FinalProject_itmd514_05_Group4

Submitted By: Abhishek Anand, Apurva Anand, Surajit Patra

2024-12-03

        - LINEAR REGRESSION MODEL FOR NO-TOPCODED
        - PLOTS FOR NO-TOPCODED
        - LINEAR REGRESSION MODEL FOR NO-TOPCODED (MARITAL STATUS)
        - PLOTS FOR NO-TOPCODED (MARITAL STATUS)
        - LINEAR REGRESSION MODEL FOR NO-TOPCODED (AGE 1996 + CITIZENSHIP)
        - PLOTS FOR NO-TOPCODED (AGE 1996 + CITIZENSHIP)
        - LINEAR REGRESSION MODEL FOR NO-TOPCODED (RACE + USED MARIJUANA + INDUSTRY CODE)
        - PLOTS FOR NO-TOPCODED (RACE + USED MARIJUANA + INDUSTRY CODE)
- PART 4 - STORYTELLING
    - METHODOLOGY
    - HANDLING MISSING VALUES
    - HANDLING TOPCODED VARIABLES
    - DID YOU PRODUCE ANY TABLES OR PLOTS THAT YOU THOUGHT WOULD REVEAL INTERESTING TRENDS BUT DIDN'T?
    - WHAT RELATIONSHIPS DID YOU INVESTIGATE THAT DON'T APPEAR IN YOUR FINDINGS SECTION?
    - WHAT'S THE ANALYSIS THAT YOU FINALLY SETTLED ON? WHAT INCOME AND GENDER RELATED FACTORS DO YOU INVESTIGATE IN THE FINAL ANALYSIS?
- PART 5 - CONCLUSION
    - In this section you should summarize your main conclusions. You should also discuss potential limitations of your analysis and findings. Are there potential confounding variables that you didn't control for? Are the models you fit believable?
    - You should also address the following question: How much confidence do you have in your analysis? Do you believe your conclusions? Are you confident enough in your analysis and findings to present them to policy makers?

PROJECT DESCRIPTION

I. Sex-related differences: Is there a significant difference in income between men and women? Does the difference vary depending on other factors (e.g., education, marital status, criminal history, drug use, childhood household factors, profession, etc.)?

INSTALL PACKAGE, LOAD LIBRARIES AND NLSY DATASET

```
# We will require the below mentioned libraries hence remove the comment(# symbol) and f
irst install the packages if not already installed on your computer:

#install.packages("tidyverse")
#install.packages("knitr")
#install.packages("gridExtra")
#install.packages("moments")

library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.5.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.1
## ✔ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
to become errors
```

```
library(knitr)
library(ggplot2)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(moments)
```

```
# Import data file for nlsy97
nlsy <- read_csv("nlsy97.csv")
```

```
## Rows: 8984 Columns: 95
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## dbl (95): B0004600, E8043100, E8043200, E8043400, R0000100, R0069400, R00700...
##
## 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.
```

```
# Looking at the data
glimpse(nlsy)
```

```
## Rows: 8,984
## Columns: 95
## $ B0004600 <dbl> -4, -5, -4, -4, -5, 41, -5, -5, 251, -5, 280, -4, -5, -5, 334…
## $ E8043100 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0…
## $ E8043200 <dbl> -4, -4, -4, -4, -4, -4, -4, -4, -4, -4, -4, -4, -4, -4, -4, -…
## $ E8043400 <dbl> -4, -4, -4, -4, -4, -4, -4, -4, -4, -4, -4, -4, -4, -4, -4, -…
## $ R0000100 <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18…
## $ R0069400 <dbl> 1, 2, 2, 2, 2, 2, 2, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 3, 4, 1, 2…
## $ R0070000 <dbl> 1, 2, 3, 3, 2, 2, 2, 1, 1, 1, 2, 2, 2, 2, 2, 3, 2, 3, 4, 2, 2…
## $ R0323900 <dbl> -4, 3, 0, -4, 3, 4, 5, -4, 1, 1, 0, -4, 1, -4, 1, 3, -4, 0, 0…
## $ R0358900 <dbl> 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1…
## $ R0513500 <dbl> 10, -4, -4, 46, -4, -4, -4, 25, -4, -4, -4, 50, -4, 90, -4, -…
## $ R0514700 <dbl> 100, -4, -4, 100, -4, -4, -4, 100, -4, -4, -4, 95, -4, 100, -…
## $ R0514800 <dbl> 10, -4, -4, 0, -4, -4, -4, 0, -4, -4, -4, 0, -4, 0, -4, -4, 0…
## $ R0514900 <dbl> 0, -4, -4, 0, -4, -4, -4, -2, -4, -4, -4, 0, -4, 0, -4, -4, 2…
## $ R0515100 <dbl> 100, -4, -4, 100, -4, -4, -4, 100, -4, -4, -4, 95, -4, 70, -4…
## $ R0536300 <dbl> 2, 1, 2, 2, 1, 2, 1, 2, 1, 1, 2, 1, 1, 1, 2, 1, 2, 1, 1, 1, 1…
## $ R0536401 <dbl> 9, 7, 9, 2, 10, 1, 4, 6, 10, 3, 6, 10, 11, 7, 1, 2, 11, 2, 4,…
## $ R0536402 <dbl> 1981, 1982, 1983, 1981, 1982, 1982, 1983, 1981, 1982, 1984, 1…
## $ R0648900 <dbl> 0, -4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -4, 0, -4, 0, 0, 0, 0, 0…
## $ R0649100 <dbl> 0, -4, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, -4, 0, -4, 0, 0, 0, 0…
## $ R0681300 <dbl> 0, -4, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, -4, 0, -4, 0, 0, 0, 0…
## $ R0690800 <dbl> -4, -4, 1, -4, -4, -4, -4, -4, -4, -4, -4, -4, -4, -4, -4, -4…
## $ R0691200 <dbl> -4, -4, -4, -4, -4, -4, 0, -4, -4, 0, -4, -4, -4, -4, -4, -4,…
## $ R1194000 <dbl> 15, 14, 13, 15, 14, 14, 13, 15, 14, 12, 14, 15, 12, 16, 13, 1…
## $ R1200100 <dbl> 19, 19, 26, 20, 34, 21, 21, 36, 36, 36, 22, 22, 26, 28, 21, 3…
## $ R1200200 <dbl> 26, 19, 26, 33, 34, 25, 26, 36, 37, 38, 22, 22, 26, 31, 21, 3…
## $ R1201300 <dbl> 3, -4, 1, 3, 3, 1, 1, 3, 3, 3, 1, 1, -4, 1, -4, 3, 1, 1, 1, 1…
## $ R1201400 <dbl> 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8…
## $ R1204700 <dbl> -3, -4, 10200, -1500, -3, 7500, 7500, -3, -3, -3, 17500, 1750…
## $ R1205300 <dbl> 1, 2, 4, 4, 1, 4, 4, 1, 1, 1, 4, 2, 4, 1, 1, 4, 2, 4, 4, 4, 4…
## $ R1210500 <dbl> 3, -4, 3, 3, 3, 3, 3, -3, -3, -3, -3, 3, -4, -3, -4, 3, 3, 3,…
## $ R1217600 <dbl> 1, -4, 1, 1, 1, 1, 1, -3, -3, -3, -3, 1, -4, -3, -4, 1, 1, 1,…
## $ R1235800 <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…
## $ R1302400 <dbl> 16, 17, -3, 12, 12, -3, -3, 6, 6, 6, 12, -3, -3, 12, 13, 10, …
## $ R1302500 <dbl> 8, 15, 12, 12, 12, 12, 12, 12, 12, 12, 14, 12, 6, 12, 12, 11,…
## $ R1302600 <dbl> 16, 14, -4, -4, 12, -4, -4, 6, 6, 6, -4, 12, -4, 12, 13, -4, …
## $ R1302700 <dbl> 8, 15, 12, 12, 12, 12, 12, 12, 12, 12, 14, 12, 6, 12, 12, 11,…
## $ R1482600 <dbl> 4, 2, 2, 2, 2, 2, 2, 4, 4, 4, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 2…
## $ R1700500 <dbl> -4, -4, 6, -4, -4, -4, 4, -4, -4, 8, -4, -4, 5, -4, 7, -4, -4…
## $ R1701100 <dbl> -4, -4, -4, -4, -4, -4, -4, -4, -4, -4, -4, -4, -4, 5, -4, -4, -4…
## $ R2191500 <dbl> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
## $ R4908500 <dbl> -4, -4, -4, -4, -4, -4, -4, 0, 20, -4, -4, 0, -4, -5, -4, -4,…
## $ S0920000 <dbl> -4, 2, 3, -4, 2, 1, 2, -4, 4, 3, 3, -4, 2, -5, 4, 1, -5, 1, 2…
## $ S0920700 <dbl> -4, 5, 5, -4, 5, 5, 4, -4, 5, 5, 3, -4, 5, -5, 5, 5, -5, 5, 5…
## $ S2011600 <dbl> -4, -4, 4250, -4, -4, -4, -5, -4, -4, -4, -4, -4, -4, -5, -4,…
## $ S3805200 <dbl> -4, 62, -4, -4, -4, -4, -5, -4, 80, 50, -4, -4, -4, -5, -3, -…
## $ S4677200 <dbl> 160, 180, 130, 250, 185, 170, -5, 150, 180, 140, 165, 230, 12…
## $ S4685800 <dbl> 4, 4, 3, 5, 3, 4, -5, 4, 3, 2, 4, 5, 2, -5, 3, 3, 5, 3, 2, 4,…
## $ S6320000 <dbl> -4, -4, -5, -4, -4, -4, -4, -4, -4, -4, -4, -4, -5, -4, -5, -4, -…
## $ T0026000 <dbl> -4, -5, -5, 1, -4, -4, -5, -4, -4, -4, 1, -4, -4, -5, -5, -4,…
## $ T1069100 <dbl> -4, -5, -5, -4, -4, -4, -5, -4, -4, 4, -4, 3, 3, -5, -5, 3, -…
```

```
## $ T1069101 <dbl> -4, -5, -5, -4, -4, -4, -5, -4, -4, 4, -4, 1, 3, -5, -5, 3, -…
## $ T1069102 <dbl> -4, -5, -5, -4, -4, -4, -5, -4, -4, 4, -4, 3, 4, -5, -5, 1, -…
## $ T1069103 <dbl> -4, -5, -5, -4, -4, -4, -5, -4, -4, 2, -4, 2, 1, -5, -5, 3, -…
## $ T3611600 <dbl> -4, -4, 1, 2, -4, 1, -4, -4, -4, -4, 1, -4, -4, -5, -4, 0, -5…
## $ T6650100 <dbl> -4, -4, -4, -4, -4, -4, -4, -4, 2, -5, 3, -4, -4, -4, 1, -4, …
## $ T6656700 <dbl> 50000, 81000, 150250, -3, 130000, 55000, 14766, 66750, 110000…
## $ T6656900 <dbl> 9, 1, 3, 4, 3, 4, 2, 2, 3, -5, 2, 3, 2, 2, 3, 2, -5, 3, 3, -5…
## $ T6657000 <dbl> 2, 0, 1, 2, 1, 1, 0, 0, 0, -5, 0, 0, 0, 0, 0, 0, -5, 1, 1, -5…
## $ T6657100 <dbl> 2, 0, 1, 1, 1, 1, 0, 0, 0, -5, 0, 0, 0, 0, 0, 0, -5, 0, 0, -5…
## $ T6657300 <dbl> 4, 2, 3, 2, 2, 2, 1, 5, 5, -5, 2, 1, 0, -3, 2, 3, -5, 1, -3, …
## $ T6663100 <dbl> -4, -4, 1, 2, 1, 1, -4, -4, -4, -5, 0, -4, -4, 0, -4, 0, -5, …
## $ T6767000 <dbl> -4, -4, -4, -4, -4, -4, -4, -4, 18, -5, 15, -4, -4, 11, 14, -…
## $ T7635600 <dbl> 5, 5, 5, 5, 5, -2, -1, 5, 5, -5, 5, 5, 5, 5, 4, 6, -5, 5, 5, …
## $ T7635700 <dbl> 7, 7, -2, 1, 6, -2, -1, 7, 11, -5, 4, 6, 5, 7, 11, 6, -5, 8, …
## $ T7635800 <dbl> 155, 175, 140, 170, 180, -2, -1, 150, 195, -5, 176, 270, 147,…
## $ T7638800 <dbl> 0, 0, 1, 0, 1, 0, -2, 1, 1, -5, 1, 0, 0, 0, 0, 1, -5, 1, 1, -…
## $ T7639200 <dbl> 0, 1, 1, 1, 1, 0, -1, 1, 1, -5, 1, 1, 1, 0, 1, 1, -5, 1, 1, -…
## $ T7639800 <dbl> -4, -4, -4, -4, -4, -4, -4, 0, -4, -5, -4, -4, -4, -4, -4, -1…
## $ T7640000 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, -5, 0, 0, 0, 0, 0, 1, -5, 0, 0, -5…
## $ T7640300 <dbl> 4, 3, 4, 3, 4, 3, -2, 4, 4, -5, 4, 5, 3, 2, 3, 5, -5, 2, 3, -…
## $ T7640400 <dbl> 1, 4, 1, 1, 3, 1, -2, 1, 1, -5, 1, 4, 3, 2, 4, 1, -5, 2, 3, -…
## $ T7731100 <dbl> 4470, 9470, 7670, 8180, 9470, 7860, -4, 4290, 7860, -5, 8190,…
## $ U0014800 <dbl> -4, -4, -5, 1, 2, 2, -4, -4, -4, -5, 0, -4, -4, 0, -5, 0, -5,…
## $ U1718000 <dbl> -5, 7680, 690, 6690, 9470, 5170, -5, -5, 7860, 6890, 8190, 44…
## $ U1719400 <dbl> -5, 3910, 620, 4840, 3740, 4760, -5, -5, 230, 1000, 5860, 485…
## $ U1852400 <dbl> -5, 0, 1, 0, 1, 1, -5, -5, 1, 1, 1, 0, 0, 0, -5, 0, -5, 0, 0,…
## $ U2857000 <dbl> -5, 1, 1, 1, 1, 1, -5, -5, 1, 1, 1, 1, 1, 0, -5, 1, -5, 1, 0,…
## $ U2857200 <dbl> -5, 116000, -1, 45000, 125000, -2, -5, -5, 59000, 75000, 3600…
## $ U2857300 <dbl> -5, -4, -1, -4, -4, 3, -5, -5, -4, -4, -4, -4, -4, -4, -5, -4…
## $ U2858100 <dbl> -5, -4, -4, -4, 1, 1, -5, -5, 1, 1, 1, 1, -4, -4, -5, -4, -5,…
## $ U2858500 <dbl> -5, -4, -4, -4, 75000, -2, -5, -5, 100000, 100000, 41000, -2,…
## $ U2858600 <dbl> -5, -4, -4, -4, -4, 4, -5, -5, -4, -4, -4, 5, -4, -4, -5, -4,…
## $ Z9031500 <dbl> -4, 1, -4, 1, -4, -4, -4, -4, -4, 1, -4, -4, -3, -4, 1, -4, -…
## $ Z9033700 <dbl> 4, 4, 2, -4, 2, 2, -3, 5, 4, -4, -4, -4, -4, -4, -3, 3, -4, -…
## $ Z9033900 <dbl> 3, 4, 2, -4, 6, 2, -3, 5, 2, -4, -4, -4, -4, -4, -3, 3, -4, -…
## $ Z9034100 <dbl> -4, -4, -4, -4, -4, -4, -4, -4, -4, -3, 4, -4, -4, 6, -3, -4,…
## $ Z9048900 <dbl> 2500, -3, 4250, -3, 11500, 7500, -3, 2500, 2500, 0, 3500, 750…
## $ Z9049000 <dbl> 42500, -3, 800, 16600, 589000, 3100, 3170, 4500, -1500, 0, 47…
## $ Z9049800 <dbl> 0, -3, 2550, 4100, 31300, 600, 0, 1000, 0, 5000, 8, 0, 7016, …
## $ Z9050100 <dbl> 0, 2000, 0, 0, 0, 0, 0, 0, 0, 0, 14000, 0, 500, -4, 0, 0, 0, …
## $ Z9121900 <dbl> 34500, 32500, -3, -3, 445000, 5000, 600, -17500, -30000, 1340…
## $ Z9122000 <dbl> 0, 0, 0, 0, 425000, 0, 0, 0, 0, 0, 0, 0, 0, 0, -4, 0, -4, 0, …
## $ Z9122200 <dbl> 6, 6, 6, 6, 1, 6, 6, 6, 6, 6, 6, 6, 6, 6, -4, 6, -4, 6, 6, 6,…
## $ Z9122300 <dbl> 30000, 45000, -3, 0, 144000, 3500, 0, 8000, 7500, 185000, 300…
## $ Z9122500 <dbl> 18000, 35000, 19300, 0, 9000, 1000, 0, 63000, 40000, 96500, 6…
```

PART 1- VARIABLES(SELECTION) AND Exploratory Data Analysis (EDA)

```
# Change column names to meaningful
colnames(nlsy)[colnames(nlsy) == "R0536300"] <- "gender"
colnames(nlsy)[colnames(nlsy) == "R0536402"] <- "birth_year"
colnames(nlsy)[colnames(nlsy) == "R0690800"] <- "is_sad_depressed_unhappy_f"
colnames(nlsy)[colnames(nlsy) == "R1201400"] <- "current_enrollment_status"
colnames(nlsy)[colnames(nlsy) == "U2857200"] <- "total_income_2016"
colnames(nlsy)[colnames(nlsy) == "R1482600"] <- "race"
colnames(nlsy)[colnames(nlsy) == "T6767000"] <- "highest_grade_completed"
colnames(nlsy)[colnames(nlsy) == "R1194000"] <- "age_1996"
colnames(nlsy)[colnames(nlsy) == "R1201300"] <- "citizenship"
colnames(nlsy)[colnames(nlsy) == "R0358900"] <- "used_marijuana"
colnames(nlsy)[colnames(nlsy) == "T7731100"] <- "industry_code"
colnames(nlsy)[colnames(nlsy) == "U1852400"] <- "marital_status"
colnames(nlsy)[colnames(nlsy) == "E8043100"] <- "total_num_incarcerations"
colnames(nlsy)[colnames(nlsy) == "T6650100"] <- "college_type"
colnames(nlsy)[colnames(nlsy) == "S0920700"] <- "trustful_or_not"
```

```
# Creating a df with renamed columns
nlsy_df <- nlsy[, c(
  "gender",
  "birth_year",
  "is_sad_depressed_unhappy_f",
  "current_enrollment_status",
  "total_income_2016",
  "race",
  "highest_grade_completed",
  "age_1996",
  "citizenship",
  "used_marijuana",
  "industry_code",
  "marital_status",
  "total_num_incarcerations",
  "college_type",
  "trustful_or_not"
)]
```

GENDER

```
# Check the unique values and if any missing values are present
unique(nlsy_df$gender)
```

```
## [1] 2 1
```

```
table(nlsy_df$gender, useNA = "ifany")
```

```
##
##    1    2
## 4599 4385
```

As we can see here we have a total of 4599 male and 4385 females which are denoted by 1 and 2 factor levels.

```
# Convert gender to a factor with Male and Female labels
nlsy_df <- nlsy_df %>%
  mutate(gender = factor(gender,
    levels = c(1, 2),
    labels = c("Male", "Female")
  ))
```

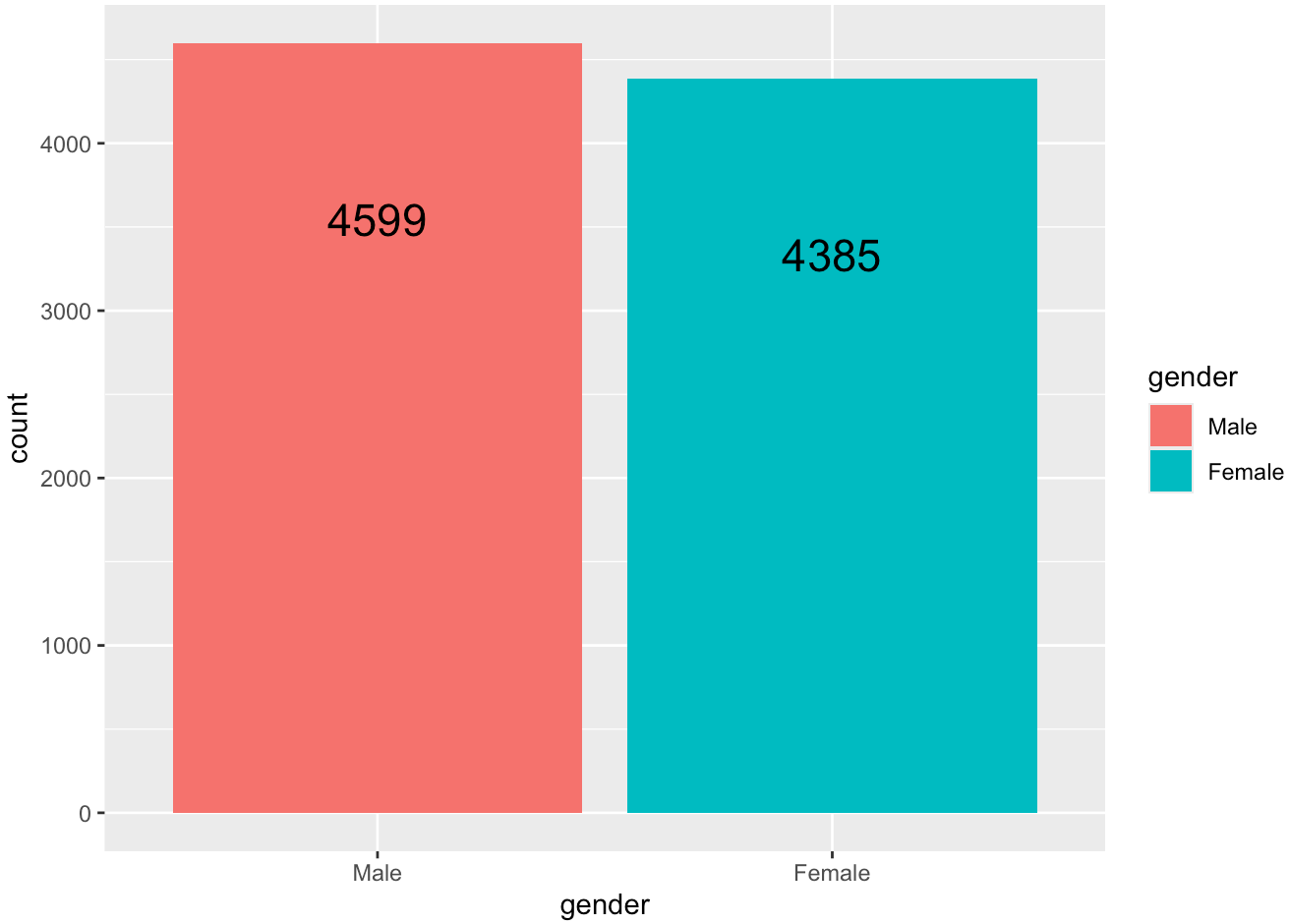
Renaming the factor level 1 and 2 in the gender column to Male and Female

```
# Verify the count of Male and Female in the data
nlsy_df %>%
  count(gender, sort = TRUE) %>%
  rename(count = n)
```

```
## # A tibble: 2 × 2
##   gender count
##   <fct>  <int>
## 1 Male    4599
## 2 Female  4385
```

```
# Gender Summary
ggplot(nlsy_df, aes(x = gender, fill = gender)) +
  geom_bar() +
  geom_text(aes(label = ..count..), stat = "count", vjust = 6, size = 6)
```

```
## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.
## ℹ Please use `after_stat(count)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
labs(title = "Gender Distribution Plot", x = "Gender", y = "Count")
```

```
## $x
## [1] "Gender"
##
## $y
## [1] "Count"
##
## $title
## [1] "Gender Distribution Plot"
##
## attr(,"class")
## [1] "labels"
```

From the plot above we can see the population distribution contains more number of Male(4599) and less Females(4385).

BIRTH YEAR

```
# Checking if birth_year column requires cleaning | unique values: 1983 1984 1980 1981 1982
# This variable birth_year does not contain any negative values and hence no cleaning is required for it. We have summarized the variable which tells minimum value expected is 1980, 1st quartile is 1981, median is 1982, 3rd quartile is 1983 and max is 1984.
sort(unique(nlsy_df$birth_year))
```

```
## [1] 1980 1981 1982 1983 1984
```

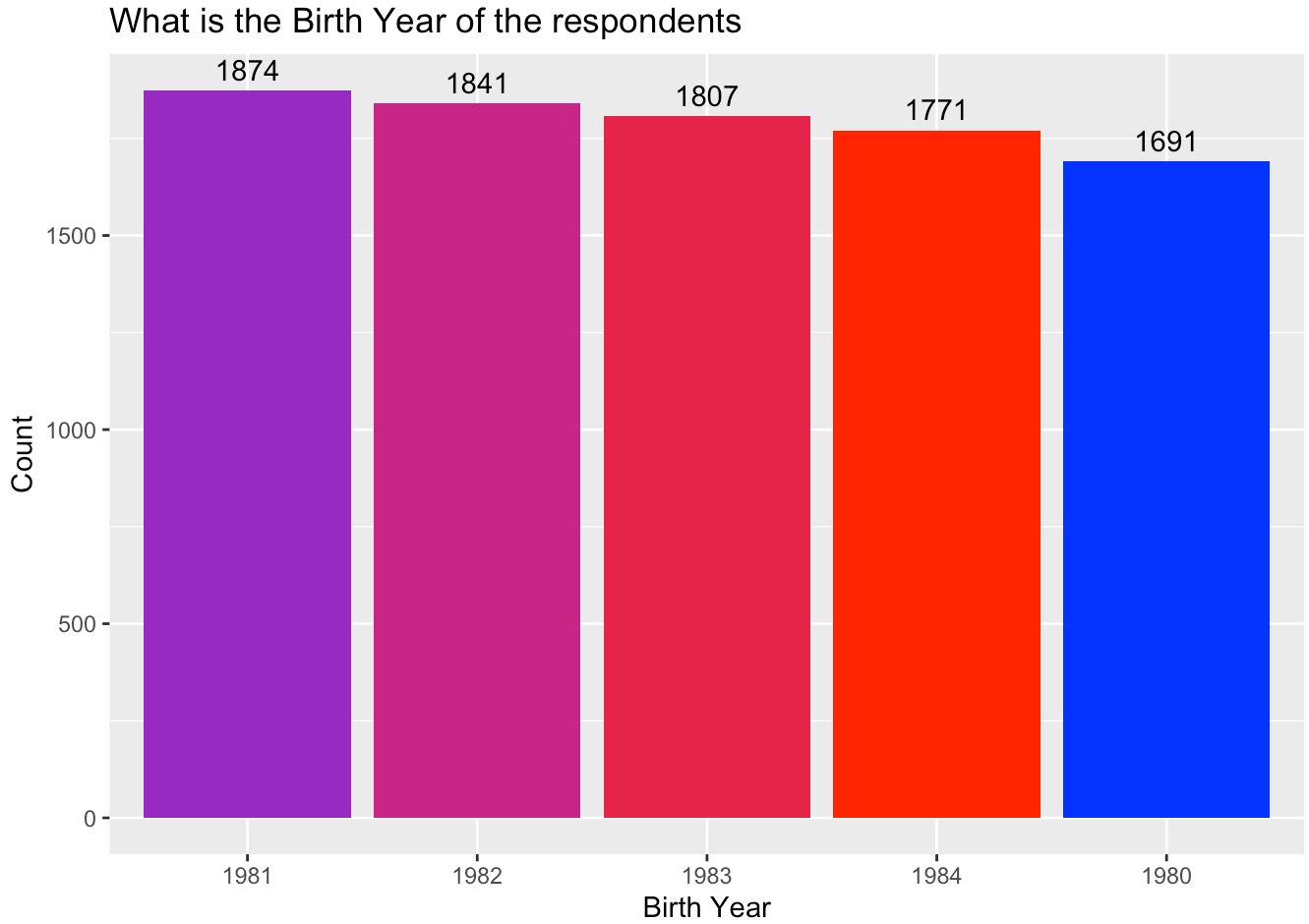
```
summary(nlsy_df$birth_year)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1980    1981    1982    1982    1983    1984
```

```
# This will give us an idea pf count of responses for each type for Birth year.
# The tibble shows the range of birth year in the survey responses and count of respondents having those birth year, like there are 1874 people having 1981 as the birth year, 1841 people having 1982 as the birth year, 1807 responses for 1983 birth year, 1771 responses for 1984 birth year, and 1691 responses for 1980 birth year.
nlsy_df %>%
  count(birth_year, sort = TRUE) %>%
  rename(count = n)
```

```
## # A tibble: 5 × 2
##   birth_year count
##        <dbl> <int>
## 1       1981  1874
## 2       1982  1841
## 3       1983  1807
## 4       1984  1771
## 5       1980  1691
```

```
# Sort the Birt Year
birth_year_sorted <- nlsy_df %>%
  count(birth_year, sort = TRUE)

# Plotting birth year and its concentration
ggplot(birth_year_sorted, aes(x = reorder(birth_year, -n), y = n, fill = birth_year)) +
  geom_bar(stat = "identity",show.legend = FALSE) +
  geom_text(aes(label = n), vjust = -0.5) +
  labs(
    title = "What is the Birth Year of the respondents",
    x = "Birth Year",
    y = "Count"
  ) +
  scale_fill_gradient(low = "blue", high = "red", name = "Birth Year")
```



The barplot shows the range of birth year and count of respondents having those birth year. For example there were 1874 respondents having 1981 as the birth year, 1841 respondents having 1982 as the birth year, 1807 responses for 1983 birth year, 1771 responses for 1984 birth year, and 1691 responses for 1980 birth year. The highest number of respondents(1874) are having birth year 1981 and the lowest number of respondents(1691) are having birth year as 1980 ## IS SAD DEPRESSED UNHAPPY F

```
# Replace -3, -4, and -5 with NA in the is_sad_depressed_unhappy_f column
# is_sad_depressed_unhappy_f is a primary variable captured in survey year 1997 to see h
ow respondents respond to the question if they are sad, unhappy, or depressed. This vari
able has 1070 responses for 0 which signifies 'Not true', 472 responses for 1 which sign
ifies 'Sometimes true', and 42 responses for which signifies 'Often true'. Out of total
8984, there are 1584 valid responses and some have negative values like -1(2),-2(2),-4
(7396),-5 which signify refusal, don't know, valid skip, non-interview, respectively. He
re we are replacing -3,-4,-5 with NA and then filtering nlsy_df with all NA for this var
iable. This means that 7400/8984 data is NA which constitutes a huge portion of data and
may introduce biases.

nlsy_df <- nlsy_df %>%
  mutate(
    is_sad_depressed_unhappy_f = ifelse(is_sad_depressed_unhappy_f %in% c(-1, -2, -3, -
4, -5), NA, is_sad_depressed_unhappy_f)
  )

# Verify the changes
table(nlsy_df$is_sad_depressed_unhappy_f, useNA = "always")
```

```
##
##    0    1    2 <NA>
## 1070  472   42 7400
```

```
# Omit rows where is_sad_depressed_unhappy_f is NA
# nlsy_df <- nlsy_df %>%
#  filter(!is.na(is_sad_depressed_unhappy_f))
sort(unique(nlsy_df$is_sad_depressed_unhappy_f))
```

```
## [1] 0 1 2
```

```
glimpse(nlsy_df)
```

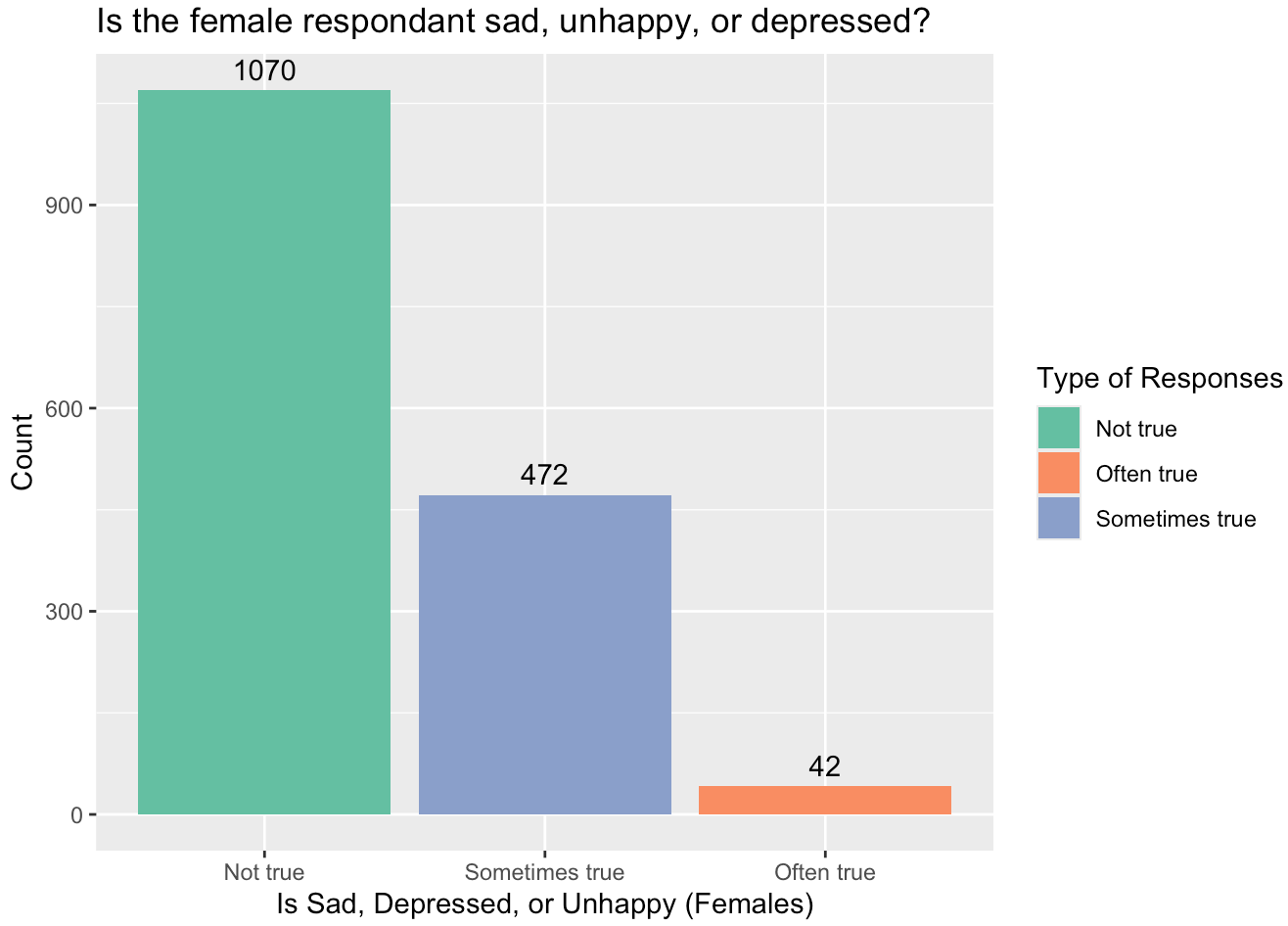
```
## Rows: 8,984
## Columns: 15
## $ gender                     <fct> Female, Male, Female, Female, Male, Female,…
## $ birth_year                 <dbl> 1981, 1982, 1983, 1981, 1982, 1982, 1983, 1…
## $ is_sad_depressed_unhappy_f <dbl> NA, NA, 1, NA, NA, NA, NA, NA, NA, NA, …
## $ current_enrollment_status  <dbl> 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8…
## $ total_income_2016          <dbl> -5, 116000, -1, 45000, 125000, -2, -5, -5, …
## $ race                       <dbl> 4, 2, 2, 2, 2, 2, 2, 4, 4, 4, 2, 2, 2, 2, 2…
## $ highest_grade_completed    <dbl> -4, -4, -4, -4, -4, -4, -4, -4, 18, -5, 15,…
## $ age_1996                   <dbl> 15, 14, 13, 15, 14, 14, 13, 15, 14, 12, 14,…
## $ citizenship                <dbl> 3, -4, 1, 3, 3, 1, 1, 3, 3, 3, 1, 1, -4, 1,…
## $ used_marijuana             <dbl> 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
## $ industry_code              <dbl> 4470, 9470, 7670, 8180, 9470, 7860, -4, 429…
## $ marital_status             <dbl> -5, 0, 1, 0, 1, 1, -5, -5, 1, 1, 1, 0, 0, 0…
## $ total_num_incarcerations   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
## $ college_type               <dbl> -4, -4, -4, -4, -4, -4, -4, -4, 2, -5, 3, -…
## $ trustful_or_not            <dbl> -4, 5, 5, -4, 5, 5, 4, -4, 5, 5, 3, -4, 5, …
```

```
# This question 'Is the respondant sad, unhappy, or depressed?' shows a tibble 4x2 where we have marked 0 as Not true, 1 as Sometimes true, 2 as Often true, and for all negative values we have used NA. The table below shows the count of each kind of responses.
nlsy_df <- nlsy_df %>%
  mutate(is_sad_depressed_unhappy_f = case_when(
    is_sad_depressed_unhappy_f == 0 ~ "Not true",
    is_sad_depressed_unhappy_f == 1 ~ "Sometimes true",
    is_sad_depressed_unhappy_f == 2 ~ "Often true",
    is_sad_depressed_unhappy_f < -2 ~ NA_character_
  ))
```

```
# This will give us an idea pf count of responses for each type of responses.
nlsy_df %>%
  count(is_sad_depressed_unhappy_f, sort = TRUE) %>%
  rename(count = n)
```

```
## # A tibble: 4 × 2
##   is_sad_depressed_unhappy_f count
##   <chr>                      <int>
## 1 <NA>                        7400
## 2 Not true                    1070
## 3 Sometimes true               472
## 4 Often true                    42
```

```
# Omit rows where is_sad_depressed_unhappy_f is NA
Is_Sad_plot_df <- nlsy_df %>%
  filter(!is.na(is_sad_depressed_unhappy_f))
# Plotting is_sad_depressed_unhappy_f and count
ggplot(data = Is_Sad_plot_df, aes(x = reorder(is_sad_depressed_unhappy_f,-table(is_sad_d
epressed_unhappy_f)[is_sad_depressed_unhappy_f]), fill = is_sad_depressed_unhappy_f)) +
  geom_bar() +
  geom_text(stat = "count", aes(label = ..count..), vjust = -0.5) +
  labs(
    title = "Is the female respondant sad, unhappy, or depressed?",
    x = "Is Sad, Depressed, or Unhappy (Females)", y = "Count",
  ) +
  scale_fill_brewer(palette = "Set2", name = "Type of Responses")
```



ENROLLMENT STATUS

```
# Check for unique values and if any missing values are present
unique(nlsy_df$current_enrollment_status)
```

```
## [1]  8  1  9  3 -2 10  2
```

```
table(nlsy_df$current_enrollment_status, useNA = "ifany")
```

```
##
##   -2    1    2    3    8    9   10
##    2  226    3    4 8742    5    2
```

There exist 2 values indicating Don't Know, denoted by -2

```
# Changing factor levels for current_enrollment_status into labels
nlsy_df <- nlsy_df %>%
  mutate(current_enrollment_status = case_when(
    current_enrollment_status == 1 ~ "Not enrolled, no high school degree, no GED",
    current_enrollment_status == 2 ~ "Not enrolled, GED",
    current_enrollment_status == 3 ~ "Not enrolled, high school degree",
    current_enrollment_status == 4 ~ "Not enrolled, some college",
    current_enrollment_status == 5 ~ "Not enrolled, 2-year college graduate",
    current_enrollment_status == 6 ~ "Not enrolled, 4-year college graduate",
    current_enrollment_status == 7 ~ "Not enrolled, graduate degree",
    current_enrollment_status == 8 ~ "Enrolled in grades 1-12, not a high school graduat
e",
    current_enrollment_status == 9 ~ "Enrolled in a 2-year college",
    current_enrollment_status == 10 ~ "Enrolled in a 4-year college",
    current_enrollment_status == 11 ~ "Enrolled in a graduate program",
    current_enrollment_status < 0 ~ NA_character_
  ))
```

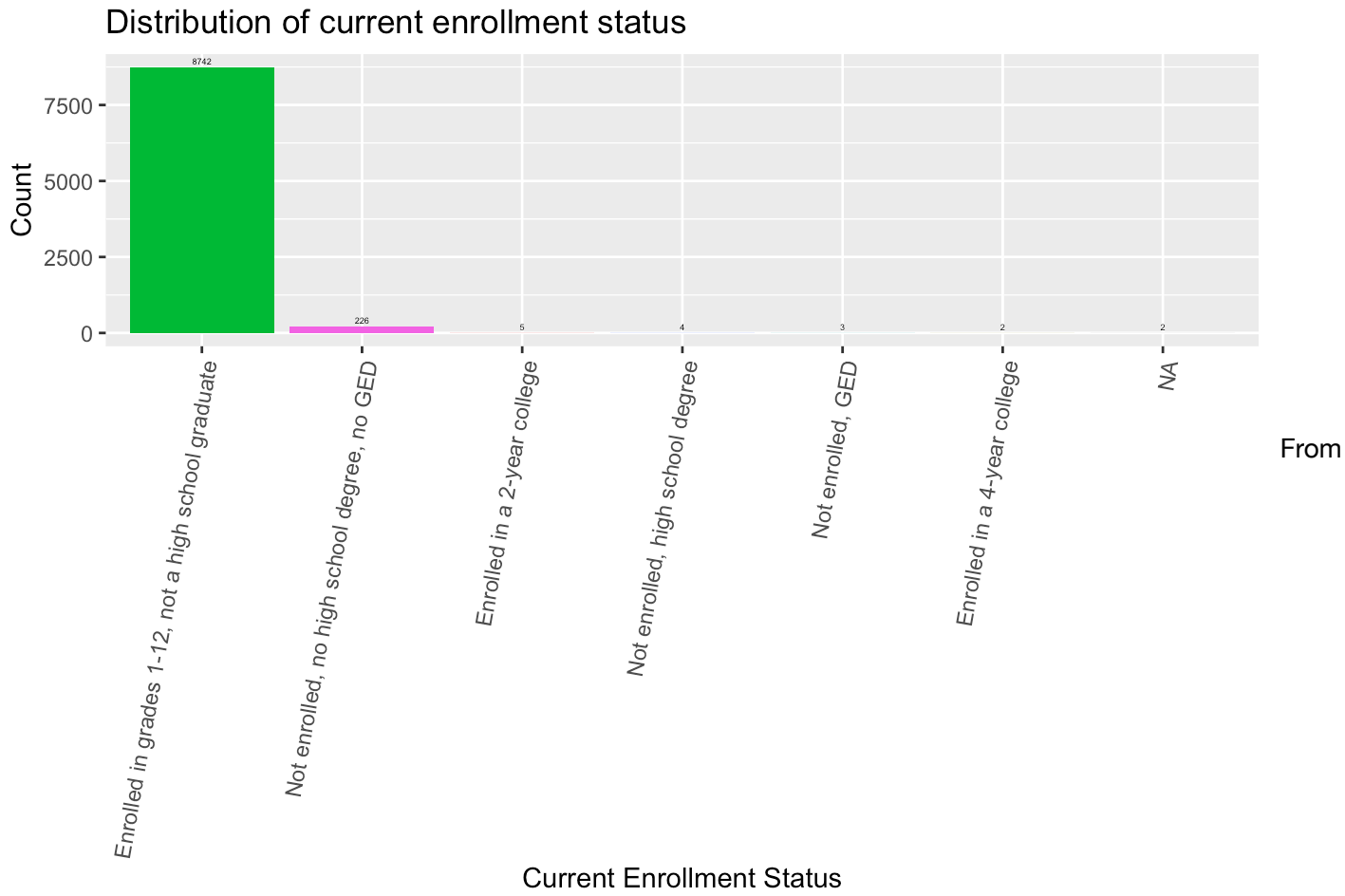
In the above we have lebelled the enrollment statuses and also changed the negative values with NA the count of which is only 2 rows which is 2/8984 rows and ~0.02% of total data.

```
# Verify the count of different enrollment statuses
nlsy_df %>%
  count(current_enrollment_status, sort = TRUE) %>%
  rename(count = n)
```

```
## # A tibble: 7 × 2
##   current_enrollment_status                          count
##   <chr>                                              <int>
## 1 Enrolled in grades 1-12, not a high school graduate  8742
## 2 Not enrolled, no high school degree, no GED           226
## 3 Enrolled in a 2-year college                            5
## 4 Not enrolled, high school degree                        4
## 5 Not enrolled, GED                                       3
## 6 Enrolled in a 4-year college                            2
## 7 <NA>                                                    2
```

From the table above we observe that 8742 of the respondants said that their current enrollment status is "Enrolled in grades 1-12, not a high school graduate" and this is followed by the ones who responsded "Not enrolled, no high school degree, no GED" which is 226.

```
# Create the plot with proper count calculation
ggplot(nlsy_df, aes(x = reorder(current_enrollment_status, -table(current_enrollment_sta
tus)[current_enrollment_status]), fill = current_enrollment_status)) +
  geom_bar(stat = "count", show.legend = FALSE) +
  geom_text(stat = "count", aes(label = ..count..), vjust = -0.5, size = 1.2) +
  labs(title = "Distribution of current enrollment status", x = "Current Enrollment Stat
us", y = "Count") +
  theme(axis.text.x = element_text(angle = 80, vjust = 1, hjust = 1))
```



From the bar plot above we observe that 8742 of the respondents confirmed their current enrollment status is “Enrolled in grades 1-12, not a high school graduate” and this is followed by the ones who responded “Not enrolled, no high school degree, no GED” which is 226.

TOTAL INCOME 2016

```
# Check the data
unique(nlsy_df$total_income_2016)
```

```
##   [1]     -5 116000     -1  45000 125000     -2  59000  75000  36000  63000
##  [11]  35000     -4  20000  55000  40000  30000  80000  15000  85000  12000
##  [21]  50000  62000  11500  74000  27000  32000  50315 101000  17000  88000
##  [31] 140000  70000 120000  73000  86000  87000  52000  64000 103000 100000
##  [41]  90000  43000  57000  77000 235884  69000  25000  53000  51000  34000
##  [51]  65000  18000  24000 135000  28000  72000  26000  41450  93000  68000
##  [61]  83000 105000  60000  78000  76000  47000  31000   5000  48000   1000
##  [71]  10000  95000  49000  44000  22000  46000  85600 119630  54000 110000
##  [81]  37000   3000  58000  33000  92951  71000  38000   4500 115000 112000
##  [91]   6000      0   3500   9300  98000  72800  38500  23000  13000  19860
## [101]  92000    500  33600  89000  93580 121000  42798  33200  42000  55500
## [111]  41000   9700  27295   4000  39000   4625   7000  16000  34500  61000
## [121]  21000 118000 130000  96000  97000  78861   8000  20905 106000  14000
## [131]  18200    672 107000  56000   1900 112700 127000 124000  67000    350
## [141]   2885  49400  57500 143000  97885  91812    200  41300  29000 100500
## [151]  62500    720  96500   9800  11000   4935  14617  91000   2000   5200
## [161] 119000    300 104000  49845  12500  18720 108000  26500  76196   4588
## [171]  62386   9000   2500  81000  84000    250  66000  88400  50715   9600
## [181]  98500 128000  47500 113000  30346  63582  71344  82000  47300  74674
## [191]  78500  19000    150  91870 122000  22780 109000  52500  94000  87500
## [201]  99000  33800  30600 102000  99500  83200  30500  30347  31500   1200
## [211]    600  72500  53100  11400  54500  34880 114000  74995  41497  45700
## [221]    173  40277  31200  22340   1500  35280  98400  22500  18500 145000
## [231]  73033  26636  15600  42500  47850  10500   4600  65500  38900  28854
## [241]  42400  21668  92866  49426  16110  10800   6500  51300  56632   6600
## [251]  14500   3200  25050  28500 124847  50543 139000    400  14100  26900
## [261]  54599  20400 146000     20  73779  24164  63200 149000  48500  79973
## [271]  18556 123800   8240  47380  79000    700   2200  27580  43750  28898
## [281]  30400  38029  17400  44400   8500  50800  16287  34600  12300  16800
## [291]   7500  19200  30490  19689  69500  61449  15500   5500  21120   7200
## [301]  58500  32500     -3  42075  61500  43500  30597  33632   2600    800
## [311]  21600   2400  28800  37500  35500  11967   3600     48  13292  46200
## [321]      2  42900  55341  35800  22800  17500 117000  29600  19241   9420
## [331]  56500  64500  33195   8900  58011  11200  44006  30617  17472    680
## [341]  25500  24624   1700  43829  59896  84665  41245  15075  76500  31139
## [351]  80500  17885  48398  46500   1800  49500   1600  39800  25376  29500
## [361]  36400  84334   4800  16500 132000  38175  19500  68250  22300
```

```
# Handle negative values and replace with NA
nlsy_df <- nlsy_df %>%
  mutate(total_income_2016 = ifelse(total_income_2016 < 0, NA, total_income_2016))
```

```
# Summarize the total_income_2016 column
summary(nlsy_df$total_income_2016)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##       0   25000   40000   49477   62000  235884    3893
```

```
# Count the number of missing values
sum(is.na(nlsy_df$total_income_2016))
```

```
## [1] 3893
```

```
# Calculate the percentage of missing values
sum(is.na(nlsy_df$total_income_2016)) / nrow(nlsy_df) * 100
```

```
## [1] 43.33259
```

From the above we can see that the missing values(NA) are 3893 which is 43.33% of the total data which might lead to bias. The mean is 49447 which is likely lower than the expected value due to the topcoding for the top 2%. The maximum income is 235884 which is also not the actual highest income but represents the average of the topcoded values. The median income is $40,000, and the interquartile range (IQR) spans from $25,000 to $62,000.

```
# Number of Topcoded values(repondents with higher income)
topcoded <- with(nlsy_df, sum(total_income_2016 == max(total_income_2016, na.rm = TRUE),
na.rm = TRUE))
topcoded
```

```
## [1] 121
```

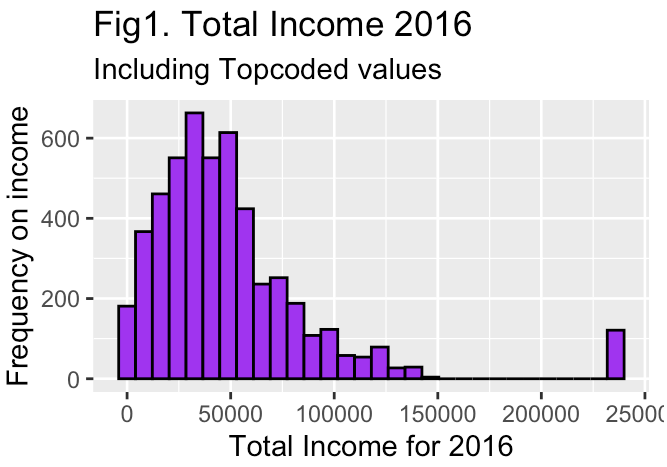
There are 121 topcoded values present with the income of $235,884. These 121 respondants are among the top 2% of the earners and are given an average value of $235,884

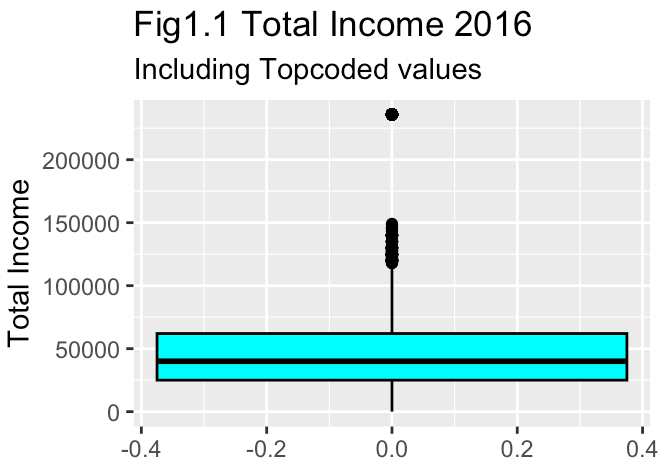
```
# Summarize the total_income_2016 column excluding topcoded values and removing NA rows
nlsy_no_topcoded <- nlsy_df %>%
  filter(total_income_2016 != max(total_income_2016, na.rm = TRUE))
summary(nlsy_no_topcoded$total_income_2016)
```

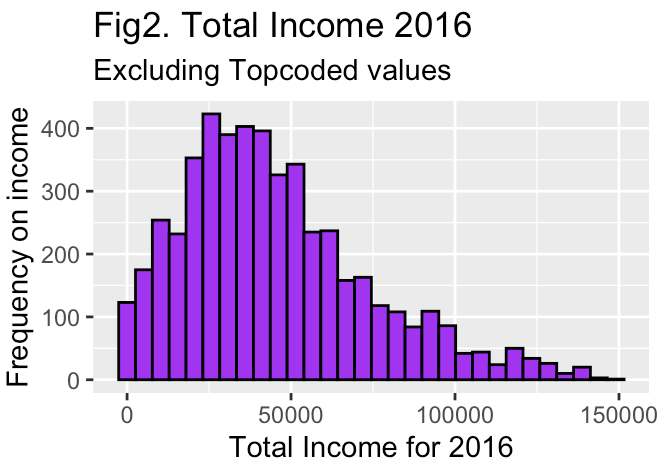
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0   25000   40000   44939   60000  149000
```

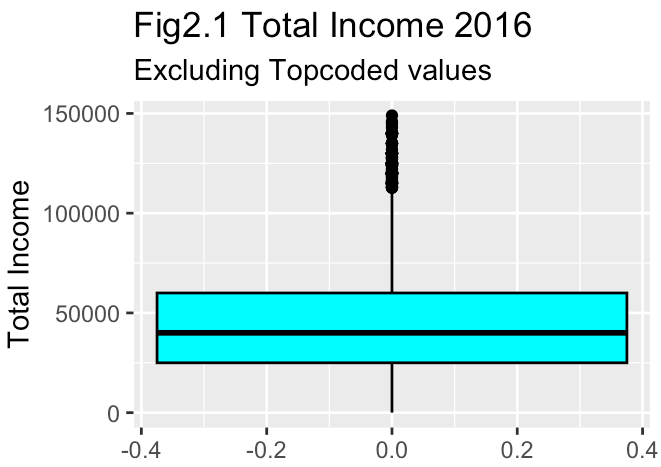
We have removed the respondents response who were hiding info or their responses could not be captured for their income in 2016. After removing the topcoded values it lowers the mean income from $49,477 to $44,939. The maximum income also drops from $235,884 to $149,000, which is the actual cutoff for the topcoded group. The median and IQR remain largely unchanged, showing that the central distribution of income is not significantly affected by the exclusion of topcoded values.

```
# Visualizing income distribution histogram including the topcoded values
plot1 <- ggplot(nlsy_df, aes(x = total_income_2016)) +
  geom_histogram(bins = 30, fill = "purple", color = "black", na.rm = TRUE) +
  labs(
    title = "Fig1. Total Income 2016",
    subtitle = "Including Topcoded values",
    x = "Total Income for 2016",
    y = "Frequency on income"
  )

# Visualizing income distribution boxplot  including the topcoded values
plot2 <- ggplot(nlsy_df, aes(y = total_income_2016)) +
  geom_boxplot(fill = "cyan", color = "black", na.rm = TRUE) +
  labs(
    title = "Fig1.1 Total Income 2016",
    subtitle = "Including Topcoded values",
    y = "Total Income"
  )

# Visualizing income distribution histogram excluding the topcoded values
nlsy_no_topcoded <- nlsy_df %>%
  filter(total_income_2016 != max(total_income_2016, na.rm = TRUE))

plot3 <- ggplot(nlsy_no_topcoded, aes(x = total_income_2016)) +
  geom_histogram(bins = 30, fill = "purple", color = "black", na.rm = TRUE) +
  labs(
    title = "Fig2. Total Income 2016",
    subtitle = "Excluding Topcoded values",
    x = "Total Income for 2016",
    y = "Frequency on income"
  )

# Visualizing income distribution boxplot excluding the topcoded values
plot4 <- ggplot(nlsy_no_topcoded, aes(y = total_income_2016)) +
  geom_boxplot(fill = "cyan", color = "black", na.rm = TRUE) +
  labs(
    title = "Fig2.1 Total Income 2016",
    subtitle = "Excluding Topcoded values",
    y = "Total Income"
  )

grid.arrange(plot1, plot2, plot3, plot4, ncol = 2)
```









In the above we see that the income is topcoded and the value for the top 2% earners is replaced with average income of this group. In Fig1. the histogram is right skewed which means that there are very few high income earners but most earners make less and this skewness shows that the majority of the population earns much less than this topcoded value.

In Fig1.1 the boxplot for the same data shows that the median income is around $40,000, and the Interquartile Range (IQR) spans from $25,000 to $62,000. This means that 50% of the individuals' incomes fall within this range. The plot also identifies the topcoded value as an outlier at $235,884, which is far outside the typical income range.

In Fig 2.when the topcoded values are removed, the histogram shows a more accurate representation of the income distribution, with the right skew still present but less extreme. This plot offers a more realistic view of the income structure, showing that the vast majority of individuals earn significantly less than $235,884.

Similarly in Fig2.1 the boxplot, now excluding the topcoded values, shows a median income of $35,000 and a more concentrated IQR of $25,000 to $50,000. By removing the topcoded value, the outlier is eliminated, and the distribution is more representative of the actual income spread. This boxplot highlights the more typical income range for most individuals, without the distortion introduced by the topcoded values.

RACE

```
# Check for unique values and if any missing values are present
unique(nlsy_df$race)
```

```
## [1] 4 2 1 3
```

```
table(nlsy_df$race, useNA = "ifany")
```

```
##
##    1    2    3    4
## 2335 1901   83 4665
```

Here we can see that we have the valid values from 1-4 and this column does not contain any missing values as well. Now further we can encode the numeric values to the race/ethinicity.

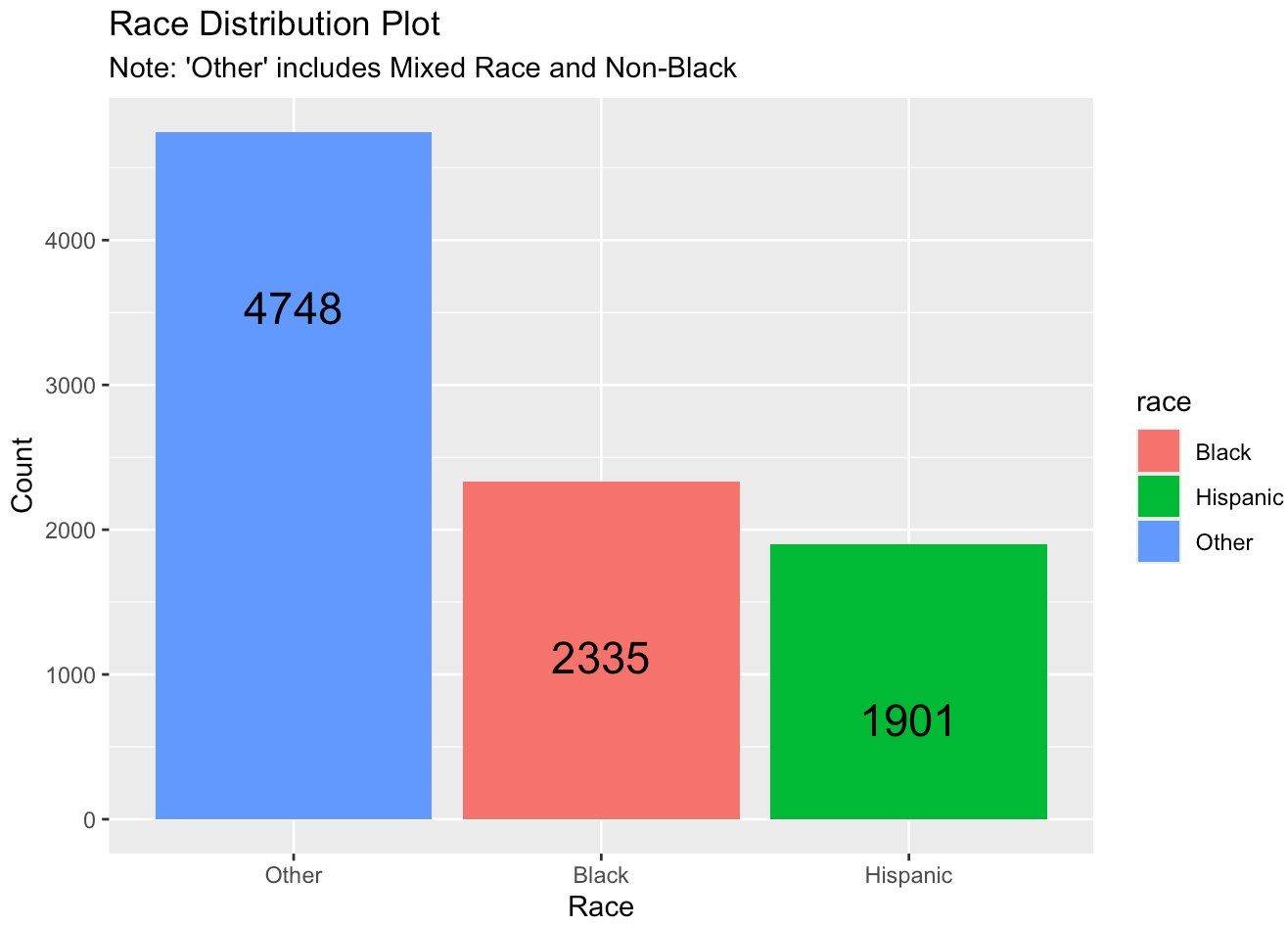
```
# Recode race values into descriptive labels
nlsy_df <- nlsy_df %>%
  mutate(race = case_when(
    race == 1 ~ "Black",
    race == 2 ~ "Hispanic",
    race %in% c(3, 4) ~ "Other" # Combine "Mixed Race" and "Non-Black" into "Other"
  )) %>%
  mutate(race = factor(race, levels = c("Black", "Hispanic", "Other")))
```

```
# Count the number of respondents in each race group
nlsy_df %>%
  count(race, sort = TRUE) %>%
  rename(count = n)
```

```
## # A tibble: 3 × 2
##   race     count
##   <fct>    <int>
## 1 Other     4748
## 2 Black     2335
## 3 Hispanic  1901
```

From the count we can infer that the total population contains 2335 Black respondents, 1901 Hispanic, 4665 Non-Black and 83 Mixed Race in Others population.

```
# Race Summary
ggplot(nlsy_df, aes(x = reorder(race, -table(race)[race]), fill = race)) +
  geom_bar() +
  geom_text(aes(label = ..count..), stat = "count", vjust = 6, size = 6) +
  labs(
    title = "Race Distribution Plot", x = "Race", y = "Count",
    subtitle = "Note: 'Other' includes Mixed Race and Non-Black"
  )
```



HIGHEST GRADE COMPLETED

```
# Check for unique values and if any missing values are present
unique(nlsy_df$highest_grade_completed)
```

```
##  [1] -4 18 -5 15 11 14 16 12 20 19 17 13  3  5  7  6  2  8 10  1  4 -1  9  0 95
## [26] -2
```

```
table(nlsy_df$highest_grade_completed, useNA = "ifany")
```

```
##
##   -5   -4   -2   -1    0    1    2    3    4    5    6    7    8    9   10   11
## 1561 5870    2    3    2    3   10    4    4    4   11    3    9   12   20   37
##   12   13   14   15   16   17   18   19   20   95
##  190  213  277  170  188  103  120   66   99    3
```

From the above we can see there are four types of negative values present which are -5, -4, -2 and -1 denoting non-interview, skipped, don't know and refusal. Further we have to encode the values with appropiriate labels.

```
# Replace negative values with NA
nlsy_df <- nlsy_df %>%
  mutate(highest_grade_completed = ifelse(highest_grade_completed < 0, NA, highest_grade
_completed))
```

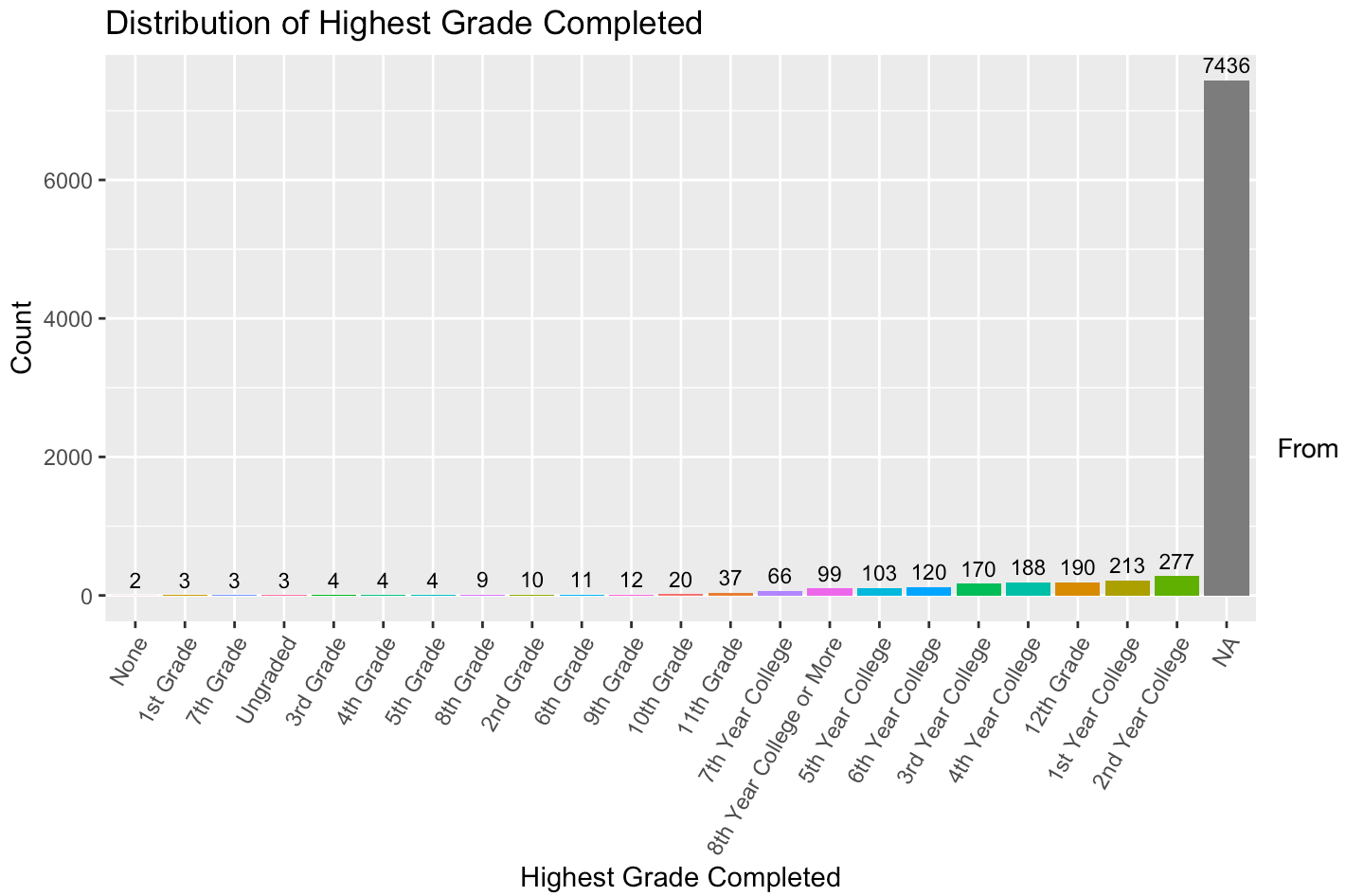
```
# Recode highest grade completed into descriptive labels
nlsy_df <- nlsy_df %>%
  mutate(highest_grade_completed = case_when(
    highest_grade_completed == 0 ~ "None",
    highest_grade_completed == 1 ~ "1st Grade",
    highest_grade_completed == 2 ~ "2nd Grade",
    highest_grade_completed == 3 ~ "3rd Grade",
    highest_grade_completed == 4 ~ "4th Grade",
    highest_grade_completed == 5 ~ "5th Grade",
    highest_grade_completed == 6 ~ "6th Grade",
    highest_grade_completed == 7 ~ "7th Grade",
    highest_grade_completed == 8 ~ "8th Grade",
    highest_grade_completed == 9 ~ "9th Grade",
    highest_grade_completed == 10 ~ "10th Grade",
    highest_grade_completed == 11 ~ "11th Grade",
    highest_grade_completed == 12 ~ "12th Grade",
    highest_grade_completed == 13 ~ "1st Year College",
    highest_grade_completed == 14 ~ "2nd Year College",
    highest_grade_completed == 15 ~ "3rd Year College",
    highest_grade_completed == 16 ~ "4th Year College",
    highest_grade_completed == 17 ~ "5th Year College",
    highest_grade_completed == 18 ~ "6th Year College",
    highest_grade_completed == 19 ~ "7th Year College",
    highest_grade_completed == 20 ~ "8th Year College or More",
    highest_grade_completed == 95 ~ "Ungraded",
    highest_grade_completed < 0 ~ NA_character_
  ))
```

```
# Verify the count of different grades
nlsy_df %>%
  count(highest_grade_completed, sort = TRUE) %>%
  rename(count = n)
```

```
## # A tibble: 23 × 2
##    highest_grade_completed  count
##    <chr>                   <int>
##  1 <NA>                     7436
##  2 2nd Year College          277
##  3 1st Year College          213
##  4 12th Grade                190
##  5 4th Year College          188
##  6 3rd Year College          170
##  7 6th Year College          120
##  8 5th Year College          103
##  9 8th Year College or More   99
## 10 7th Year College           66
## # ℹ 13 more rows
```

From above we see that majority of the population did not answer about their grades which is 7436. From the remaining we can see that the majority of population completed 2nd year college(277) followed by 1st year college(213).

```
# Create the plot with proper count calculation
ggplot(nlsy_df, aes(x = reorder(highest_grade_completed, table(highest_grade_completed)
[highest_grade_completed]), fill = highest_grade_completed)) +
  geom_bar(stat = "count", show.legend = FALSE) +
  geom_text(stat = "count", aes(label = ..count..), vjust = -0.5, size = 3) +
  labs(title = "Distribution of Highest Grade Completed", x = "Highest Grade Completed",
y = "Count") +
  theme(axis.text.x = element_text(angle = 60, vjust = 1, hjust = 1))
```



the above we see that theere are missing values for the majority of responses which is 7436. From the remaining we can see that the majority of population completed 2nd year college(277) followed by 1st year college(213).

AGE AS OF 1996 DECEMBER 31ST

```
# Checking if Age as of 1996 december 31st needs cleaning | unique values: 13 12 16 15 1
4
# The variable Age shows the minimum expected value in survey response is 12, 1st quarti
le is 13, median is 14, mean is 13.99, 3rd quartile is 15, and max is 16. This variable
does not have any missing values and hence no cleaning is required.
unique(nlsy_df$age_1996)
```

```
## [1] 15 14 13 12 16
```

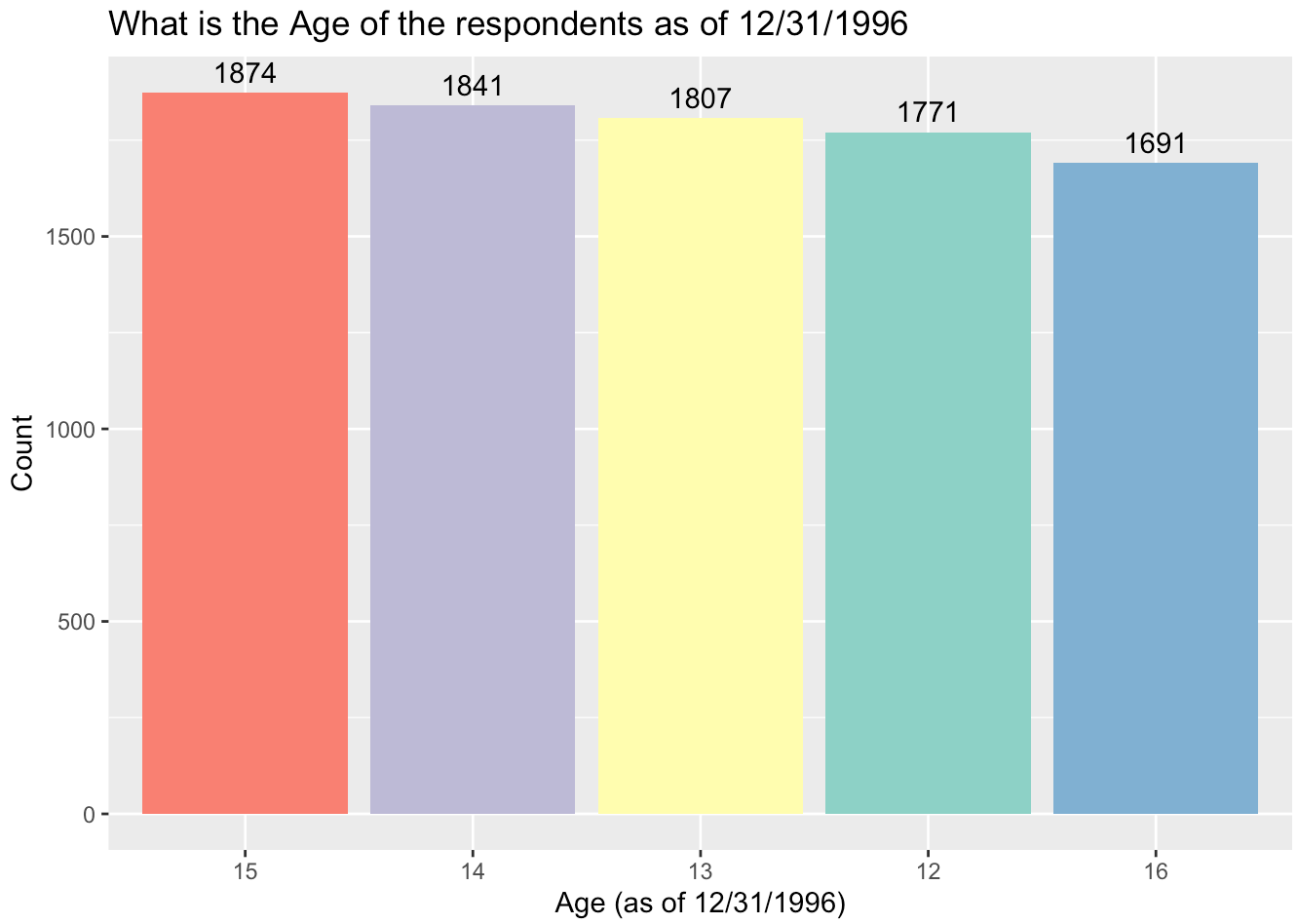
```
summary(nlsy_df$age_1996)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   12.00   13.00   14.00   13.99   15.00   16.00
```

```
# The tabular form of data displayed below shows the age and number of people having tha
t age.
nlsy_df %>%
  count(age_1996, sort = TRUE) %>%
  rename(count = n)
```

```
## # A tibble: 5 × 2
##   age_1996 count
##      <dbl> <int>
## 1       15  1874
## 2       14  1841
## 3       13  1807
## 4       12  1771
## 5       16  1691
```

```
# Create a summarized dataset to sort the age counts
age_1996_sorted <- nlsy_df %>%
  count(age_1996, sort = TRUE) %>%
  arrange(desc(n))

# Plotting age and its concentration
ggplot(data = age_1996_sorted, aes(x = reorder(age_1996, -n), y = n, fill = factor(age_1
996))) +
  geom_bar(stat = "identity",show.legend = FALSE) +
  geom_text(aes(label = n), vjust = -0.5) +
  labs(
    title = "What is the Age of the respondents as of 12/31/1996",
    x = "Age (as of 12/31/1996)",
    y = "Count"
  ) +
  scale_fill_brewer(palette = "Set3", name = "Age")
```



CITIZENSHIP

```
# Checking if Citizenship requires cleaning | unique values:  1 3 2 -4
# Citizenship is a variable have 1,2,3 as response in the survey and -4, a negative valu
e indicating Valid Skip. We will be cleaning this variable to replace -4 in the data wit
h NA. Furthermore, summarizing this variable we get min value as -4 ( which will be trea
ted later), 1st quartile is 1, median is 1, mean is 0.6279, 3rd quartile is 1, and max i
s 3.
unique(nlsy_df$citizenship)
```

```
## [1]  3 -4  1  2
```

```
summary(nlsy_df$citizenship)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -4.0000  1.0000  1.0000  0.6279  1.0000  3.0000
```

```
# The tabular form of data displayed below shows the age and number of people having tha
t age.
# Here we see that there is a significant number of responses with negative value in our
data set for Citizenship variable. If we change -4 to NA, then 1042 data will be omitted
when we plot it which might introduce biases in the data and the output might not be acc
urate.
nlsy_df %>%
  count(citizenship, sort = TRUE) %>%
  rename(count = n)
```

```
## # A tibble: 4 × 2
##   citizenship count
##         <dbl> <int>
## 1           1  6869
## 2          -4  1042
## 3           3   794
## 4           2   279
```

```
# Replace -4 with NA in the Citizenship column
nlsy_df <- nlsy_df %>%
  mutate(
    citizenship = ifelse(citizenship %in% c(-4), NA, citizenship)
  )

# Verify the changes
table(nlsy_df$citizenship, useNA = "always")
```

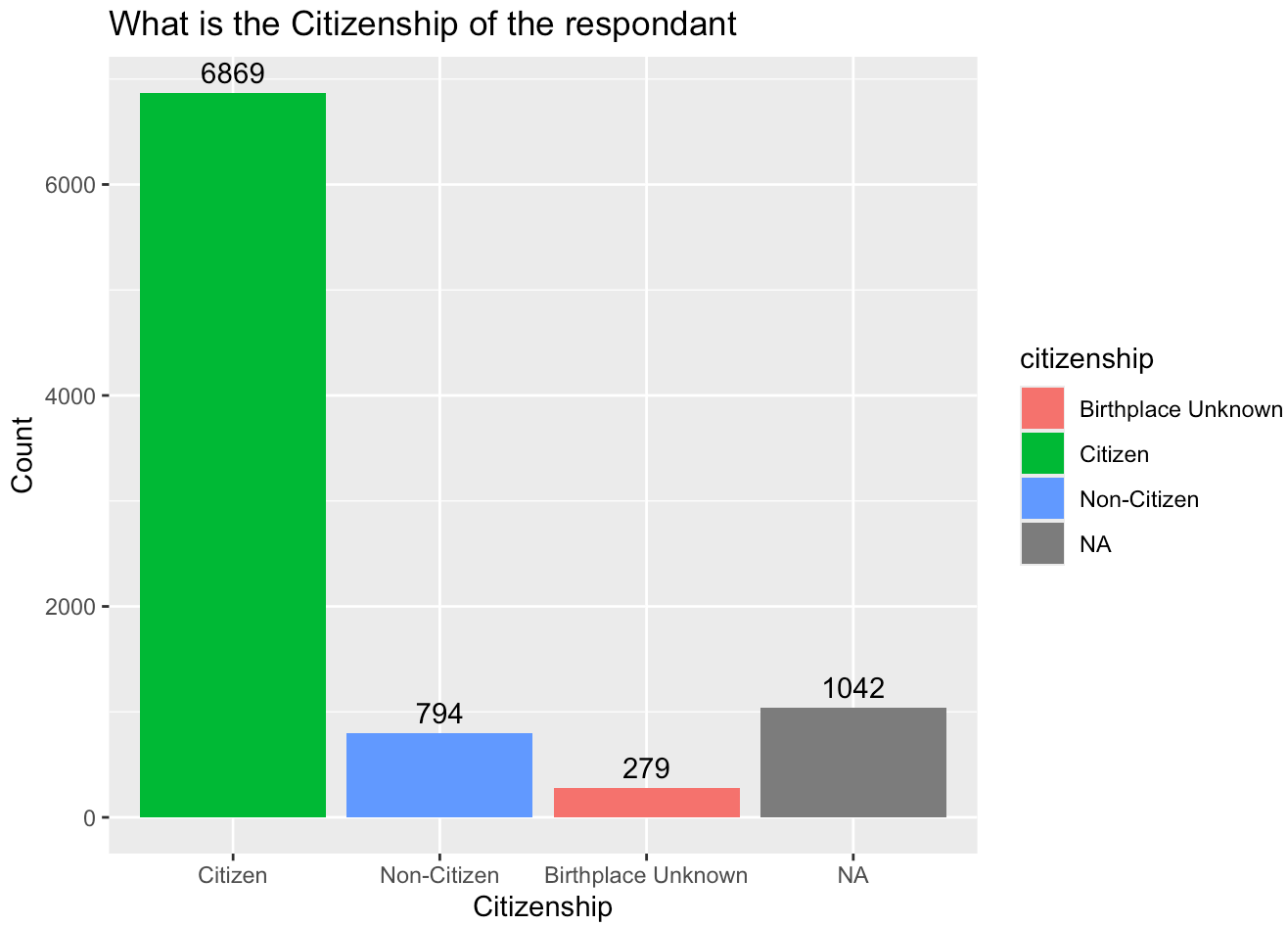
```
##
##    1    2    3 <NA>
## 6869  279  794 1042
```

```
# For this question 'What is the citizenship', we have marked 1 as Citizen, born in the
U.S., 2 as Unknown, not born in U.S., 3 as Unknown, not born in U.S., and for all negati
ve values we have used NA. The table below shows the count of each kind of responses.
nlsy_df <- nlsy_df %>%
  mutate(citizenship = case_when(
    citizenship == 1 ~ "Citizen",
    citizenship == 2 ~ "Birthplace Unknown",
    citizenship == 3 ~ "Non-Citizen",
    citizenship < 1 ~ NA_character_
  ))
```

```
# Upon mutating values for citizenship to make it categorical we have replaced -4 with N
A for 1042 rows, 1 with Citizen, born in the U.S., 2 with Unknown, not born in U.S., 3 w
ith Unknown, can't determine birthplace, and anything less than 1 with NA. This helps us
in easily reading the analysis.
nlsy_df %>%
  count(citizenship, sort = TRUE) %>%
  rename(count = n)
```

```
## # A tibble: 4 × 2
##   citizenship        count
##   <chr>              <int>
## 1 Citizen             6869
## 2 <NA>                1042
## 3 Non-Citizen          794
## 4 Birthplace Unknown   279
```

```
# Plotting citizenship and its concentration
ggplot(data = nlsy_df, aes(x = reorder(citizenship,-table(citizenship)[citizenship]), fi
ll = citizenship)) +
  geom_bar() +
  geom_text(stat = "count", aes(label = ..count..), vjust = -0.5) +
  labs(
    title = "What is the Citizenship of the respondant",
    x = "Citizenship", y = "Count",
  )
```




```
# The graph below shows how the response is ranged across. It illustrates that majority of the respondants have a citizenship of the U.S. and constitutes to 6869 out of 8984, some don't know their birthplace constitute to 794, some who were not born in the U.S. adds up to 279, and the remaining 1042 were all missing data. This, we can say that 76.45% of the people were born in the U.S., 3.1% were not born in the U.S, 8.83% cannot determine their birthplace, and 11.5% of significant population skipped the question.
```

EVER USE MARIJUANA

```
# This variable Citizenship has different values like Non-Citizen, Birthplace Unknown, Citizen , and NA. Possible responses are 0,1,-1,-2,-3 where 0 indicates No, 1 indicates Yes, -1 indicates Refusal, -2 indicates Don't know, and -3 indicates Invalid Skip. To clean this we will be taking -1 and -2 into consideration as it can tell us if respondents might be hiding the truth. However, we can replace -3 with NA to get the accurate analysis done for this variable.
unique(nlsy_df$used_marijuana)
```

```
## [1]  0  1 -1 -2 -3
```

```
summary(nlsy_df$used_marijuana)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -3.0000  0.0000  0.0000  0.1942  0.0000  1.0000
```

```
# To clean the variable Used Marijuana, we are performing following steps like convertin
g to categorical values for ease of analysis.
# Replace -3 with NA in the Used_Marijuana
nlsy_df <- nlsy_df %>%
  mutate(
    used_marijuana = ifelse(used_marijuana %in% c(-3), NA, used_marijuana)
  )

# Verify the changes
table(nlsy_df$used_marijuana, useNA = "always")
```

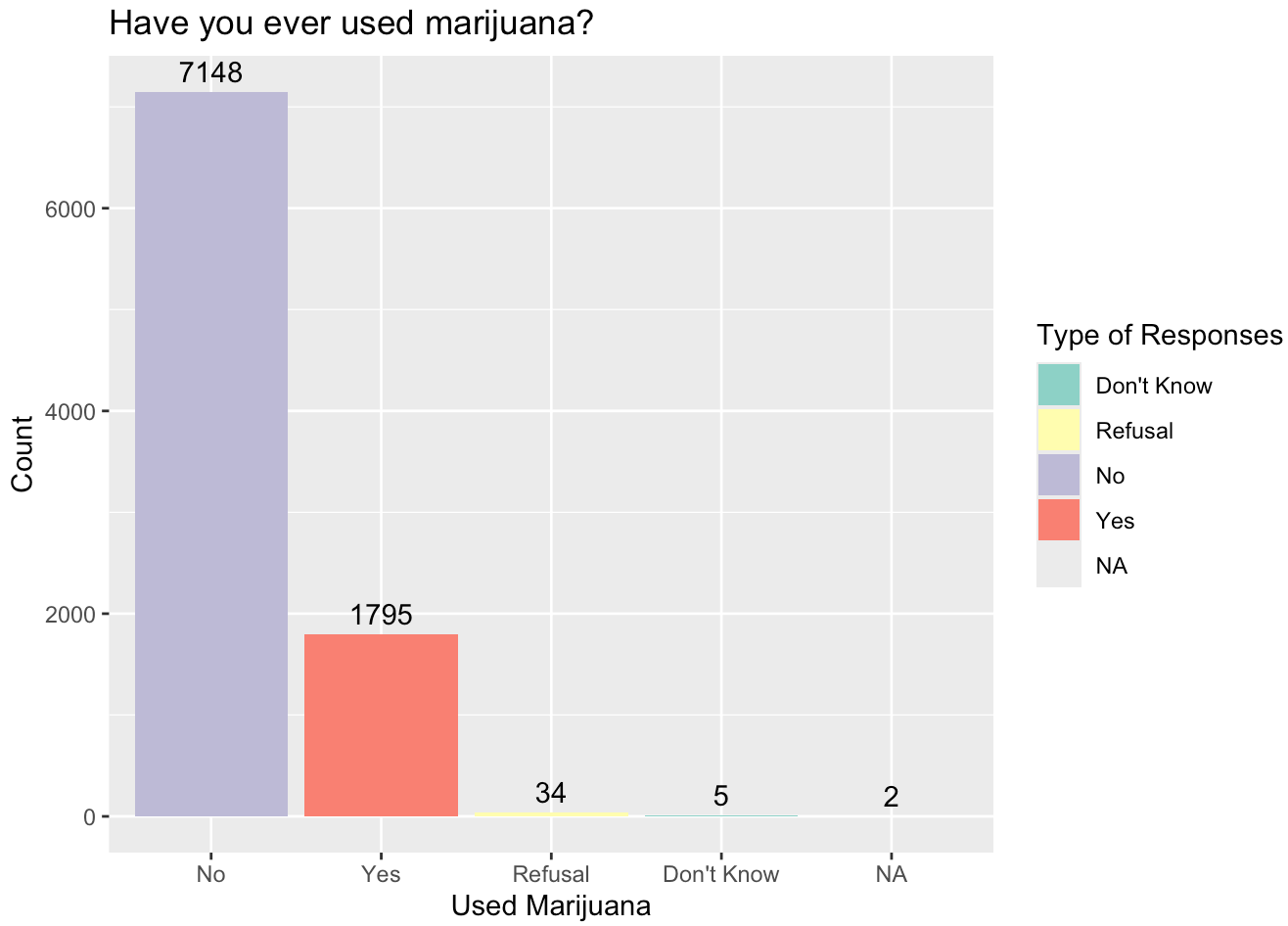
```
##
##   -2   -1    0    1 <NA>
##    5   34 7148 1795    2
```

```
# Here we have different factor levels like Don't know, Refusal, No, Yes for Factor used
_marijuana.
nlsy_df <- nlsy_df %>%
  mutate(used_marijuana = factor(used_marijuana, levels = c(-2, -1, 0, 1), labels = c("D
on't Know", "Refusal", "No", "Yes")))
```

```
# Upon factoring values for used_marihjuana to make it categorical we have replaced -3 w
ith NA for 2 rows, 1 with Yes, 0 with No, -1 with Refusal, and -2 with don't know. This
helps us in easily reading the analysis.
nlsy_df %>%
  count(used_marijuana, sort = TRUE) %>%
  rename(count = n)
```

```
## # A tibble: 5 × 2
##   used_marijuana count
##   <fct>          <int>
## 1 No              7148
## 2 Yes             1795
## 3 Refusal           34
## 4 Don't Know         5
## 5 <NA>               2
```

```
# Plotting used_marijuana
ggplot(data = nlsy_df, aes(x = reorder(used_marijuana, -table(used_marijuana)[used_marij
uana]), fill = used_marijuana)) +
  geom_bar() +
  geom_text(stat = "count", aes(label = ..count..), vjust = -0.5) +
  labs(
    title = "Have you ever used marijuana?",
    x = "Used Marijuana", y = "Count",
  ) +
  scale_fill_brewer(palette = "Set3", name = "Type of Responses")
```




```
# The graph below illustrates that 79.56% (7148) of respondents do not use marijuana, 1
9.97% (1795) use marijuana, 0.05% (5) don't know if the used it, 0.37% (34) refused to a
nswer the question, and the question was a invalid skip for 2 people. By this plot, we c
an say that 0.43% (39) people might be hiding some information on use of marijuana. Ther
e is a significant number of respondents who use marijuana.
```

INDUSTRY CODE

```
# Checking unique values in industry_code
unique(nlsy_df$industry_code)
```

```
##   [1] 4470 9470 7670 8180 7860   -4 4290   -5 8190 8680 6870 5380 6170 6990 8370
##  [16]  770 7390 7890 7070 7270 6180 7580 9170 5070 7870 7680 6070 6570 8090 9590
##  [31] 5270 7370 1990 7780 6290 7690 9290   -3 7380 1270 9570 5170 8660 7790 7470
##  [46] 6480 8980 4560 8270 7280 8470 1880 9890 6680 6690  590 2290  280 7080 8970
##  [61] 8590 8770 1680 4690 5390 6390 6670 6470 6970 9480 4790 7980 4770 9990 8170
##  [76] 1370 6890 6695 6380 7770 7970 8290 8560 8990 5580 7290 4970 4980 8390 2190
##  [91] 7480 5590 6880 9490 7490 5090 9390 2870 1870 5190 3580  670 8080 3190 9370
## [106]  180 3590 1490 3890 4380 4270 4370 7590 1180 6770 8780 4090 5480 8880 6190
## [121] 5490 7460 8570 4670 4680 5080 6692 4580 8690  570 3390 4180 9190 8870 9160
## [136] 3980 2890 6080 7990 2370 5370 4870 2380 3370 8070  490 5290 7570 5180 4390
## [151] 3080 5690 4190 3960 6370 3170 2670 3570 4170 5592 3180 2990 3970 3090 5470
## [166] 4080 2590 4070 1190 3870 2270 7880 5570 7180 4280 8790 2690 6490 2170 5790
## [181] 2070 4890 9090 9380 2980 1590 3490 2770 2570 6675 1280  290  170 2390 2780
## [196] 1170 4585 5591 3360  580 4260 1890 2680 2880 7170 8670 6590 1290  270 9070
## [211]  690 1790 2790 4990 7190 3670  680 3680 1470 3770 4780 2280 1480 1670 5670
## [226]  370 3380 6090 8380 4880 4490 4480 1090 2490  190 1770 5680 3470 3780  470
## [241]  380 3790 6270 1070 3690 1080 9080 2480 1690 2090 1390
```

```
table(nlsy_df$industry_code, useNA = "ifany")
```

```
##
##   -5   -4   -3  170  180  190  270  280  290  370  380  470  490  570  580  590
## 1561 1128  129   16   20    1    3    1    4   10    1    2   32   16    5    2
##  670  680  690  770 1070 1080 1090 1170 1180 1190 1270 1280 1290 1370 1390 1470
##   12    2    1  430    6    1    4    4   24    5    9    8    1   10    1    2
## 1480 1490 1590 1670 1680 1690 1770 1790 1870 1880 1890 1990 2070 2090 2170 2190
##    1    2    2    2    8    2    1    2   10    8    3   17    4    1    7   13
## 2270 2280 2290 2370 2380 2390 2480 2490 2570 2590 2670 2680 2690 2770 2780 2790
##    2    3   14   14    5    1    2    1    3    3    7    3    7    1    1    1
## 2870 2880 2890 2980 2990 3080 3090 3170 3180 3190 3360 3370 3380 3390 3470 3490
##   16    8    4   11    1    8    1    6    4   24    5    4    3    9    3   10
## 3570 3580 3590 3670 3680 3690 3770 3780 3790 3870 3890 3960 3970 3980 4070 4080
##   29    9   10    1    6    2    2    1    1    9   14    5    5   14   10   11
## 4090 4170 4180 4190 4260 4270 4280 4290 4370 4380 4390 4470 4480 4490 4560 4580
##   10   12    2    7    7   18    3    9    4   17    6   34    2    1   15    7
## 4585 4670 4680 4690 4770 4780 4790 4870 4880 4890 4970 4980 4990 5070 5080 5090
##    1   37    4   19   21    4   27   47    4    6   95   13    4   44   15   23
## 5170 5180 5190 5270 5290 5370 5380 5390 5470 5480 5490 5570 5580 5590 5591 5592
##   51    5    7   20    1   10  110   30    1    8   16    5   31    3    1    7
## 5670 5680 5690 5790 6070 6080 6090 6170 6180 6190 6270 6290 6370 6380 6390 6470
##    3    2   16    2   20    8    3   72   16    3    2   37   12   19   10   10
## 6480 6490 6570 6590 6670 6675 6680 6690 6692 6695 6770 6870 6880 6890 6970 6990
##   15    4   18    4   25    1   16   36    3    8    9   88   15   48   45  111
## 7070 7080 7170 7180 7190 7270 7280 7290 7370 7380 7390 7460 7470 7480 7490 7570
##   83   11    1    7    6   57   38   50   13   80   67   24   28   12   22    6
## 7580 7590 7670 7680 7690 7770 7780 7790 7860 7870 7880 7890 7970 7980 7990 8070
##  141   55    7   63   51   59   19   21  303  135    2   45   66   36    4    2
## 8080 8090 8170 8180 8190 8270 8290 8370 8380 8390 8470 8560 8570 8590 8660 8670
##    7   62   58   71  226  106   53   88    5   12   97   67   10   93   86    5
## 8680 8690 8770 8780 8790 8870 8880 8970 8980 8990 9070 9080 9090 9160 9170 9190
##  429   28   60   17    8   10    5   14   52   25    6    2   19   28   44    9
## 9290 9370 9380 9390 9470 9480 9490 9570 9590 9890 9990
##   34   25   13    9  113   33   11    7   38   22   10
```

From the above, we observe:

1561 values are NON-INTERVIEW (denoted by -5),
1128 values are VALID SKIP (denoted by -4),
129 values are Invalid Skip (denoted by -3).

```
# Recoding industry_code with descriptive labels
nlsy_df <- nlsy_df %>%
  mutate(industry_code = case_when(
    industry_code >= 170 & industry_code <= 290 ~ "Agriculture, Forestry, and Fisherie
s",
    industry_code >= 370 & industry_code <= 490 ~ "Mining",
    industry_code >= 570 & industry_code <= 690 ~ "Utilities",
    industry_code == 770 ~ "Construction",
    industry_code >= 1070 & industry_code <= 3990 ~ "Manufacturing",
    industry_code >= 4070 & industry_code <= 4590 ~ "Wholesale Trade",
    industry_code >= 4670 & industry_code <= 5790 ~ "Retail Trade",
    industry_code == 5890 ~ "Arts, Entertainment, and Recreation Services",
    industry_code >= 6070 & industry_code <= 6390 ~ "Transportation and Warehousing",
    industry_code >= 6470 & industry_code <= 6780 ~ "Information and Communication",
    industry_code >= 6870 & industry_code <= 7190 ~ "Finance, Insurance, and Real Estat
e",
    industry_code >= 7270 & industry_code <= 7790 ~ "Professional and Related Services",
    industry_code >= 7860 & industry_code <= 8470 ~ "Educational, Health, and Social Ser
vices",
    industry_code >= 8560 & industry_code <= 8690 ~ "Entertainment, Accommodations, and
Food Services",
    industry_code >= 8770 & industry_code <= 9290 ~ "Other Services",
    industry_code >= 9370 & industry_code <= 9590 ~ "Public Administration",
    industry_code >= 9670 & industry_code <= 9890 ~ "Active Duty Military",
    industry_code >= 9950 & industry_code <= 9990 ~ "ACS Special Codes",
    industry_code < 0 ~ NA_character_ # Handling special codes (-5, -4, -3)
  ))
```

In the above we have lebelled the industry code and also changed the negative values with NA.

```
# Counting missing values
sum(is.na(nlsy_df$industry_code))
```

```
## [1] 2818
```

The count of missing values is 1561+1128+129 = 2818 rows.

```
# Calculating the percentage of missing values
sum(is.na(nlsy_df$industry_code)) / nrow(nlsy_df) * 100
```

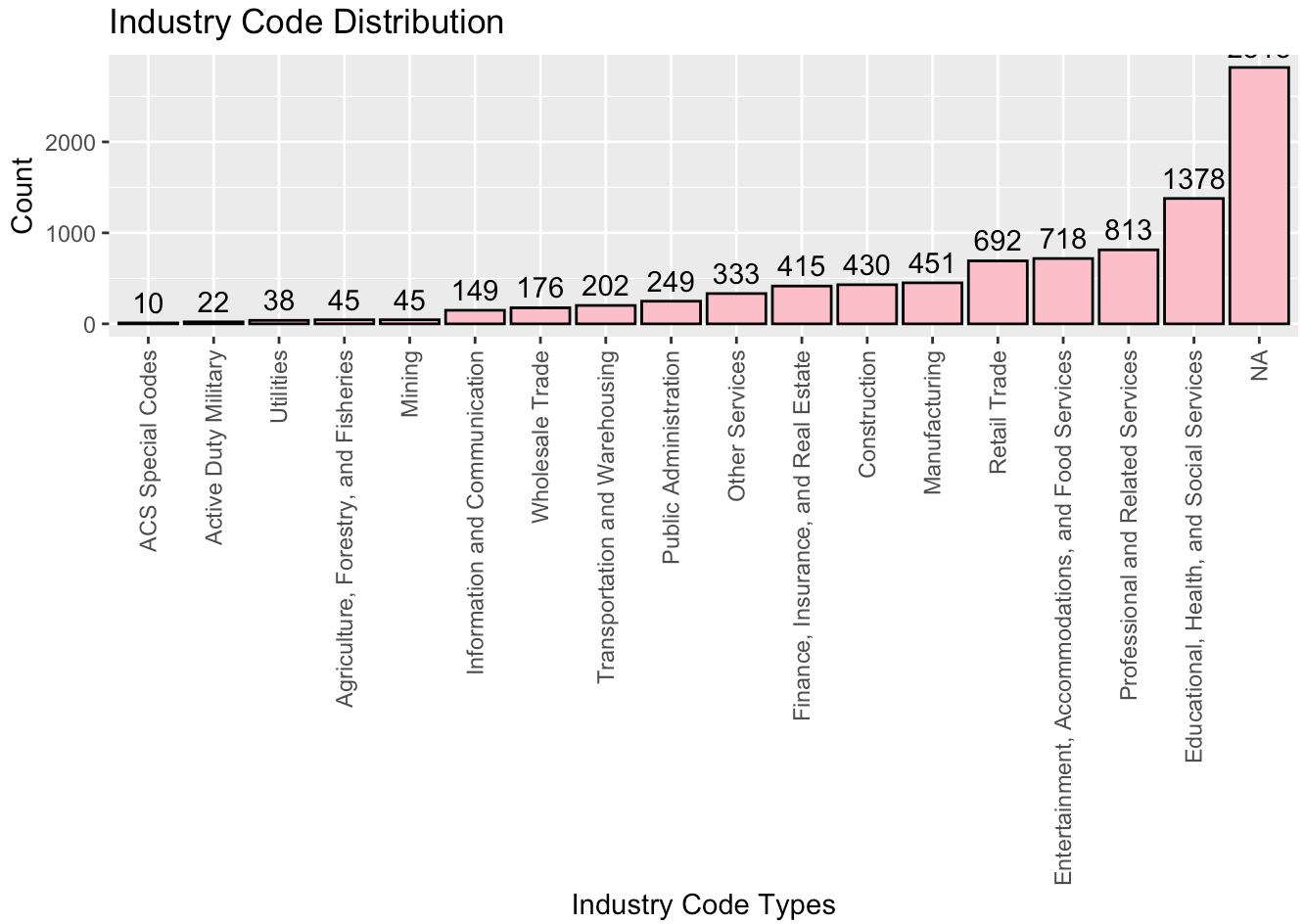
```
## [1] 31.36687
```

From the above we can see that the missing values(NA) are 2818 and the percentage can be calculated as 2818/8984*100 which is 31.37% of the total data which causes the data to be baised.

```
# Verifying the count of each industry_code
nlsy_df %>%
  count(industry_code, sort = TRUE) %>%
  rename(count = n)
```

```
## # A tibble: 18 × 2
##    industry_code                                  count
##    <chr>                                          <int>
##  1 <NA>                                            2818
##  2 Educational, Health, and Social Services        1378
##  3 Professional and Related Services                813
##  4 Entertainment, Accommodations, and Food Services  718
##  5 Retail Trade                                     692
##  6 Manufacturing                                    451
##  7 Construction                                     430
##  8 Finance, Insurance, and Real Estate              415
##  9 Other Services                                   333
## 10 Public Administration                            249
## 11 Transportation and Warehousing                   202
## 12 Wholesale Trade                                  176
## 13 Information and Communication                    149
## 14 Agriculture, Forestry, and Fisheries              45
## 15 Mining                                            45
## 16 Utilities                                         38
## 17 Active Duty Military                              22
## 18 ACS Special Codes                                 10
```

```
# Visualizing industry code distribution with a bar plot
ggplot(data = nlsy_df, aes(x = reorder(industry_code,table(industry_code)[industry_cod
e]))) +
  geom_bar(fill = "pink", color = "black") +
  geom_text(stat = "count", aes(label = ..count..), vjust = -0.5) +
  labs(
    title = "Industry Code Distribution",
    x = "Industry Code Types",
    y = "Count"
  ) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))
```



The industry_code variable shows that 2818 rows are missing, accounting for 31.38% of the dataset. The largest group belongs to Educational, Health, and Social Services (1378 respondents), followed by Professional and Related Services (813) and Entertainment, Accommodations, and Food Services (718). Smaller industries, like Utilities (38) and Active Duty Military (22), are least represented. The missing data introduces potential bias and should be accounted for in analysis.

MARITAL STATUS

```
# Checking unique values
unique(nlsy_df$marital_status)
```

```
## [1] -5  0  1  2  3 -3  4
```

```
table(nlsy_df$marital_status, useNA = "ifany")
```

```
##
##   -5   -3    0    1    2    3    4
## 2250   62 2766 3066  154  663   23
```

There exists 2250 values indicating NON-INTERVIEW denoted by -5 and 62 values indicating Invalid Skip denoted by -3.

```
# Recode marital_status values into descriptive labels
nlsy_df <- nlsy_df %>%
  mutate(marital_status = case_when(
    marital_status == 0 ~ "Never-married",
    marital_status == 1 ~ "Married",
    marital_status %in% c(2, 3, 4) ~ "Other" # Combine "Separated", "Divorced", "Widowe
d"
  )) %>%
  mutate(marital_status = factor(marital_status, levels = c("Never-married", "Married",
"Other")))
```

In the above we have lebelled the marital status and also changed the negative values with NA.

```
# Counting the number of missing values
sum(is.na(nlsy_df$marital_status))
```

```
## [1] 2312
```

The count of missing values is 2250+62 = 2312 rows.

```
# Calculating the percentage of missing values
sum(is.na(nlsy_df$marital_status)) / nrow(nlsy_df) * 100
```

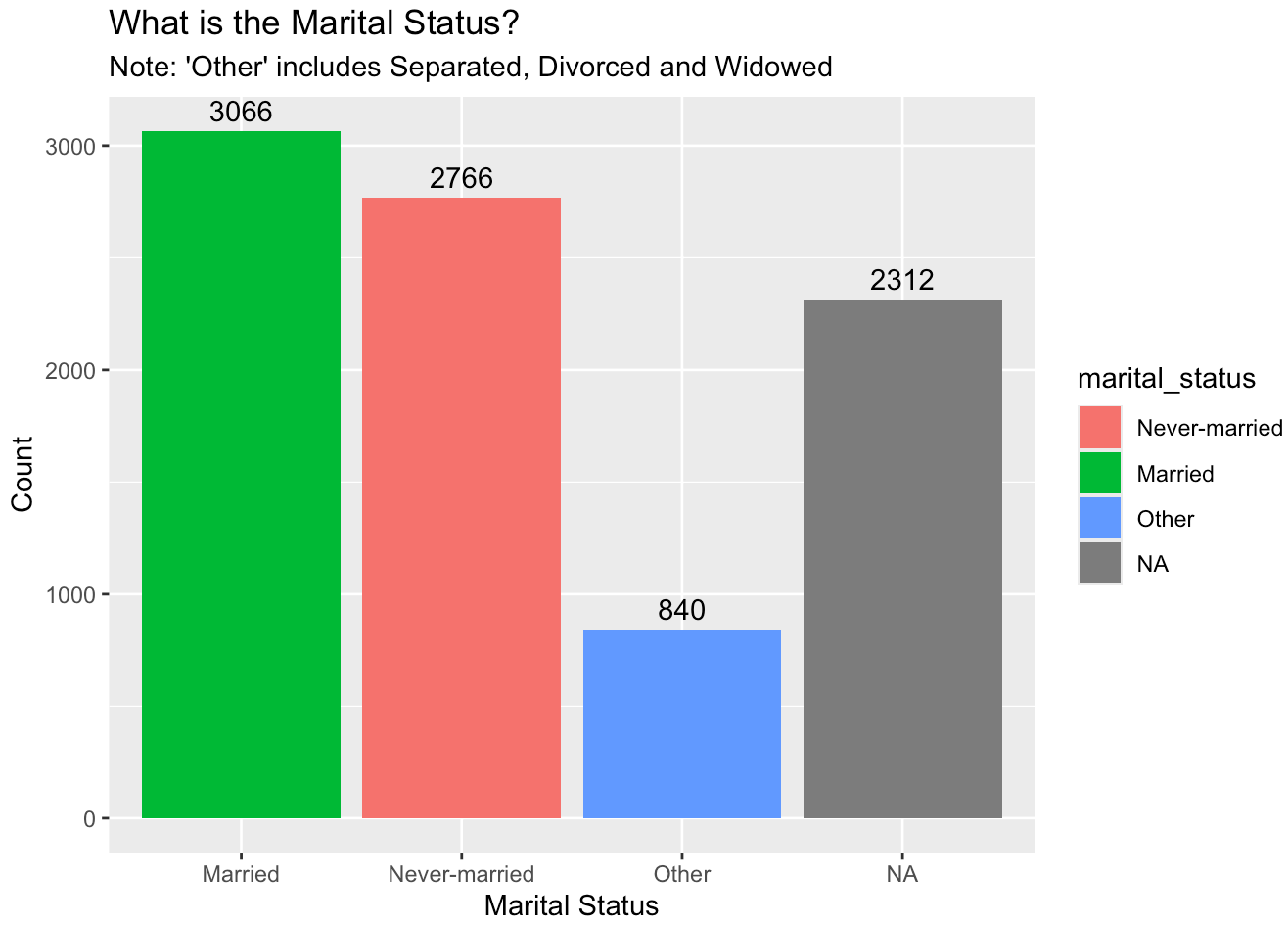
```
## [1] 25.73464
```

From the above we can see that the missing values(NA) are 2312 and the percentage can be calculated as 2312/8984*100 i.e. 25.73% of the total data which causes the data to be baised.

```
# Verifying the count of each marital_status
nlsy_df %>%
  count(marital_status, sort = TRUE) %>%
  rename(count = n)
```

```
## # A tibble: 4 × 2
##   marital_status count
##   <fct>          <int>
## 1 Married         3066
## 2 Never-married   2766
## 3 <NA>            2312
## 4 Other            840
```

```
# Visualizing marital status distribution with a bar plot
ggplot(data = nlsy_df, aes(x = reorder(marital_status,-table(marital_status)[marital_sta
tus]), fill = marital_status)) +
  geom_bar() +
  geom_text(stat = "count", aes(label = ..count..), vjust = -0.5) +
  labs(
    title = "What is the Marital Status?",
    x = "Marital Status",
    y = "Count",
    subtitle = "Note: 'Other' includes Separated, Divorced and Widowed"
  )
```



The marital_status distribution shows that the majority of respondents are Never-married (2766) or Married (3066). Smaller numbers are classified as “Other” (e.g., Separated, Divorced, Widowed). About 25.75% of the data is missing, which may introduce bias.

In the bar plot above, we can see that a significant portion of the dataset is missing marital status information, with about 25.75% of respondents not providing data due to non-interviews (-5) and invalid skips (-3). This missing data introduces potential bias in any analysis of marital status and should be carefully addressed. The largest groups in the marital status distribution are “Never-married” (2766 respondents) and “Married” (3066 respondents), with smaller numbers for “Divorced” (663), “Separated” (154), and “Widowed” (23). The distribution is relatively skewed towards those who are “Never-married” or “Married,” while the other categories are underrepresented. The presence of missing data in a large proportion of respondents calls for caution, as it may affect the generalizability of any findings related to marital status in this sample.

TOTAL NUM INCARCERATION

```
# Checking unique values in total_num_incarcerations
unique(nlsy_df$total_num_incarcerations)
```

```
##  [1]  0  1  2  3 -3  4  5  8  7  6 11  9
```

```
table(nlsy_df$total_num_incarcerations, useNA = "ifany")
```

```
##
##   -3    0    1    2    3    4    5    6    7    8    9   11
##   21 8054  511  205   94   52   29    9    4    3    1    1
```

From the data, 21 rows are Invalid Skip (denoted by -3).

```
# Recoding total_num_incarcerations into descriptive labels
nlsy_df <- nlsy_df %>%
  mutate(total_num_incarcerations = case_when(
    total_num_incarcerations == 0 ~ "No incarcerations",
    total_num_incarcerations >= 1 & total_num_incarcerations <= 2 ~ "1 to 2 incarceratio
ns",
    total_num_incarcerations >= 3 & total_num_incarcerations <= 4 ~ "3 to 4 incarceratio
ns",
    total_num_incarcerations >= 5 & total_num_incarcerations <= 6 ~ "5 to 6 incarceratio
ns",
    total_num_incarcerations >= 7 & total_num_incarcerations <= 8 ~ "7 to 8 incarceratio
ns",
    total_num_incarcerations >= 9 & total_num_incarcerations <= 10 ~ "9 to 10 incarcerat
ions",
    total_num_incarcerations >= 11 & total_num_incarcerations <= 12 ~ "11 to 12 incarcer
ations",
    total_num_incarcerations >= 13 & total_num_incarcerations <= 14 ~ "13 to 14 incarcer
ations",
    total_num_incarcerations >= 15 & total_num_incarcerations <= 16 ~ "15 to 16 incarcer
ations",
    total_num_incarcerations >= 17 & total_num_incarcerations <= 18 ~ "17 to 18 incarcer
ations",
    total_num_incarcerations >= 19 & total_num_incarcerations <= 20 ~ "19 to 20 incarcer
ations",
    total_num_incarcerations < 0 ~ NA_character_ # Handling special codes (-1, -2, -3)
  ))
```

In the above we have lebelled the total_num_incarcerations and also changed the negative values with NA.

```
# Counting the number of missing values
sum(is.na(nlsy_df$total_num_incarcerations))
```

```
## [1] 21
```

The count of missing values is 21 rows.

```
# Calculating the percentage of missing values
sum(is.na(nlsy_df$total_num_incarcerations)) / nrow(nlsy_df) * 100
```

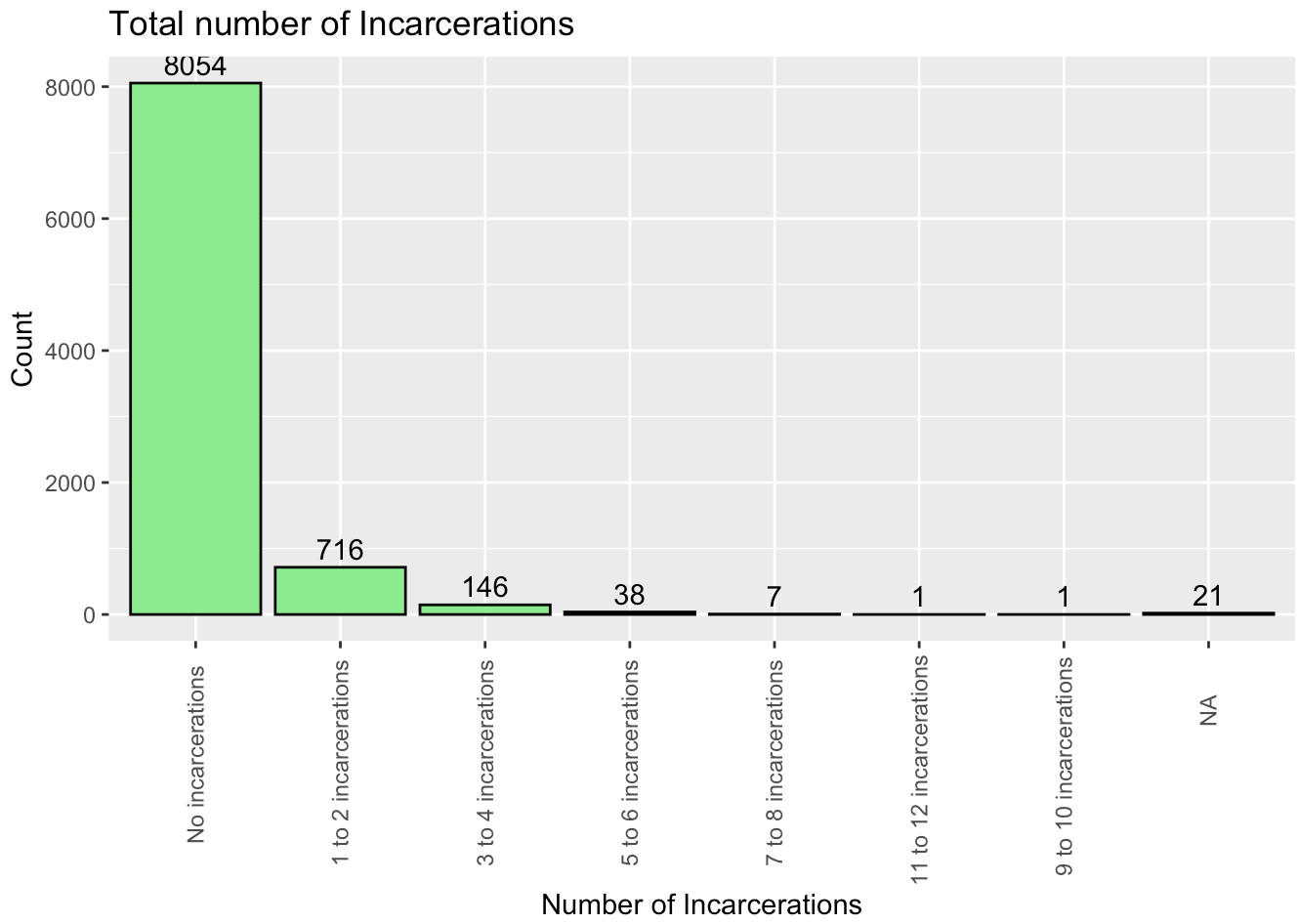
```
## [1] 0.2337489
```

From the above we can see that the missing values(NA) are 21 and the percentage can be calculated as 21/8984*100 i.e. 0.23% of the total data.

```
# Verifying the count of total_num_incarcerations for different levels
nlsy_df %>%
  count(total_num_incarcerations, sort = TRUE) %>%
  rename(count = n)
```

```
## # A tibble: 8 × 2
##   total_num_incarcerations count
##   <chr>                    <int>
## 1 No incarcerations         8054
## 2 1 to 2 incarcerations      716
## 3 3 to 4 incarcerations      146
## 4 5 to 6 incarcerations       38
## 5 <NA>                        21
## 6 7 to 8 incarcerations        7
## 7 11 to 12 incarcerations      1
## 8 9 to 10 incarcerations       1
```

```
# Bar plot for total_num_incarcerations
ggplot(data = nlsy_df, aes(x = reorder(total_num_incarcerations,-table(total_num_incarce
rations)[total_num_incarcerations]))) +
  geom_bar(fill = "lightgreen", color = "black") +
  geom_text(stat = "count", aes(label = ..count..), vjust = -0.4) +
  labs(
    title = "Total number of Incarcerations",
    x = "Number of Incarcerations",
    y = "Count"
  ) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 0.8))
```



The distribution of total_num_incarcerations reveals that the majority of respondents (8054) reported “No incarcerations”, indicating no contact with the criminal justice system. Smaller groups reported 1 to 2 incarcerations (716 respondents) and 3 to 4 incarcerations (146 respondents). Only 38 respondents reported 5 to 6 incarcerations, and even fewer experienced higher levels of incarceration: 7 respondents for 7 to 8 incarcerations, 1 respondent for 9 to 10 incarcerations, and 1 respondent for 11 to 12 incarcerations. With only 21 missing values (0.23%), the data is relatively complete for this variable.

COLLEGE TYPE

```
# Checking unique values in college_type
unique(nlsy_df$college_type)
```

```
## [1] -4  2 -5  3  1 -3
```

```
table(nlsy_df$college_type, useNA = "ifany")
```

```
##
##   -5   -4   -3    1    2    3
## 1561 5937   52  960  213  261
```

From the data:

1561 rows are NON-INTERVIEW (denoted by -5),
5937 rows are VALID SKIP (denoted by -4),
52 rows are Invalid Skip (denoted by -3).

```
# Changing factor levels for college_type into labels
nlsy_df <- nlsy_df %>%
  mutate(college_type = case_when(
    college_type == 1 ~ "Public institution",
    college_type == 2 ~ "Private not-for-profit institution",
    college_type == 3 ~ "Private for-profit institution",
    college_type < 0 ~ NA_character_ # Handling special codes (-1, -2, -3, -4, -5)
  ))
```

In the above we have lebelled the college types and also changed the negative values with NA.

```
# Counting the number of missing values
sum(is.na(nlsy_df$college_type))
```

```
## [1] 7550
```

The count of missing values is 1561+5937+52 = 7550 rows.

```
# Calculating the percentage of missing values
sum(is.na(nlsy_df$college_type)) / nrow(nlsy_df) * 100
```

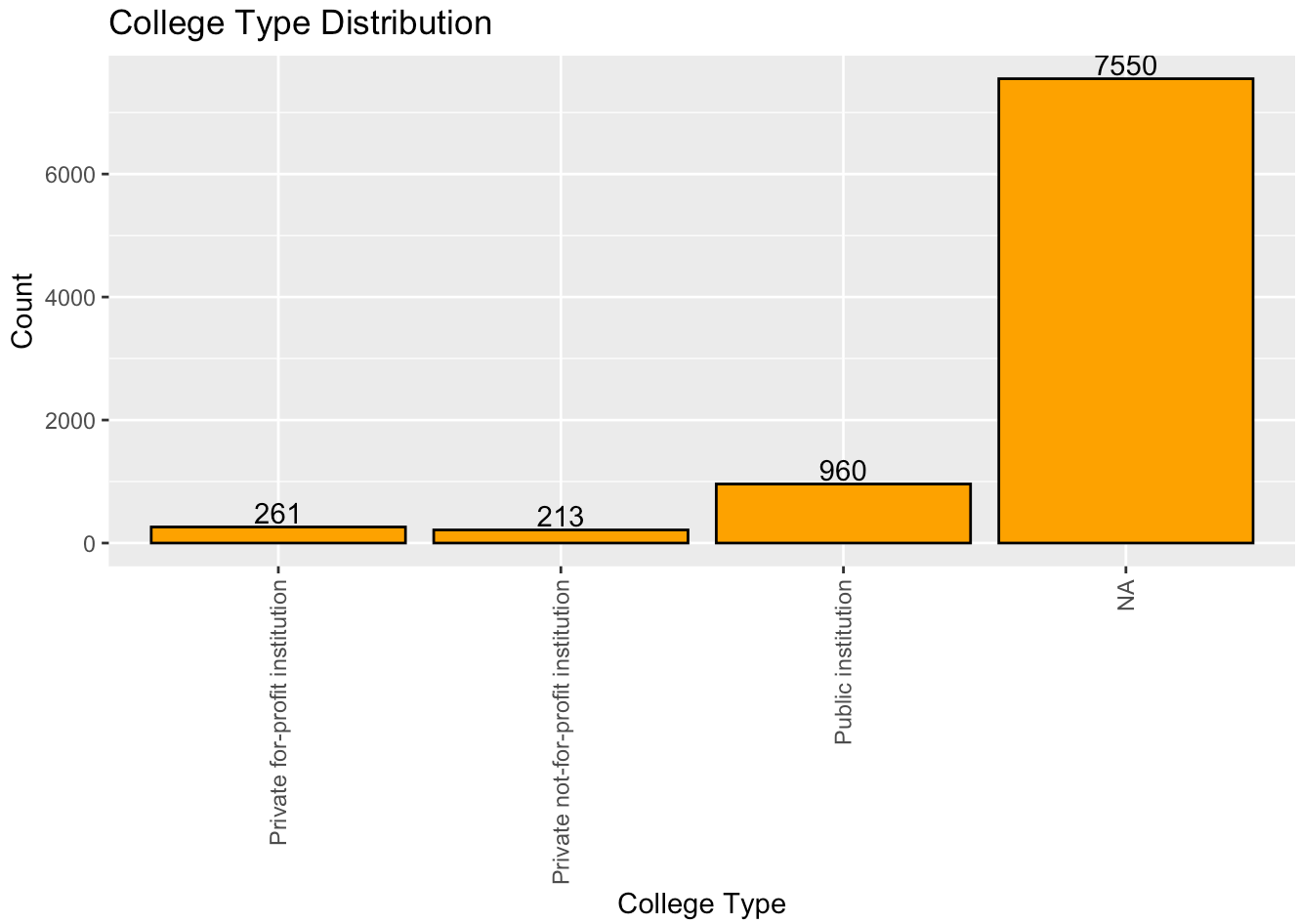
```
## [1] 84.03829
```

From the above we can see that the missing values(NA) are 7550 and the percentage given by 7550/8984*100 i.e. 84.04% of the total data which causes the data to be baised.

```
# Verifying the count of each college_type
nlsy_df %>%
  count(college_type, sort = TRUE) %>%
  rename(count = n)
```

```
## # A tibble: 4 × 2
##   college_type                      count
##   <chr>                             <int>
## 1 <NA>                               7550
## 2 Public institution                  960
## 3 Private for-profit institution      261
## 4 Private not-for-profit institution  213
```

```
# Bar plot for college_type
ggplot(data = nlsy_df, aes(x = college_type)) +
  geom_bar(fill = "orange", color = "black") +
  geom_text(stat = "count", aes(label = ..count..), vjust = -0.2) +
  labs(
    title = "College Type Distribution",
    x = "College Type",
    y = "Count"
  ) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))
```



The college_type variable indicates the type of institution respondents attended. The majority of respondents (960) attended a Public institution, followed by 261 respondents who attended a Private for-profit institution, and 213 respondents who attended a Private not-for-profit institution. A significant portion of the data (7550 rows, approximately 84.04%) is missing or marked as NA due to NON-INTERVIEW, VALID SKIP, or Invalid Skip. This high proportion of missing data introduces potential bias and limits the generalizability of findings. The distribution suggests that public institutions dominate the sample, with relatively few respondents attending private institutions of either type.

TRUSTFUL OR NOT

```
# Checking the data for unique values
unique(nlsy_df$trustful_or_not)
```

```
## [1] -4  5  4  3 -5  1  2 -1 -2
```

```
table(nlsy_df$trustful_or_not, useNA = "ifany")
```

```
##
##   -5   -4   -2   -1    1    2    3    4    5
## 1088 3005    3   13  128  189  271 1344 2943
```

It can be seen that 1088 values are NON-INTERVIEW denoted by -5, 3005 values are VALID SKIP denoted by -4, 3 values are Don't Know denoted by -2 and 13 values are Refusal denoted by -1.

```
# Recode trustful_or_not values into descriptive labels
nlsy_df <- nlsy_df %>%
  mutate(trustful_or_not = case_when(
    trustful_or_not >= 1 & trustful_or_not <= 3 ~ "Distrustful",
    trustful_or_not >= 4 ~ "Trustful",
    trustful_or_not < 0 ~ NA_character_ # Handling special codes (-1, -2, -3, -4, -5)
  )) %>%
  mutate(trustful_or_not = factor(trustful_or_not, levels = c("Distrustful", "Trustfu
l")))
```

In the above we have lebelled the college types and also changed the negative values with NA.

```
# Counting the number of missing values
sum(is.na(nlsy_df$trustful_or_not))
```

```
## [1] 4109
```

The count of missing values is 4109 rows.

```
# Calculating the percentage of missing values
sum(is.na(nlsy_df$trustful_or_not)) / nrow(nlsy_df) * 100
```

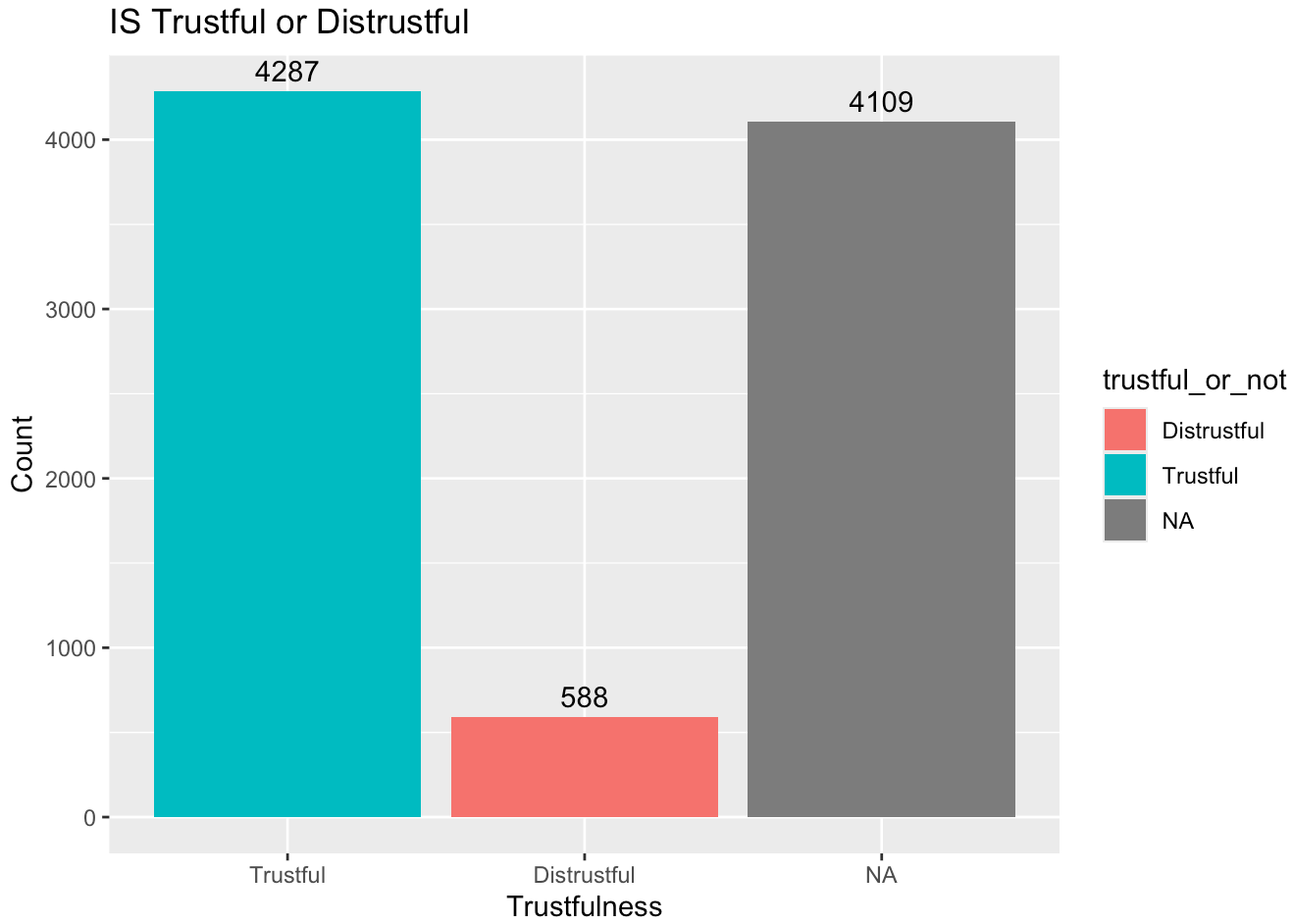
```
## [1] 45.73687
```

From the above we can see that the missing values(NA) are 4109 and the percentage given by 4109/8984*100 i.e. 45.73% of the total data which causes the data to be baised.

```
# Verifying the count of each trustful_or_not
nlsy_df %>%
  count(trustful_or_not, sort = TRUE) %>%
  rename(count = n)
```

```
## # A tibble: 3 × 2
##   trustful_or_not count
##   <fct>           <int>
## 1 Trustful         4287
## 2 <NA>             4109
## 3 Distrustful       588
```

```
# Bar plot for trustful_or_not
ggplot(data = nlsy_df, aes(x = reorder(trustful_or_not,-table(trustful_or_not)[trustful_
or_not]), fill = trustful_or_not)) +
  geom_bar() +
  geom_text(stat = "count", aes(label = ..count..), vjust = -0.5) +
  labs(
    title = "IS Trustful or Distrustful",
    x = "Trustfulness",
    y = "Count"
  )
```



The trustful_or_not variable shows a high percentage of missing data (45.73%). Among the valid data, the “Trustful” group (values >= 4) has a higher count than the “Distrustful” group (values 1–3). The missing data should be carefully addressed to ensure reliable insights.

SUMMARY OF EDA

```
summary(nlsy_df)
```

```
##     gender       birth_year   is_sad_depressed_unhappy_f
##  Male  :4599   Min.   :1980   Length:8984
##  Female:4385   1st Qu.:1981   Class :character
##                Median :1982   Mode  :character
##                Mean   :1982
##                3rd Qu.:1983
##                Max.   :1984
##
##  current_enrollment_status total_income_2016       race
##  Length:8984               Min.   :     0    Black   :2335
##  Class :character          1st Qu.: 25000    Hispanic:1901
##  Mode  :character          Median : 40000    Other   :4748
##                            Mean   : 49477
##                            3rd Qu.: 62000
##                            Max.   :235884
##                            NA's   :3893
##  highest_grade_completed    age_1996     citizenship          used_marijuana
##  Length:8984             Min.   :12.00   Length:8984        Don't Know:   5
##  Class :character        1st Qu.:13.00   Class :character   Refusal   :  34
##  Mode  :character        Median :14.00   Mode  :character   No        :7148
##                          Mean   :13.99                      Yes       :1795
##                          3rd Qu.:15.00                      NA's      :   2
##                          Max.   :16.00
##
##  industry_code            marital_status total_num_incarcerations
##  Length:8984        Never-married:2766   Length:8984
##  Class :character   Married      :3066   Class :character
##  Mode  :character   Other        : 840   Mode  :character
##                     NA's         :2312
##
##
##
##  college_type          trustful_or_not
##  Length:8984        Distrustful: 588
##  Class :character   Trustful   :4287
##  Mode  :character   NA's       :4109
##
##
##
##
```

For our analysis, we selected 15 key variables from the original dataset, nlsy, which contains 95 variables and 8,984 rows of data. These selected variables were stored in a new dataframe, nlsy_df, where all subsequent analyses were conducted. To ensure accuracy in our results, we addressed missing and negative values during the cleaning process.

The variable total_income_2016 presented topcoded values, representing the top 2% of earners. To account for this, we performed analyses both with and without truncating these topcoded values. Truncation resulted in a more accurate mean for this variable. Additionally, the summary of nlsy_df provides an overview of the data types and characteristics of the selected variables.

For better interpretability, categorical variables like gender, marital_status, and trustful_or_not were converted into factors with appropriate levels. This preparation ensures that the dataset is ready for thorough and precise analysis in the subsequent sections.

PART 2 - RELATIONSHIP AND TREND ANALYSIS

INCOME vs GENDER

```
# Summarize income by gender including topcoded variable, excluding NA

gender_income_summary <- nlsy_df %>%
  filter(!is.na(total_income_2016)) %>% # Exclude rows with NA in income
  group_by(gender) %>%
  summarize(
    count = n(),
    mean_income = round(mean(total_income_2016, na.rm = TRUE)),
    median_income = round(median(total_income_2016, na.rm = TRUE)),
    sd_income = round(sd(total_income_2016, na.rm = TRUE))
  )
print(gender_income_summary)
```

```
## # A tibble: 2 × 5
##   gender count mean_income median_income sd_income
##   <fct>  <int>       <dbl>         <dbl>     <dbl>
## 1 Male    2621       57203         47000     44712
## 2 Female  2470       41279         35000     34047
```

```
# Summarize income by gender excluding topcoded variable, excluding NA

# Excuding the top coded values
nlsy_no_topcoded <- nlsy_df %>%
  filter(!is.na(total_income_2016)) %>%
  filter(total_income_2016 != max(total_income_2016, na.rm = TRUE))

gender_income_summary <- nlsy_no_topcoded %>%
  group_by(gender) %>%
  summarize(
    count = n(), # Count rows after filtering
    mean_income = round(mean(total_income_2016, na.rm = TRUE)),
    median_income = round(median(total_income_2016, na.rm = TRUE)),
    sd_income = round(sd(total_income_2016, na.rm = TRUE))
  )
print(gender_income_summary)
```

```
## # A tibble: 2 × 5
##   gender count mean_income median_income sd_income
##   <fct>  <int>       <dbl>         <dbl>     <dbl>
## 1 Male    2530       50776         45000     29681
## 2 Female  2440       38886         35000     26494
```

The above tabular summary for income by gender shows that males have a higher average income ($57,203) compared to females ($41,279), with a median income of $47,000 for males and $35,000 for females. The standard deviation is also higher for males ($44,712) than for females ($34,047), indicating greater income variability among males. Overall, males consistently earn more on average and have more variation in income compared to females.

```
plot1 <- ggplot(nlsy_df, aes(x = gender, y = total_income_2016, fill = gender)) +
  geom_boxplot(show.legend = FALSE, na.rm = TRUE) +
  labs(
    title = "Fig1.Income Distribution by Gender",
    subtitle = "Including Topcoded values",
    x = "Gender",
    y = "Total Income"
  )

# Plot of Income with Gender excluding topcoded values
nlsy_no_topcoded <- nlsy_df %>%
  filter(total_income_2016 != max(total_income_2016, na.rm = TRUE))

plot2 <- ggplot(nlsy_no_topcoded, aes(x = gender, y = total_income_2016, fill = gender))
+
  geom_boxplot(show.legend = FALSE) +
  labs(
    title = "Fig2.Income Distribution by Gender",
    subtitle = "Excluding Topcoded values",
    x = "Gender",
    y = "Total Income"
  )

grid.arrange(plot1, plot2, ncol = 2)
```

The box plot shows the distribution of income by gender, highlighting key differences between males and females. Males have a higher median income (indicated by the middle line of the box) compared to females. The overall spread of income is wider for males, as shown by the larger interquartile range and longer whiskers, reflecting greater income variability. From this boxplot we can say that males tend to earn more and have more income variability compared to females.

INCOME vs RACE

```
# Calculate mean income by race and gender, excluding rows with NA in income
income_by_race_gender <- nlsy_df %>%
  filter(!is.na(total_income_2016)) %>% # Exclude rows where income is NA
  group_by(race, gender) %>%
  summarize(
    average_income = round(mean(total_income_2016, na.rm = TRUE)), # Calculate mean inco
me
    count = n() # Count rows after filtering
  )
```

```
## `summarise()` has grouped output by 'race'. You can override using the
## `.groups` argument.
```

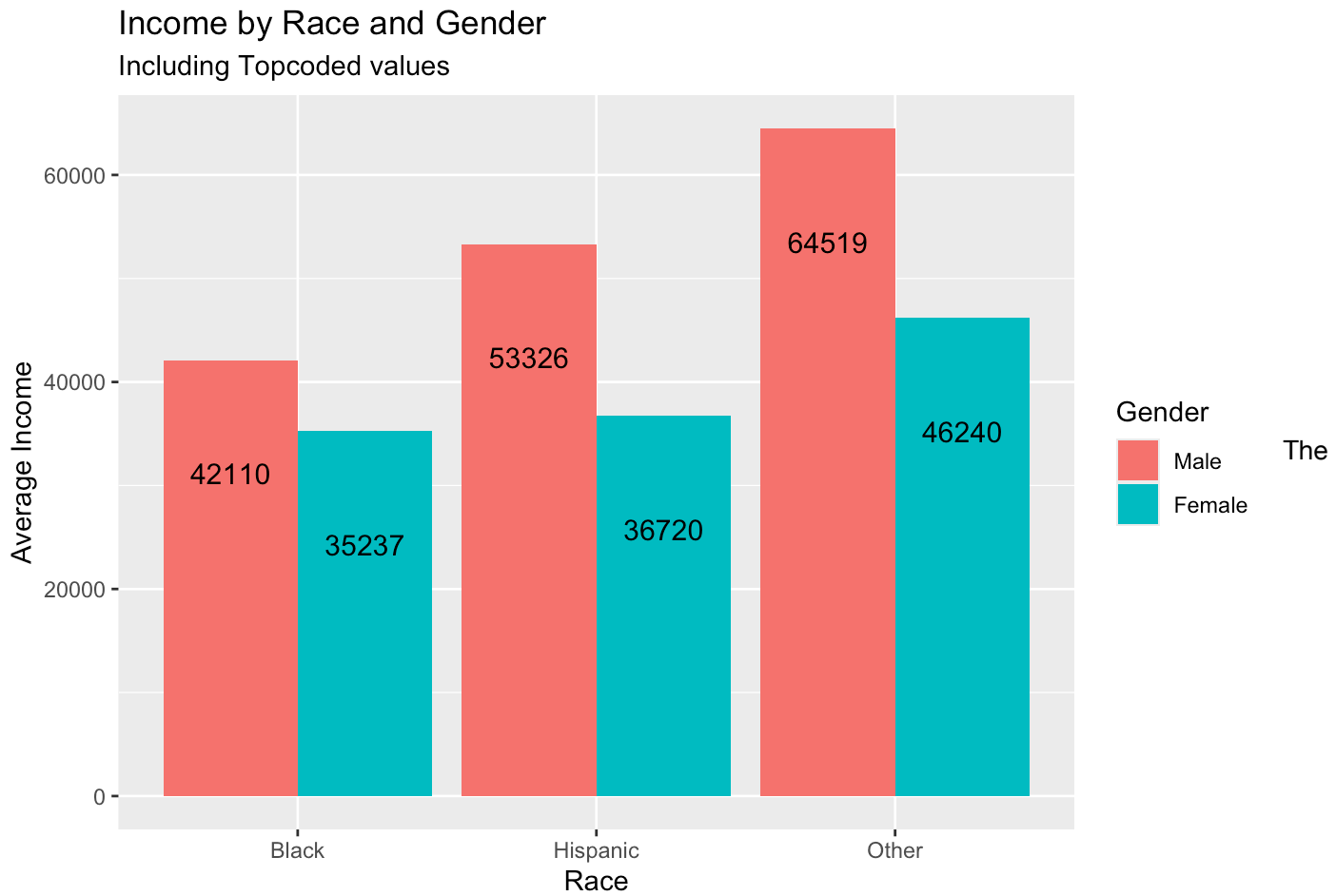
```
# Display the tabular summary
income_by_race_gender
```

```
## # A tibble: 6 × 4
## # Groups:   race [3]
##   race     gender average_income count
##   <fct>    <fct>           <dbl> <int>
## 1 Black    Male            42110   585
## 2 Black    Female          35237   669
## 3 Hispanic Male            53326   542
## 4 Hispanic Female          36720   514
## 5 Other    Male            64519  1494
## 6 Other    Female          46240  1287
```

From the above we can see that the average income and the number of individuals for each race-gender combination, highlighting clear disparities. Males consistently earn more than females across all racial groups, with the income gap being the largest in the “Other” group ($18,279 difference) and smallest in the “Black” group ($6,873 difference). Among racial groups, individuals in the “Other” category have the highest average income for both males ($64,519) and females ($46,240), followed by “Hispanic” and “Black” groups, where males earn $53,326 and $42,110, respectively. Within each race, males and females are nearly equally represented, though males slightly outnumber females in most cases. This data reveals significant income disparities by both race and gender.

```
# Create the bar plot for income by race and gender including topcoded values
ggplot(nlsy_df, aes(x = race, y = total_income_2016, fill = gender)) +
  geom_bar(stat = "summary", fun = "mean", position = "dodge", show.legend = TRUE, na.rm
= TRUE) +
  geom_text(
    aes(label = round(..y..)),
    stat = "summary", fun = "mean", position = position_dodge(width = 0.9), vjust = 6, s
ize = 4
  ) +
  labs(
    title = "Income by Race and Gender",
    subtitle = "Including Topcoded values",
    x = "Race",
    y = "Average Income",
    fill = "Gender"
  )
```

```
## Warning: Removed 3893 rows containing non-finite outside the scale range
## (`stat_summary()`).
```



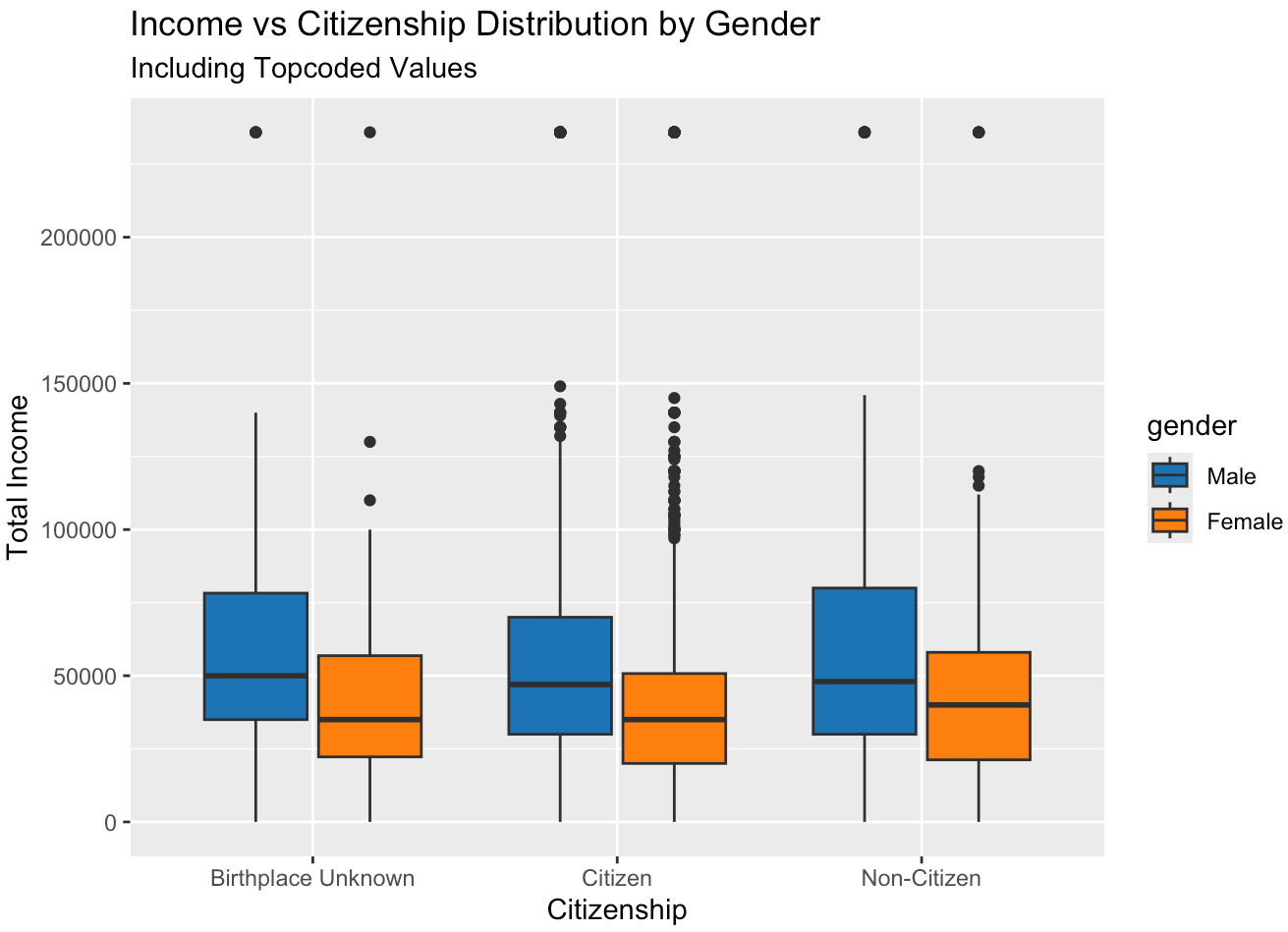
The bar plot shows the average income by race and gender, with distinct bars for males and females within each racial group. The “Other” group has the highest average income for both males ($64,519) and females ($46,240), followed by “Hispanic” and “Black” groups. Males consistently earn more than females across all groups, with the gender gap largest in the “Other” group ($18,279) and smallest in the “Black” group ($6,873). The plot highlights significant income disparities both between races and genders.

CITIZENSHIP vs INCOME

```
# Summarize income by citizenship status by gender distribution
# The table below shows the count of respondents for different category of citizenship, mean_income, median_income, and standard deviation. Here we can see that the mean income for citizen is less than the non-citizen and people who don't know their birthplace have the highest mean income. However, the median income for non-citizen is the highest.Also, standard deviation for citizens is lowest compared to non-citizen and unknown birthplace category respondents. The NA consists of valid skip which means the question was not app licable to those respondants and hence we have omitted it. Therefore, the data we see fo r citizenship status in table or the graph will be for the 7942 responses as out of 8984 we omitted 1042 rows. Also, the male and female distribution is shown for income based o n citizenship.

# Omit rows where Citizenship is NA
citizen_nlsy_df <- nlsy_df %>%
  filter(!is.na(citizenship))

citizenship_income_summary <- citizen_nlsy_df %>%
  group_by(citizenship) %>%
  summarize(
    count = n(),
    count_male = sum(gender == "Male", na.rm = TRUE),
    count_female = sum(gender == "Female", na.rm = TRUE),
    mean_income = mean(total_income_2016, na.rm = TRUE),
    median_income = median(total_income_2016, na.rm = TRUE),
    sd_income = sd(total_income_2016, na.rm = TRUE)
  )
print(citizenship_income_summary)
```

```
## # A tibble: 3 × 7
##   citizenship  count count_male count_female mean_income median_income sd_income
##   <chr>        <int>      <int>        <int>       <dbl>         <dbl>     <dbl>
## 1 Birthplace …   279        147          132      55871.        42914.    48567.
## 2 Citizen       6869       3538         3331      48879.        40000     39807.
## 3 Non-Citizen    794        396          398      54610.        45000     46762.
```

```
# Analyzing income with citizenship based on Gender. Here, citizenship is categorical va
riable. The graph shows a range from $0-$200,000. The pink box represent male and turquo
ise boxes represent females. Each box plot shows median, quantiles, and outliers. The gr
aphs show population into four groups called birthplace unknown, citizen, and non-citize
n. Males show high median incomes across all citizenship categories as compared to femal
es in turquoise boxes. The median income for males is around $50K, while for females it
is lower around $35K. We can also say that the 1st quartile for females lies even below
the 25K mark, while for males, the 1st quantile is above the 30K mark. The outliers show
dots above the whiskers reaching up to $235K. We can also say that the 1st quantile for
female lies even below 25K mark while for male 1st quantile is above 30K mark. We can sa
y that the income disparity between genders appear to be relatively consistent across di
fferent citizenship status. Citizens and non-citizens show a kind of similar distributio
n of income patterns. Also, the boxes are larger for males, which indicates more income
variability for them. Having topcoded values gives more accurate median. If we truncate
the top 2%, the mean might underestimate the true average.
ggplot(citizen_nlsy_df, aes(x = citizenship, y = total_income_2016, fill = gender)) +
  geom_boxplot(na.rm = TRUE) +
  scale_fill_manual(values = c("Male" = "#1f77b4", "Female" = "#ff7f0e")) +
  labs(
    title = "Income vs Citizenship Distribution by Gender",
    subtitle = "Including Topcoded Values",
    x = "Citizenship",
    y = "Total Income"
  )
```



AGE vs INCOME

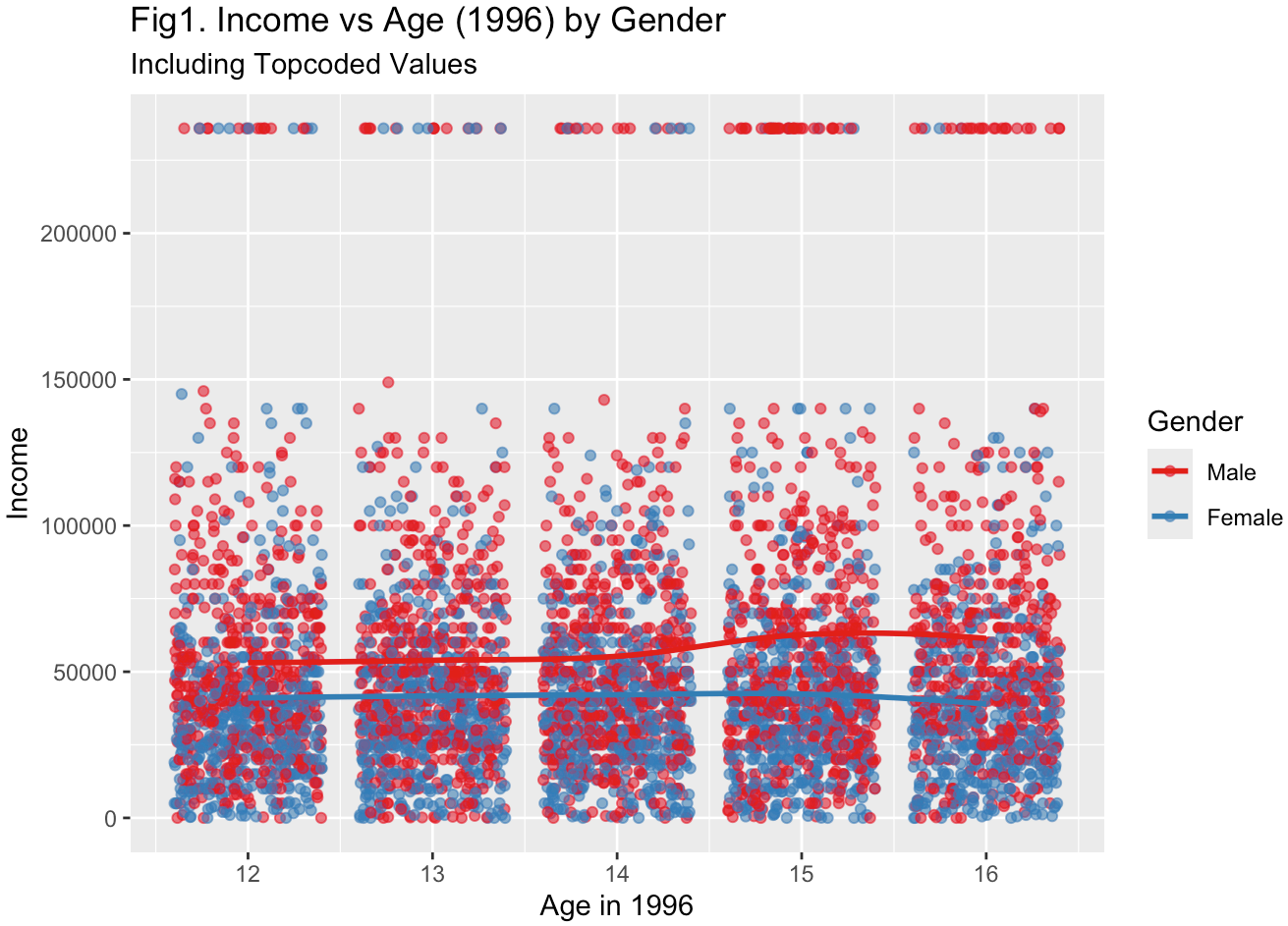
```
# This will give us a tabular representation of age, mean, median, and sd. The possible age values are 12 to 16 where there are reasonable number of respondents for each age category. We can see that mean income is the highest for respondents of age 15 and lowest for age 12. The lowest median income is again for 12 years respondants and is 40000 for respondents of age 13, 14, and 16. We have filtered the rows where income is NA to get accurate number of male and female having respective mean, median, and sd. The table shows that there are 523 males and 499 females with mean income of 47192.36, median 39500, and sd 39106.02. Similarly we have data for different age across our respondents for a total of 2621 males and 2470 females.
age_income_summary <- nlsy_df %>%
  filter(!is.na(total_income_2016)) %>% # Exclude rows with NA in income
  group_by(age_1996) %>%
  summarize(
    count = n(),
    count_male = sum(gender == "Male", na.rm = TRUE),
    count_female = sum(gender == "Female", na.rm = TRUE),
    mean_income = mean(total_income_2016, na.rm = TRUE),
    median_income = median(total_income_2016, na.rm = TRUE),
    sd_income = sd(total_income_2016, na.rm = TRUE)
  )
print(age_income_summary)
```

```
## # A tibble: 5 × 7
##   age_1996 count count_male count_female mean_income median_income sd_income
##      <dbl> <int>      <int>        <int>       <dbl>         <dbl>     <dbl>
## 1       12  1022        523          499      47192.         39500    39106.
## 2       13  1044        547          497      48114.         40000    37696.
## 3       14  1050        530          520      48774.         40000    37811.
## 4       15  1041        552          489      53142.         41000    45829.
## 5       16   934        469          465      50206.         40000    42338.
```

```
# Filter the data before creating the plot to remove the topcoded values. As we know there are 121 people having topcoded which is 2% level. There income has been average to 235884 and we plan to analyse the data for both with topcoded and without.
no_topcoded_nlsy_df <- nlsy_df %>%
  filter(total_income_2016 < 235884)

# Graph with topcoded income values
ggplot(nlsy_df, aes(x = age_1996, y = total_income_2016, color = gender)) +
  geom_jitter(alpha = 0.6, width = 0.4, height = 0, na.rm = TRUE) +
  geom_smooth(method = "loess", se = FALSE) +
  labs(
    title = "Fig1. Income vs Age (1996) by Gender",
    subtitle = "Including Topcoded Values",
    x = "Age in 1996",
    y = "Income"
  ) +
  scale_color_brewer(palette = "Set1", name = "Gender")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 3893 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```




```
# Graph with no-topcoded income values will exclude the people with higher income and th
is will help us have even more accurate data as the topcoded income for 2% level is aver
age and it might interfere in our analysis.
ggplot(no_topcoded_nlsy_df, aes(x = age_1996, y = total_income_2016, color = gender)) +
  geom_jitter(alpha = 0.6, na.rm = TRUE, width = 0.3, height = 0) +
  geom_smooth(method = "loess", se = FALSE) +
  labs(
    title = "Fig2. Income vs Age (1996) by Gender",
    subtitle = "Excluding Topcoded Values",
    x = "Age in 1996",
    y = "Income"
  ) +
  scale_color_brewer(palette = "Set2", name = "Gender")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```


```
# Fig1. illustrates a plot for topcoded income vs age distributed by gender. Red point i
s for male and blue is for female. It shows income in the range $0-$236K. The age of res
pondents here is as of 12/31/1996. The graphs shows that the income for male is consiste
ntly higher than that for females across different ages 12,13,14,15,16. The trend line i
ndicates that income is approximately $50K for males of age 12 to 14 whereas for females
it is around $40K from the age 12 to 15. However, we see that income increased to $60K
(approx) for male above the age of 14 and decreased to $35K(approx) for females above th
e age of 15.
# Fig2. depicts a plot for excluding topcoded income for 2% level. It shows income range
in $0-$150K. Green point is for male and orange is for female. Here as the data has topc
oded values truncated, the remains similar to the previous one. Male income starts from
a little below $50K while females start around $37K which is lower as compared to fig1.
The income goes slightly up for males above 14 and income goes down for females above th
e age of 15.
```

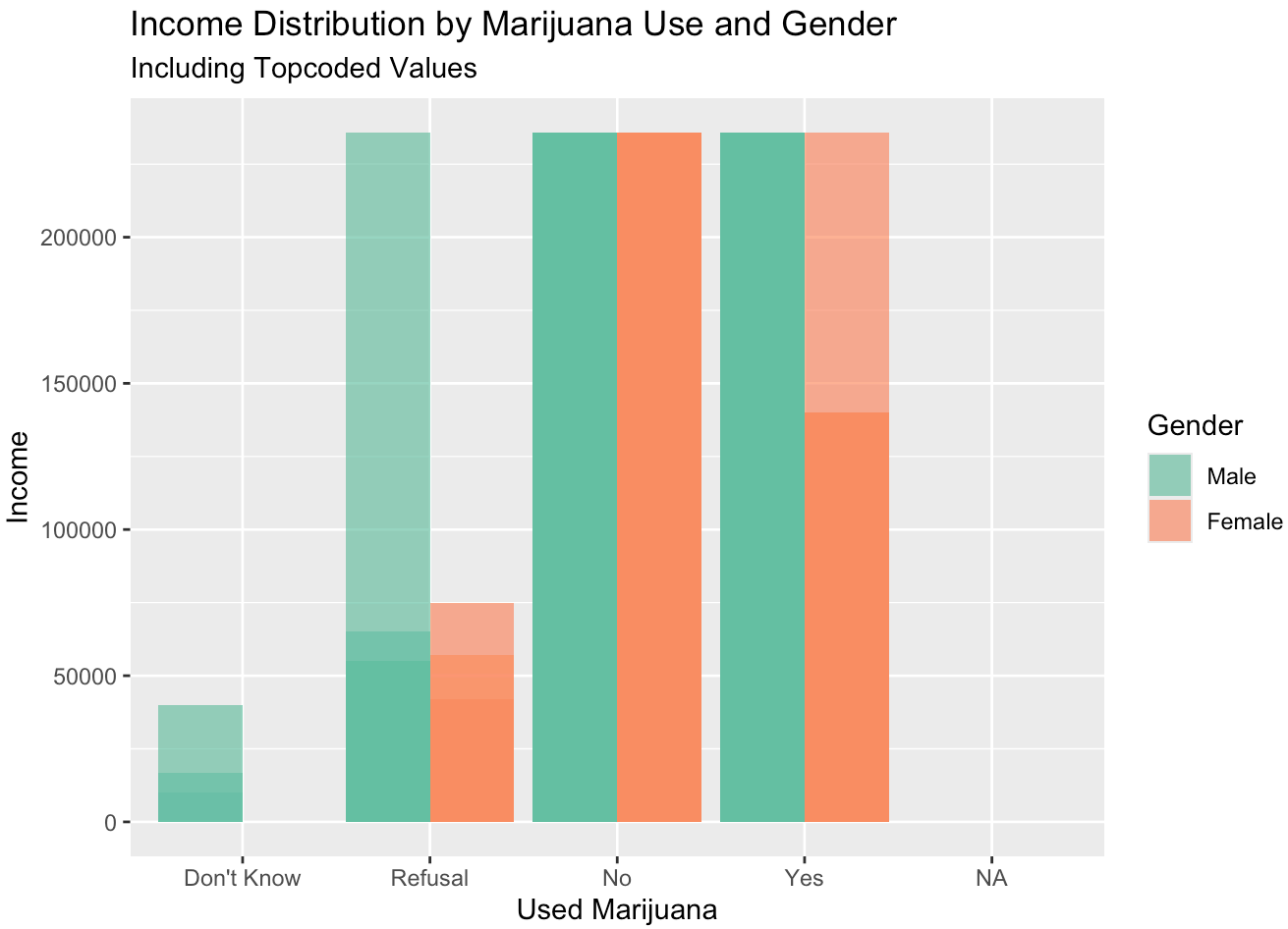
USED MARIJUANA vs INCOME

```
# This tabular below shows a tibble of 4x7 where we can see number of male and female ha
ving mean income based on usage of marijuana. There is a huge number of people 4078 who
do not use marijuana and have a mean income of 49843.55. People who use marijuana consti
tutes to 994 out of which 534 are male and 460 are female who together have a mean incom
e of 47958 which is lower when compared to people who don't use marijuana. However, cate
gory using marijuana and not using marijuana have same median of 40000. There are 3 male
s who say don't know have lowest mean income of 22266.67 and 11 male and 5 female refuse
d to answer this question have the highest mean income of 55524.
mari_income_summary <- nlsy_df %>%
  filter(!is.na(total_income_2016)) %>% # Exclude rows with NA in income
  group_by(used_marijuana) %>%
  summarize(
    count = n(),
    count_male = sum(gender == "Male", na.rm = TRUE),
    count_female = sum(gender == "Female", na.rm = TRUE),
    mean_income = mean(total_income_2016, na.rm = TRUE),
    median_income = median(total_income_2016, na.rm = TRUE),
    sd_income = sd(total_income_2016, na.rm = TRUE)
  )
print(mari_income_summary)
```

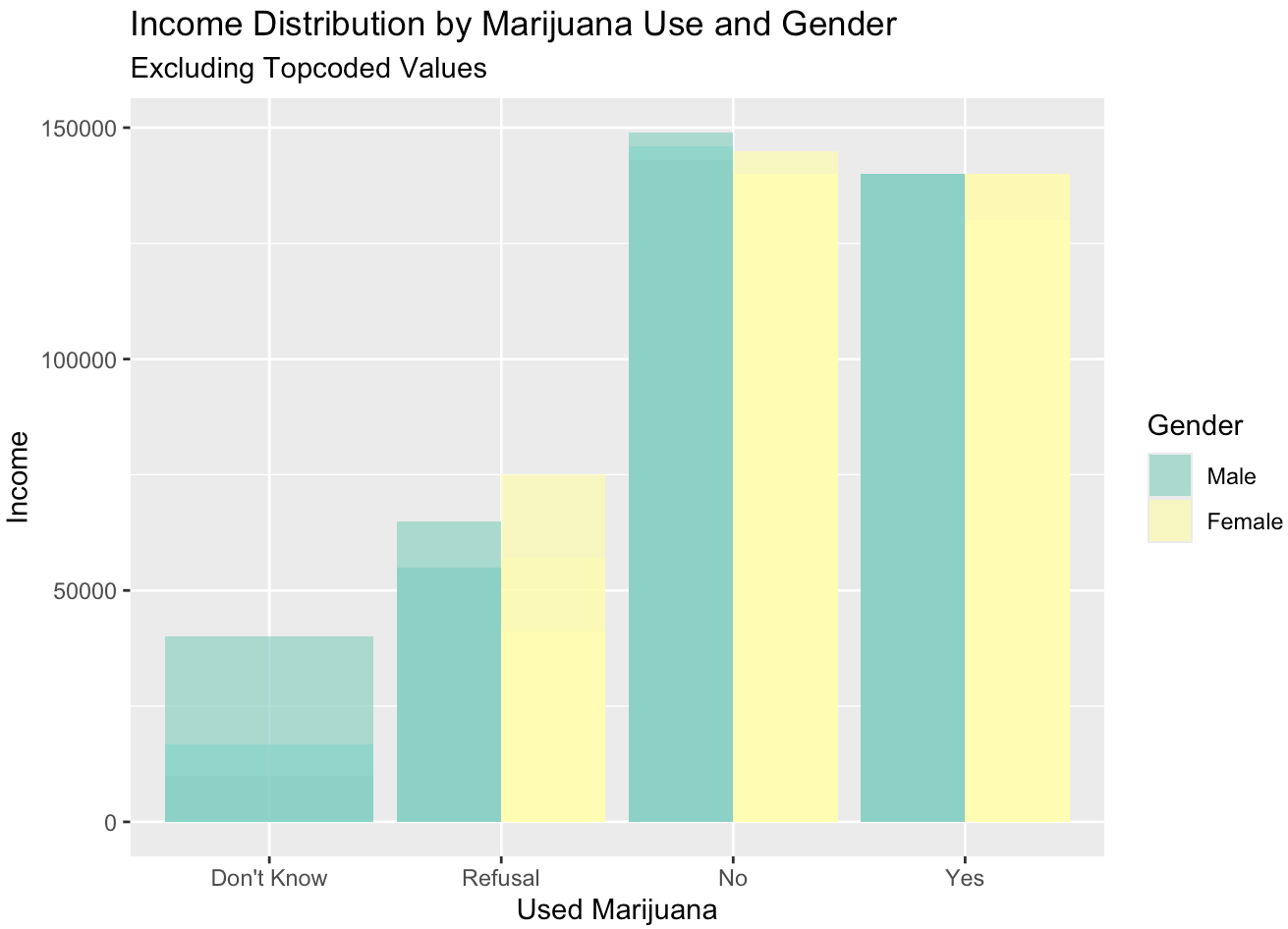
```
## # A tibble: 4 × 7
##   used_marijuana count count_male count_female mean_income median_income
##   <fct>          <int>      <int>        <int>       <dbl>         <dbl>
## 1 Don't Know         3          3            0      22267.         16800
## 2 Refusal           16         11            5      55524          43000
## 3 No              4078       2073         2005      49844.         40000
## 4 Yes              994        534          460      47958.         40000
## # ℹ 1 more variable: sd_income <dbl>
```

```
# This graph shows that males consistently show higher income levels across all marijuan
a use categories, with the most pronounced differences in the "Refusal" category. The in
come ranges from approximately $0 to $200,000. Don't know category shows lowest income w
ith only male respondents, refusal shows a significant gender gap where males are making
higher income than females, the No category shows both genders have similar high income
with huge number of population lying in this category which can be seen by dark green an
d orange color and the Yes category also shows similar high income but for lower density
of people. Hence, the data suggests that marijuana use itself doesn't show a clear negat
ive correlation with income, as both "Yes" and "No" categories display similar income le
vels. The most significant variations appear to be driven by gender rather than marijuan
a use status.

ggplot(nlsy_df, aes(x = used_marijuana, y = total_income_2016, fill = gender)) +
  geom_col(position = "dodge", alpha = 0.7) +
  labs(
    title = "Income Distribution by Marijuana Use and Gender",
    subtitle = "Including Topcoded Values",
    x = "Used Marijuana",
    y = "Income"
  ) +
  scale_fill_brewer(palette = "Set2", name = "Gender")
```

```
## Warning: Removed 3893 rows containing missing values or values outside the scale rang
e
## (`geom_col()`).
```




```
# Similarly, let's see if truncating topcoded values show a difference.
ggplot(no_topcoded_nlsy_df, aes(x = used_marijuana, y = total_income_2016, fill = gende
r)) +
  geom_col(position = "dodge", alpha = 0.7) +
  labs(
    title = "Income Distribution by Marijuana Use and Gender",
    subtitle = "Excluding Topcoded Values",
    x = "Used Marijuana",
    y = "Income"
  ) +
  scale_fill_brewer(palette = "Set3", name = "Gender")
```




```
# If we analyze the graph we can say that the trend of income shows some variation when compared to graph with topcoded values. After removing extreme values, the income differences become more moderate. Gender disparities persist but are less dramatic when compared to graph with topcoded values. Here, the male in refusal category shows lower income as compared to female, for No category males earn slightly more than female and the highest income can be near approx 150K for males and 140K for females, for Yes category we see that both have similar income range.
```

IS SAD DEPRESSED UNHAPPY vs INCOME

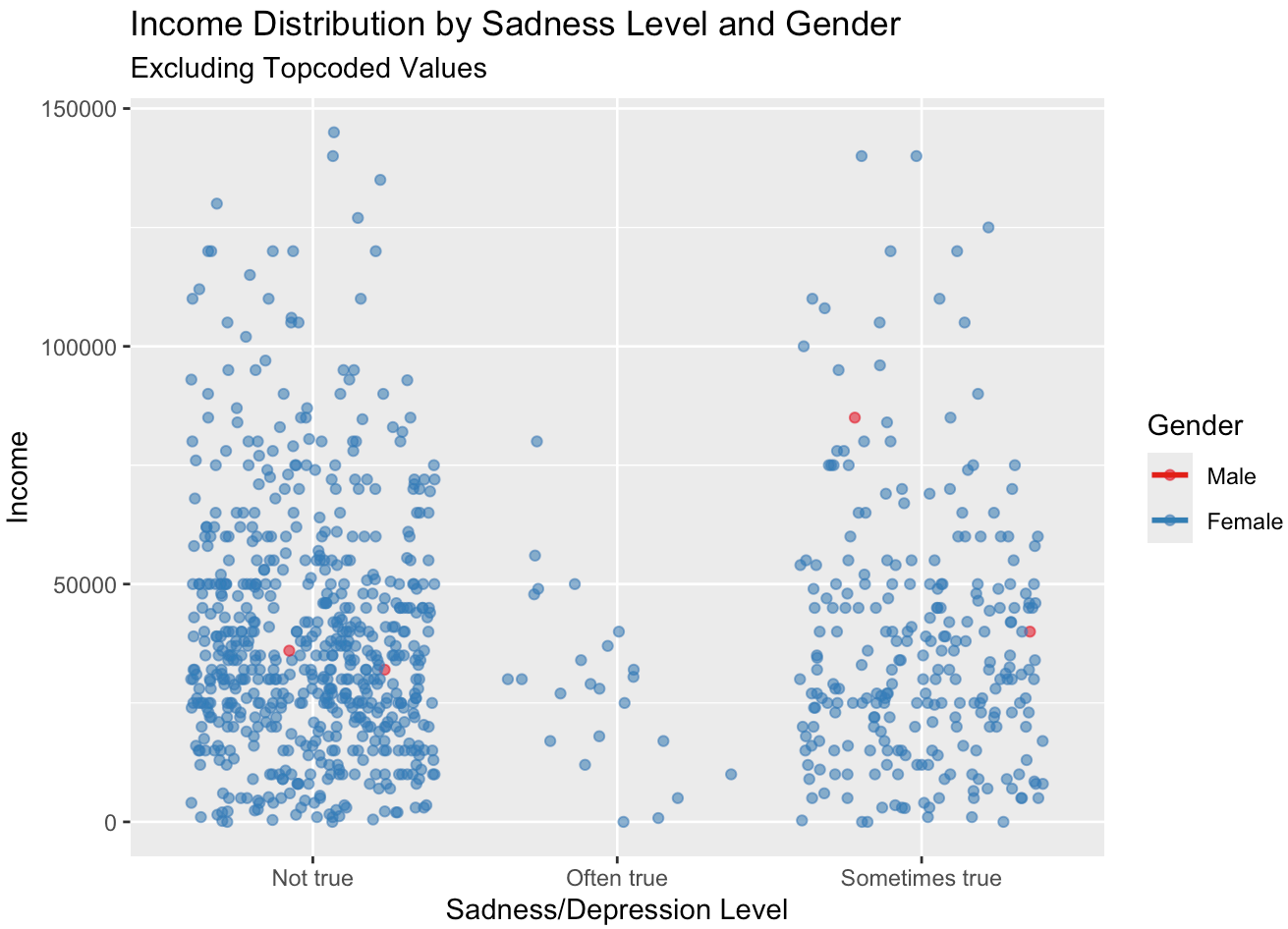
BASED ON GENDER

```
# This table analyzes the relationship between income and reported feelings of sadness/depression, broken down by gender. The data shows 631 responses for 'Not true', 259 responses for 'Sometimes true', and 24 responses for 'Often true'. This question shows sample is heavily skewed towards females, with only 4 male respondents. Mean income level is highest for people who don't feel depressed and lowest income is for respondents who often feel depressed or unhappy. Overall we get the idea that people who reported frequently feeling sadness or depression show lower income level among females. There is huge mean income gap of $13,383 between 'Not true' and 'Often true' category respondents.
sad_income_summary <- nlsy_df %>%
  filter(!is.na(total_income_2016)) %>% # Exclude rows with NA in income
  filter(!is.na(is_sad_depressed_unhappy_f)) %>% # Exclude rows with NA in is_sad
  group_by(is_sad_depressed_unhappy_f) %>%
  summarize(
    count = n(),
    count_male = sum(gender == "Male", na.rm = TRUE),
    count_female = sum(gender == "Female", na.rm = TRUE),
    mean_income = mean(total_income_2016, na.rm = TRUE),
    median_income = median(total_income_2016, na.rm = TRUE),
    sd_income = sd(total_income_2016, na.rm = TRUE)
  )
print(sad_income_summary)
```

```
## # A tibble: 3 × 7
##   is_sad_depressed_unh…¹ count count_male count_female mean_income median_income
##   <chr>                  <int>      <int>        <int>       <dbl>         <dbl>
## 1 Not true                 631          2          629      42763.         35000
## 2 Often true                24          0           24      29381.         29500
## 3 Sometimes true           259          2          257      40826.         32000
## # ℹ abbreviated name: ¹is_sad_depressed_unhappy_f
## # ℹ 1 more variable: sd_income <dbl>
```

```
# Removing all NAs from is_sad variable and apply it to no-topcoded df
sad_income_df <- no_topcoded_nlsy_df %>%
  filter(!is.na(is_sad_depressed_unhappy_f))

# Here, we plan to exclude the topcoded values to avoid truncation of data. The graph di
splays the relationship between income distribution, sadness/depression levels, and gend
er using a scatter plot. The visualization shows income levels plotted against four cate
gories of sadness/depression namely not true, sometimes true, and often true. We can say
that income variability is the highest in 'Not true' and lowest in 'Often true'. 'Someti
mes true' shows moderate spread of income ranging $0-$100000, 'Often true' shows smalles
t cluster with incomes ranging $0-$50000, and 'Not true' shows income ranging from $0-$1
50000. The data suggests that sadness/depression levels show some correlation with lower
income levels, particularly for those reporting frequent sadness.
ggplot(sad_income_df, aes(x = is_sad_depressed_unhappy_f, y = total_income_2016, color =
gender)) +
  geom_jitter(alpha = 0.6, width = 0.4, height = 0, na.rm = TRUE) +
  geom_smooth(method = "loess", se = FALSE) +
  labs(
    title = "Income Distribution by Sadness Level and Gender",
    subtitle = "Excluding Topcoded Values",
    x = "Sadness/Depression Level",
    y = "Income"
  ) +
  scale_color_brewer(palette = "Set1", name = "Gender")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



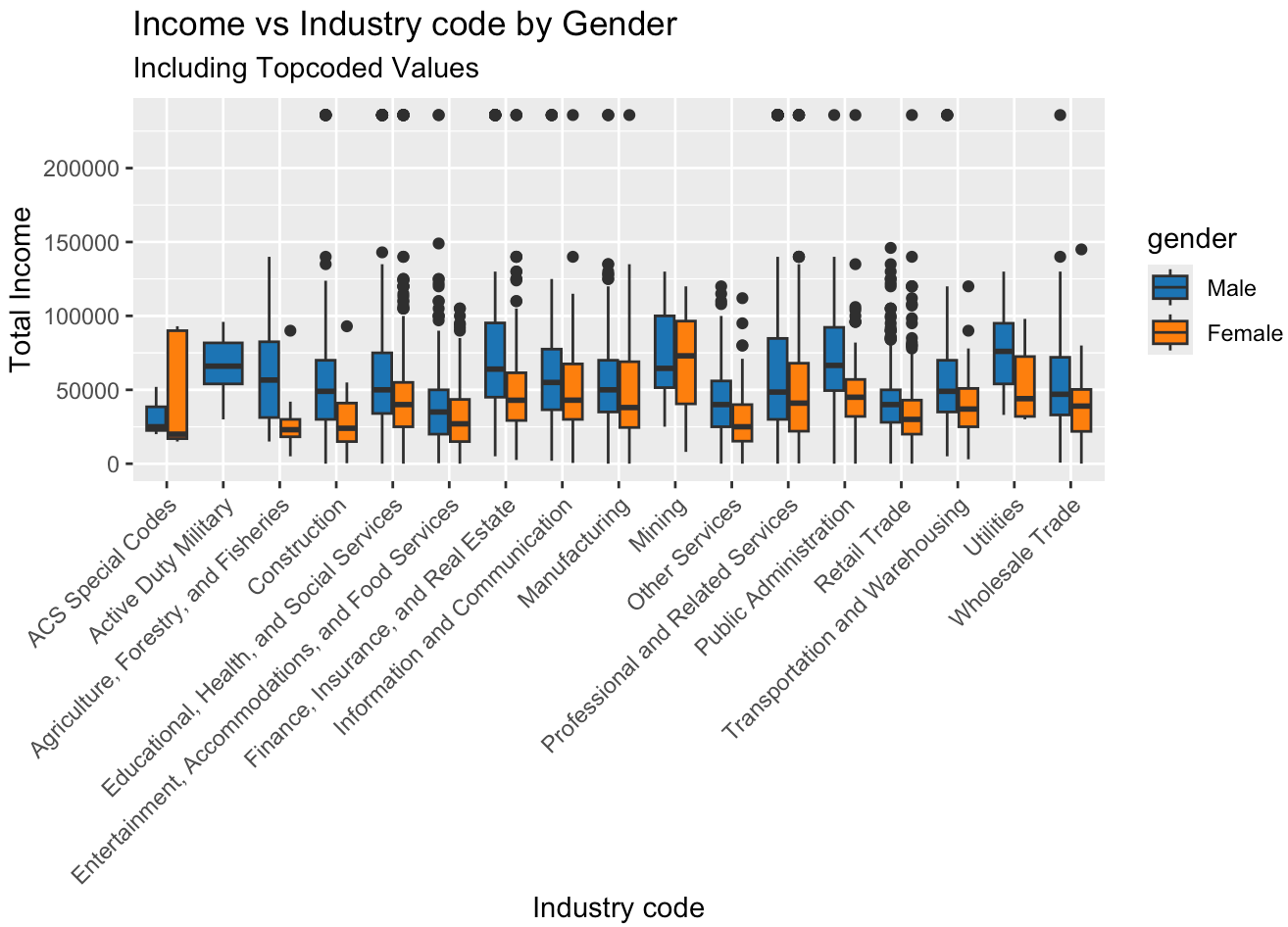
INDUSTRY CODE vs INCOME

```
# Filtering NA from Industry Code
industry_code_nlsy_df <- nlsy_df %>%
  filter(!is.na(industry_code))%>%
  filter(!is.na(total_income_2016))

# Summarizing the Industry Code
industry_code_income_summary <- industry_code_nlsy_df %>%
group_by(industry_code) %>%
summarize(
count = n(),
count_male = sum(gender == "Male", na.rm = TRUE),
count_female = sum(gender == "Female", na.rm = TRUE),
mean_income = mean(total_income_2016, na.rm = TRUE),
median_income = median(total_income_2016, na.rm = TRUE),
sd_income = sd(total_income_2016, na.rm = TRUE)
)%>%
  arrange(desc(count))
print(industry_code_income_summary)
```

```
## # A tibble: 17 × 7
##    industry_code         count count_male count_female mean_income median_income
##    <chr>                 <int>      <int>        <int>       <dbl>         <dbl>
##  1 Educational, Health,…  1005        243          762      48459.         41000
##  2 Professional and Rel…   562        342          220      61159.         45500
##  3 Entertainment, Accom…   484        233          251      36019          30000
##  4 Retail Trade            472        239          233      39551.         35000
##  5 Manufacturing           332        245           87      53811.         47000
##  6 Finance, Insurance, …   318        144          174      66746.         50000
##  7 Construction            260        239           21      54822.         45000
##  8 Other Services          215        113          102      36834.         32000
##  9 Public Administration   201        116           85      61657          55000
## 10 Transportation and W…   142        104           38      55030.         45000
## 11 Information and Comm…   118         67           51      62443.         50000
## 12 Wholesale Trade         117         93           24      52171.         45000
## 13 Agriculture, Forestr…    32         18           14      46431.         32500
## 14 Mining                   27         24            3      72778.         65000
## 15 Utilities                23         17            6      71378.         75000
## 16 Active Duty Military     18         18            0      67056.         66000
## 17 ACS Special Codes         8          3            5      41500          22500
## # ℹ 1 more variable: sd_income <dbl>
```

The table above shows the count of respondents for different category of industry code, mean_income, median_income, and standard deviation. Here we can see that the mean income for different industry codes. The mean income for Mining is the highest and lowest for Entertainment, Accommodations, and Food Services. However, the median income for Utilities is the highest. Also, standard deviation for Active Duty Military is lowest compared to Finance, Insurance, and Real Estate respondents. The NA consists of valid skip which means the question was not applicable to those respondants and hence we have omitted it. Therefore, the data we see for industry codes in table or the graph will be for the 4650 responses as out of 8984 we omitted 4334 rows.

```
# Create the bar plot for total income by industry code and gender
ggplot(industry_code_nlsy_df, aes(x = industry_code, y = total_income_2016, fill = gende
r)) +
  geom_boxplot(na.rm=TRUE)+
  scale_fill_manual(values = c("Male" = "#1f77b4", "Female" = "#ff7f0e")) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1)) +
  labs(title = "Income vs Industry code by Gender",
       subtitle = "Including Topcoded Values",
    x = "Industry code",
    y = "Total Income")
```



The above graph illustrates a plot for topcoded income vs industry code by gender. The blue boxes are for male and the orange ones are for females. The graphs shows that the income for male is consistently higher than that for females across different industry codes apart from ACS Special codes where it can be seen that females earn more than males.

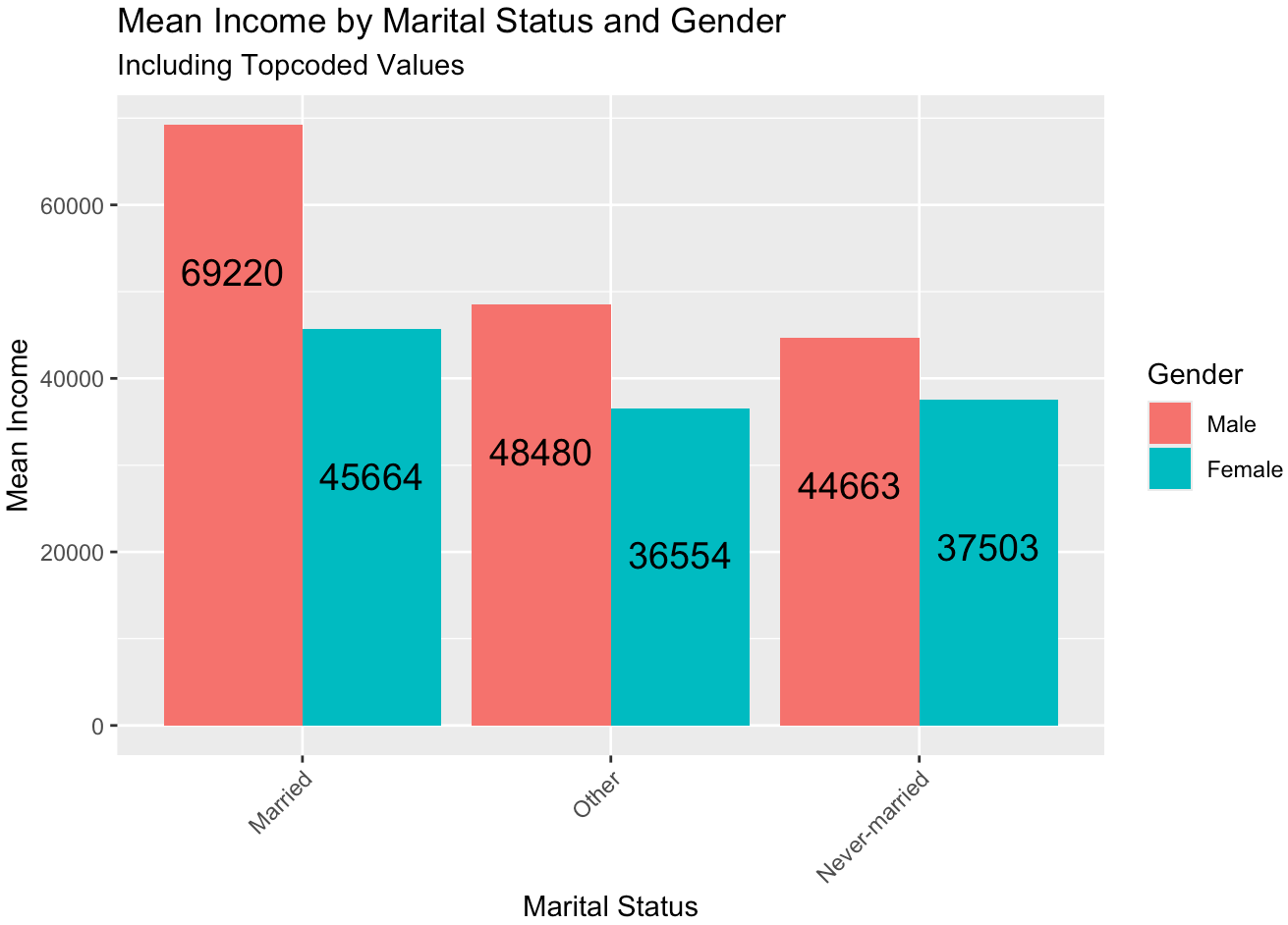
MARTIAL STATUS vs INCOME

```
# Filtering NA from marital status
marital_status_nlsy_df <- nlsy_df %>%
  filter(!is.na(marital_status))%>%
filter(!is.na(total_income_2016))

# Summarizing the marital status
marital_status_income_summary <- marital_status_nlsy_df %>%
group_by(marital_status) %>%
summarize(
count = n(),
count_male = sum(gender == "Male", na.rm = TRUE),
count_female = sum(gender == "Female", na.rm = TRUE),
mean_income = round(mean(total_income_2016, na.rm = TRUE)),
median_income = round(median(total_income_2016, na.rm = TRUE)),
sd_income = round(sd(total_income_2016, na.rm = TRUE)), .groups = "drop"
)%>%
  arrange(desc(count))
print(marital_status_income_summary)
```

```
## # A tibble: 3 × 7
##   marital_status count count_male count_female mean_income median_income
##   <fct>          <int>      <int>        <int>       <dbl>         <dbl>
## 1 Married         2494       1297         1197       57915         48000
## 2 Never-married   1925       1023          902       41308         35000
## 3 Other            641        282          359       41801         35000
## # ℹ 1 more variable: sd_income <dbl>
```

The table above shows the count of respondents for different category of marital status, mean_income, median_income, and standard deviation. Here we can see that the mean income for different Marital statuses. The mean income for Married is the highest and lowest for Never-Married. However, the median income for Married is the highest but same for Never-Married and Others (combined: Separated, Divorced, Widowed). Also, standard deviation for Never-married is lowest and for Married is highest. The NA consists of valid skip which means the question was not applicable to those respondants and hence we have omitted it. Therefore, the data we see for marital status in table or the graph will be for the 3924 responses as out of 8984 we omitted 5060 rows.

```
# Filter data to exclude missing income values and focus on marital status
nlsy_married <- nlsy_df %>%
  filter(!is.na(total_income_2016)) %>%
  filter(!is.na(marital_status))

# Create the bar plot for mean income by marital status and gender
ggplot(nlsy_married, aes(x = reorder(marital_status, -total_income_2016, FUN = mean), y
= total_income_2016, fill = gender)) +
  geom_bar(stat = "summary", fun = "mean", position = "dodge") +
  geom_text(aes(label = round(..y..)), stat = "summary", fun = "mean", position = positi
on_dodge(width = 0.9), vjust = 6, size= 5 ) +
  labs(
    title = "Mean Income by Marital Status and Gender",
    subtitle = "Including Topcoded Values",
    x = "Marital Status",
    y = "Mean Income",
    fill = "Gender"
  ) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```



The above graph illustrates a plot for topcoded income vs marital status by gender. The red boxes are for male and the blue ones are for females. The graphs shows that the mean income for male is consistently higher than that for females across all marital statuses. Therefore we see that married male earn more than any different

marital status male like never married or other categories. On the other hand the income for female does not vary much although the married females are earning a bit more compared to the never married and other marital status females.

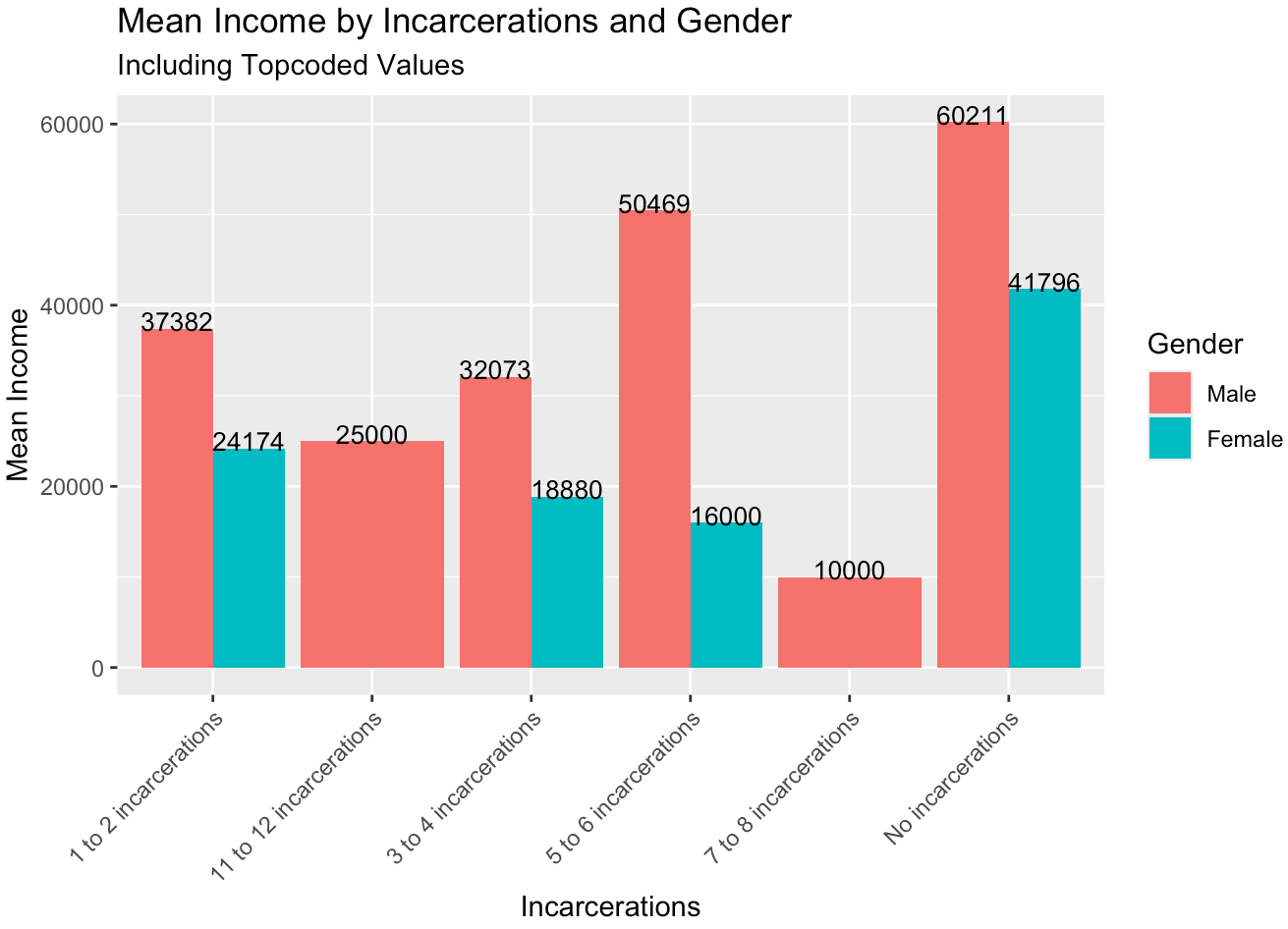
INCARCERATIONS vs INCOME

```
# Filtering NA from total number of Incarcerations
total_num_incarcerations_nlsy_df <- nlsy_df %>%
  filter(!is.na(total_num_incarcerations))%>%
  filter(!is.na(total_income_2016))

# Summarizing total number of Incarcerations
total_num_incarcerations_income_summary <- total_num_incarcerations_nlsy_df %>%
group_by(total_num_incarcerations) %>%
summarize(
count = n(),
count_male = sum(gender == "Male", na.rm = TRUE),
count_female = sum(gender == "Female", na.rm = TRUE),
mean_income = mean(total_income_2016, na.rm = TRUE),
median_income = median(total_income_2016, na.rm = TRUE),
sd_income = sd(total_income_2016, na.rm = TRUE)
)%>%
  arrange(desc(count))
print(total_num_incarcerations_income_summary)
```

```
## # A tibble: 6 × 7
##   total_num_incarcerat…¹ count count_male count_female mean_income median_income
##   <chr>                  <int>      <int>        <int>       <dbl>         <dbl>
## 1 No incarcerations       4681       2280         2401      50766.         42000
## 2 1 to 2 incarcerations    328        270           58      35046.         30000
## 3 3 to 4 incarcerations     62         52           10      29945.         27790
## 4 5 to 6 incarcerations     11         10            1      47335.         16000
## 5 11 to 12 incarceratio…     1          1            0      25000          25000
## 6 7 to 8 incarcerations      1          1            0      10000          10000
## # ℹ abbreviated name: ¹total_num_incarcerations
## # ℹ 1 more variable: sd_income <dbl>
```

The table above shows the count of respondents with total number of incarcerations, mean_income, median_income, and standard deviation. Here we can see that the mean income for No incarcerations is the highest and lowest for 11 to 12 and 9 to 10 incarcerations. Similarly, the median income for No incarcerations is the highest and lowest for 11 to 12 incarcerations. Also, standard deviation for No incarcerations is the highest. We did remove the NA values but the NA values in the summary are likely occurring because some groups have only a single observation (e.g., the “11 to 12 incarcerations” and “9 to 10 incarcerations” categories). When calculating statistics like the standard deviation, the formula requires at least two values to compute a valid result. With only one value in a group, the standard deviation cannot be calculated, so it returns NA. The data we see for different incarcerations in table or the graph will be for the 3900 responses as out of 8984 we omitted 5084 rows.

```
# Filter data to exclude missing income values and focus on incacerations
nlsy_incarcerations <- nlsy_df %>%
  filter(!is.na(total_income_2016)) %>%
  filter(!is.na(total_num_incarcerations))

# Create the bar plot for mean income by  incacerations and gender
ggplot(nlsy_incarcerations, aes(x = total_num_incarcerations, y = total_income_2016, fil
l = gender)) +
  geom_bar(stat = "summary", fun = "mean", position = "dodge") +
  geom_text(aes(label = round(..y..)), stat = "summary", fun = "mean", position = positi
on_dodge(width = 0.9), vjust = 0.1, size= 3.5 ) +
  labs(
    title = "Mean Income by Incarcerations and Gender",
    subtitle = "Including Topcoded Values",
    x = "Incarcerations",
    y = "Mean Income",
    fill = "Gender"
  ) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```



The above graph illustrates a plot for topcoded income vs incarcerations by gender. The red boxes are for male and the blue ones are for females. The graphs shows that the mean income for male is consistently higher than that for females even with different incarcerations they have had. However we do not see any mean income for females in the 7 to 8 and 11 to 12 incarcerations since we only have one respondent and we can assume that it was a male.

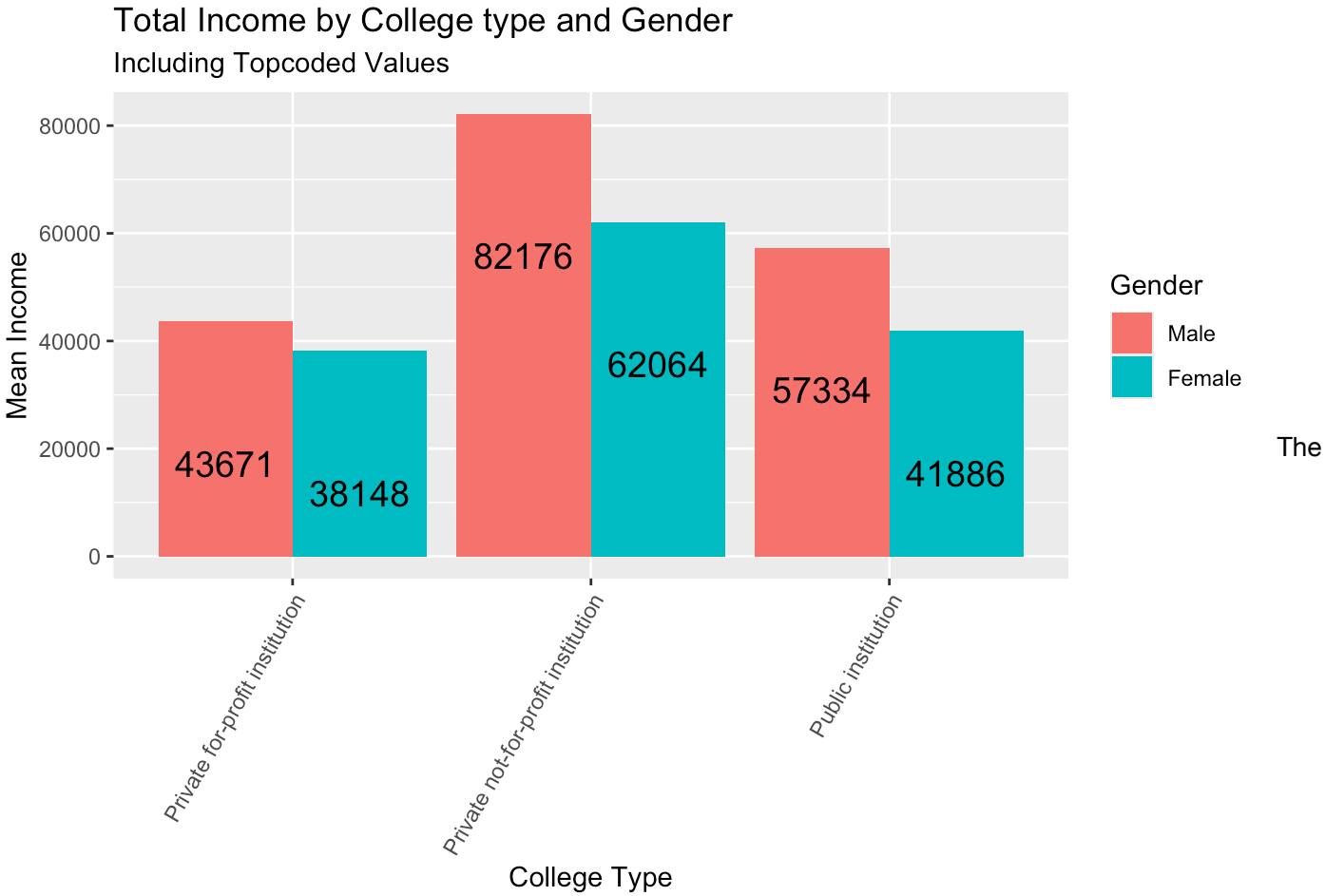
COLLEGE TYPE vs INCOME

```
# Filtering NA from college types
college_type_nlsy_df <- nlsy_df %>%
  filter(!is.na(college_type))%>%
  filter(!is.na(total_income_2016))

# Summarizing college types
college_type_income_summary <- college_type_nlsy_df %>%
group_by(college_type) %>%
summarize(
count = n(),
count_male = sum(gender == "Male", na.rm = TRUE),
count_female = sum(gender == "Female", na.rm = TRUE),
mean_income = mean(total_income_2016, na.rm = TRUE),
median_income = median(total_income_2016, na.rm = TRUE),
sd_income = sd(total_income_2016, na.rm = TRUE)
)
print(college_type_income_summary)
```

```
## # A tibble: 3 × 7
##   college_type count count_male count_female mean_income median_income sd_income
##   <chr>        <int>      <int>        <int>       <dbl>         <dbl>     <dbl>
## 1 Private for…   170         79           91      40715.         37000    24560.
## 2 Private not…   155         63           92      70238.         55000    55861.
## 3 Public inst…   701        300          401      48497.         41000    36493.
```

The table above shows the count of respondents for different category of colleges, mean_income, median_income, and standard deviation. Here we can see that the mean income, median income and sd income for Private not-for-profit institution is the highest and lowest for Private for-profit institution. The NA consists of valid skip which means the question was not applicable to those respondants and hence we have omitted it. Therefore, the data we see for marital status in table or the graph will be for the 7958 responses as out of 8984 we omitted 1026 rows.

```
# Filter data to exclude missing income values and focus on college types
nlsy_college <- nlsy_df %>%
  filter(!is.na(total_income_2016)) %>%
  filter(!is.na(college_type))

# Create the bar plot for mean income by college types and gender
ggplot(nlsy_college, aes(x = college_type, y = total_income_2016, fill = gender)) +
  geom_bar(stat = "summary", fun = "mean", position = "dodge") +
  geom_text(aes(label = round(..y..)), stat = "summary", fun = "mean", position = positi
on_dodge(width = 0.9), vjust = 6, size= 5 ) +
  labs(
    title = "Total Income by College type and Gender",
    subtitle = "Including Topcoded Values",
    x = "College Type",
    y = "Mean Income",
    fill = "Gender"
  ) +
  theme(axis.text.x = element_text(angle = 60, vjust = 1, hjust = 1))
```



The above graph illustrates a plot for topcoded income vs industry code by gender. The red boxes are for male and the blue ones are for females. The graphs shows that the income for male is consistently higher than that for females for all the three types of institutions.

TRUSTFUL/DISTRUSTFUL vs INCOME

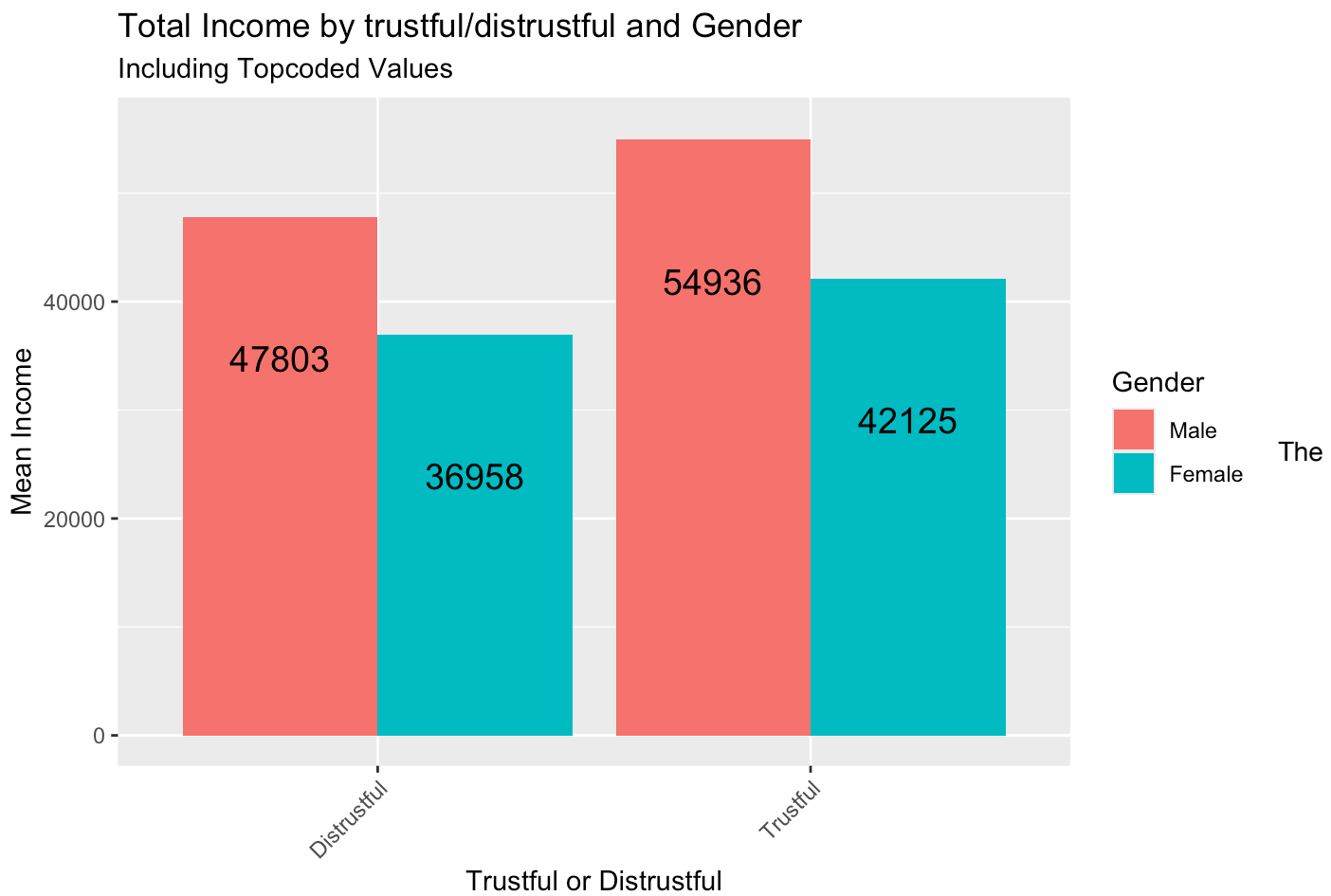
```
# Filtering NA from trustful/distrustful
trustful_distrustful_nlsy_df <- nlsy_df %>%
  filter(!is.na(trustful_or_not))%>%
  filter(!is.na(total_income_2016))

# Summarizing trustful/distrustful
trustful_distrustful_income_summary <- trustful_distrustful_nlsy_df %>%
group_by(trustful_or_not) %>%
summarize(
count = n(),
count_male = sum(gender == "Male", na.rm = TRUE),
count_female = sum(gender == "Female", na.rm = TRUE),
mean_income = mean(total_income_2016, na.rm = TRUE),
median_income = median(total_income_2016, na.rm = TRUE),
sd_income = sd(total_income_2016, na.rm = TRUE)
)
print(trustful_distrustful_income_summary)
```

```
## # A tibble: 2 × 7
##   trustful_or_not count count_male count_female mean_income median_income
##   <fct>           <int>      <int>        <int>       <dbl>         <dbl>
## 1 Distrustful       329        180          149      42891.         35000
## 2 Trustful         2611       1321         1290      48607.         40000
## # ℹ 1 more variable: sd_income <dbl>
```

The table above shows the count of respondents for Distrustful and Trustful, mean_income, median_income, and standard deviation. The mean income, median income and sd income for Trustful is the higher and lower for Distrustful. The NA consists of valid skip which means the question was not applicable to those respondants and hence we have omitted it. Therefore, the data we see for marital status in table or the graph will be for the 6044 responses as out of 8984 we omitted 2940 rows.

```
# Filter data to exclude missing income values and focus on trustful/distrustful
trustful_distrustful_nlsy_df <- nlsy_df %>%
  filter(!is.na(trustful_or_not))%>%
  filter(!is.na(total_income_2016))

# Create the bar plot for mean income by trustful/distrustful and gender
ggplot(trustful_distrustful_nlsy_df, aes(x = trustful_or_not, y = total_income_2016, fil
l = gender)) +
  geom_bar(stat = "summary", position = "dodge") +
  geom_text(aes(label = round(..y..)), stat = "summary", position = position_dodge(width
= 0.9), vjust = 6, size= 5 ) +
  labs(
    title = "Total Income by trustful/distrustful and Gender",
    subtitle = "Including Topcoded Values",
    x = "Trustful or Distrustful",
    y = "Mean Income",
    fill = "Gender"
  ) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```

```
## No summary function supplied, defaulting to `mean_se()`
## No summary function supplied, defaulting to `mean_se()`
```



The above graph illustrates a plot for topcoded income vs respondents who are trustful/distrustful by gender. The red boxes are for male and the blue ones are for females. The graphs shows that the income for male is consistently

higher than that for females for both who are trustful as well as distrustful.

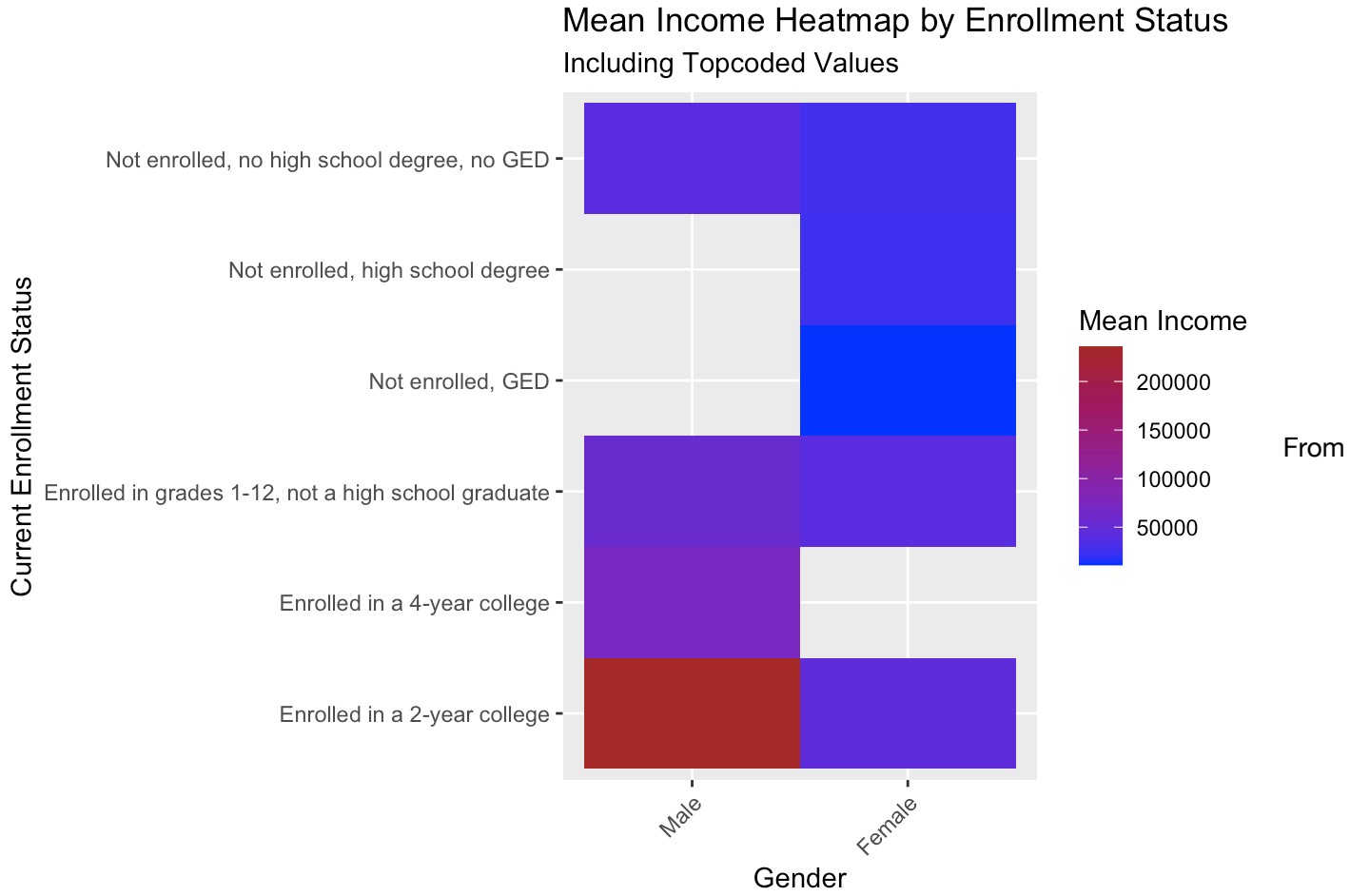
MEAN INCOME BY CURRENT ENROLLMENT STATUS AND GENDER

```
# Calculate mean income for each enrollment status and gender, excluding missing values
nlsy_current_enr_df <- nlsy_df %>%
  filter(!is.na(total_income_2016), !is.na(current_enrollment_status)) %>%
  group_by(current_enrollment_status, gender) %>%
  summarize(
    `Mean Income` = mean(total_income_2016, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  rename(`Enrollment Status` = current_enrollment_status)

# Print the results
print(nlsy_current_enr_df)
```

```
## # A tibble: 9 × 3
##   `Enrollment Status`                                  gender `Mean Income`
##   <chr>                                                <fct>          <dbl>
## 1 Enrolled in a 2-year college                         Male         235884
## 2 Enrolled in a 2-year college                         Female        45500
## 3 Enrolled in a 4-year college                         Male          73000
## 4 Enrolled in grades 1-12, not a high school graduate Male          57438.
## 5 Enrolled in grades 1-12, not a high school graduate Female        41622.
## 6 Not enrolled, GED                                    Female        11000
## 7 Not enrolled, high school degree                     Female        24000
## 8 Not enrolled, no high school degree, no GED          Male          41218.
## 9 Not enrolled, no high school degree, no GED          Female        26972
```

From the above we can infer that the respondents who are currently enrolled in a 2-year college tend to earn highest average income(this includes the topcoded values). However there is a significant difference in the average income of Male and Female although they attended the same level of college and the Male population are earning more than Female.

```
# Calculate mean income for each enrollment status and gender
nlsy_heatmap <- nlsy_df %>%
  filter(!is.na(total_income_2016), !is.na(current_enrollment_status)) %>%
  group_by(current_enrollment_status, gender) %>%
  summarize(mean_income = mean(total_income_2016, na.rm = TRUE), .groups = "drop")

# Create heatmap for mean income by the current enrollment status
ggplot(nlsy_heatmap, aes(x = gender, y = current_enrollment_status, fill = mean_income))
+
  geom_tile() +
  scale_fill_gradient(low = "blue", high = "brown", name = "Mean Income") +
  labs(
    title = "Mean Income Heatmap by Enrollment Status",
    subtitle = "Including Topcoded Values",
    x = "Gender",
    y = "Current Enrollment Status"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



the Heatmap we can observe that Male and Female respondents who are attending same 2 year college degree but the male population is earning more average income compared to female respondents.

PART 3 - NORMALITY CHECK, CI, HYPOTHESE TESTING AND REGRESSION

NORMALITY CHECK

```
# To check for normality of total_income_2016 we have performed the Skewness and Kurtosi
s tests to check for normality.

# Skewness:
# Close to 0: Symmetric distribution (normal-like).
# Positive: Right-skewed (tail extends to the right).
# Negative: Left-skewed (tail extends to the left).

skewness(nlsy_df$total_income_2016, na.rm = TRUE)
```

```
## [1] 2.435323
```

```
#Skewness = 2.435323: This indicates that the distribution of total_income_2016 is posit
ively skewed, meaning it has a long right tail. The majority of the income values are co
ncentrated toward the lower end, with fewer values extending toward the higher end. Henc
e it can be determined that the data deviates from normality.

# Kurtosis:
# Normal distribution has a kurtosis of 3.
# >3: Heavy-tailed distribution (leptokurtic).
# <3: Light-tailed distribution (platykurtic).

kurtosis(nlsy_df$total_income_2016, na.rm = TRUE)
```

```
## [1] 11.26869
```

```
# Kurtosis = 11.26869: This is significantly greater than 3, suggesting a leptokurtic di
stribution. This means the data has heavier tails and a sharper peak compared to a norma
l distribution. Hence it can be determined that the data deviates from normality.
```

TOBIT REGRESSION TEST

We ran Tobit regression test as our data has top coded values. As $p < 0.05$, we can reject the Null Hypothesis (Ho) in support of Alternate Hypothesis (Ha) which states the difference in income for males and females is not equal to 0. We analyzed that females earn approximately $16180 less than males in terms of latent income (before censoring). Both the intercept, latent income and scale, and gender coefficient are highly statistically significant which means these effects are unlikely to be due to random chances but is something which is consistent across data. The 95% confidence interval [13747.84, 18099], t-value is 14.346, and p-value < 2.2e-16. The mean income for males is 57202.82 and females is 41278.92.

WITH TOPCODED VALUES

CONFIDENCE INTERVAL FOR TOPCODED

```
# Filter dataset to exclude rows with missing income
filtered_data <- nlsy_df %>%
  filter(!is.na(total_income_2016))

# Perform t-test to calculate confidence interval for income by gender
gender_ci_top <- t.test(
  total_income_2016 ~ gender,
  data = filtered_data,
  conf.level = 0.95
)

# Print the t-test results
print(gender_ci_top)
```

```
##
##  Welch Two Sample t-test
##
## data:  total_income_2016 by gender
## t = 14.346, df = 4876.8, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group Male and group Female
is not equal to 0
## 95 percent confidence interval:
##  13747.84 18099.96
## sample estimates:
##   mean in group Male mean in group Female
##             57202.82             41278.92
```

The Welch Two-Sample t-test was conducted to examine the difference in mean income between males and females in the dataset with top-coded income values. We will consider the Null Hypothesis (Ho) as the mean income for males and females is equal. Alternate Hypothesis (Ha) as mean income for males and females are unequal. The test statistic t is 14.346, with a p-value < 2.2e-16 which indicates a highly significant result. This means we reject the null hypothesis and conclude that there is a statistically significant difference in mean income between males and females. A 95% confidence interval was calculated for the difference in mean income between males and females. The confidence interval is (13,747.84, 18,099.96). We can interpret that we are 95% confident that the true difference in mean income between males and females lies between $13,747.84 and $18,099.96, with males earning more on average. The average income for males is $57202.82 while for females it is $41278.92 and it highlights a substantial income disparity with top-coded values between the groups, with males earning approximately $15923 more than females on average.

HYPOTHESES TEST FOR TOPCODED

```
# Decision Rule
if (gender_ci_top$p.value < 0.05) {
  print("Reject the null hypothesis: Income significantly differs between males and fema
les.")
} else {
  print("Fail to reject the null hypothesis: No significant income difference between ma
les and females.")
}
```

```
## [1] "Reject the null hypothesis: Income significantly differs between males and femal
es."
```

The hypothesis test for top-coded income values examined whether there is a significant difference in mean income between males and females. The null hypothesis (Ho) assumes no difference, while the alternative hypothesis (Ha) suggests a significant difference. The Welch Two-Sample t-test yielded a p-value < 2.2e-16, far below the significance threshold (0.05). Consequently, the null hypothesis is rejected, indicating a statistically significant income disparity between genders, with males earning more on average.

LINEAR REGRESSION MODEL FOR TOPCODED

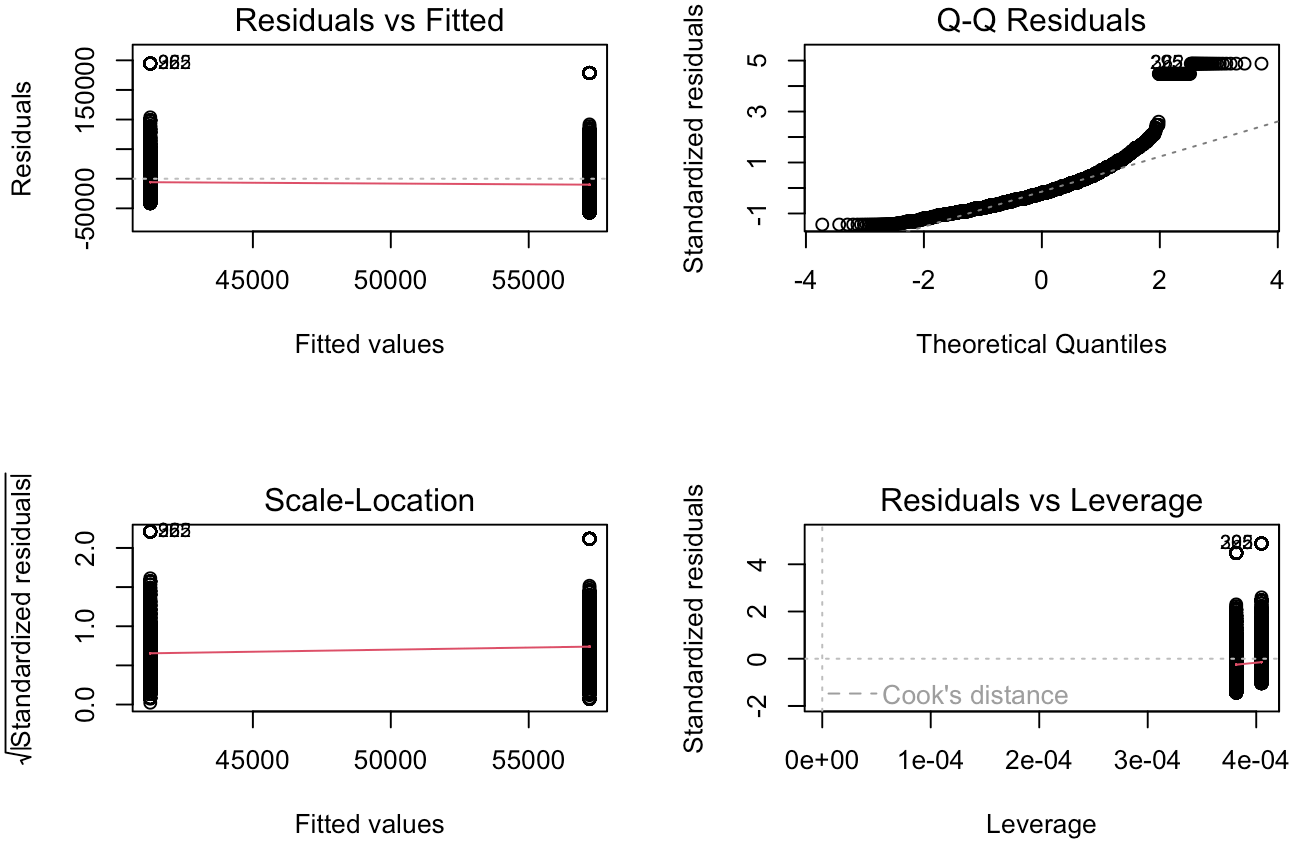
```
# Fit the regression model
lm_model <- lm(total_income_2016 ~ gender, data = filtered_data)

# Summarize the regression model
 summary(lm_model)
```

```
##
## Call:
## lm(formula = total_income_2016 ~ gender, data = filtered_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -57203 -24203  -8203  12797 194605
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   57202.8      779.3   73.41   <2e-16 ***
## genderFemale -15923.9     1118.8  -14.23   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39900 on 5089 degrees of freedom
## Multiple R-squared:  0.03829,    Adjusted R-squared:  0.0381
## F-statistic: 202.6 on 1 and 5089 DF,  p-value: < 2.2e-16
```

The linear regression model was used to analyze the relationship between gender and total income for top-coded values. The intercept (57,202.8) represents the average income for males, while the coefficient for genderFemale (−15,923.9) indicates that females earn, on average, $15,923.9 less than males. The p-value for the gender coefficient (<2.2×10 ^ −16) is highly significant, confirming that gender is a strong predictor of income in this dataset. However, the model explains only 3.8% of the variability in income (Adjusted R^2 =0.0381), suggesting other factors also contribute to income differences.

PLOTS FOR TOPCODED

```
# Diagnostic plots for regression model
par(mfrow = c(2, 2))  # Arrange plots in 2x2 grid
plot(lm_model)
```



1. Residuals vs. Fitted Plot: This plot assesses the linearity assumption. The residuals appear scattered but exhibit some clustering at extreme values, indicating possible deviations from linearity.
2. Q-Q Plot: The Q-Q plot evaluates the normality of residuals. The residuals deviate significantly from the diagonal line, particularly at the tails, suggesting the residuals are not normally distributed.
3. Scale-Location Plot: This plot checks for homoscedasticity (constant variance of residuals). The residuals' spread appears uneven, with greater variation at extreme fitted values, indicating heteroscedasticity.
4. Residuals vs. Leverage Plot: This plot identifies influential points. Several observations near the top right corner with high leverage suggest some data points may disproportionately impact the model. These plots collectively highlight potential violations of regression assumptions, including non-normality, heteroscedasticity, and influential outliers. Adjustments or alternative modeling approaches may improve the model's validity and so we have also tested it for income truncating topcoded values.

WITHOUT TOPCODED VALUES

CONFIDENCE INTERVAL FOR NO-TOPCODED

```
nlsy_no_topcoded <- nlsy_df %>%
  filter(total_income_2016 != max(total_income_2016, na.rm = TRUE))
# Filter dataset to exclude rows with missing income
notop_filtered_data <- nlsy_no_topcoded %>%
  filter(!is.na(total_income_2016))

# Perform t-test to calculate confidence interval for income by gender
gender_ci <- t.test(
  total_income_2016 ~ gender,
  data = notop_filtered_data,
  conf.level = 0.95
)

# Print the t-test results
print(gender_ci)
```

```
##
##  Welch Two Sample t-test
##
## data:  total_income_2016 by gender
## t = 14.91, df = 4938.7, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group Male and group Female
is not equal to 0
## 95 percent confidence interval:
##  10326.42 13453.00
## sample estimates:
##   mean in group Male mean in group Female
##             50775.95             38886.23
```

The Welch Two Sample t-test was performed to calculate the confidence interval for the difference in mean income between males and females, excluding topcoded values. After running the calculation, the t-statistic is 14.91, with degrees of freedom approximately 4938.7. The p-value is less than 2.2e-16, indicating strong evidence to reject the null hypothesis of no difference in mean income between males and females. The 95% confidence interval for the difference in means is [10,326.42, 13,453.00]. This interval does not include 0, further supporting the conclusion that a significant difference exists. The mean income for males is $50775 (approx) and $38886 (approx) for females. There is a statistically significant difference in mean income between males and females, with males earning, on average, significantly more than females in the dataset after excluding topcoded values which is similar to the trend as shown by topcoded test.

HYPOTHESES TEST FOR NO-TOPCODED

```
# Decision Rule
if (gender_ci$p.value < 0.05) {
  print("Reject the null hypothesis: Income significantly differs between males and fema
les.")
} else {
  print("Fail to reject the null hypothesis: No significant income difference between ma
les and females.")
}
```

```
## [1] "Reject the null hypothesis: Income significantly differs between males and femal
es."
```

The hypothesis test shows a statistically significant income difference between males and females after excluding topcoded values, as the p-value is less than 0.05 and so we reject the Ho in support of Ha.

LINEAR REGRESSION MODEL FOR NO-TOPCODED

```
# Fit the regression model
notop_lm_model <- lm(total_income_2016 ~ gender, data = notop_filtered_data)

# Summarize the regression model
 summary(notop_lm_model)
```

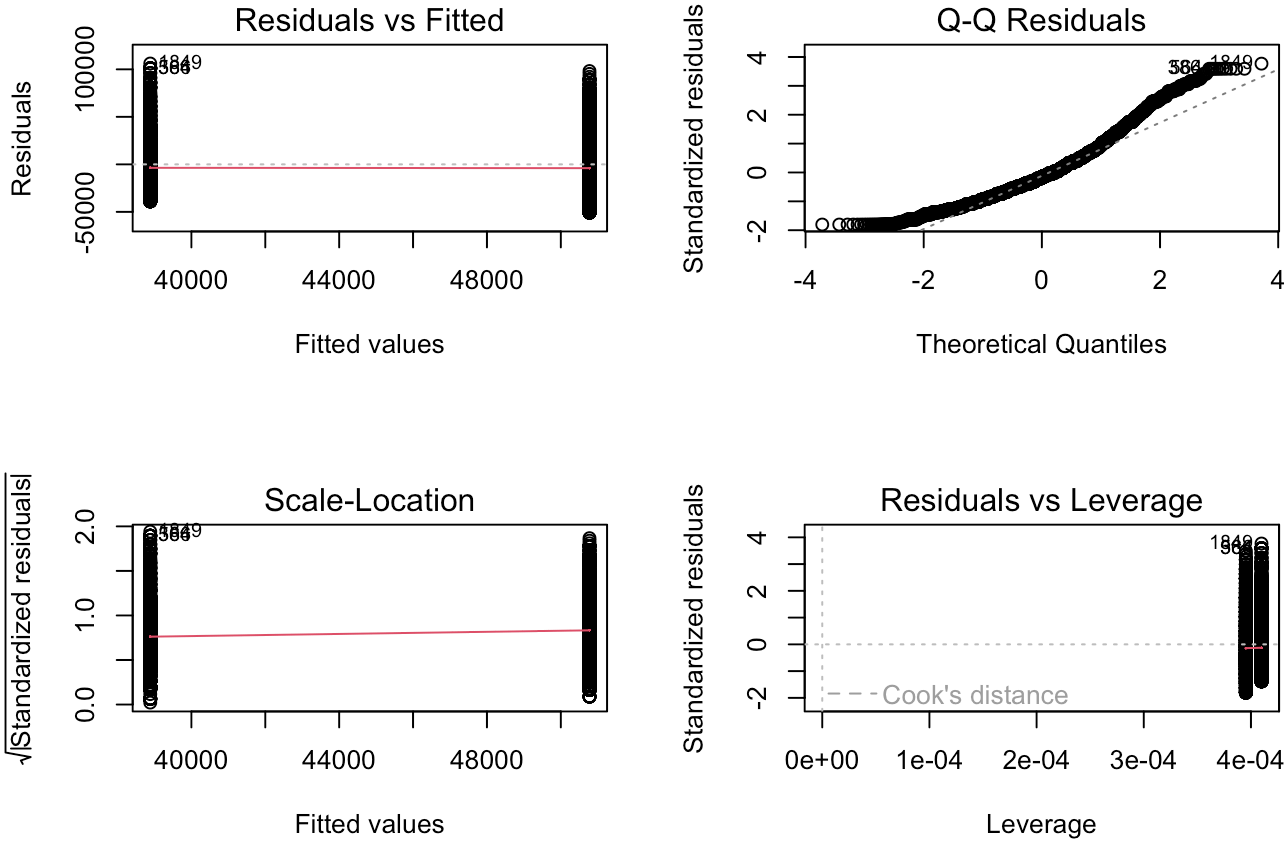
```
##
## Call:
## lm(formula = total_income_2016 ~ gender, data = notop_filtered_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -50776 -20776  -4776  14224 106114
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   50775.9      559.9   90.69   <2e-16 ***
## genderFemale -11889.7      799.1  -14.88   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 28160 on 4968 degrees of freedom
## Multiple R-squared:  0.04267,    Adjusted R-squared:  0.04247
## F-statistic: 221.4 on 1 and 4968 DF,  p-value: < 2.2e-16
```

The linear regression model examines the relationship between total income (2016) and gender. The intercept, represents the average total income for males, is estimated at $50,775.90. The coefficient for genderFemale is -11,889.70, which indicates that, on average, females earn $11,889.70 less than males in 2016. This difference is significant, as evidenced by a p-value < 2e-16. The residual standard error is 28,160, reflecting variability in

income not explained by the model. The model explains only 4.27% of the variance in total income (R^2 = 0.04267), suggesting that gender alone is not a strong predictor of income. The overall model is significant (F(1, 4968) = 221.4, p-value < 2.2e-16).

PLOTS FOR NO-TOPCODED

```
# Diagnostic plots for regression model
par(mfrow = c(2, 2))  # Arrange plots in 2x2 grid
plot(notop_lm_model)
```



1. Residuals vs. Fitted Plot: The residuals are clustered tightly for smaller fitted values but become more spread at the upper end. Indicates heteroscedasticity (non-constant variance of residuals). There may also be outliers present, as some points deviate significantly from the zero line.
2. Normal Q-Q Plot: The residuals deviate from the straight diagonal line at both ends, particularly in the tails. Suggests the residuals are not normally distributed.
3. Scale-Location Plot: The spread of residuals increases slightly as fitted values increase. Indicates heteroscedasticity, confirming the findings from the Residuals vs. Fitted plot.
4. Residuals vs. Leverage Plot: Points near Cook's distance lines indicate high influence, though most points are clustered in low-leverage areas. A few data points may have high leverage or strong influence, warranting further investigation.

LINEAR REGRESSION MODEL FOR NO-TOPCODED (MARITAL STATUS)

```
# Fit the regression model
notop_lm_model_ms <- lm(total_income_2016 ~ gender + marital_status, data = notop_filter
ed_data)

# Summarize the regression model
 summary(notop_lm_model_ms)
```

```
##
## Call:
## lm(formula = total_income_2016 ~ gender + marital_status, data = notop_filtered_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -56614 -19708  -4708  14928 106834
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            45071.7      733.3  61.467   <2e-16 ***
## genderFemale          -11905.5      787.8 -15.112   <2e-16 ***
## marital_statusMarried  11542.0      847.9  13.612   <2e-16 ***
## marital_statusOther     1308.8     1268.9   1.031    0.302
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27630 on 4935 degrees of freedom
##   (31 observations deleted due to missingness)
## Multiple R-squared:  0.08079,    Adjusted R-squared:  0.08023
## F-statistic: 144.6 on 3 and 4935 DF,  p-value: < 2.2e-16
```

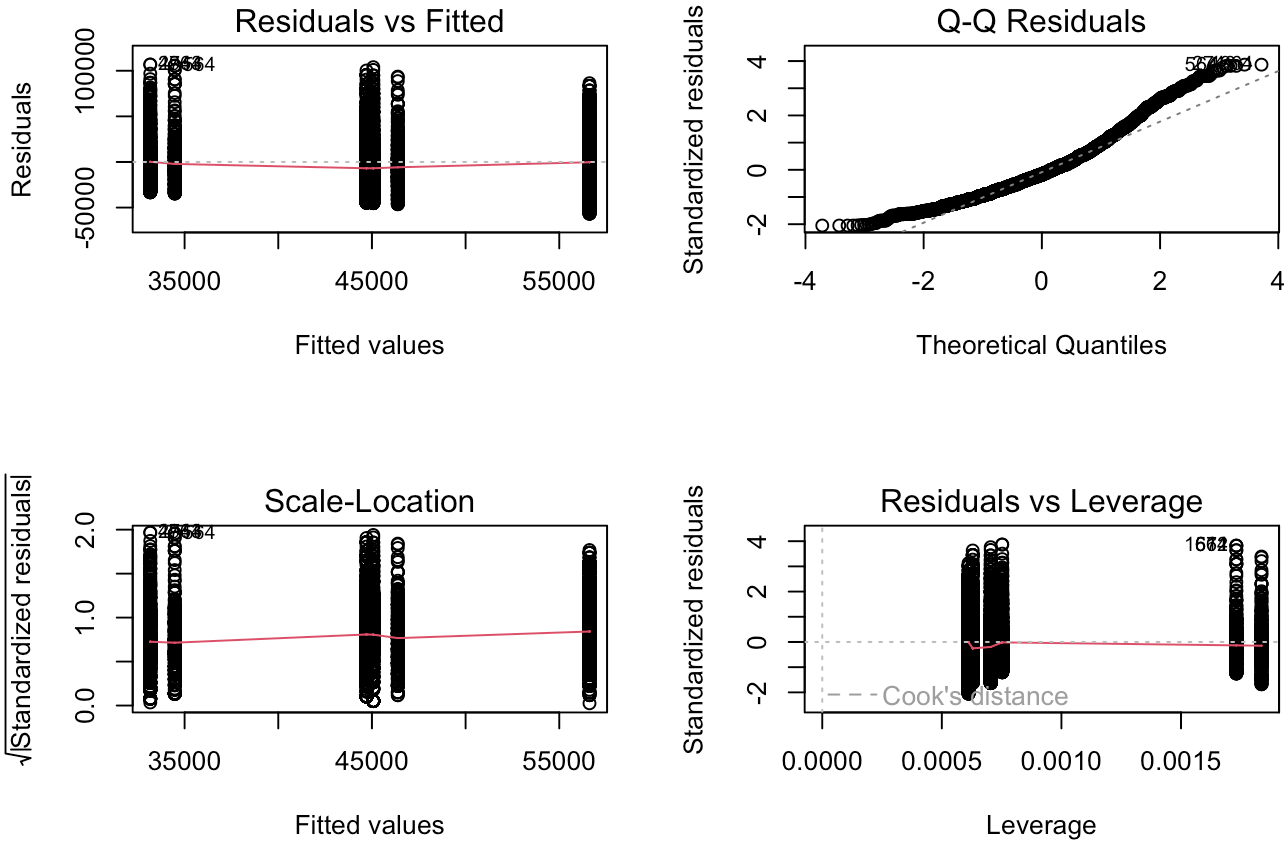
The linear regression model examines the relationship between total income (2016) and the predictors: gender and marital status. The intercept, represents the estimated average income for males who are unmarried, is $45,071.70 and is statistically significant (p-value < 2e-16).

Gender: Females earn $11,905.50 less than males on average, a statistically significant difference (p-value < 2e-16). Marital Status: Married individuals earn $11,542.00 more on average than unmarried individuals, a statistically significant effect (p-value < 2e-16). Individuals categorized as “Other” marital status earn $1,308.80 more than unmarried individuals, but this is not statistically significant (p-value = 0.302).

The residual standard error is 27,630, indicating the unexplained variability in income. The model explains 8.08% of the variance in income (R^2 = 0.08079), suggesting a modest explanatory power. The overall model is statistically significant (F(3, 4935) = 144.6, p-value < 2.2e-16). A total of 31 observations were excluded due to missing data.

PLOTS FOR NO-TOPCODED (MARITAL STATUS)

```
# Diagnostic plots for regression model
par(mfrow = c(2, 2))  # Arrange plots in 2x2 grid
plot(notop_lm_model_ms)
```



1. Residuals vs. Fitted: The residuals are scattered unevenly across the range of fitted values. There appears to be some clustering or systematic patterns, especially at lower fitted values. There may be issues with non-linearity or heteroscedasticity, suggesting that the model assumptions of linearity or equal variance might be violated.
2. Normal Q-Q Plot: Residuals deviate significantly from the theoretical quantile line in the tails. A visible “S” shape suggests non-normality of residuals. Statistical tests and confidence intervals derived from the model may be less reliable due to this non-normality.
3. Scale-Location (Spread-Location): The spread of residuals increases slightly with larger fitted values, as indicated by the upward trend in the red line. The model may suffer from mild heteroscedasticity, meaning the variance of residuals is not constant across levels of the predictors.
4. Residuals vs. Leverage: Points like 1672 and 1674 are potential high-leverage points, but they do not exceed Cook’s distance threshold. While these points are not excessively influential, they should be examined for potential outlier effects.

LINEAR REGRESSION MODEL FOR NO-TOPCODED (AGE 1996

+ CITIZENSHIP)

```
# Fit the regression model
notop_lm_model_ag_ctz <- lm(total_income_2016 ~ gender + age_1996 + citizenship, data =
notop_filtered_data)

# Summarize the regression model
 summary(notop_lm_model_ag_ctz)
```

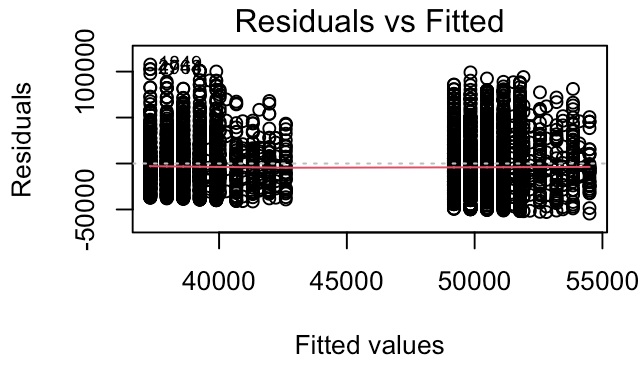
```
##
## Call:
## lm(formula = total_income_2016 ~ gender + age_1996 + citizenship,
##     data = notop_filtered_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -54494 -19836  -4605  14518 107693
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)             43763.8     4876.5   8.974   <2e-16 ***
## genderFemale           -11883.0      841.1 -14.129   <2e-16 ***
## age_1996                  646.0      301.2   2.145    0.032 *
## citizenshipCitizen      -2326.0     2498.2  -0.931    0.352
## citizenshipNon-Citizen    394.3     2818.2   0.140    0.889
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 28020 on 4439 degrees of freedom
##   (526 observations deleted due to missingness)
## Multiple R-squared:  0.04467,    Adjusted R-squared:  0.04381
## F-statistic: 51.89 on 4 and 4439 DF,  p-value: < 2.2e-16
```

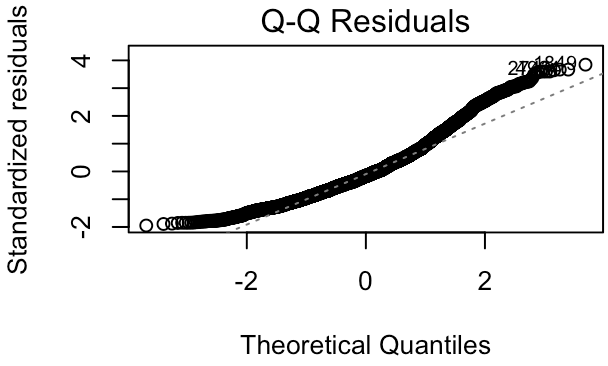
The linear regression model evaluates the relationship between total income (2016) and three predictors: gender, age (1996), and citizenship status. The intercept, represents the average total income for males of age 0 (reference age) with unknown citizenship, is estimated at $43,763.80. The coefficient for genderFemale is -11,883.00, indicating that females earn $11,883.00 less than males on average, holding other variables constant. This difference is statistically significant (p-value < 2e-16). The coefficient for age_1996 is 646.00, suggesting that income increases by $646.00 for each additional year of age, which is statistically significant (p-value = 0.032). However, the coefficients for citizenshipCitizen and citizenshipNon-Citizen are not statistically significant (p-values of 0.352 and 0.889, respectively), indicating no significant effect of citizenship status on income.

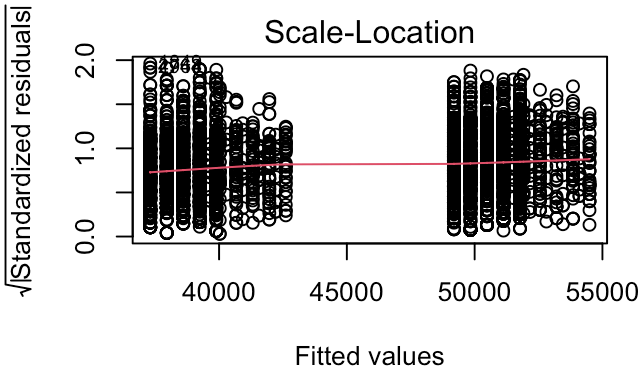
The residual standard error is 28,020, showing variability in income not explained by the model. The model explains 4.47% of the variance in total income (R^2 = 0.04467), indicating a weak predictive capability. The overall model is statistically significant (F(4, 4439) = 51.89, p-value < 2.2e-16). Note that 526 observations were excluded due to missing data.

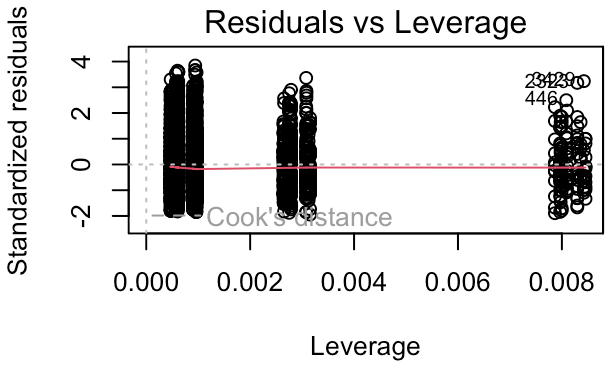
PLOTS FOR NO-TOPCODED (AGE 1996 + CITIZENSHIP)

```
# Diagnostic plots for regression model
par(mfrow = c(2, 2))  # Arrange plots in 2x2 grid
plot(notop_lm_model_ag_ctz)
```









1. Residuals vs. Fitted: Residuals are scattered around zero but appear to show non-random patterns. Some clustering may indicate mild issues with linearity or omitted variables. There may be slight violations of linearity or homoscedasticity.
2. Normal Q-Q Plot: The residuals follow the diagonal line reasonably well in the middle but deviate in the tails. The departure at both ends indicates potential non-normality of residuals. Inferences might be slightly affected, particularly for tests or confidence intervals.
3. Scale-Location (Spread-Location): The red line is mostly flat, but the spread of residuals increases slightly as fitted values grow. Indicates mild heteroscedasticity. Variance may not be constant, which could affect model accuracy.
4. Residuals vs. Leverage: Points like 244, 1200, and 3220 show relatively high leverage. None of the observations exceed Cook's distance threshold, meaning their influence is limited. These points should still be reviewed for potential impact on model performance.

LINEAR REGRESSION MODEL FOR NO-TOPCODED (RACE +

USED MARIJUANA + INDUSTRY CODE)

```
# Fit the regression model
notop_lm_model_rc_um_ic <- lm(total_income_2016 ~ gender + race + used_marijuana + indus
try_code, data = notop_filtered_data)

# Summarize the regression model
 summary(notop_lm_model_rc_um_ic)
```

```
##
## Call:
## lm(formula = total_income_2016 ~ gender + race + used_marijuana +
##     industry_code, data = notop_filtered_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -69125 -18596  -4090  14309 106309
##
## Coefficients:
##                                                          Estimate
## (Intercept)                                               21000.0
## genderFemale                                             -10252.5
## raceHispanic                                               6606.7
## raceOther                                                 12904.6
## used_marijuanaRefusal                                     14115.1
## used_marijuanaNo                                          19715.8
## used_marijuanaYes                                         18717.8
## industry_codeActive Duty Military                         16306.7
## industry_codeAgriculture, Forestry, and Fisheries           514.4
## industry_codeConstruction                                  -211.4
## industry_codeEducational, Health, and Social Services      3227.7
## industry_codeEntertainment, Accommodations, and Food Services  -8132.4
## industry_codeFinance, Insurance, and Real Estate          10536.4
## industry_codeInformation and Communication                 8805.3
## industry_codeManufacturing                                 4185.4
## industry_codeMining                                       23903.5
## industry_codeOther Services                               -6861.8
## industry_codeProfessional and Related Services             5968.9
## industry_codePublic Administration                        15504.7
## industry_codeRetail Trade                                 -4530.6
## industry_codeTransportation and Warehousing                3203.7
## industry_codeUtilities                                    26103.0
## industry_codeWholesale Trade                               3233.7
##                                                          Std. Error
## (Intercept)                                                 21450.8
## genderFemale                                                  910.5
## raceHispanic                                                 1244.7
## raceOther                                                    1025.4
## used_marijuanaRefusal                                       20735.4
## used_marijuanaNo                                            19195.8
## used_marijuanaYes                                           19213.7
## industry_codeActive Duty Military                           11539.5
## industry_codeAgriculture, Forestry, and Fisheries           10720.9
## industry_codeConstruction                                    9751.0
## industry_codeEducational, Health, and Social Services        9625.9
## industry_codeEntertainment, Accommodations, and Food Services    9666.7
## industry_codeFinance, Insurance, and Real Estate             9714.7
## industry_codeInformation and Communication                   9924.2
## industry_codeManufacturing                                   9710.5
## industry_codeMining                                         10925.6
## industry_codeOther Services                                  9764.0
```

```
## industry_codeProfessional and Related Services                    9660.3
## industry_codePublic Administration                                9778.7
## industry_codeRetail Trade                                         9667.4
## industry_codeTransportation and Warehousing                       9866.9
## industry_codeUtilities                                           11134.1
## industry_codeWholesale Trade                                      9918.0
##                                                                t value Pr(>|t|)
## (Intercept)                                                      0.979   0.3276
## genderFemale                                                   -11.261  < 2e-16
## raceHispanic                                                     5.308 1.17e-07
## raceOther                                                       12.585  < 2e-16
## used_marijuanaRefusal                                            0.681   0.4961
## used_marijuanaNo                                                 1.027   0.3044
## used_marijuanaYes                                                0.974   0.3300
## industry_codeActive Duty Military                                1.413   0.1577
## industry_codeAgriculture, Forestry, and Fisheries                0.048   0.9617
## industry_codeConstruction                                       -0.022   0.9827
## industry_codeEducational, Health, and Social Services            0.335   0.7374
## industry_codeEntertainment, Accommodations, and Food Services   -0.841   0.4002
## industry_codeFinance, Insurance, and Real Estate                 1.085   0.2782
## industry_codeInformation and Communication                       0.887   0.3750
## industry_codeManufacturing                                       0.431   0.6665
## industry_codeMining                                              2.188   0.0287
## industry_codeOther Services                                     -0.703   0.4822
## industry_codeProfessional and Related Services                   0.618   0.5367
## industry_codePublic Administration                               1.586   0.1129
## industry_codeRetail Trade                                       -0.469   0.6393
## industry_codeTransportation and Warehousing                      0.325   0.7454
## industry_codeUtilities                                           2.344   0.0191
## industry_codeWholesale Trade                                     0.326   0.7444
##
## (Intercept)
## genderFemale                                                   ***
## raceHispanic                                                   ***
## raceOther                                                      ***
## used_marijuanaRefusal
## used_marijuanaNo
## used_marijuanaYes
## industry_codeActive Duty Military
## industry_codeAgriculture, Forestry, and Fisheries
## industry_codeConstruction
## industry_codeEducational, Health, and Social Services
## industry_codeEntertainment, Accommodations, and Food Services
## industry_codeFinance, Insurance, and Real Estate
## industry_codeInformation and Communication
## industry_codeManufacturing
## industry_codeMining                                            *
## industry_codeOther Services
## industry_codeProfessional and Related Services
## industry_codePublic Administration
## industry_codeRetail Trade
## industry_codeTransportation and Warehousing
```

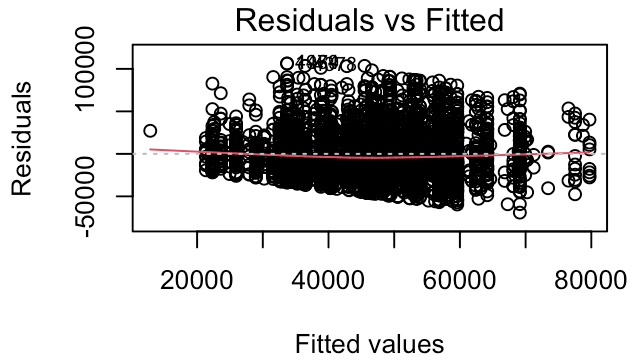
```
## industry_codeUtilities                                          *
## industry_codeWholesale Trade
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27110 on 4204 degrees of freedom
##   (743 observations deleted due to missingness)
## Multiple R-squared:  0.1277, Adjusted R-squared:  0.1231
## F-statistic: 27.96 on 22 and 4204 DF,  p-value: < 2.2e-16
```

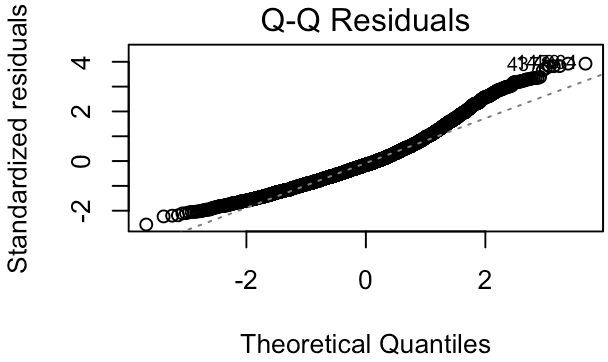
The linear regression model explores the relationship between total income (2016) and the predictors: gender, race, used_marijuana, and industry_code. The intercept, representing the estimated average income for males of the “White” race, who have not responded about marijuana usage, and with the reference industry, is $21,000.00 but is not statistically significant (p-value = 0.3276).

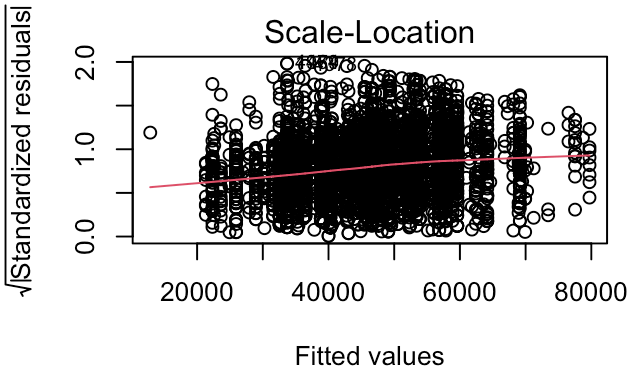
Gender: Females earn $10,252.50 less than males on average, a statistically significant difference (p-value < 2e-16). Race: Compared to the reference race (“White”), Hispanic individuals earn $6,606.70 more on average (p-value = 1.17e-07), while individuals of “Other” races earn $12,904.60 more on average (p-value < 2e-16). Used_marijuana: None of the levels (Refusal, No, Yes) showed a significant effect on income. Industry_code: Two industries showed significant effects on income. Workers in the “Mining” industry earn $23,903.50 more on average (p-value = 0.0287), and those in “Utilities” earn $26,103.00 more on average (p-value = 0.0191) compared to the reference industry. The residual standard error is 27,110, indicating the variation in income not explained by the model. The model explains 12.77% of the variance in income (R^2 = 0.1277), suggesting moderate explanatory power. The overall model is statistically significant (F(22, 4204) = 27.96, p-value < 2.2e-16). Note that 743 observations were excluded due to missing data.

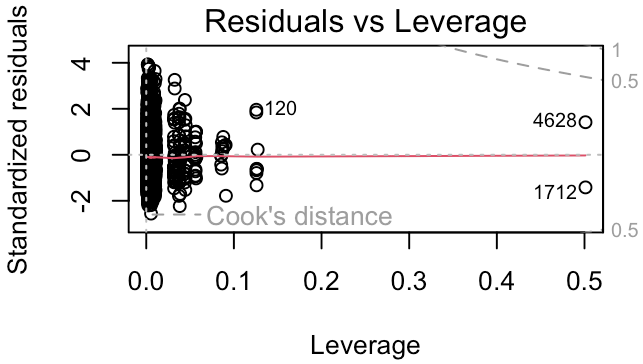
PLOTS FOR NO-TOPCODED (RACE + USED MARIJUANA + INDUSTRY CODE)

```
# Diagnostic plots for regression model
par(mfrow = c(2, 2))  # Arrange plots in 2x2 grid
plot(notop_lm_model_rc_um_ic)
```









1. Residuals vs. Fitted: The residuals are scattered around zero, but there is some noticeable spread variation. There may be mild signs of heteroscedasticity, as the variance increases slightly with the fitted values. This suggests that the model may not perfectly meet the assumption of homoscedasticity.
2. Normal Q-Q Plot: The residuals deviate from the diagonal line, especially at the tails, indicating that the residuals are not perfectly normal. This might affect inference (e.g., confidence intervals and p-values).
3. Scale-Location (Spread-Location): The red line is relatively flat, but there is some variation in the spread of points. The heteroscedasticity detected in the first plot is confirmed here. Indicates that the model's variance might not be constant across the range of fitted values.
4. Residuals vs. Leverage: Points like 120, 4628, and 1712 are flagged as having high leverage. No point crosses the Cook's distance threshold (indicated by the dashed lines), so no observation appears to be highly influential. No extreme leverage points, but observations with high leverage should be examined.

PART 4 - STORYTELLING

METHODOLOGY

Our theme question is to analyze sex-related income gap. We have taken many other factors into consideration to get the confidence in our analysis based on tables, graphical representations, and summaries. Now, to identify different variables that will help us get insights, we explored codebook and factored required variable. We renamed the columns to relevant names, mutated some to get better plots, and identified the count of male and female to bifercate how many of them fall in different category. We have used different plots like bar, histogram, boxplots, jitter plot, and heatmap that reveals trends.

HANDLING MISSING VALUES

The data setset contains negative and missing values where the responses indicated Refusal(-1), Don't Know(-2), Invalid Skip(-3), VALID SKIP(-4) and NON-INTERVIEW(-5). Initially, we conducted a data audit to identify missing values in key variables (e.g., income, highest grade completed, etc.). For the numerical variables such as income, the negative or invalid values were straightforwardly recoded as NA to ensure statistical analyses were not biased. Also this was done to maintain consistency in the regression model, as incomplete data would lead to computational errors and distort the results.

HANDLING TOPCODED VARIABLES

In the dataset, for example the income variable was topcoded, where the incomes of the top 2% earners were replaced with an average value of $235,884. This led to distortion in measures like the mean and maximum income, as the topcoded values artificially inflated these metrics. Specifically, the mean income was $49,447, and the maximum income was $235,884, both skewed by the 121 topcoded respondents.

To address this, we excluded the 121 topcoded values, lowering the mean income to $44,939 and the maximum income to $149,000, which was the cutoff for the topcoded group. This adjustment reduced distortion while preserving the dataset's overall structure. Visualizations also reflected these changes: the histogram became less extreme while still showing a right-skewed distribution, and the boxplot, after removing outliers, provided a clearer representation of typical incomes, with a median of $35,000 and an IQR of $25,000 to $50,000.

By handling topcoded values in this way, we ensured a more accurate and realistic analysis of income distribution which allowed for better insights into the income range and inequality without the distortion caused by artificially high values.

DID YOU PRODUCE ANY TABLES OR PLOTS THAT YOU THOUGHT WOULD REVEAL INTERESTING TRENDS BUT DIDN'T?

Yes, we found one relationship between income and marital status which did not show a significant trend as I thought. The distribution of incomes for both groups, male and female, overlapped but the differences were not as huge as expected. For males, married people make more money than unmarried people and other categories. At the same time, for females, there is not much difference seen in income between married females and other categories.

WHAT RELATIONSHIPS DID YOU INVESTIGATE THAT DON'T APPEAR IN YOUR FINDINGS SECTION?

In our findings section where plotted our regression test, we didn't see a trend of females earning more than male for refusal category of use of marijuana. However, when we analyzed use of marijuana with income and gender, it was shown that females earn more than males.

WHAT'S THE ANALYSIS THAT YOU FINALLY SETTLED ON? WHAT INCOME AND GENDER RELATED FACTORS DO YOU INVESTIGATE IN THE FINAL ANALYSIS?

We have analyzed many different combination of variables with income and gender which helped us stick to the observation that males have more income as compared to females. The factors that were taken into consideration are race, marital status, enrollment status, use of marijuana, birth year, industry code, age, is sad/depressed, trustful or not, citizenship, college type, highest grade completed, and total number of incarcerations. We plotted almost all of the variables against income and gender to find out if our hypothesis of male and female having same income is true or not. For our final analysis we applied regression plots for income ~ gender for both topcoded and no topcoded values, income ~ gender with race, marijuana, and industry code, income ~ gender with age & citizenship, income ~ gender with marital status.It turns out that we see males have more income across the dataset and even alayzing different types of variable, the reading remain same. Furthermore, there is a huge difference in the income of both groups.

PART 5 - CONCLUSION

In this section you should summarize your main conclusions. You should also discuss potential limitations of your analysis and findings. Are there potential confounding variables that you didn't control for? Are the models you fit believable?

Females consistently earned less than males, throughout our analysis while incorporating different parameters such as race, citizenship, highest grade completed, industry code, marital status, use of marijuana etc. The income varied significantly in most of the analysis, such as in race, citizenship, use of marijuana, if they are sad or depressed, throughout different industries, marital status, incarcerations, college type, and if they were trustful or not, however we had certain circumstances where a female earned more which is the industry code- ASC special codes. A significant portion of the data was missing which posed as limitations for us. The missing data and the topcoded income values may have also introduced potential biases. While these values were excluded to prevent distortion of income distributions, it also meant that the analysis might not fully reflect the income dynamics of the top 2% earners. Out of the 95 variables we selected 15 so there could have been many Key confounders, few of them could be education level, occupation, and location, which were not controlled for in the models. These factors likely influence income and could have lead to residual confounding in the analysis. The regression model that we have used calculates the linear relationships between income and the predictors. However, some relationships may be non-linear or have interactions that were not captured in the model.

You should also address the following question: How much confidence do you have in your analysis? Do you believe your conclusions? Are

you confident enough in your analysis and findings to present them to policy makers?

After doing our analysis we could see that there is a trend that men have more income than women, and we are moderately confident in the findings, particularly regarding the gender pay gap and income disparities. While the findings show significant differences in the income of males and females, the missing data and the top coded values makes the analysis incomplete and a whole picture can not be drawn. We are definitely confident in out findings for the data on which we could perform our analysis and if we could have got the complete data a full analysis could have been done and we would have been more confident about the complete analysis. The findings, especially those related to the gender income gap, racial income disparities, and industry-specific income gaps, are important and could be valuable for policy makers. However, the missing data and topcoding limitations must be acknowledged, as they might impact the generalizability of the results.