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
  )
```

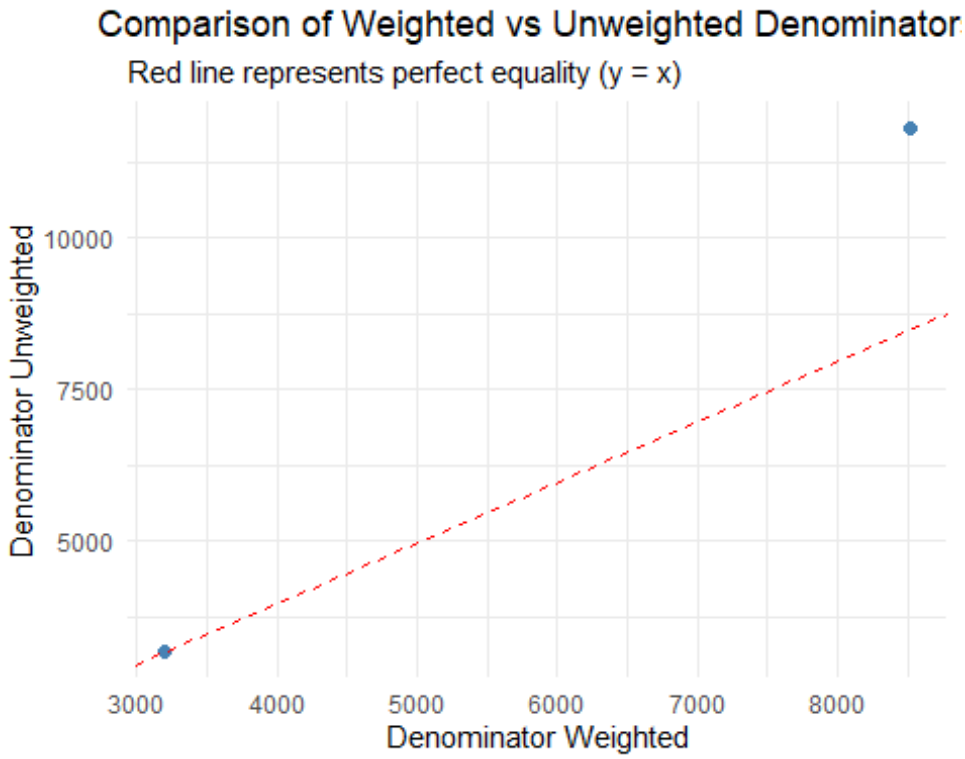
#Missing Value Handling

```
lit_df <- lit_df %>%
  fill(DenominatorWeighted, DenominatorUnweighted, .direction = "downup")

lit_df[
       c("DenominatorWeighted", "DenominatorUnweighted")]
```

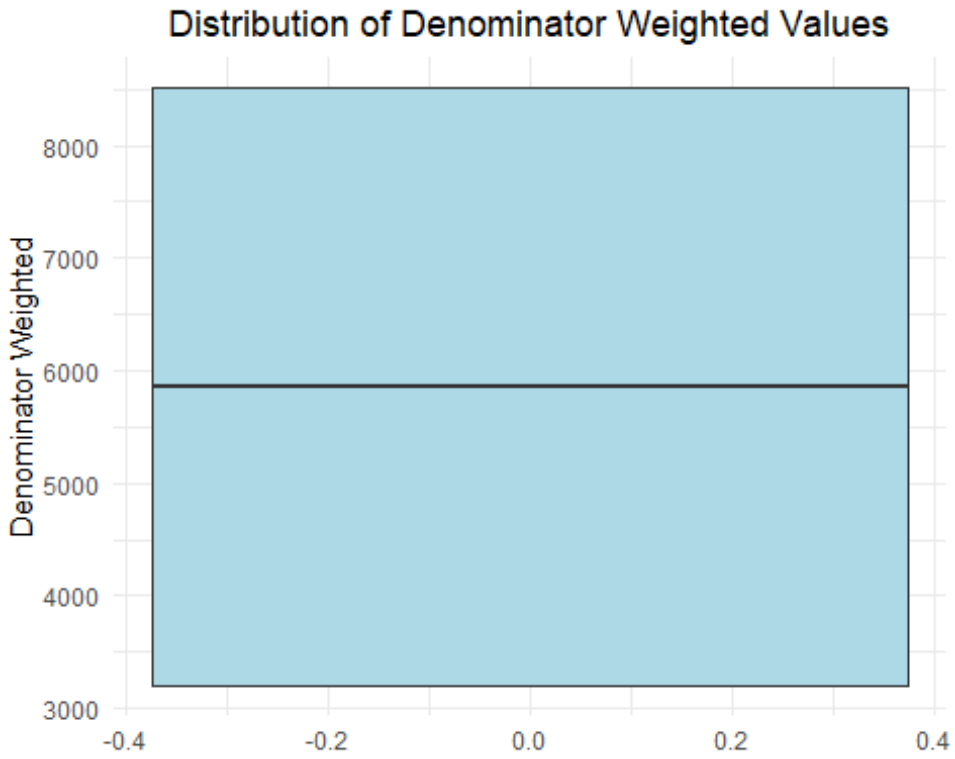
```
## # A tibble: 20 × 2
##    DenominatorWeighted DenominatorUnweighted
##                  <dbl>                 <dbl>
```

```
##  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
```

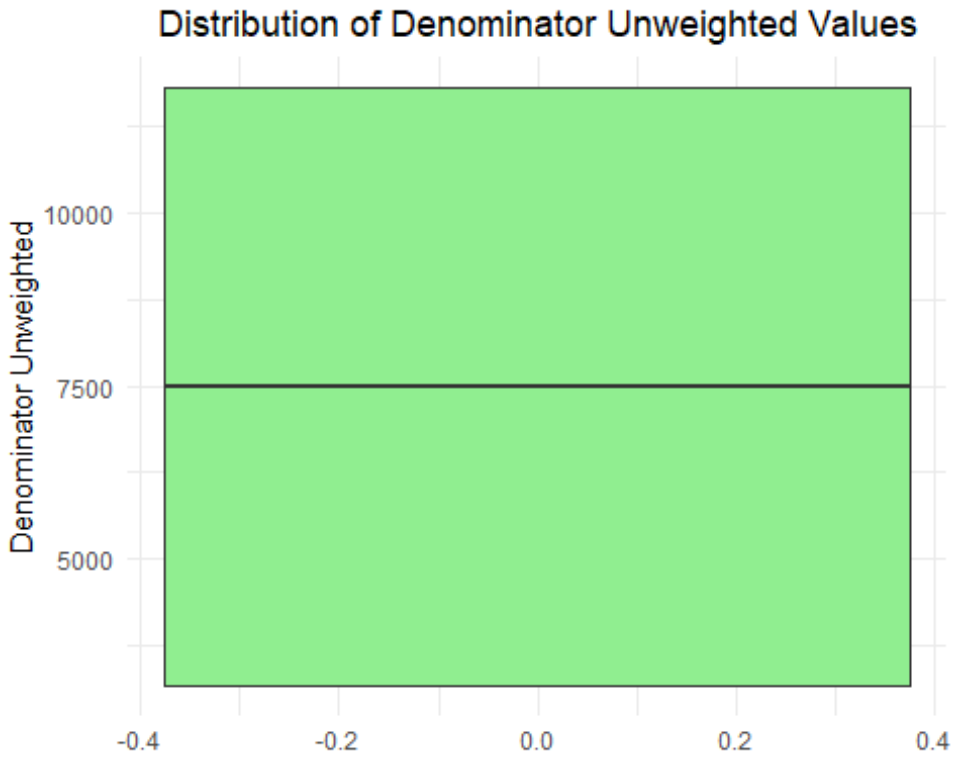
```r
# 1. Scatterplot comparing weighted vs unweighted denominators
if(all(c("DenominatorWeighted", "DenominatorUnweighted") %in% names(lit_df)))
{
  scatter_plot <- ggplot(lit_df, aes(x = DenominatorWeighted, y =
DenominatorUnweighted)) +
    geom_point(alpha = 0.6, color = "steelblue", size = 2) +
    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)
}
```



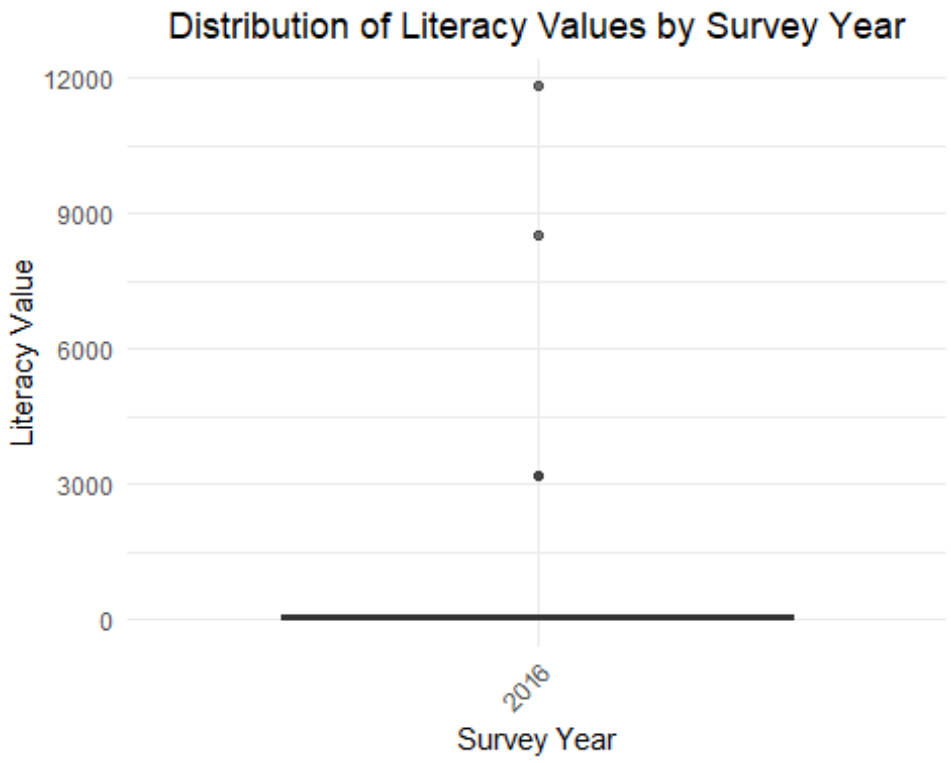

```r
# 2. Boxplot for DenominatorWeighted
if("DenominatorWeighted" %in% names(lit_df)) {
  boxplot_weighted <- ggplot(lit_df, aes(y = DenominatorWeighted)) +
    geom_boxplot(fill = "lightblue", outlier.color = "red", outlier.shape =
16) +
    labs(title = "Distribution of Denominator Weighted Values",
         y = "Denominator Weighted") +
    theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5))

  print(boxplot_weighted)
}
```



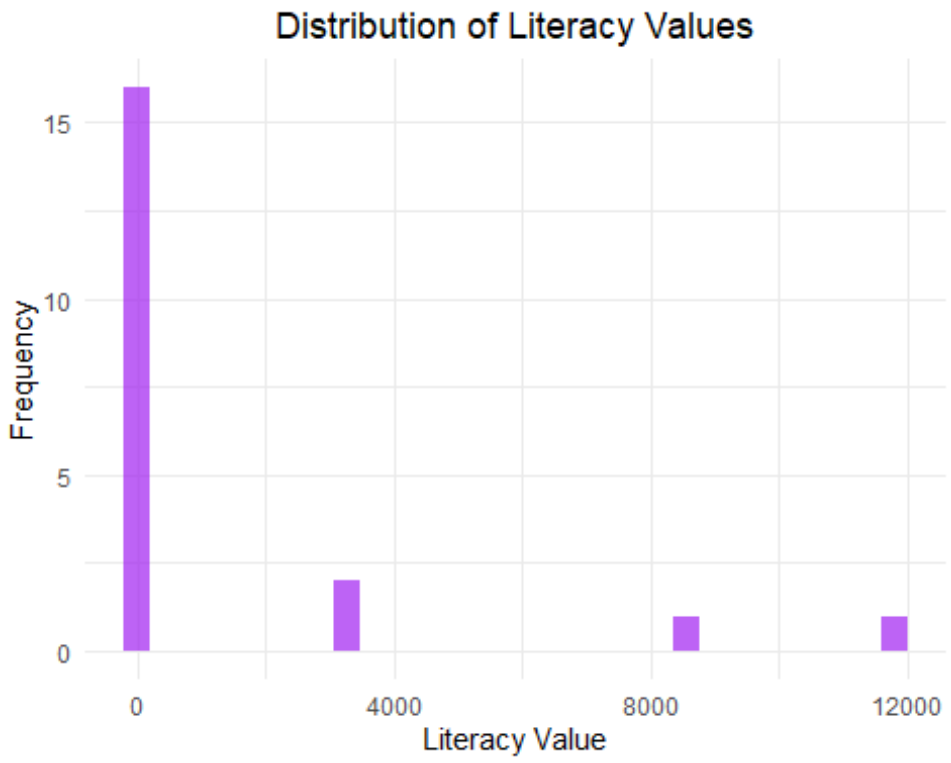

```r
# 3. Boxplot for DenominatorUnweighted
if("DenominatorUnweighted" %in% names(lit_df)) {
  boxplot_unweighted <- ggplot(lit_df, aes(y = DenominatorUnweighted)) +
    geom_boxplot(fill = "lightgreen", outlier.color = "red", outlier.shape =
16) +
    labs(title = "Distribution of Denominator Unweighted Values",
         y = "Denominator Unweighted") +
    theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5))

  print(boxplot_unweighted)
}
```




```r
# 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)
}
```



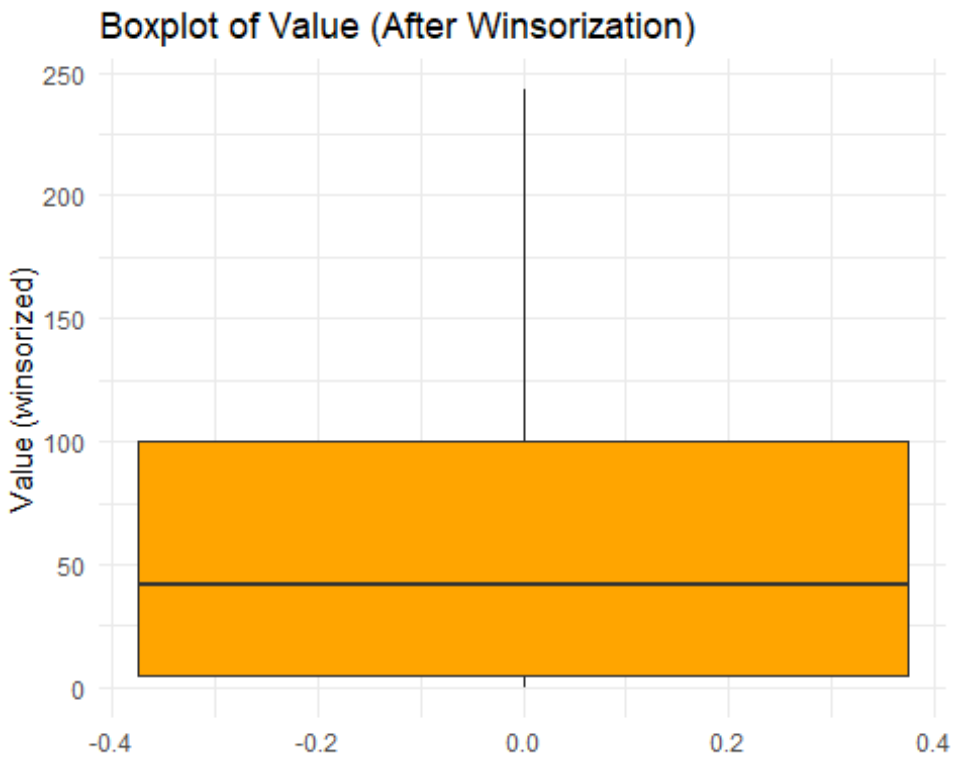

```r
# 5. Histogram of literacy values
if("Value" %in% names(lit_df)) {
  value_histogram <- ggplot(lit_df, aes(x = Value)) +
    geom_histogram(fill = "purple", alpha = 0.7, bins = 30) +
    labs(title = "Distribution of Literacy Values",
         x = "Literacy Value",
         y = "Frequency") +
    theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5))

  print(value_histogram)
}
```

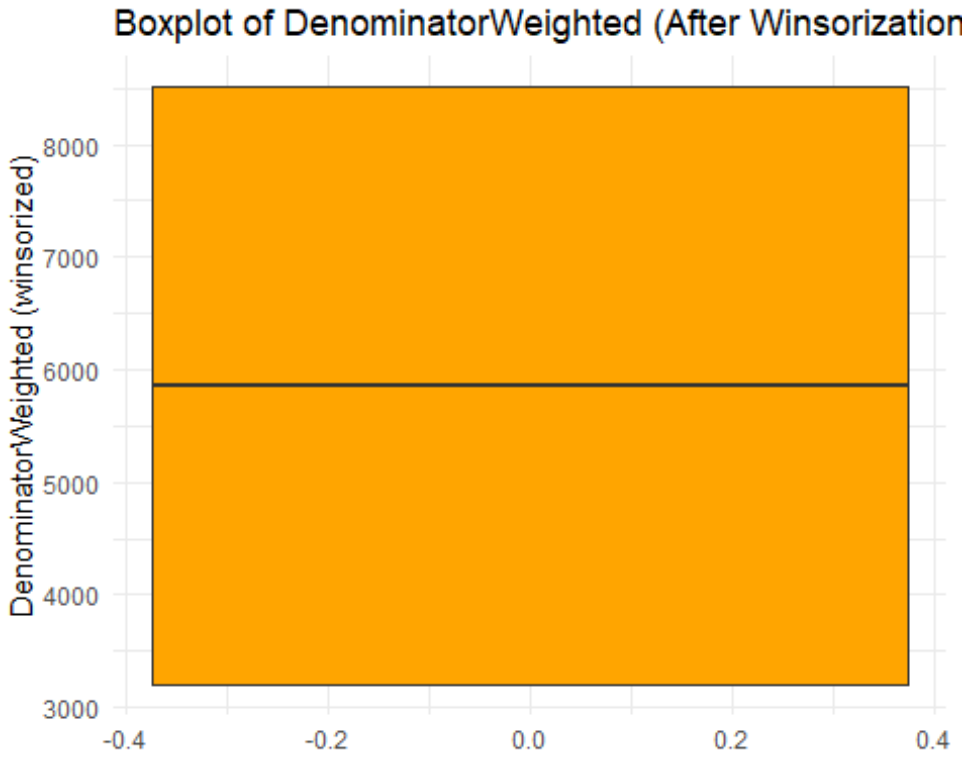


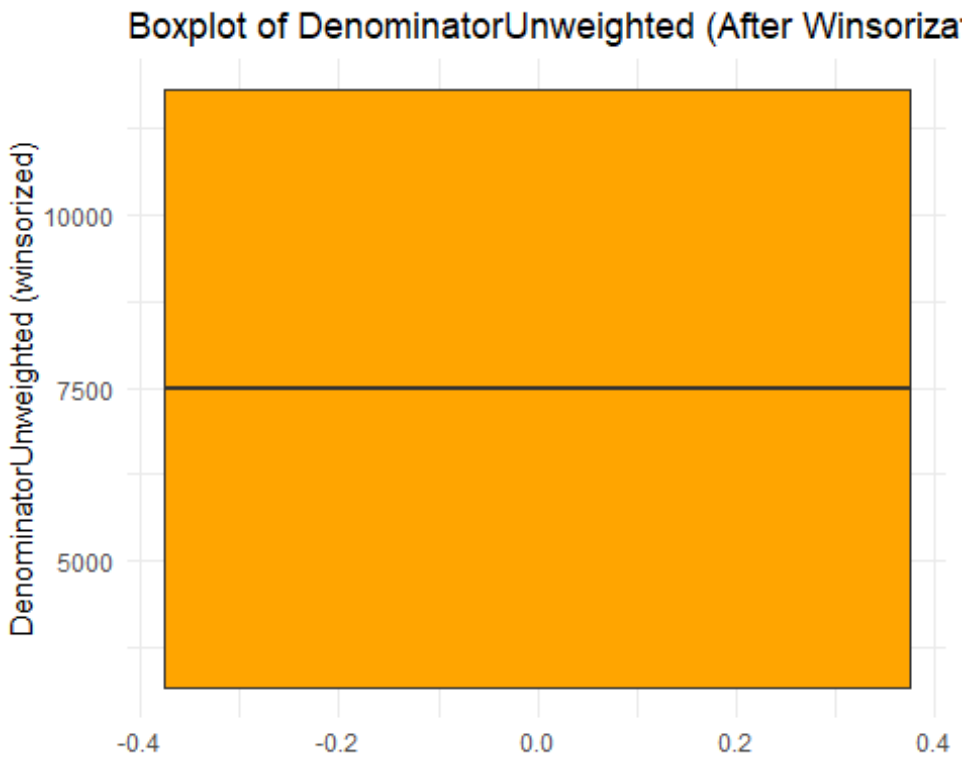

```r
# Create a copy for comparison
lit_df_original <- lit_df
numerical_cols <- c("Value", "DenominatorWeighted", "DenominatorUnweighted")

# 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)

    }
  }
}
```


Boxplot of Value (After Winsorization)
Value (winsorized)
250
200
150
100
50
0
-0.4
-0.2
0.0
0.2
0.4


Boxplot of DenominatorWeighted (After Winsorization
DenominatorWeighted (winsorized)
8000
7000
6000
5000
4000
3000
-0.4
-0.2
0.0
0.2
0.4



#save cleaned data

```
write_csv(lit_df, here("data","processed", "literacy_cleaned.csv"))
```