CL_IYCF_09

Group K

2025-09-19

Firts 5 rows

```
## # A tibble: 5 x 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    795971 Children~ 87.4  1         ZA              South Afri~ 1998
## 3 ZAF    795973 Children~ 38.9  1         ZA              South Afri~ 1998
## 4 ZAF    621666 Children~ 6.9   1         ZA              South Afri~ 1998
## 5 ZAF    621667 Children~ 6.3   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>
```

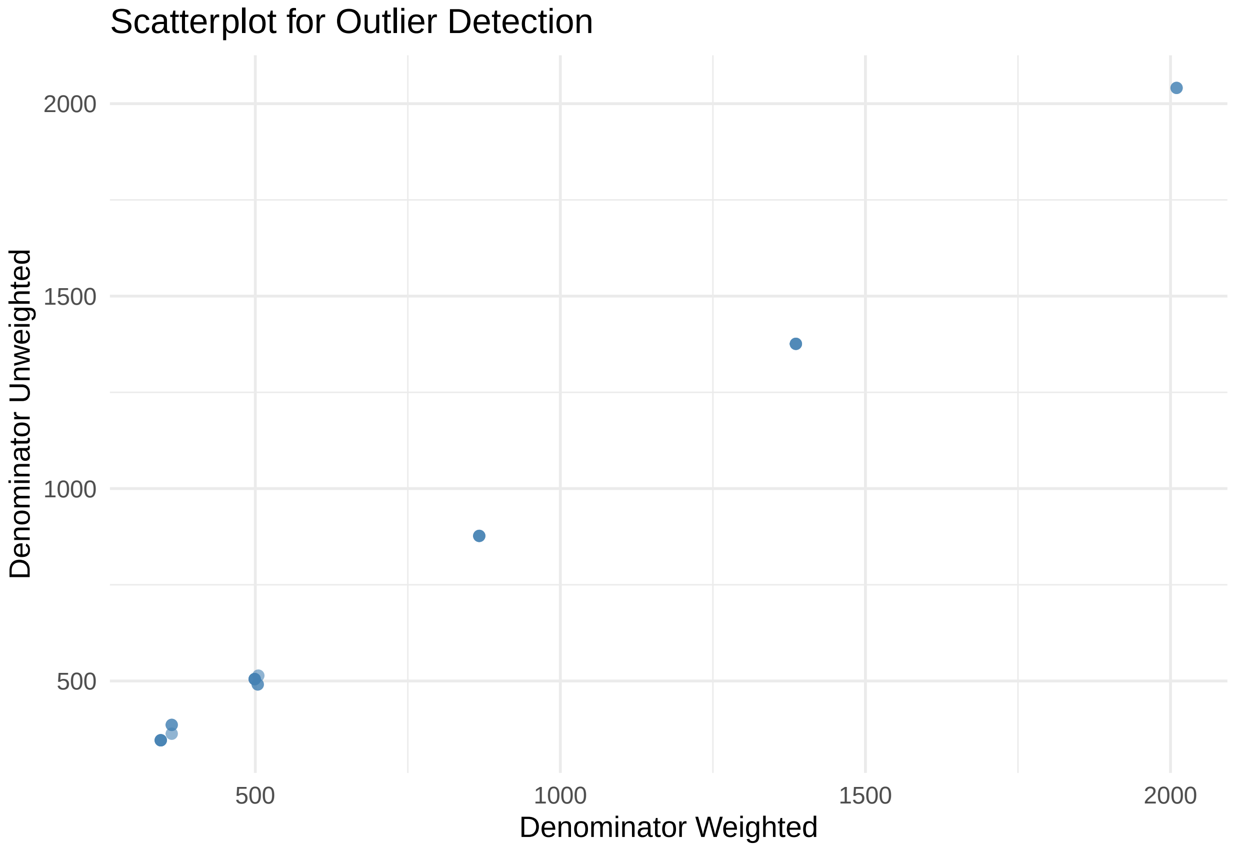
Convert Data Types

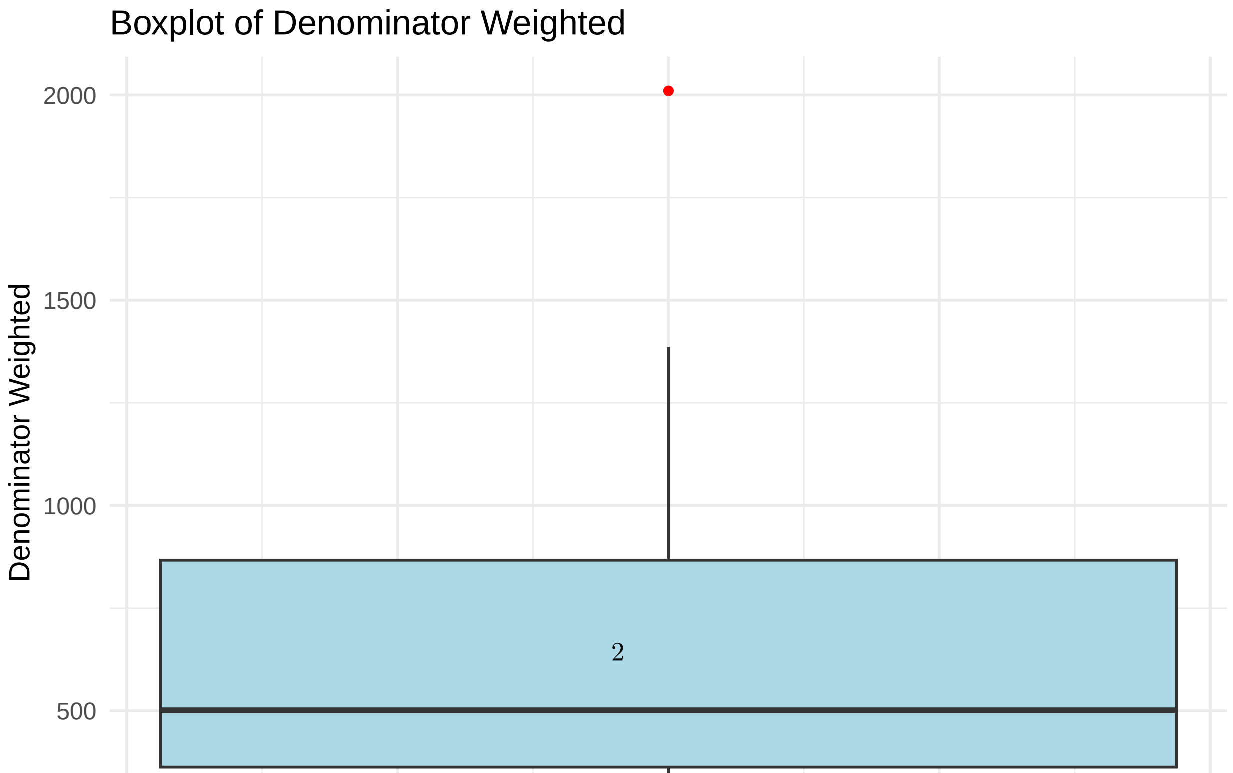
```
icy_df <- icy_df %>%
  mutate(
        Value = as.numeric(Value),
    Precision = as.numeric(Precision),
    SurveyYear = as.integer(SurveyYear),
    IndicatorOrder = as.integer(IndicatorOrder),
    CharacteristicId = as.integer(CharacteristicId),
    CharacteristicOrder = as.integer(CharacteristicOrder),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

check for unique values

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

Assumed pattern, the missing values can be filled with the previous non missing value in the opposite attribute





Outlier Handling

```r
# Calculate IQR boundaries
Q1_w <- quantile(icy_df$DenominatorWeighted, 0.25, na.rm = TRUE)
Q3_w <- quantile(icy_df$DenominatorWeighted, 0.75, na.rm = TRUE)
IQR_w <- Q3_w - Q1_w
lower_w <- Q1_w - 1.5 * IQR_w
upper_w <- Q3_w + 1.5 * IQR_w

Q1_uw <- quantile(icy_df$DenominatorUnweighted, 0.25, na.rm = TRUE)
Q3_uw <- quantile(icy_df$DenominatorUnweighted, 0.75, na.rm = TRUE)
IQR_uw <- Q3_uw - Q1_uw
lower_uw <- Q1_uw - 1.5 * IQR_uw
upper_uw <- Q3_uw + 1.5 * IQR_uw

# Cap values to the IQR limits
icy_df <- icy_df %>%
  mutate(
    DenominatorWeighted = pmin(pmax(DenominatorWeighted, lower_w), upper_w),
    DenominatorUnweighted = pmin(pmax(DenominatorUnweighted, lower_uw), upper_uw)
  )
```