09\_Literacy

# 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(purrr)  
library(ggplot2)

# Load Dataset

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

## Rows: 21 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.

# Disdplay Dataset content

head(lit\_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 563770 Women wi… 11.8 1 ZA South Afri… 2016   
## 3 ZAF 563771 Women wh… 76.2 1 ZA South Afri… 2016   
## 4 ZAF 563772 Women wh… 8.2 1 ZA South Afri… 2016   
## 5 ZAF 563773 Women wh… 3.5 1 ZA South Afri… 2016   
## 6 ZAF 563769 Women fo… 0.1 1 ZA South Afri… 2016   
## # 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)

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

# dimensions

dim(lit\_df)

## [1] 20 29

# Inspect Duplicated rows

dup\_check <- lit\_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>, …

# perc na values

data.frame(  
 Column = names(lit\_df),  
 Missing\_Percentage = paste0(round(colMeans(is.na(lit\_df)) \* 100, 2), "%")  
 )

## Column Missing\_Percentage  
## 1 ISO3 0%  
## 2 DataId 0%  
## 3 Indicator 0%  
## 4 Value 0%  
## 5 Precision 0%  
## 6 DHS\_CountryCode 0%  
## 7 CountryName 0%  
## 8 SurveyYear 0%  
## 9 SurveyId 0%  
## 10 IndicatorId 0%  
## 11 IndicatorOrder 0%  
## 12 IndicatorType 0%  
## 13 CharacteristicId 0%  
## 14 CharacteristicOrder 0%  
## 15 CharacteristicCategory 0%  
## 16 CharacteristicLabel 0%  
## 17 ByVariableId 0%  
## 18 ByVariableLabel 100%  
## 19 IsTotal 0%  
## 20 IsPreferred 0%  
## 21 SDRID 0%  
## 22 RegionId 100%  
## 23 SurveyYearLabel 0%  
## 24 SurveyType 0%  
## 25 DenominatorWeighted 10%  
## 26 DenominatorUnweighted 10%  
## 27 CILow 100%  
## 28 CIHigh 100%  
## 29 LevelRank 100%

data.frame(  
 Column = names(lit\_df),  
 Missing\_Data = paste0(colSums(is.na(lit\_df)))  
 )

## Column Missing\_Data  
## 1 ISO3 0  
## 2 DataId 0  
## 3 Indicator 0  
## 4 Value 0  
## 5 Precision 0  
## 6 DHS\_CountryCode 0  
## 7 CountryName 0  
## 8 SurveyYear 0  
## 9 SurveyId 0  
## 10 IndicatorId 0  
## 11 IndicatorOrder 0  
## 12 IndicatorType 0  
## 13 CharacteristicId 0  
## 14 CharacteristicOrder 0  
## 15 CharacteristicCategory 0  
## 16 CharacteristicLabel 0  
## 17 ByVariableId 0  
## 18 ByVariableLabel 20  
## 19 IsTotal 0  
## 20 IsPreferred 0  
## 21 SDRID 0  
## 22 RegionId 20  
## 23 SurveyYearLabel 0  
## 24 SurveyType 0  
## 25 DenominatorWeighted 2  
## 26 DenominatorUnweighted 2  
## 27 CILow 20  
## 28 CIHigh 20  
## 29 LevelRank 20

# check data types

data.frame(  
 Column = names(lit\_df),  
 paste0(sapply(lit\_df, typeof))  
)

## Column paste0.sapply.lit\_df..typeof..  
## 1 ISO3 character  
## 2 DataId character  
## 3 Indicator character  
## 4 Value character  
## 5 Precision character  
## 6 DHS\_CountryCode character  
## 7 CountryName character  
## 8 SurveyYear character  
## 9 SurveyId character  
## 10 IndicatorId character  
## 11 IndicatorOrder double  
## 12 IndicatorType character  
## 13 CharacteristicId double  
## 14 CharacteristicOrder double  
## 15 CharacteristicCategory character  
## 16 CharacteristicLabel character  
## 17 ByVariableId character  
## 18 ByVariableLabel character  
## 19 IsTotal double  
## 20 IsPreferred double  
## 21 SDRID character  
## 22 RegionId logical  
## 23 SurveyYearLabel double  
## 24 SurveyType character  
## 25 DenominatorWeighted double  
## 26 DenominatorUnweighted double  
## 27 CILow logical  
## 28 CIHigh logical  
## 29 LevelRank logical

#Convert Data Types

lit\_df <- lit\_df %>%  
 mutate(  
 Value = as.numeric(Value),  
 Precision = as.numeric(Precision),  
 SurveyYear = as.integer(SurveyYear),  
 IndicatorOrder = as.integer(IndicatorOrder),  
 CharacteristicId = as.integer(CharacteristicId),  
 CharacteristicOrder = as.integer(CharacteristicOrder),  
 IsTotal = as.logical(as.integer(IsTotal)),  
 IsPreferred = as.logical(as.integer(IsPreferred)),  
 SurveyYearLabel = as.integer(SurveyYearLabel),  
 DenominatorWeighted = as.numeric(DenominatorWeighted),  
 DenominatorUnweighted = as.numeric(DenominatorUnweighted),  
 )

# Summary table: column name, number of unique values, sample of unique values

library(purrr)  
n\_sample <- 3  
  
summary\_tbl <- lit\_df %>%  
 map\_df(~ tibble(  
 n\_unique = n\_distinct(.),  
 sample\_values = paste(head(unique(.), n\_sample), collapse = ", ")  
 ), .id = "column")  
  
  
summary\_tbl

## # A tibble: 29 × 3  
## column n\_unique sample\_values   
## <chr> <int> <chr>   
## 1 ISO3 1 ZAF   
## 2 DataId 20 563770, 563771, 563772   
## 3 Indicator 20 Women with secondary or higher education, Women who…  
## 4 Value 17 11.8, 76.2, 8.2   
## 5 Precision 2 1, 0   
## 6 DHS\_CountryCode 1 ZA   
## 7 CountryName 1 South Africa   
## 8 SurveyYear 1 2016   
## 9 SurveyId 1 ZA2016DHS   
## 10 IndicatorId 20 ED\_LITR\_W\_SCH, ED\_LITR\_W\_RDW, ED\_LITR\_W\_RDP   
## # ℹ 19 more rows

#Drop redundant columns

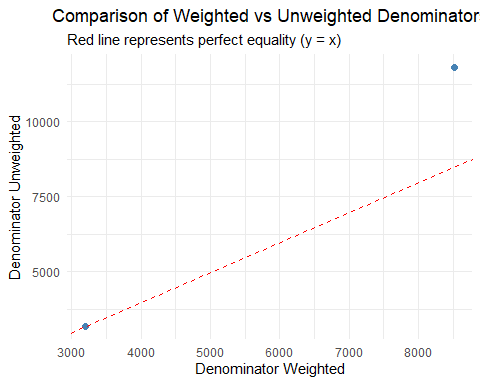
lit\_df <- lit\_df %>%  
   
 select(  
 -ISO3,   
 -DHS\_CountryCode,   
 -CountryName,   
 -SurveyId,  
 -ByVariableId,   
 -ByVariableLabel,   
 -IsTotal,  
 -RegionId,   
 -SurveyYearLabel,   
 -SurveyType,  
 -CharacteristicOrder  
 )

#Missing Value Handling

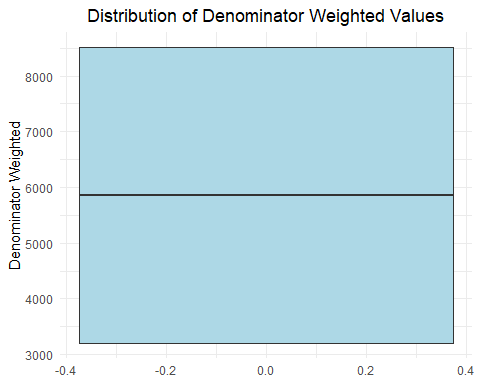
lit\_df <- lit\_df %>%  
 fill(DenominatorWeighted, DenominatorUnweighted, .direction = "downup")  
  
lit\_df[  
 c("DenominatorWeighted", "DenominatorUnweighted")]

## # A tibble: 20 × 2  
## DenominatorWeighted DenominatorUnweighted  
## <dbl> <dbl>  
## 1 8514 11805  
## 2 8514 11805  
## 3 8514 11805  
## 4 8514 11805  
## 5 8514 11805  
## 6 8514 11805  
## 7 8514 11805  
## 8 8514 11805  
## 9 8514 11805  
## 10 8514 11805  
## 11 3202 3179  
## 12 3202 3179  
## 13 3202 3179  
## 14 3202 3179  
## 15 3202 3179  
## 16 3202 3179  
## 17 3202 3179  
## 18 3202 3179  
## 19 3202 3179  
## 20 3202 3179

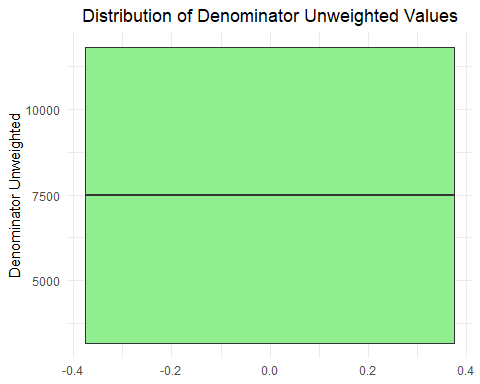
# 1. Scatterplot comparing weighted vs unweighted denominators  
if(all(c("DenominatorWeighted", "DenominatorUnweighted") %in% names(lit\_df))) {  
 scatter\_plot <- ggplot(lit\_df, aes(x = DenominatorWeighted, y = DenominatorUnweighted)) +  
 geom\_point(alpha = 0.6, color = "steelblue", size = 2) +  
 geom\_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +  
 labs(title = "Comparison of Weighted vs Unweighted Denominators",  
 x = "Denominator Weighted",  
 y = "Denominator Unweighted",  
 subtitle = "Red line represents perfect equality (y = x)") +  
 theme\_minimal() +  
 theme(plot.title = element\_text(hjust = 0.5))  
   
 print(scatter\_plot)  
}



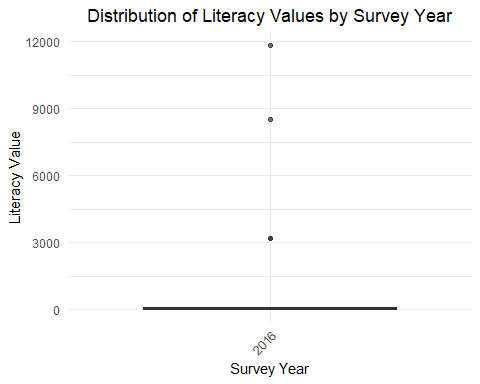
# 2. Boxplot for DenominatorWeighted  
if("DenominatorWeighted" %in% names(lit\_df)) {  
 boxplot\_weighted <- ggplot(lit\_df, aes(y = DenominatorWeighted)) +  
 geom\_boxplot(fill = "lightblue", outlier.color = "red", outlier.shape = 16) +  
 labs(title = "Distribution of Denominator Weighted Values",  
 y = "Denominator Weighted") +  
 theme\_minimal() +  
 theme(plot.title = element\_text(hjust = 0.5))  
   
 print(boxplot\_weighted)  
}



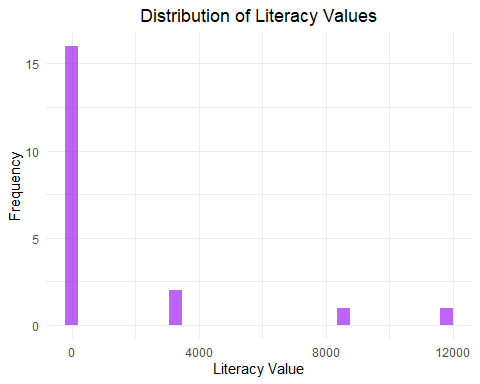
# 3. Boxplot for DenominatorUnweighted  
if("DenominatorUnweighted" %in% names(lit\_df)) {  
 boxplot\_unweighted <- ggplot(lit\_df, aes(y = DenominatorUnweighted)) +  
 geom\_boxplot(fill = "lightgreen", outlier.color = "red", outlier.shape = 16) +  
 labs(title = "Distribution of Denominator Unweighted Values",  
 y = "Denominator Unweighted") +  
 theme\_minimal() +  
 theme(plot.title = element\_text(hjust = 0.5))  
   
 print(boxplot\_unweighted)  
}



# 4. Distribution of literacy values by survey year (if available)  
if(all(c("Value", "SurveyYear") %in% names(lit\_df))) {  
 value\_distribution <- ggplot(lit\_df, aes(x = as.factor(SurveyYear), y = Value)) +  
 geom\_boxplot(fill = "orange", alpha = 0.7) +  
 labs(title = "Distribution of Literacy Values by Survey Year",  
 x = "Survey Year",  
 y = "Literacy Value") +  
 theme\_minimal() +  
 theme(plot.title = element\_text(hjust = 0.5),  
 axis.text.x = element\_text(angle = 45, hjust = 1))  
   
 print(value\_distribution)  
}



# 5. Histogram of literacy values  
if("Value" %in% names(lit\_df)) {  
 value\_histogram <- ggplot(lit\_df, aes(x = Value)) +  
 geom\_histogram(fill = "purple", alpha = 0.7, bins = 30) +  
 labs(title = "Distribution of Literacy Values",  
 x = "Literacy Value",  
 y = "Frequency") +  
 theme\_minimal() +  
 theme(plot.title = element\_text(hjust = 0.5))  
   
 print(value\_histogram)  
}



# Create a copy for comparison  
lit\_df\_original <- lit\_df  
numerical\_cols <- c("Value", "DenominatorWeighted", "DenominatorUnweighted")  
  
# Outlier treatment for each numerical column  
for(col in numerical\_cols) {  
 if(!all(is.na(lit\_df[[col]]))) {  
 # Calculate IQR bounds  
 q1 <- quantile(lit\_df[[col]], 0.25, na.rm = TRUE)  
 q3 <- quantile(lit\_df[[col]], 0.75, na.rm = TRUE)  
 iqr <- q3 - q1  
 lower\_bound <- q1 - 1.5 \* iqr  
 upper\_bound <- q3 + 1.5 \* iqr  
   
 # Method 1: Winsorization (cap outliers at bounds)  
 lit\_df <- lit\_df %>%  
 mutate(!!paste0(col, "\_winsorized") := case\_when(  
 .data[[col]] < lower\_bound ~ lower\_bound,  
 .data[[col]] > upper\_bound ~ upper\_bound,  
 TRUE ~ .data[[col]]  
 ))  
   
 # Method 2: Log transformation (for positive values only)  
 if(all(lit\_df[[col]] > 0, na.rm = TRUE)) {  
 lit\_df <- lit\_df %>%  
 mutate(!!paste0(col, "\_log") := log(.data[[col]] + 1)) # +1 to avoid log(0)  
 }  
 }  
}  
  
# Compare summary statistics before and after outlier treatment  
cat("Summary statistics before outlier treatment:\n")

## Summary statistics before outlier treatment:

summary(lit\_df\_original %>% select(all\_of(numerical\_cols)))

## Value DenominatorWeighted DenominatorUnweighted  
## Min. : 0.00 Min. :3202 Min. : 3179   
## 1st Qu.: 4.55 1st Qu.:3202 1st Qu.: 3179   
## Median : 42.10 Median :5858 Median : 7492   
## Mean : 1364.56 Mean :5858 Mean : 7492   
## 3rd Qu.: 100.00 3rd Qu.:8514 3rd Qu.:11805   
## Max. :11805.00 Max. :8514 Max. :11805

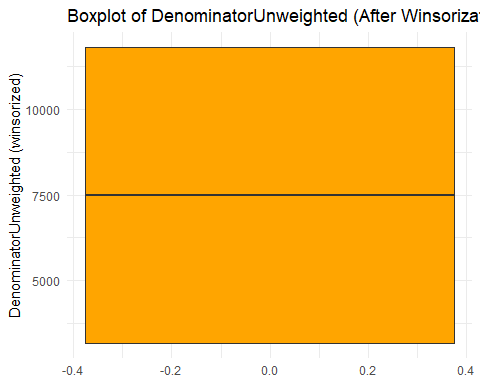
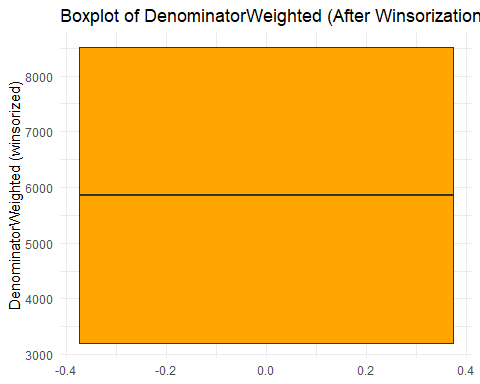
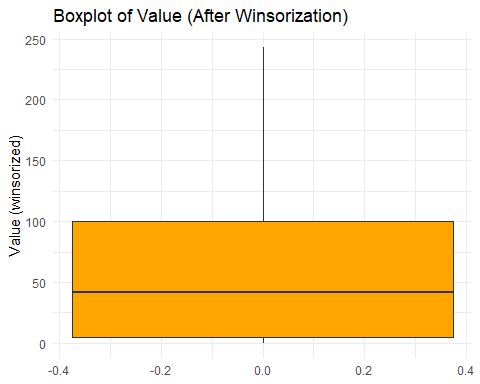
cat("\nSummary statistics after winsorization:\n")

##   
## Summary statistics after winsorization:

winsorized\_cols <- paste0(numerical\_cols, "\_winsorized")  
summary(lit\_df %>% select(all\_of(winsorized\_cols)))

## Value\_winsorized DenominatorWeighted\_winsorized  
## Min. : 0.00 Min. :3202   
## 1st Qu.: 4.55 1st Qu.:3202   
## Median : 42.10 Median :5858   
## Mean : 78.19 Mean :5858   
## 3rd Qu.:100.00 3rd Qu.:8514   
## Max. :243.18 Max. :8514   
## DenominatorUnweighted\_winsorized  
## Min. : 3179   
## 1st Qu.: 3179   
## Median : 7492   
## Mean : 7492   
## 3rd Qu.:11805   
## Max. :11805

# Visualize after winsorization  
if(length(winsorized\_cols) > 0) {  
 for(i in seq\_along(winsorized\_cols)) {  
 col <- winsorized\_cols[i]  
 orig\_col <- numerical\_cols[i]  
   
 if(!all(is.na(lit\_df[[col]]))) {  
 # Boxplot after treatment  
 p\_box\_after <- ggplot(lit\_df, aes(y = .data[[col]])) +  
 geom\_boxplot(fill = "orange", outlier.color = "red", outlier.shape = 16) +  
 labs(title = paste("Boxplot of", orig\_col, "(After Winsorization)"),  
 y = paste(orig\_col, "(winsorized)")) +  
 theme\_minimal()  
   
  
   
 print(p\_box\_after)  
   
 }  
 }  
}



#save cleaned data

write\_csv(lit\_df, here("data","processed", "literacy\_cleaned.csv"))