Income Bilingual

2024-07-29

Loading in Data

```
#all crosswalks
ddi <- read_ipums_ddi("usa_00009.xml")
all_indicator_data <- read_ipums_micro(ddi)
```

```
## Use of data from IPUMS USA is subject to conditions including that users should cite
the data appropriately. Use command `ipums_conditions()` for more details.
```

```
#location data
regions <- read.csv("../location_data/County_12_Regions.csv")
rural_urban <-read.csv("../location_data/rural_urban.csv")

#language micro with bilungual
language_micro_data <- read.csv("Microdata_Bilingualism.csv")
```

FUNCTIONS

```
weighted_median <- function(values, weights) {
  sorted_indices <- order(values) #finding ascending order
  sorted_values <- values[sorted_indices] #sorting them by the order
  sorted_weights <- weights[sorted_indices]
  cumulative_weight <- cumsum(sorted_weights)
  cutoff <- sum(sorted_weights) / 2
  median_value <- sorted_values[which(cumulative_weight >= cutoff)[1]]
  return(median_value)
}
```

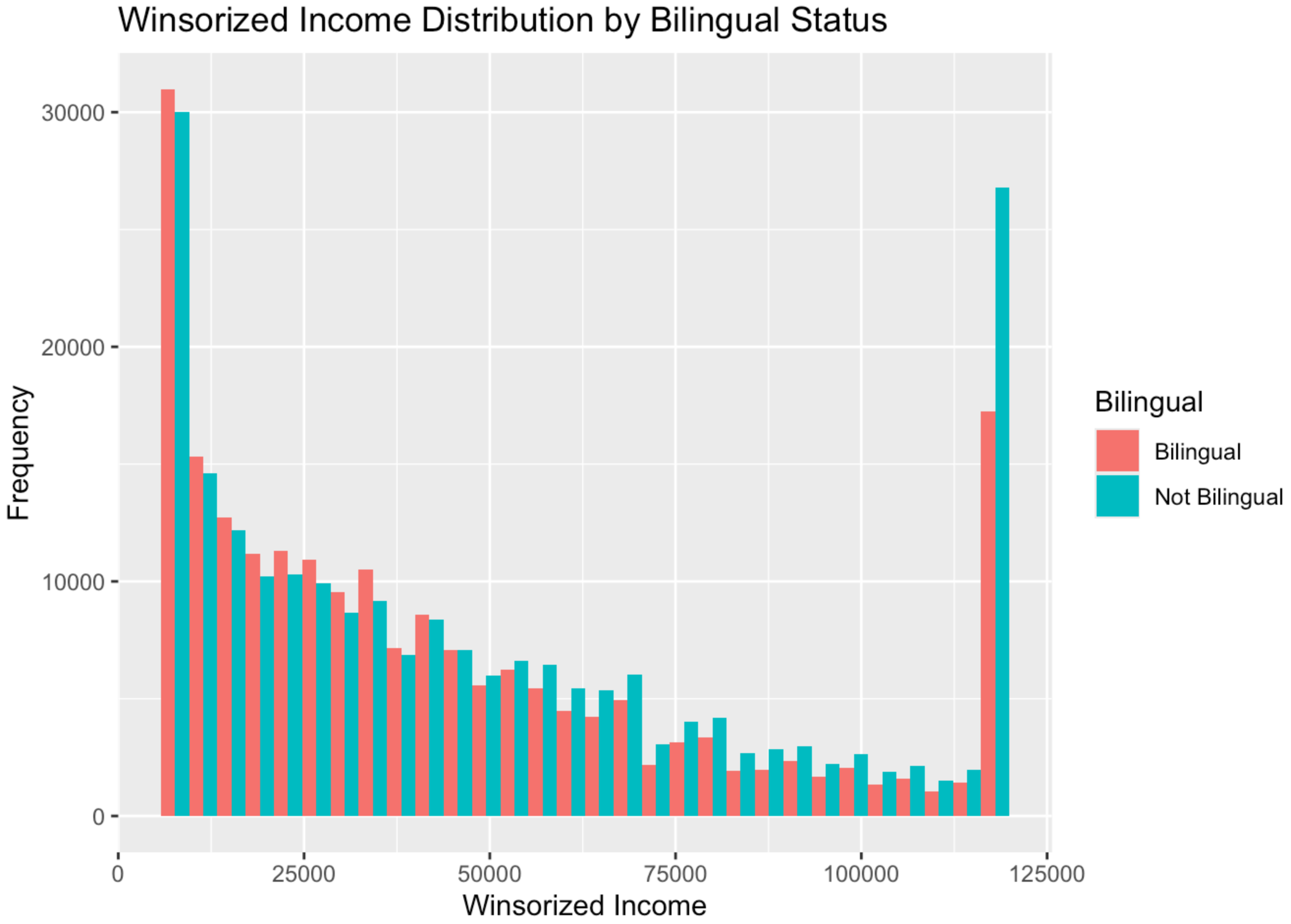
INCOMES

```
income_data <- language_micro_data |>
  select(AGE, INCTOT, Bilingual, PERWT, SEX, EDUCD, AGE) |>
  filter(INCTOT != 9999999 & INCTOT > 0) |>
  filter(AGE > 14)

sum_weights_filtered <- sum(income_data$PERWT, na.rm = TRUE)

sum_weights_original <- sum(language_micro_data$PERWT, na.rm = TRUE)

#recalibrate weights
income_data <- income_data |>
  mutate(recalibrated_weight = PERWT * (sum_weights_filtered / sum_weights_original))

income_data_weighted <- income_data |>
  mutate(Weighted_Income = INCTOT * (recalibrated_weight / 100))
```

```
write.csv(file = "Income_Bilingualism_Weighted_Data.csv", income_data_weighted)
```

TRYING WINSORIZATION

```
library(dplyr)
library(ggplot2)

# Custom winsorization function
winsorize <- function(x, trim = 0.1) {
  lower_bound <- quantile(x, trim, na.rm = TRUE)
  upper_bound <- quantile(x, 1 - trim, na.rm = TRUE)
  x[x < lower_bound] <- lower_bound
  x[x > upper_bound] <- upper_bound
  return(x)
}

# Apply winsorization
income_data_winsor <- income_data |>
  mutate(WINSORIZED_INCTOT = winsorize(INCTOT, trim = 0.1))

# Visualize the winsorized data
income_data_winsor |>
  ggplot(aes(x = WINSORIZED_INCTOT, fill = Bilingual)) +
  geom_histogram(bins = 30, position = "dodge") +
  ggtitle("Winsorized Income Distribution by Bilingual Status") +
  xlab("Winsorized Income") +
  ylab("Frequency")
```



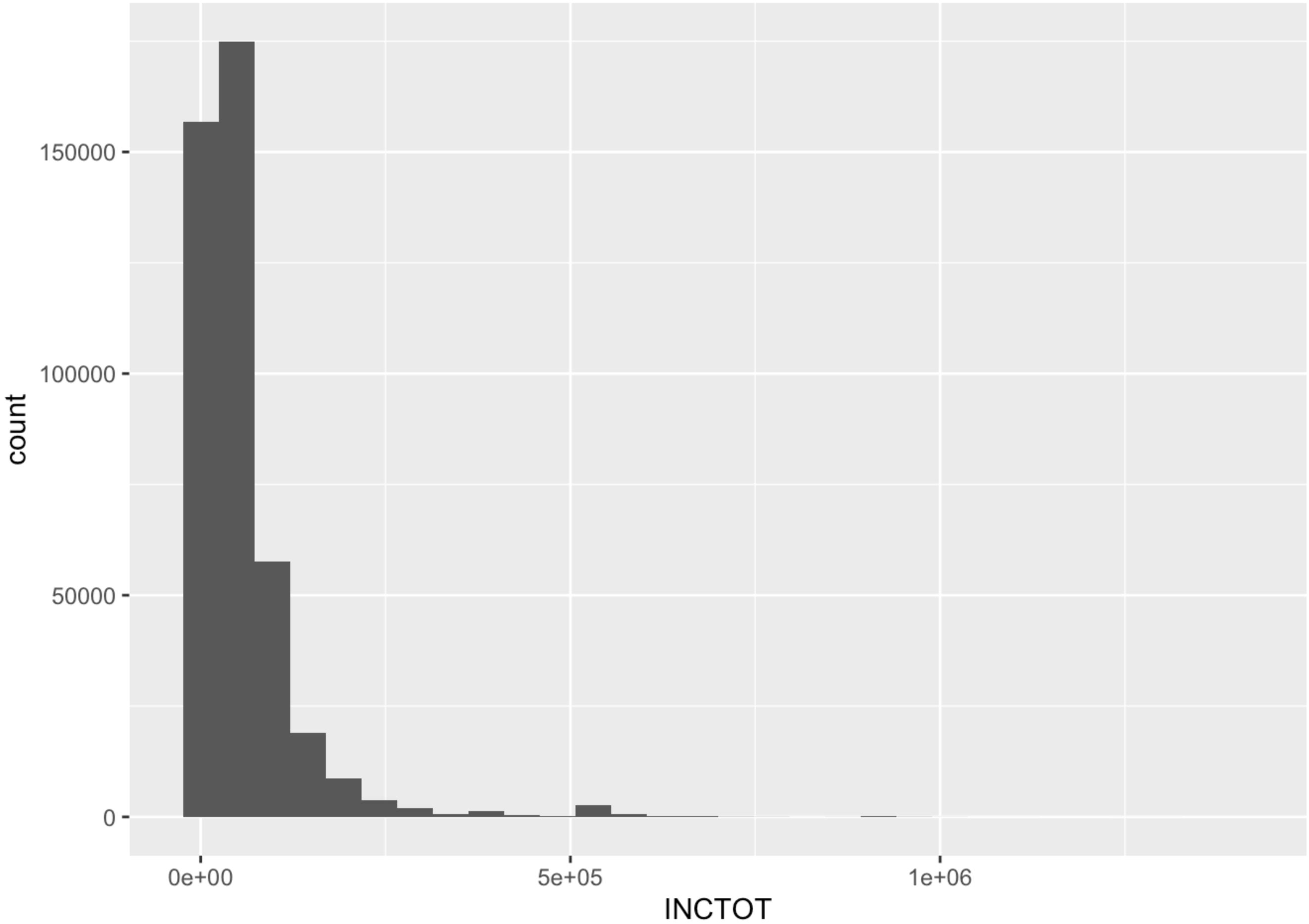
```
# Calculate weighted median for winsorized data
median_income_bilingual_winsor <- weighted_median(
  income_data_winsor |> filter(Bilingual == "Bilingual") |> pull(WINSORIZED_INCTOT),
  income_data_winsor |> filter(Bilingual == "Bilingual") |> pull(recalibrated_weight)
)

median_income_english_winsor <- weighted_median(
  income_data_winsor |> filter(Bilingual == "Not Bilingual") |> pull(WINSORIZED_INCTOT),
  income_data_winsor |> filter(Bilingual == "Not Bilingual") |> pull(recalibrated_weigh
t)
)

# Summarize winsorized data
income_summary_winsor <- income_data_winsor |>
  group_by(Bilingual) |>
  summarize(
    Mean_Income = mean(WINSORIZED_INCTOT, na.rm = TRUE),
    Median_Income = weighted_median(WINSORIZED_INCTOT, recalibrated_weight))

print(income_summary_winsor)
```

```
## # A tibble: 2 × 3
##   Bilingual     Mean_Income Median_Income
##   <chr>               <dbl>         <dbl>
## 1 Bilingual          43019.         33994
## 2 Not Bilingual      49036.         40000
```

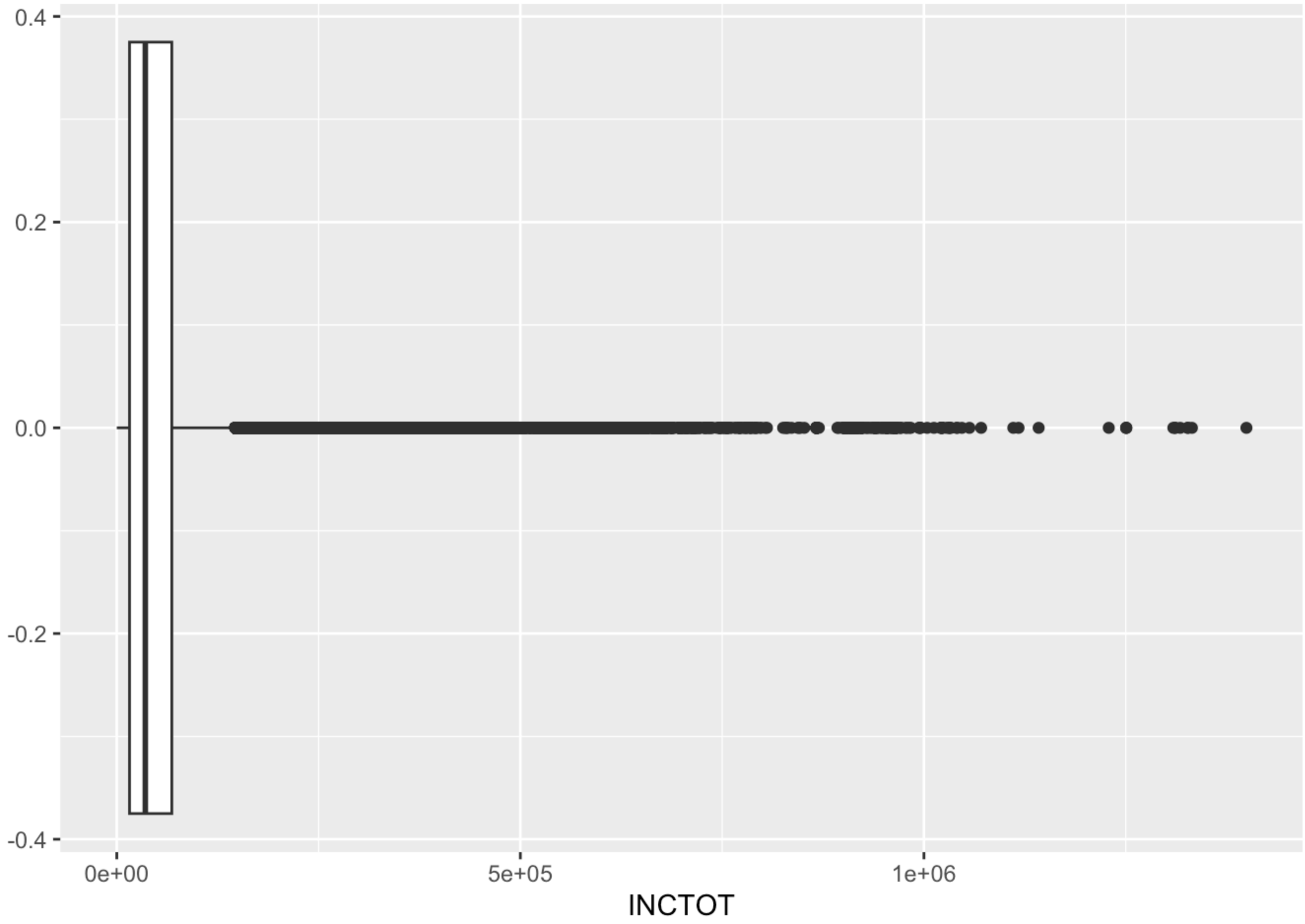
```
write.csv(file = "Income_Data_Winsorization.csv", income_data_winsor)
```

```
income_data |>
  ggplot(aes(x = INCTOT)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

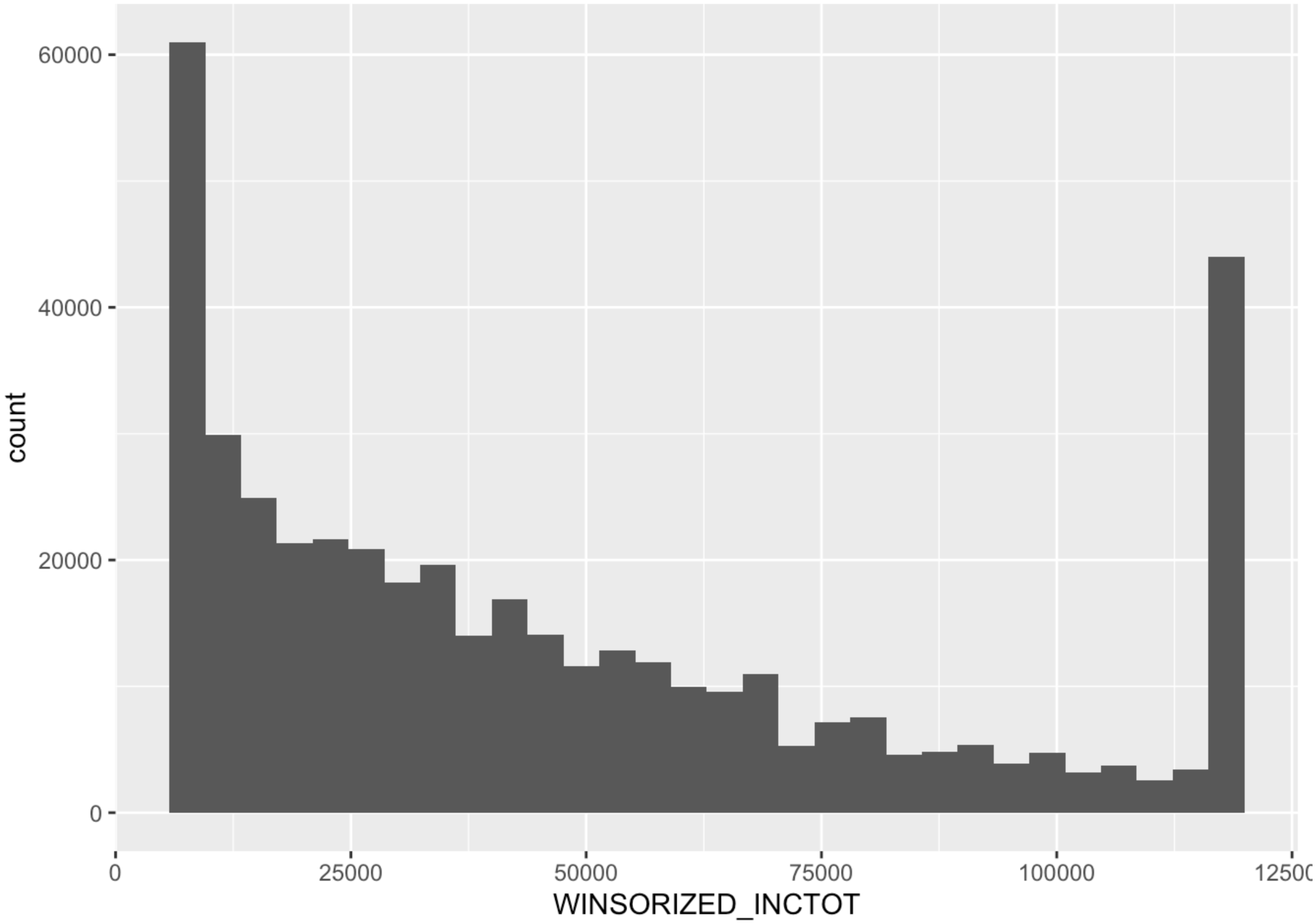


```
income_data |>
  ggplot(aes(x = INCTOT)) +
  geom_boxplot()
```

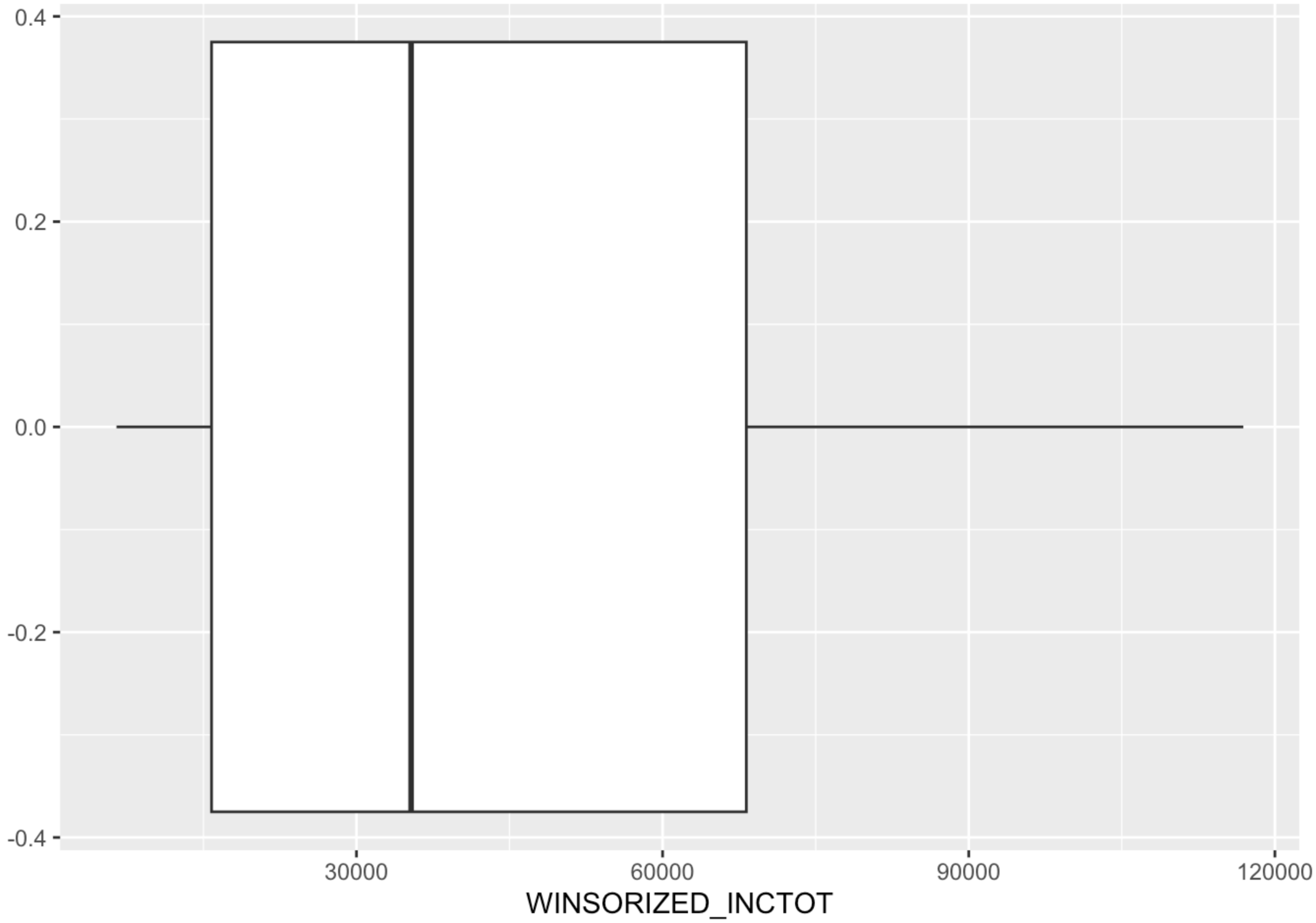


```
income_data_winsor |>
  ggplot(aes(x = WINSORIZED_INCTOT)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
income_data_winsor |>
  ggplot(aes(x = WINSORIZED_INCTOT)) +
  geom_boxplot()
```


0.4
0.2
0.0
-0.2
-0.4
30000
60000
90000
120000
WINSORIZED_INCTOT

```
# calculating weighted median attempt
#should grab indices for sorted values
#apply indices to the values AND the weights so they should be lined up accordinly
# going to find median by where 50% of weight is hit and can figure this out w cumulativ
e sum
# once this place is reached will assign it as cutoff and then
# will have median by first value that is greater than that
weighted_median <- function(values, weights) {
  sorted_indices <- order(values) #finding ascending order
  sorted_values <- values[sorted_indices] #sorting them by the order
  sorted_weights <- weights[sorted_indices]
  cumulative_weight <- cumsum(sorted_weights)
  cutoff <- sum(sorted_weights) / 2
  median_value <- sorted_values[which(cumulative_weight >= cutoff)[1]]
  return(median_value)
}

median_income_bilingual <- weighted_median(income_data |>
                          filter(Bilingual == "Bilingual") |> pull(INCTOT),
                          income_data |> filter(Bilingual == "Bilingual")
                          |> pull(recalibrated_weight))

median_income_monolingual <- weighted_median(income_data |>
                            filter(Bilingual == "Not Bilingual") |>
                            pull(INCTOT),income_data |>
                            filter(Bilingual == "Not Bilingual") |>
                            pull(recalibrated_weight))

result_df <- tibble(
  Category = c("Bilingual", "Not Bilingual"),
  Median_Income = c(median_income_bilingual, median_income_monolingual)
)

result_df
```

```
## # A tibble: 2 × 2
##   Category      Median_Income
##   <chr>                 <dbl>
## 1 Bilingual             33994
## 2 Not Bilingual         40000
```

seems rather low compared to what 2022 5 year ACS says, around 60,000 median personal

```
#trimmed mean function
trimmed_mean <- function(x, trim = 0.1) {
  return(mean(x, trim = trim, na.rm = TRUE))
}

#summary statistics with trimmed mean
income_summary_trimmed <- income_data_weighted |>
  group_by(Bilingual) |>
  summarize(Trimmed_Mean_Income = trimmed_mean(INCTOT, 0.1))

#print trimmed mean results
print(income_summary_trimmed)
```

```
## # A tibble: 2 × 2
##   Bilingual     Trimmed_Mean_Income
##   <chr>                       <dbl>
## 1 Bilingual                  38472.
## 2 Not Bilingual              46000.
```

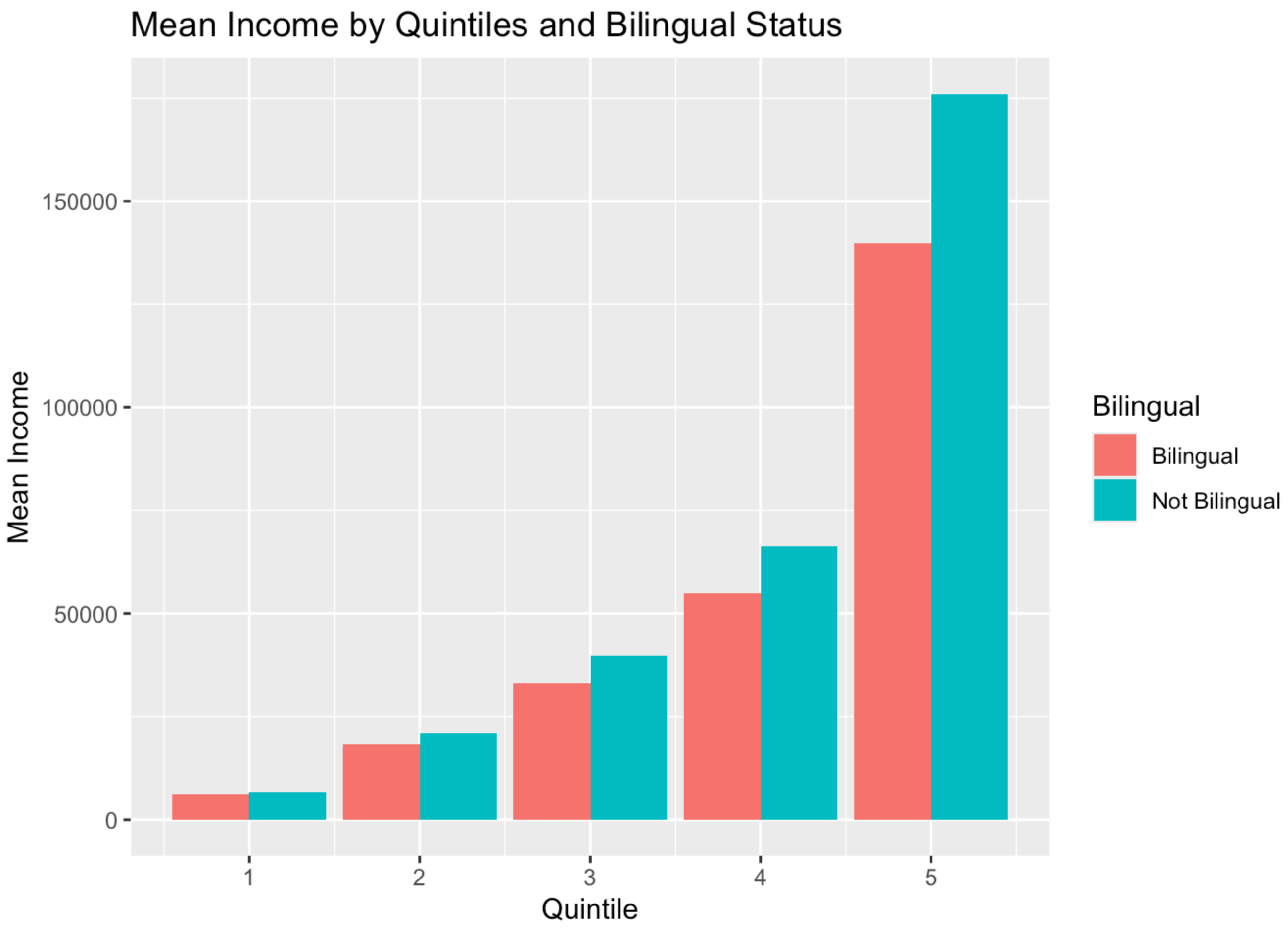
seeing if trimmed mean shows greater change

Quintiles

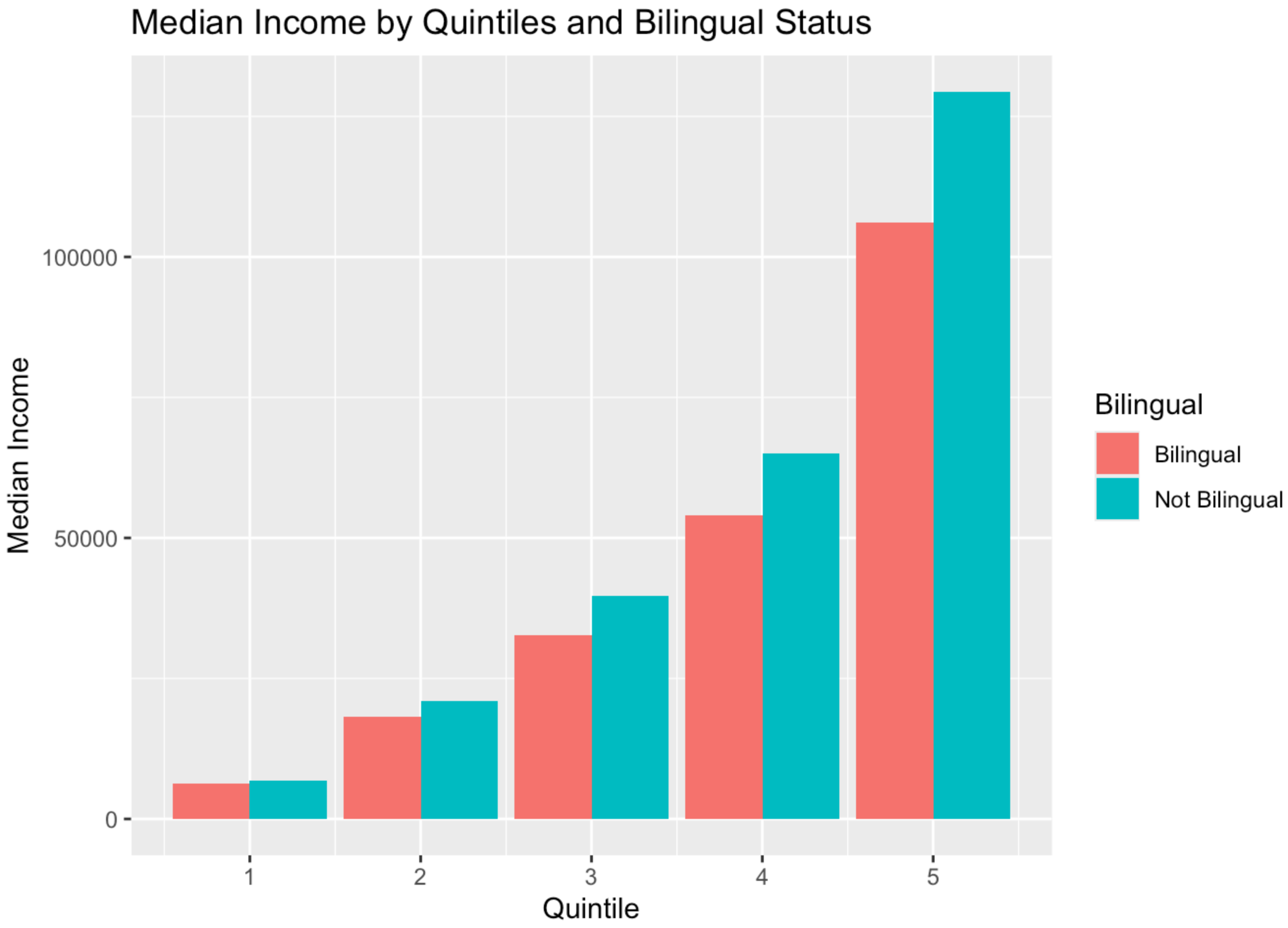
```r
options(scipen = 999)
#quints for bilingual
quintiles_bilingual <- quantile(
  income_data |> filter(Bilingual == "Bilingual") |> pull(INCTOT),
  probs = seq(0, 1, 0.2),
  na.rm = TRUE
)

#quints for non bilingual
quintiles_monolingual <- quantile(
  income_data |> filter(Bilingual == "Not Bilingual") |> pull(INCTOT),
  probs = seq(0, 1, 0.2),
  na.rm = TRUE
)

#assign them
assign_quintile <- function(x, quintiles) {
  cut(x, breaks = quintiles, include.lowest = TRUE, labels = FALSE)
}

#add a variable for quints
income_data_quint <- income_data |>
  mutate(
    Quintile = case_when(
      Bilingual == "Bilingual" ~ assign_quintile(INCTOT, quintiles_bilingual),
      Bilingual == "Not Bilingual" ~ assign_quintile(INCTOT, quintiles_monolingual)
    )
  )

#create summary
income_summary_quintiles <- income_data_quint |>
  group_by(Bilingual, Quintile) |>
  summarize(
    Mean_Income = mean(INCTOT, na.rm = TRUE),
    Median_Income = median(INCTOT, na.rm = TRUE),
    .groups = 'drop'
  )

print(income_summary_quintiles)
```

```
## # A tibble: 10 × 4
##    Bilingual     Quintile Mean_Income Median_Income
##    <chr>            <int>       <dbl>         <dbl>
##  1 Bilingual            1       6149.          6300
##  2 Bilingual            2      18323.         18130
##  3 Bilingual            3      33049.         32734
##  4 Bilingual            4      54905.         54046
##  5 Bilingual            5     139829.        106095
##  6 Not Bilingual        1       6605.          6810
##  7 Not Bilingual        2      20960.         20898
##  8 Not Bilingual        3      39642.         39660
##  9 Not Bilingual        4      66426.         65000
## 10 Not Bilingual        5     176011.        129400
```

```
# mean
income_summary_quintiles |>
  ggplot(aes(x = Quintile, y = Mean_Income, fill = Bilingual)) +
  geom_bar(stat = "identity", position = "dodge") +
  ggtitle("Mean Income by Quintiles and Bilingual Status") +
  xlab("Quintile") +
  ylab("Mean Income")
```



```
# median
income_summary_quintiles |>
  ggplot(aes(x = Quintile, y = Median_Income, fill = Bilingual)) +
  geom_bar(stat = "identity", position = "dodge") +
  ggtitle("Median Income by Quintiles and Bilingual Status") +
  xlab("Quintile") +
  ylab("Median Income")
```



WEIGHTED QUINTILES

```
#making quints based on weight
weighted_quantile <- function(values, weights, probs) {
  sorted_indices <- order(values)
  sorted_values <- values[sorted_indices]
  sorted_weights <- weights[sorted_indices]
  cumulative_weight <- cumsum(sorted_weights)
  total_weight <- sum(sorted_weights)

  quantiles <- sapply(probs, function(p) { #makign them based ont he probability
    cutoff <- total_weight * p
    return(sorted_values[which(cumulative_weight >= cutoff)[1]])
  })

  return(quantiles)
}

#how it works with 0th iteration
# cutoff = total_weight * 0 = 15 * 0 = 0
# index = which(cumulative_weight >= cutoff)[1] = which(cumulative_weight >= 0)[1] = 1
# quantile value = sorted_values[1] = 10

weighted_quintiles_bilingual <- weighted_quantile(
  income_data |>
    filter(Bilingual == "Bilingual") |>
    pull(INCTOT), income_data |>        #pulling values
    filter(Bilingual == "Bilingual") |>
    pull(recalibrated_weight),   #pulling weights
  probs = seq(0, 1, 0.2) #sending probs from 0 to 1 increase by 0.2 for quintiles
)

# english -- same thing as above
weighted_quintiles_english <- weighted_quantile(
  income_data |>
    filter(Bilingual == "Not Bilingual") |>
    pull(INCTOT), income_data |>
    filter(Bilingual == "Not Bilingual") |>
    pull(recalibrated_weight), probs = seq(0, 1, 0.2)
)

print("Quintiles")
```

```
## [1] "Quintiles"
```

```
weighted_quintiles_bilingual
```

```
## [1]       1   12860   26304   42600   70144 1326908
```

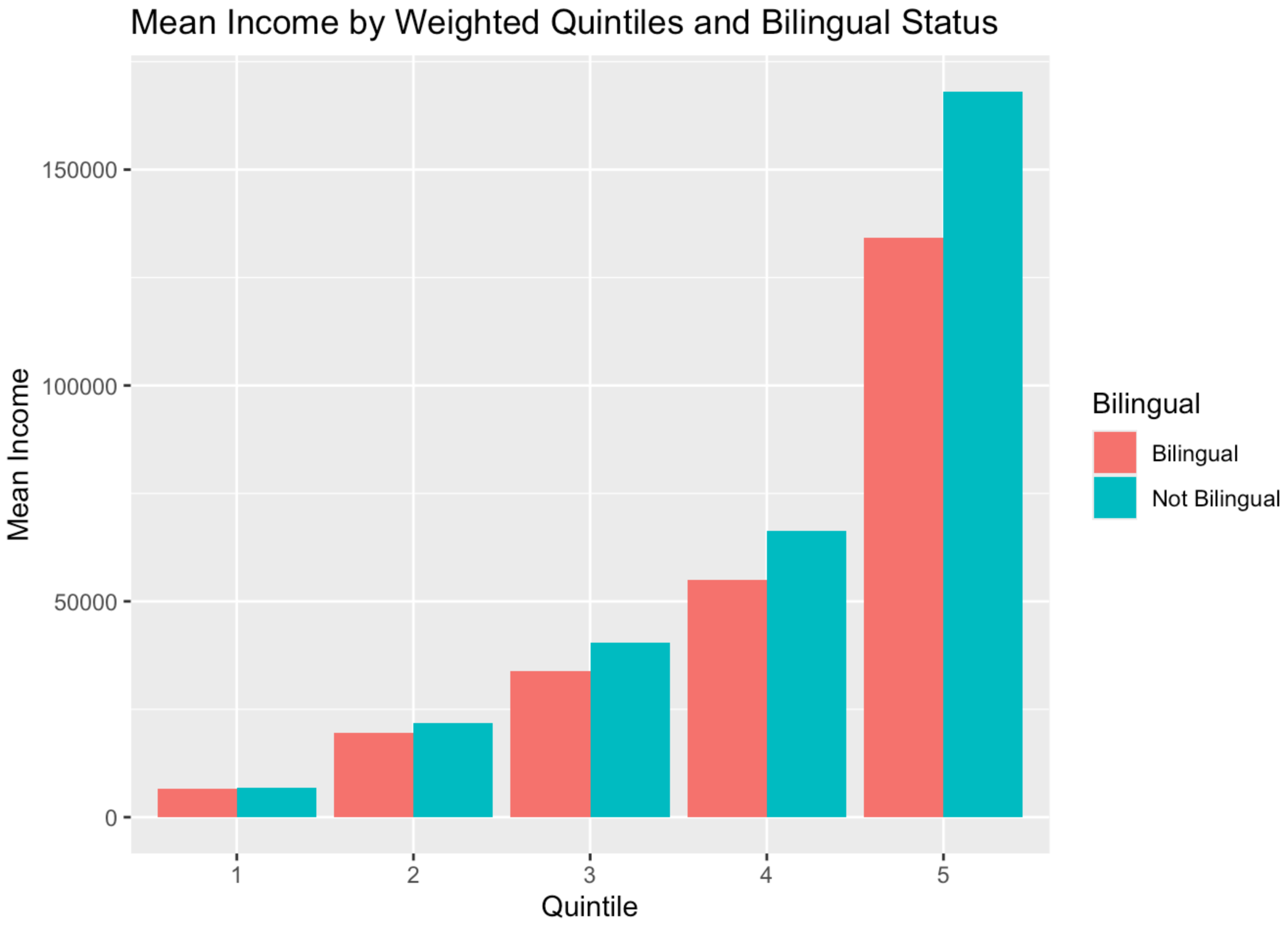
```
weighted_quintiles_english
```

```
## [1]       1   13678   30000   51670   85000 1399542
```

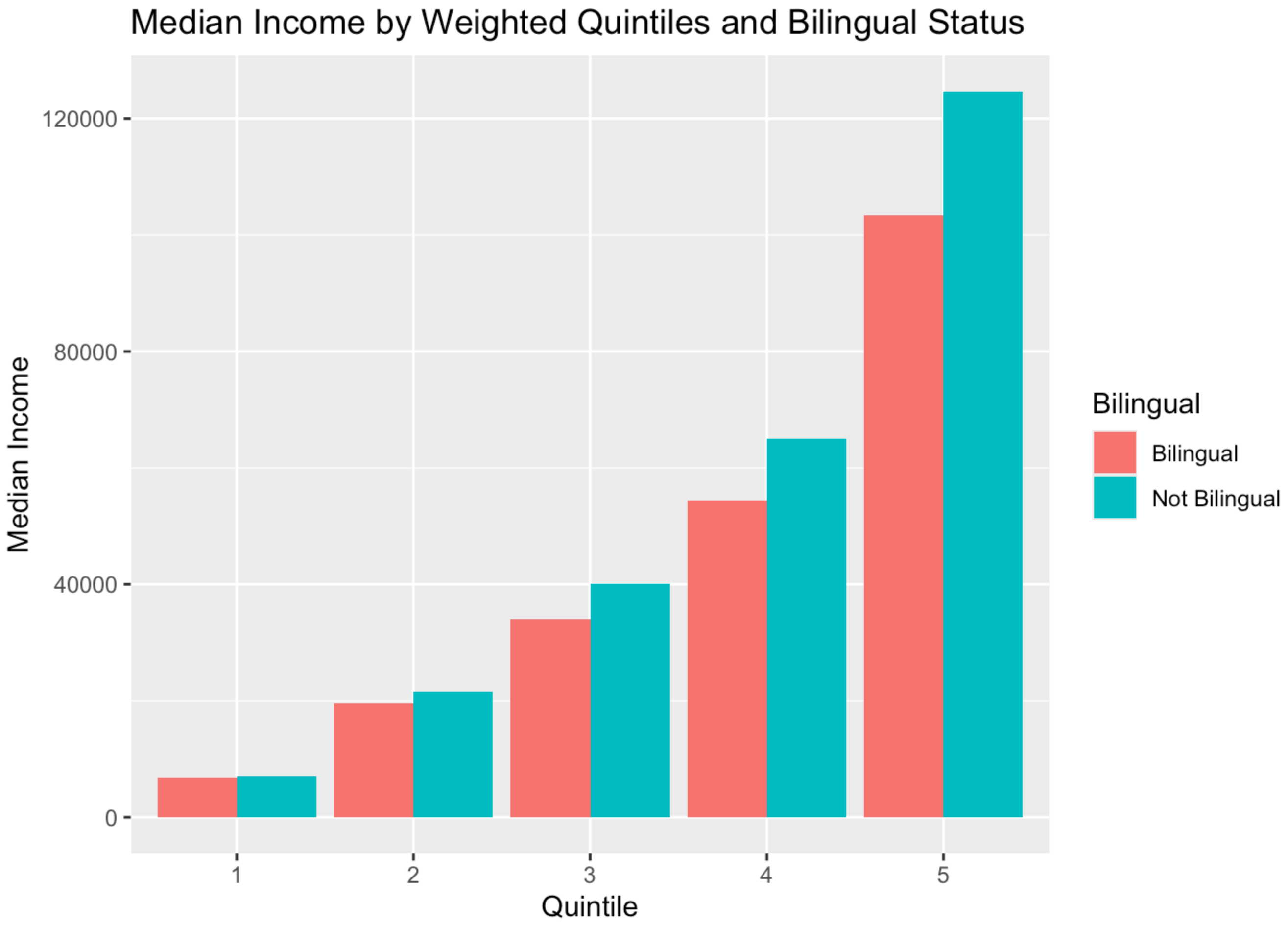
```
#maybe will have to section out the top % cause that's such a huge distribution

# Function to assign weighted quintiles
assign_weighted_quintile <- function(x, quintiles) {
  cut(x, breaks = quintiles, include.lowest = TRUE, labels = FALSE)
}

# Assign weighted quintiles to income data
income_data_quint <- income_data |>
  mutate(
    Weighted_Quintile = case_when(
      Bilingual == "Bilingual" ~ assign_weighted_quintile(INCTOT, weighted_quintiles_bilingual),
      Bilingual == "Not Bilingual" ~ assign_weighted_quintile(INCTOT, weighted_quintiles_english)
    )
  )

# Summarize data by weighted quintiles
income_summary_weighted_quintiles <- income_data_quint |>
  group_by(Bilingual, Weighted_Quintile) |>
  summarize(
    Mean_Income = weighted.mean(INCTOT, recalibrated_weight, na.rm = TRUE),
    Median_Income = weighted_median(INCTOT, recalibrated_weight),
    IQR_Income = IQR(INCTOT, na.rm = TRUE),
    .groups = 'drop'
  )

print(income_summary_weighted_quintiles)
```

```
## # A tibble: 10 × 5
##    Bilingual     Weighted_Quintile Mean_Income Median_Income IQR_Income
##    <chr>                     <int>       <dbl>         <dbl>      <dbl>
##  1 Bilingual                     1       6602.          6810       7085
##  2 Bilingual                     2      19433.         19520       6758
##  3 Bilingual                     3      33880.         33994       7917
##  4 Bilingual                     4      55039.         54350      13416
##  5 Bilingual                     5     134273.        103340      61164
##  6 Not Bilingual                 1       6809.          7000       7689
##  7 Not Bilingual                 2      21701.         21618       8612
##  8 Not Bilingual                 3      40350.         40000      10506
##  9 Not Bilingual                 4      66273.         65000      15631
## 10 Not Bilingual                 5     168094.        124645      82122
```

```
# Plot mean income by weighted quintiles
income_summary_weighted_quintiles |>
  ggplot(aes(x = as.factor(Weighted_Quintile), y = Mean_Income, fill = Bilingual)) +
  geom_bar(stat = "identity", position = "dodge") +
  ggtitle("Mean Income by Weighted Quintiles and Bilingual Status") +
  xlab("Quintile") +
  ylab("Mean Income")
```



```
# Plot median income by weighted quintiles
income_summary_weighted_quintiles |>
  ggplot(aes(x = as.factor(Weighted_Quintile), y = Median_Income, fill = Bilingual)) +
  geom_bar(stat = "identity", position = "dodge") +
  ggtitle("Median Income by Weighted Quintiles and Bilingual Status") +
  xlab("Quintile") +
  ylab("Median Income")
```



#winsorization not crazy change, will keep normal

FINAL INCOME MEDIAN

```
#reshaping final data
income_summary_wide <- income_summary_weighted_quintiles |>
  select(Bilingual, Weighted_Quintile, Median_Income) |>
  pivot_wider(names_from = Weighted_Quintile, values_from = Median_Income, names_prefix = "Quintile_")

#renaming cols for clarity
colnames(income_summary_wide) <- c("Bilingual", "Quintile_1", "Quintile_2", "Quintile_3", "Quintile_4", "Quintile_5")

print(income_summary_wide)
```

```
## # A tibble: 2 × 6
##   Bilingual     Quintile_1 Quintile_2 Quintile_3 Quintile_4 Quintile_5
##   <chr>              <dbl>      <dbl>      <dbl>      <dbl>      <dbl>
## 1 Bilingual           6810      19520      33994      54350     103340
## 2 Not Bilingual       7000      21618      40000      65000     124645
```

```
write.csv(file = "Income_Median_Personal_Income_Quintiles.csv", income_summary_wide)
```

BY EDUCATION

CODES:

CODE	Educational Level
001 & 999	N/a & missing
002	no schooling completed
10-61	grade school
062	high school or GED
65-100	one or more years of college, no degree
101	bachelors
114	masters
115	professional degree beyond bachelors
116	doctoral

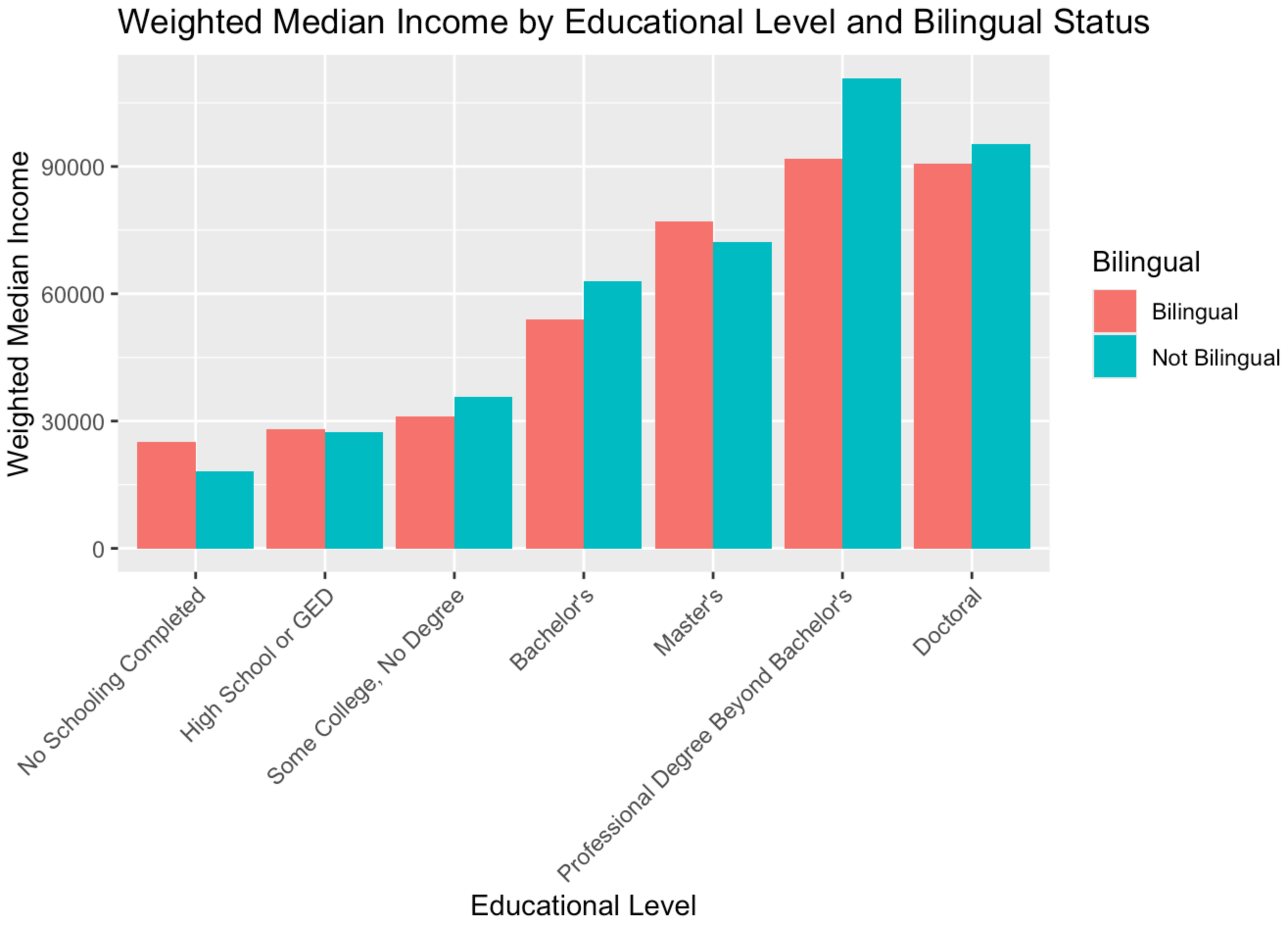
```
#calrity for the education codes
map_educational_level <- function(code) {
  if (code %in% c(1, 999)) {
    return("N/A or Missing")
  } else if (code == 2) {
    return("No Schooling Completed")
  } else if (code >= 10 & code < 62) {
    return("No High School Degree or GED")
  } else if (code >= 62 & code < 65) {
    return("High School or GED")
  } else if (code >= 65 & code <= 100) {
    return("Some College, No Degree")
  } else if (code == 101) {
    return("Bachelor's")
  } else if (code == 114) {
    return("Master's")
  } else if (code == 115) {
    return("Professional Degree Beyond Bachelor's")
  } else if (code == 116) {
    return("Doctoral")
  } else {
    return(NA)
  }
}

income_data_education <- income_data |>
  mutate(Educational_Level = sapply(EDUCD, map_educational_level))


income_data_education |>
  filter(is.na(Educational_Level))
```

```
## [1] AGE                 INCTOT              Bilingual
## [4] PERWT               SEX                 EDUCD
## [7] recalibrated_weight Educational_Level
## <0 rows> (or 0-length row.names)
```

```
education_levels <- c("N/A or Missing", "No Schooling Completed", "High School or GED",
                      "Some College, No Degree", "Bachelor's", "Master's",
                      "Professional Degree Beyond Bachelor's", "Doctoral")

#weighted medians fro each group
income_summary_education <- income_data_education |>
  mutate(Educational_Level = factor(sapply(EDUCD, map_educational_level), levels = educa
tion_levels)) |>
  group_by(Bilingual, Educational_Level) |>
  summarize(
    Median_Income = weighted_median(INCTOT, recalibrated_weight),
    .groups = 'drop'
  ) |>
  arrange(Educational_Level) |>
  filter(Educational_Level != "N/A")

#summary
print(income_summary_education)
```

```
## # A tibble: 14 × 3
##    Bilingual     Educational_Level                     Median_Income
##    <chr>         <fct>                                         <dbl>
##  1 Bilingual     No Schooling Completed                        25000
##  2 Not Bilingual No Schooling Completed                        18237
##  3 Bilingual     High School or GED                            28057
##  4 Not Bilingual High School or GED                            27280
##  5 Bilingual     Some College, No Degree                       31200
##  6 Not Bilingual Some College, No Degree                       35670
##  7 Bilingual     Bachelor's                                    54046
##  8 Not Bilingual Bachelor's                                    63000
##  9 Bilingual     Master's                                      76931
## 10 Not Bilingual Master's                                      72131
## 11 Bilingual     Professional Degree Beyond Bachelor's         91857
## 12 Not Bilingual Professional Degree Beyond Bachelor's        110803
## 13 Bilingual     Doctoral                                      90651
## 14 Not Bilingual Doctoral                                      95302
```

```
#visualize
ggplot(income_summary_education, aes(x = Educational_Level, y = Median_Income, fill = Bi
lingual)) +
  geom_bar(stat = "identity", position = "dodge") +
  ggtitle("Weighted Median Income by Educational Level and Bilingual Status") +
  xlab("Educational Level") +
  ylab("Weighted Median Income") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```


Weighted Median Income by Educational Level and Bilingual Status
Weighted Median Income
No Schooling Completed
High School or GED
Some College, No Degree
Bachelor's
Master's
Professional Degree Beyond Bachelor's
Doctoral
Educational Level
Bilingual
Bilingual
Not Bilingual