeric %in% names(imm\_df)]  
cols\_to\_integer <- cols\_to\_integer[cols\_to\_integer %in% names(imm\_df)]  
cols\_to\_logical <- cols\_to\_logical[cols\_to\_logical %in% names(imm\_df)]  
  
# Convert  
imm\_df <- imm\_df %>%  
 mutate(  
 across(all\_of(cols\_to\_numeric), as.numeric),  
 across(all\_of(cols\_to\_integer), as.integer),  
 across(all\_of(cols\_to\_logical), ~as.logical(as.integer(.)))  
 )  
  
# Check structure  
str(imm\_df)

## tibble [116 × 25] (S3: tbl\_df/tbl/data.frame)  
## $ ISO3 : chr [1:116] "ZAF" "ZAF" "ZAF" "ZAF" ...  
## $ DataId : chr [1:116] "330965" "139796" "330966" "139797" ...  
## $ Indicator : chr [1:116] "BCG vaccination received" "BCG vaccination received" "DPT 1 vaccination received" "DPT 1 vaccination received" ...  
## $ Value : num [1:116] 96.8 94.9 93.3 93.1 86.2 82.4 76.4 73.8 91.2 87.7 ...  
## $ Precision : num [1:116] 1 1 1 1 1 1 1 1 1 1 ...  
## $ DHS\_CountryCode : chr [1:116] "ZA" "ZA" "ZA" "ZA" ...  
## $ CountryName : chr [1:116] "South Africa" "South Africa" "South Africa" "South Africa" ...  
## $ SurveyYear : int [1:116] 1998 1998 1998 1998 1998 1998 1998 1998 1998 1998 ...  
## $ SurveyId : chr [1:116] "ZA1998DHS" "ZA1998DHS" "ZA1998DHS" "ZA1998DHS" ...  
## $ IndicatorId : chr [1:116] "CH\_VACS\_C\_BCG" "CH\_VACS\_C\_BCG" "CH\_VACS\_C\_DP1" "CH\_VACS\_C\_DP1" ...  
## $ IndicatorOrder : int [1:116] 93886010 93886010 93886020 93886020 93886030 93886030 93886040 93886040 93886050 93886050 ...  
## $ IndicatorType : chr [1:116] "I" "I" "I" "I" ...  
## $ CharacteristicId : int [1:116] 268002 268002 268002 268002 268002 268002 268002 268002 268002 268002 ...  
## $ CharacteristicOrder : num [1:116] 268002 268002 268002 268002 268002 ...  
## $ CharacteristicCategory: chr [1:116] "Source of vaccination information" "Source of vaccination information" "Source of vaccination information" "Source of vaccination information" ...  
## $ CharacteristicLabel : chr [1:116] "Either source" "Either source" "Either source" "Either source" ...  
## $ ByVariableId : chr [1:116] "258001" "258002" "258001" "258002" ...  
## $ ByVariableLabel : chr [1:116] "12-23" "24-35" "12-23" "24-35" ...  
## $ IsTotal : num [1:116] 1 1 1 1 1 1 1 1 1 1 ...  
## $ IsPreferred : logi [1:116] TRUE FALSE TRUE FALSE TRUE FALSE ...  
## $ SDRID : chr [1:116] "CHVACSCBCG" "CHVACSCBCG" "CHVACSCDP1" "CHVACSCDP1" ...  
## $ SurveyYearLabel : num [1:116] 1998 1998 1998 1998 1998 ...  
## $ SurveyType : chr [1:116] "DHS" "DHS" "DHS" "DHS" ...  
## $ DenominatorWeighted : num [1:116] 973 933 973 933 973 933 973 933 973 933 ...  
## $ DenominatorUnweighted : num [1:116] 971 951 971 951 971 951 971 951 971 951 ...

* Numeric: Value, Precision, DenominatorWeighted, DenominatorUnweighted
* Integer: SurveyYear, IndicatorOrder, CharacteristicId
* Logical: IsPreferred

#Drop the countries only onw unqiue value: reason, there is no useful information - county is also always za

# Columns to remove and rationale  
cols\_to\_remove <- c(  
 "ISO3", # Only one value "ZAF" → provides no useful variation  
 "DHS\_CountryCode", # Only one value "ZA" → redundant  
 "CountryName", # Always "South Africa" → redundant  
 "SurveyId", # Encodes survey metadata, not needed for analysis  
 "ByVariableId", # IDs for subgroup variables; not used in analysis  
 "ByVariableLabel", # Labels for subgroup variables; not needed  
 "IsTotal", # Logical flag that is either 0/1 for all → redundant  
 "RegionId", # Missing or NA → no information  
 "SurveyYearLabel", # Duplicate of SurveyYear → redundant  
 "SurveyType", # Always "DHS" → no variation  
 "CharacteristicOrder" # IDs for order only; not analytically useful  
)  
  
# Remove only columns that exist to avoid errors  
imm\_df <- imm\_df %>% select(-any\_of(cols\_to\_remove))  
  
# Documenting action  
cat("Removed columns that were either redundant, constant, or not analytically useful:\n")

## Removed columns that were either redundant, constant, or not analytically useful:

cat(paste(cols\_to\_remove, collapse = ", "), "\n")

## ISO3, DHS\_CountryCode, CountryName, SurveyId, ByVariableId, ByVariableLabel, IsTotal, RegionId, SurveyYearLabel, SurveyType, CharacteristicOrder

Columns removed because they were constant, redundant, or not analytically useful:

ISO3, DHS\_CountryCode, CountryName, SurveyId, ByVariableId, ByVariableLabel, IsTotal, RegionId, SurveyYearLabel, SurveyType, CharacteristicOrder

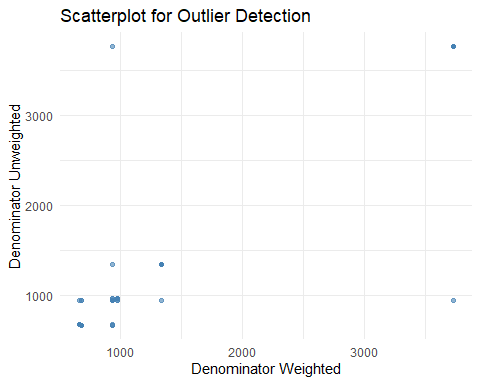
These columns either contained a single value or metadata that does not impact analysis.

#the missing values can be filled with the previous non missing value in the opposite attribute

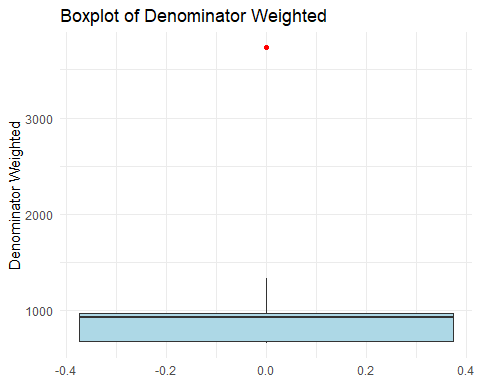
imm\_df <- imm\_df %>%  
 fill(DenominatorWeighted, DenominatorUnweighted, .direction = "down")  
  
imm\_df[  
 c("DenominatorWeighted", "DenominatorUnweighted")]

## # A tibble: 116 × 2  
## DenominatorWeighted DenominatorUnweighted  
## <dbl> <dbl>  
## 1 973 971  
## 2 933 951  
## 3 973 971  
## 4 933 951  
## 5 973 971  
## 6 933 951  
## 7 973 971  
## 8 933 951  
## 9 973 971  
## 10 933 951  
## # ℹ 106 more rows

ggplot(imm\_df, aes(x = DenominatorWeighted, y = DenominatorUnweighted)) +  
 geom\_point(alpha = 0.6, color = "steelblue") +  
 labs(title = "Scatterplot for Outlier Detection",  
 x = "Denominator Weighted",  
 y = "Denominator Unweighted") +  
 theme\_minimal()



ggplot(imm\_df, aes(y = DenominatorWeighted)) +  
 geom\_boxplot(fill = "lightblue", outlier.color = "red", outlier.shape = 16) +  
 labs(title = "Boxplot of Denominator Weighted",  
 y = "Denominator Weighted") +  
 theme\_minimal()

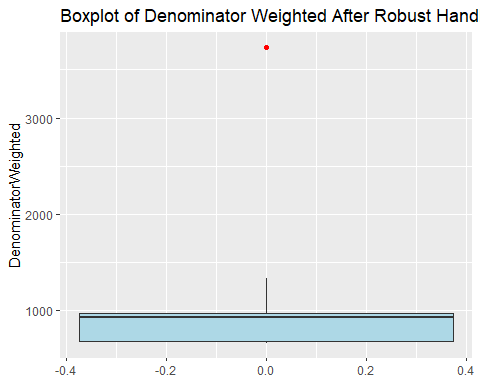


dim(imm\_df)

## [1] 116 15

#Outlier Handling

# Winsorize at 1st and 99th percentiles  
lower\_w <- quantile(imm\_df$DenominatorWeighted, 0.01, na.rm = TRUE)  
upper\_w <- quantile(imm\_df$DenominatorWeighted, 0.99, na.rm = TRUE)  
  
lower\_uw <- quantile(imm\_df$DenominatorUnweighted, 0.01, na.rm = TRUE)  
upper\_uw <- quantile(imm\_df$DenominatorUnweighted, 0.99, na.rm = TRUE)  
  
imm\_df <- imm\_df %>%  
 mutate(  
 DenominatorWeighted = pmax(pmin(DenominatorWeighted, upper\_w), lower\_w),  
 DenominatorUnweighted = pmax(pmin(DenominatorUnweighted, upper\_uw), lower\_uw)  
 )  
  
ggplot(imm\_df, aes(y = DenominatorWeighted)) +  
 geom\_boxplot(fill = "lightblue", outlier.color = "red") +  
 labs(title = "Boxplot of Denominator Weighted After Robust Handling")



summary(imm\_df$DenominatorWeighted)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 660 677 933 1185 973 3734

summary(imm\_df$DenominatorUnweighted)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 670 670 951 1192 971 3761

## Robust Outlier Handling via Winsorization

**Purpose:**  
The dataset contains extreme values in DenominatorWeighted and DenominatorUnweighted that skew the distribution. Instead of removing rows, we Winsorize the data to limit extreme values while keeping all observations intact.

**Steps:**

1. **Calculate bounds:**
   * 1st percentile (0.01) → lower bound
   * 99th percentile (0.99) → upper bound  
     This ensures the extreme 1% of values on either side are capped.
2. **Apply Winsorization:**
   * Values below the lower bound are set to the lower bound.
   * Values above the upper bound are set to the upper bound.
3. **Visual Check:**
   * A boxplot is created to verify the effect of Winsorization on DenominatorWeighted.
4. **Summary Statistics:**
   * summary() is used to compare min, max, mean, and quartiles after capping, confirming that extreme outliers have been mitigated.

**Code Explanation:**

* quantile(..., 0.01, na.rm = TRUE) → computes the 1st percentile ignoring missing values.
* pmin() and pmax() → ensure values stay within the specified bounds.
* geom\_boxplot(outlier.color = "red") → visualizes remaining extreme points (if any).

**Outcome:**  
The extreme skew in denominator columns is reduced, improving stability for subsequent analyses, while retaining all rows in the dataset.

## Final check and Save Dataset

#save cleaned data

write\_csv(imm\_df, here("data","processed", "immunization\_cleaned.csv"))