Project_1_456

Anthony Yasan, Preston O'Connor, Khoa Dao

2025-02-20

We are modeling the linear regression of the Dependent Income, Independent Age in our model

Introduction

Installing the R-packages

```
# remove comments out these blocks to install the R packages that are being used
#install.packages("ipumsr") # for the data set
#install.packages("dplyr") # for the data set
#install.packages("caTools") # use this for the set seed of the training set
#install.packages("ggplot2")

# Code to implement the R packages
library(ipumsr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2) # visial displays of the Boxplot, and Q-Q plots
library(caTools)
```

Data description

The dataset is sourced from IPUMS USA, which provides microdata extracted from the U.S. Census and American Community Survey (ACS). It contains demographic and economic data at both household and individual levels.

The dataset consists of 3,405,809 rows and 15 columns.

However, for model implementation we randomly selected a seed of 200,000 rows to utilize for the modeling portion of our data.

Each observation represents a household.

Here are the key variables included in the dataset:
- YEAR: Census year (e.g., 2023).
- SAMPLE: IPUMS sample identifier.
- SERIAL: Unique household serial number.
- CBSERIAL: Original Census Bureau household serial number.
- HHWT: Household weight for proper representation.
- CLUSTER: Household cluster for variance estimation.
- STRATA: Household strata for variance estimation.
- GQ: Group quarters status (e.g., household, institution).
- HHINCOME: Total household income for all members over 15 years old. - PERNUM: Person number within the household.
- PERWT: Person weight for population estimates.
- SEX: Gender classification (Male/Female).
- AGE: Individual's age in years.
- RACE: General race classification.
- RACED: Detailed race classification.

Information about the Data set

```
ddi <- read_ipums_ddi("usa_00001.xml")
data <- read_ipums_micro(ddi)
```

```
## Use of data from IPUMS USA is subject to conditions including that users should cite the data appropr
```

```
dim(data)
```

```
## [1] 3405809      15
```

```
summary(data)
```

```
##       YEAR          SAMPLE           SERIAL           CBSERIAL
##  Min.   :2023   Min.   :202301   Min.   :      1   Min.   :2.023e+12
##  1st Qu.:2023   1st Qu.:202301   1st Qu.: 372386   1st Qu.:2.023e+12
##  Median :2023   Median :202301   Median : 756830   Median :2.023e+12
##  Mean   :2023   Mean   :202301   Mean   : 758992   Mean   :2.023e+12
##  3rd Qu.:2023   3rd Qu.:202301   3rd Qu.:1147002   3rd Qu.:2.023e+12
##  Max.   :2023   Max.   :202301   Max.   :1519010   Max.   :2.023e+12
##       HHWT            CLUSTER              STRATA             GQ
##  Min.   :   1.00   Min.   :2.023e+12   Min.   :  10001   Min.   :1.000
##  1st Qu.:  48.00   1st Qu.:2.023e+12   1st Qu.: 100005   1st Qu.:1.000
##  Median :  71.00   Median :2.023e+12   Median : 231248   Median :1.000
##  Mean   :  97.24   Mean   :2.023e+12   Mean   : 488810   Mean   :1.134
##  3rd Qu.: 115.00   3rd Qu.:2.023e+12   3rd Qu.: 480148   3rd Qu.:1.000
##  Max.   :2225.00   Max.   :2.023e+12   Max.   :8100351   Max.   :5.000
##     HHINCOME          PERNUM           PERWT              SEX
##  Min.   : -16800   Min.   : 1.000   Min.   :   1.00   Min.   :1.000
```

```
##  1st Qu.:  54000   1st Qu.: 1.000   1st Qu.:  47.00   1st Qu.:1.000
##  Median : 100100   Median : 2.000   Median :  72.00   Median :2.000
##  Mean   : 638692   Mean   : 2.058   Mean   :  98.34   Mean   :1.509
##  3rd Qu.: 176400   3rd Qu.: 3.000   3rd Qu.: 117.00   3rd Qu.:2.000
##  Max.   :9999999   Max.   :20.000   Max.   :2225.00   Max.   :2.000
##       AGE             RACE           RACED
##  Min.   : 0.00   Min.   :1.000   Min.   :100.0
##  1st Qu.:22.00   1st Qu.:1.000   1st Qu.:100.0
##  Median :44.00   Median :1.000   Median :100.0
##  Mean   :43.11   Mean   :2.535   Mean   :257.9
##  3rd Qu.:63.00   3rd Qu.:2.000   3rd Qu.:200.0
##  Max.   :96.00   Max.   :9.000   Max.   :990.0
```

Table of Data

```
#View(data)
#The Code struggles to run the data set with 2 million points is two extensive to run
set.seed(11)

s <- sample(1:nrow(data), size = 200000)
data <- data[s, ]
dim(data)
```

```
## [1] 200000     15
```

Data Cleaning and Outlier Removal

```
# select the age and the Total Household income as the main columns of interest, then filter based of 1

# ask if the filter crashes out after a certain amount on the computer and if we need to shrink the tra
data <- data %>%
  select(AGE, HHINCOME) %>%
  mutate(HHINCOME = as.numeric(HHINCOME), AGE = as.numeric(AGE)) %>%
  filter(!is.na(HHINCOME), !is.na(AGE)) %>%
  filter(between(AGE, 18, 65)) %>%
  filter(HHINCOME > 0)

dim(data)  #if you want to view the two filtered columns
```

```
## [1] 117981      2
```

```
IQR_of_AGE <- IQR(data$AGE, na.rm = TRUE)
IQR_of_HHINCOME <- IQR(data$HHINCOME, na.rm = TRUE)

# calculating the upper and lower bounds of both of the data sets to filter the data
AGE_lower <- quantile(data$AGE, 0.25, na.rm = TRUE) - 1.5 * IQR_of_AGE
AGE_upper <- quantile(data$AGE, 0.75, na.rm = TRUE) + 1.5 * IQR_of_AGE

HHINCOME_lower <- quantile(data$HHINCOME, 0.25, na.rm = TRUE) - 1.5 * IQR_of_HHINCOME
```

```r
HHINCOME_upper <- quantile(data$HHINCOME, 0.75, na.rm = TRUE) + 1.5 * IQR_of_HHINCOME

#continue to filter any of the outliers that are presents in the data set
filtered_data <- data %>%
  filter(AGE >= AGE_lower & AGE <= AGE_upper) %>%
  filter(HHINCOME >= HHINCOME_lower & HHINCOME <= HHINCOME_upper)

dim(filtered_data)
```

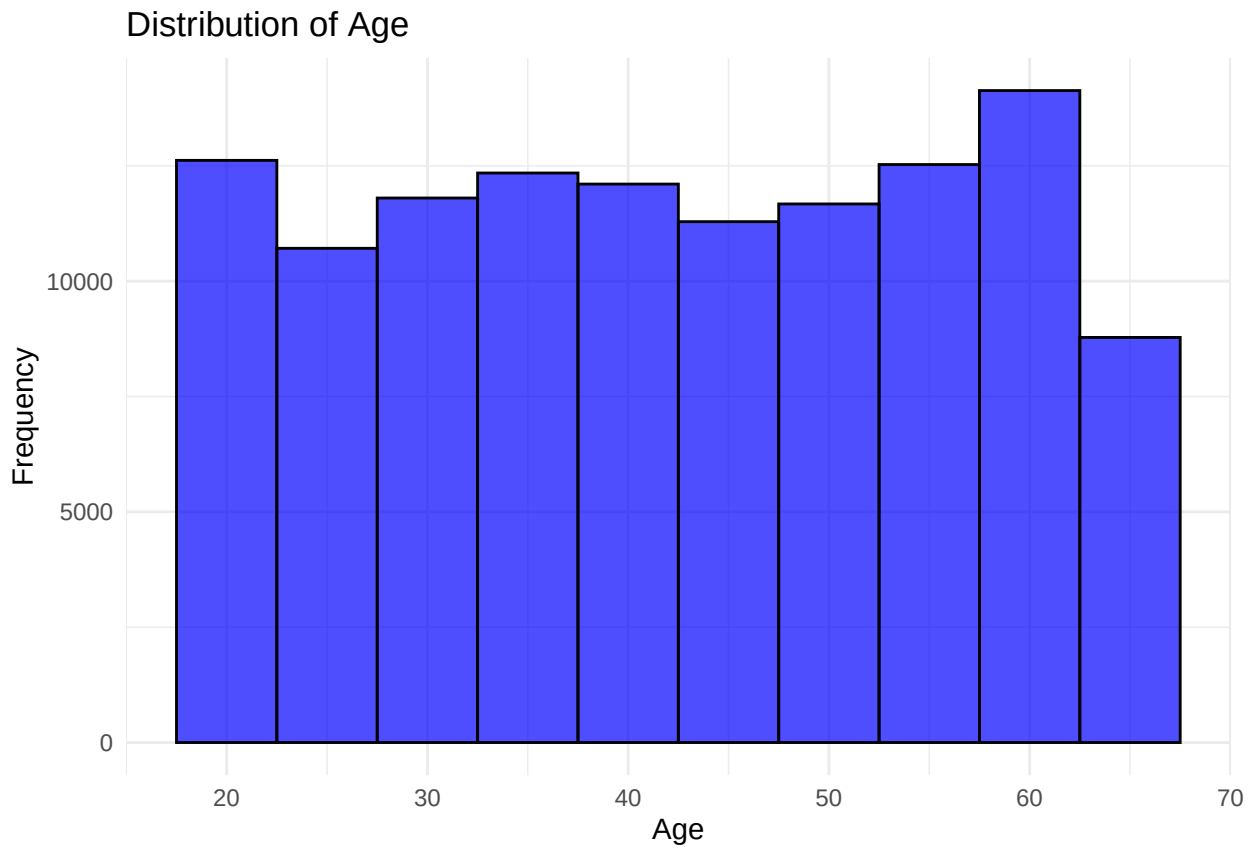
```
## [1] 105383      2
```

Original Histogram

Age

We can see in the histogram for Age that there is a relatively normal distribution with no extreme skew or any otliers present at all.

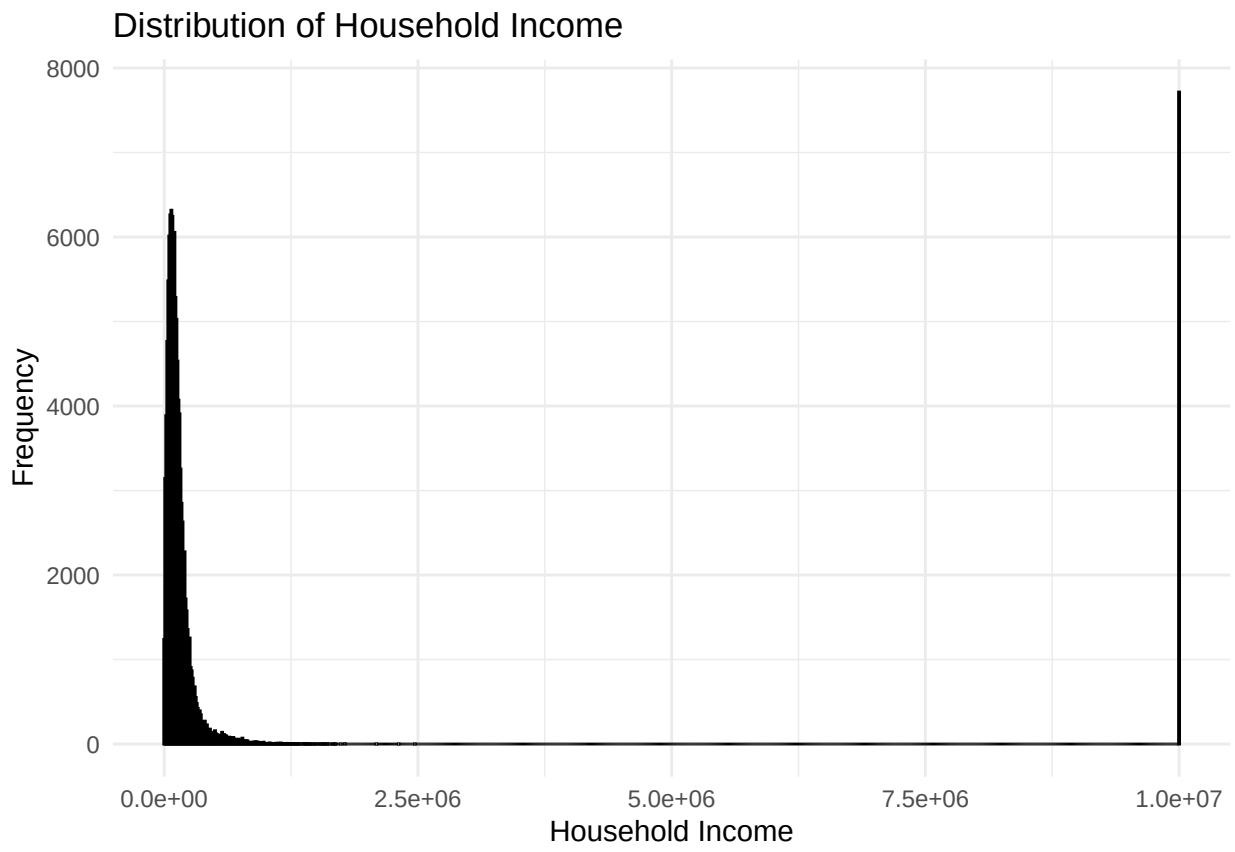
```r
ggplot(data, aes(x = AGE)) +
  geom_histogram(binwidth = 5, fill = "blue", color = "black", alpha = 0.7) +
  labs(title = "Distribution of Age", x = "Age", y = "Frequency") +
  theme_minimal()
```



Household Income We can see here that pre filtering our data has a very large right skew and an extremly large outlier present in the data set.This is als due to the fact that most individuals earn a

relatively modest and moderate incomes. It's likely this data set interviewed a lot of people who come from this category

```
ggplot(data, aes(x = HHINCOME)) +
  geom_histogram(binwidth = 10000, fill = "green", color = "black", alpha = 0.7) +
  labs(title = "Distribution of Household Income", x = "Household Income", y = "Frequency") +
  theme_minimal()
```

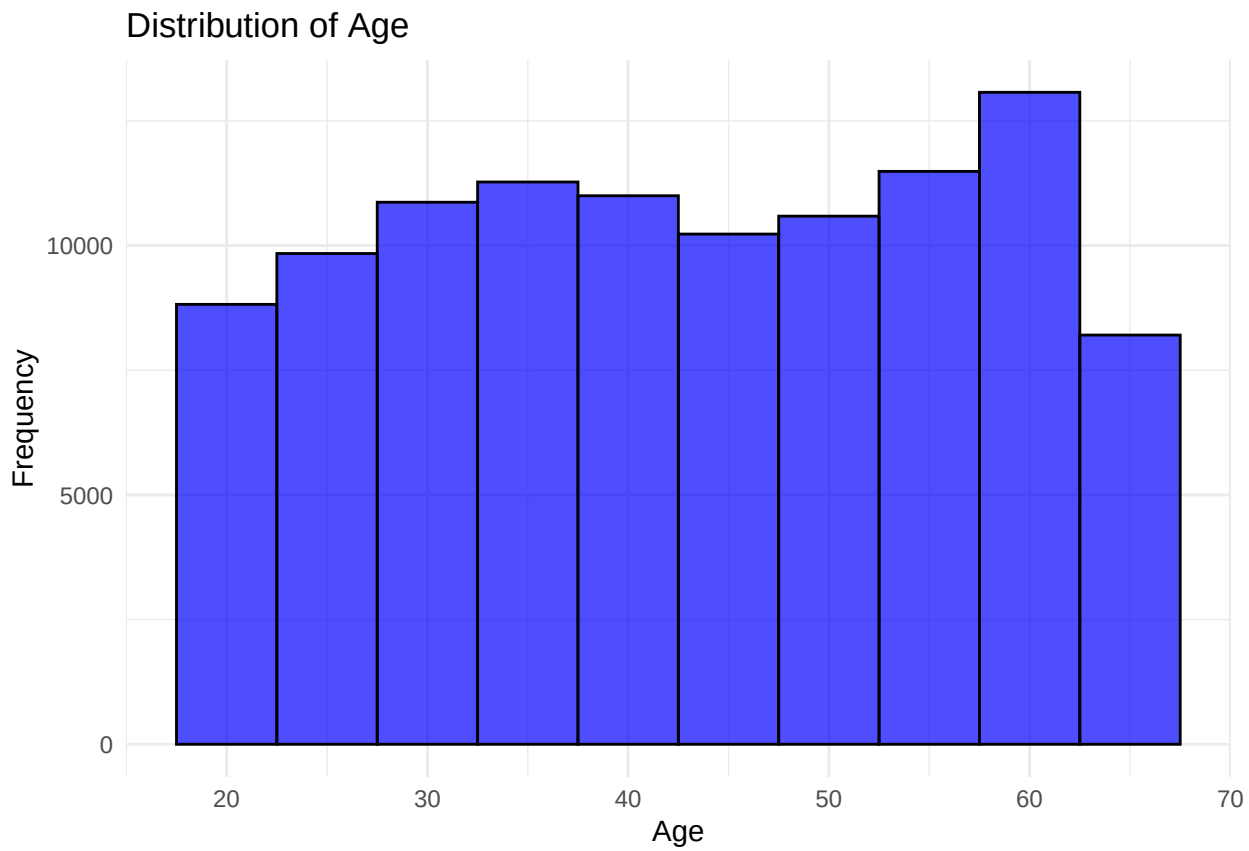


Filtered Histogram

AGE

We can see in the histogram for Age that there is a relatively normal distribution with no skew and major outliers. so we have a fairly balanced data set.

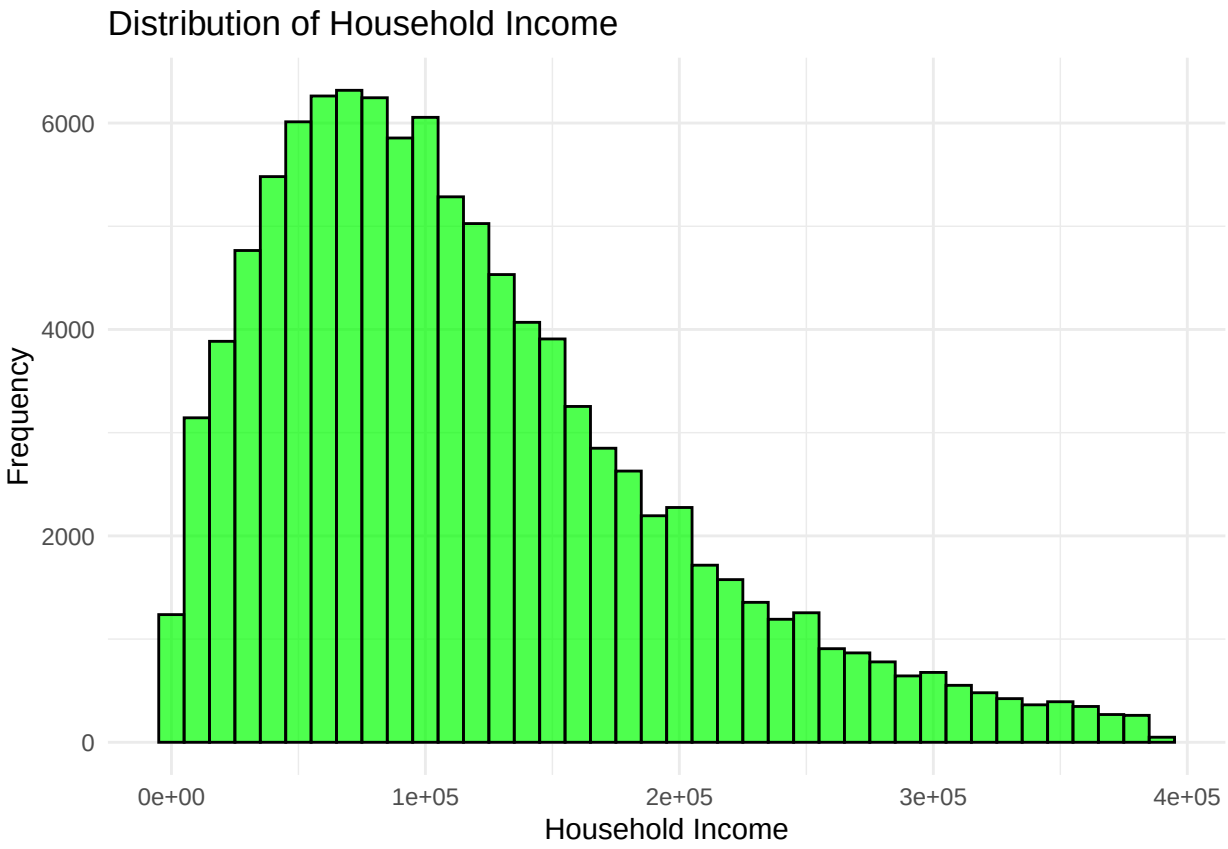
```
ggplot(filtered_data, aes(x = AGE)) +
  geom_histogram(binwidth = 5, fill = "blue", color = "black", alpha = 0.7) +
  labs(title = "Distribution of Age", x = "Age", y = "Frequency") +
  theme_minimal()
```



Household Income

althought the data set still contains a right skew, The data is alot better of a fit for this instance. There are no extreme outliers and actualy as mentioned before that skew is bound to be prevelant over the individual as most house holds in the data set earn a relatively modest income

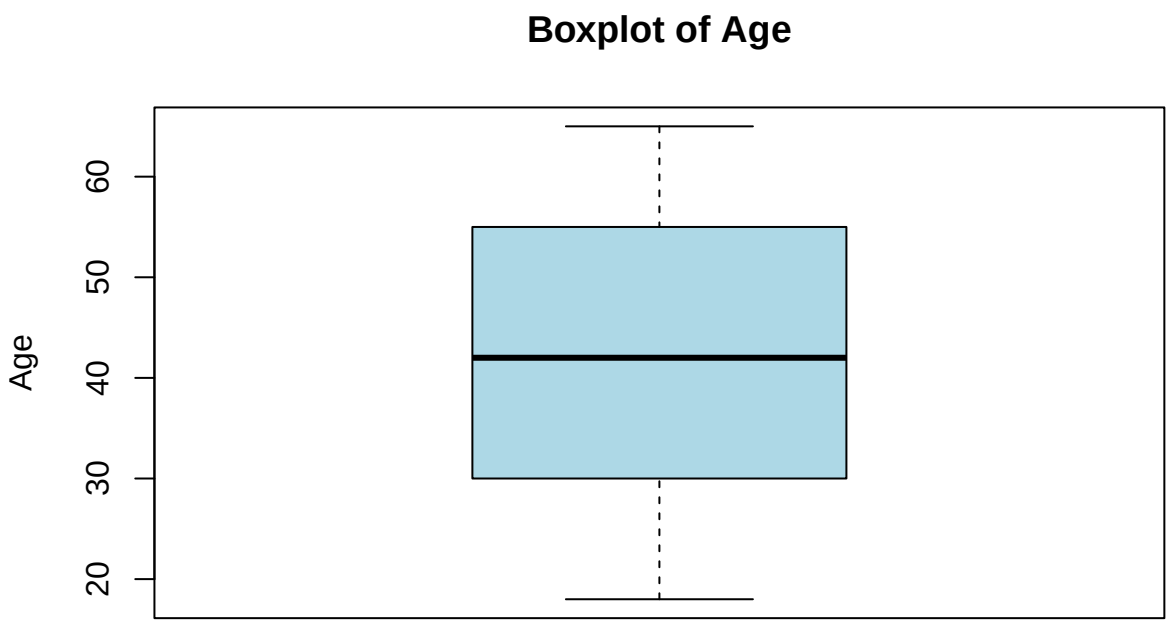
```
ggplot(filtered_data, aes(x = HHINCOME)) +
  geom_histogram(binwidth = 10000, fill = "green", color = "black", alpha = 0.7) +
  labs(title = "Distribution of Household Income", x = "Household Income", y = "Frequency") +
  theme_minimal()
```



Orignal Box Plots

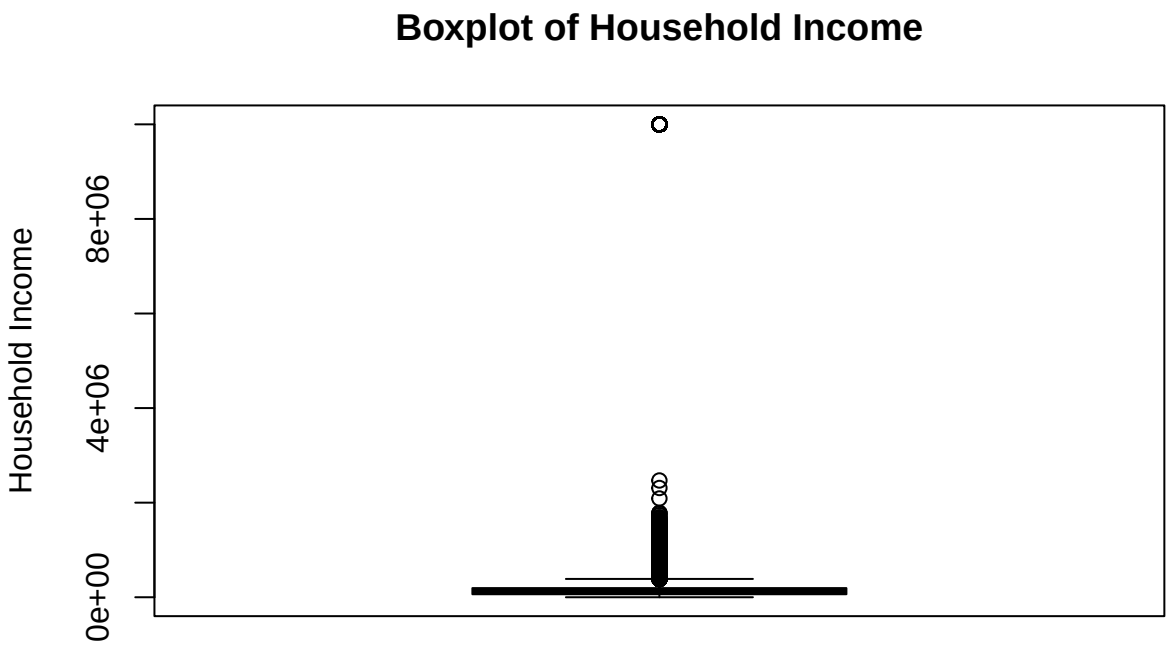
AGE

```
boxplot(data$AGE, main = "Boxplot of Age", col = "lightblue", ylab = "Age")
```



Household Income

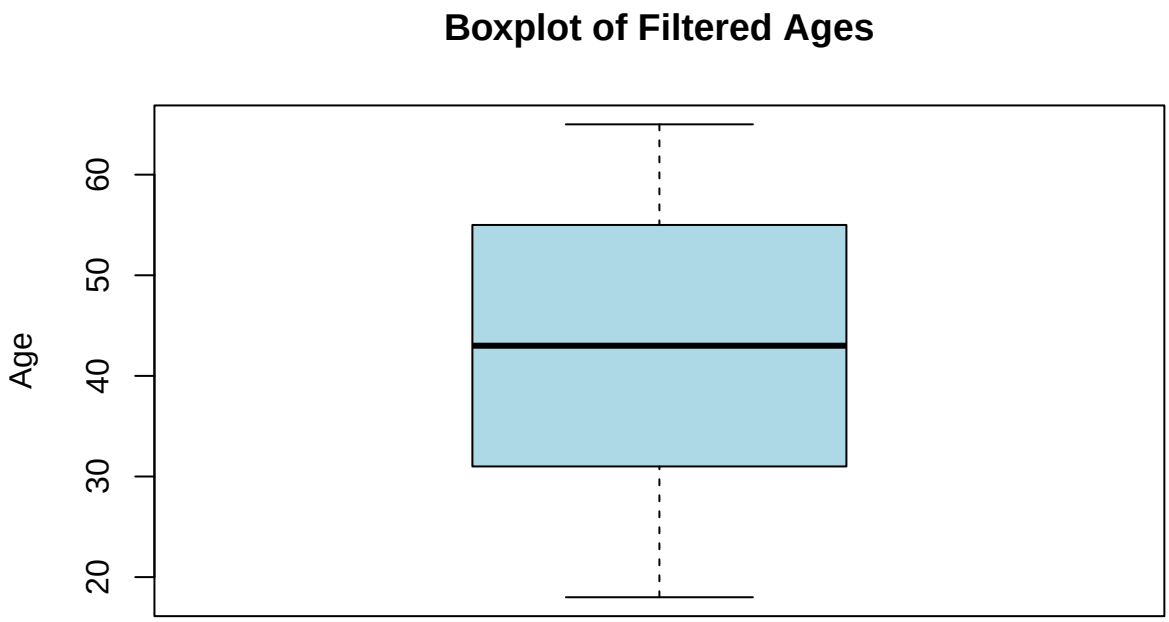
```
boxplot(data$HHINCOME, main = "Boxplot of Household Income", col = "lightgreen", ylab = "Household Incor
```



Filtered Box Plots

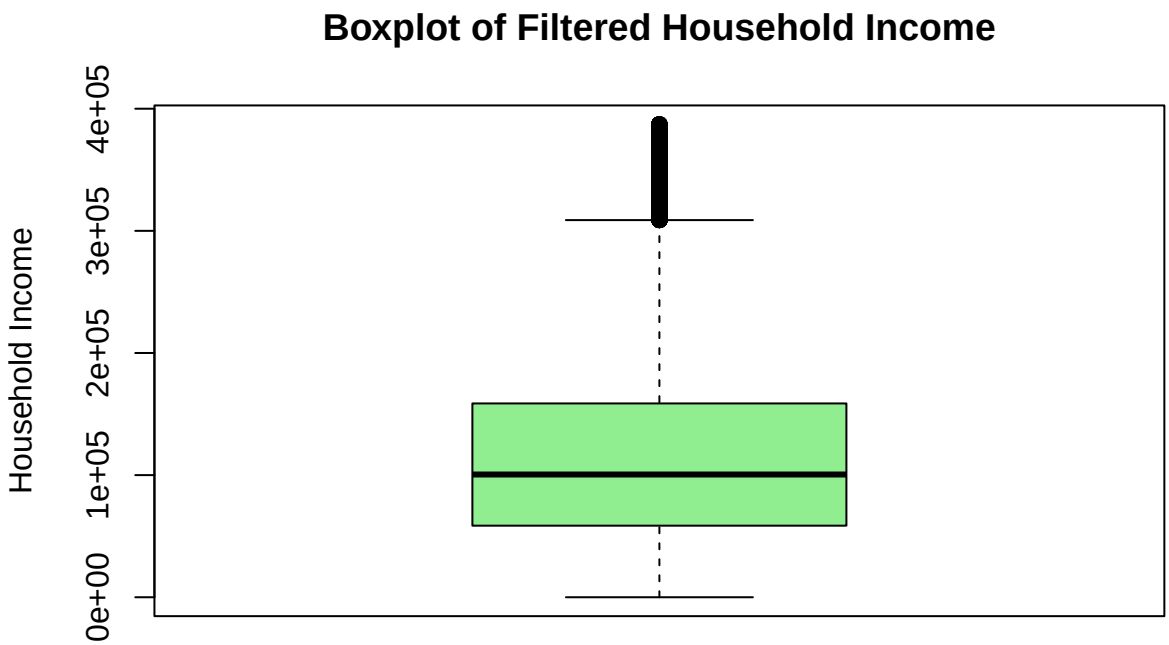
AGE

```
boxplot(filtered_data$AGE, main = "Boxplot of Filtered Ages", col = "lightblue", ylab = "Age")
```



Household Income

```
# someone edit this box plot for the outlier data that is present here
boxplot(filtered_data$HHINCOME, main = "Boxplot of Filtered Household Income", col = "lightgreen",
ylab = "Household Income")
```



#Analysis

```
# modifying data into a training set and a testing set
set.seed(1)
# ask about a good metric for the split of the data
split <- sample.split(filtered_data$HHINCOME, SplitRatio = 0.98)
train_set <- subset(filtered_data, split == TRUE)
test_set <- subset(filtered_data, split == FALSE)

#sized of the sets
dim(train_set)
```

```
## [1] 103723      2
```

```
dim(test_set)
```

```
## [1] 1660    2
```

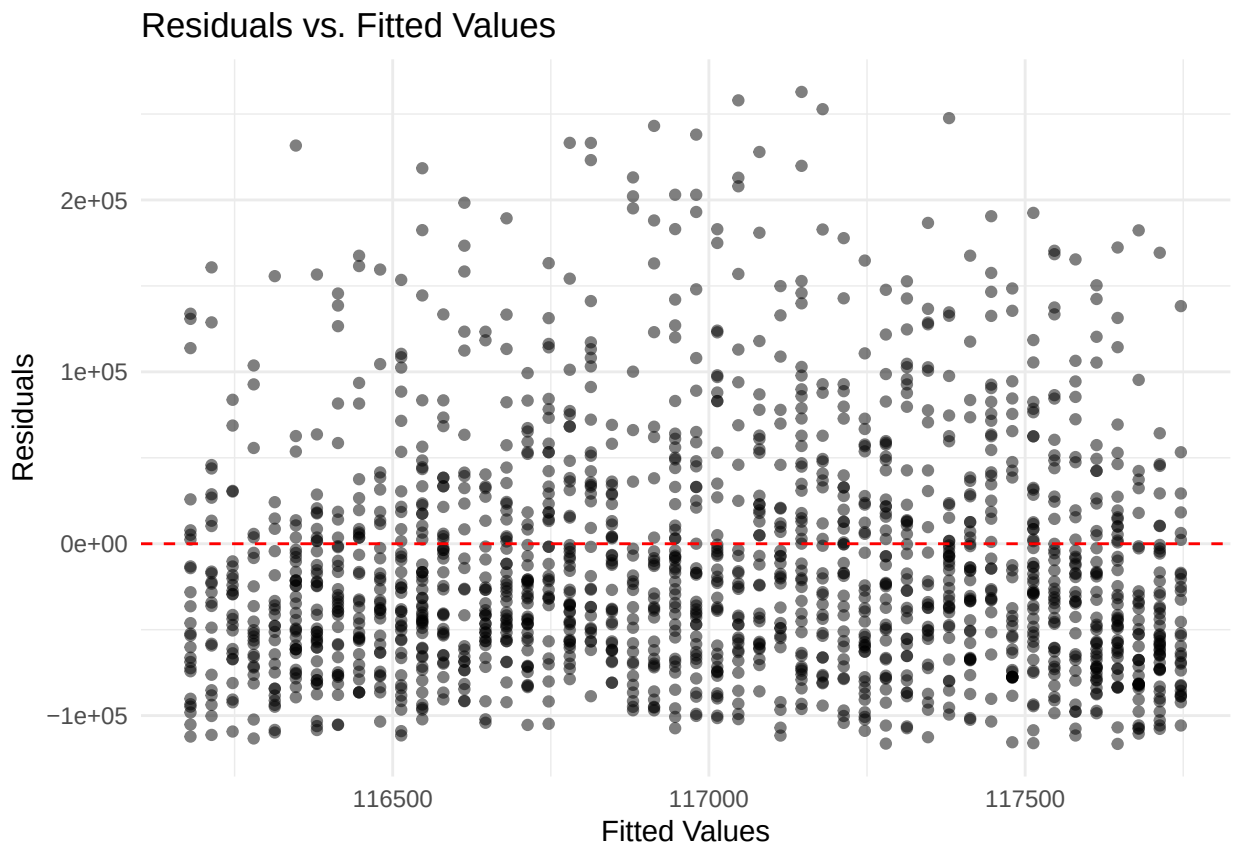
```
#model from the training data
linear_model <- lm(HHINCOME ~ AGE, data = train_set)

# Prediceted values on the test set
test_set$predicted_HHI <- predict(linear_model, newdata = test_set)

# calculate residuals for the test set
```

```r
test_set$residuals <- test_set$HHINCOME - test_set$predicted_HHI


#implement the diagonal plot

# implement the various forms of analysis to show and explain what is going on in the data set
```

Implementing the Plots

Risidual vs. Fitted Values Plot

```r
ggplot(test_set, aes(x = predicted_HHI, y = residuals)) +
  geom_point(alpha = 0.5, color = 'black') +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Residuals vs. Fitted Values",
       x = "Fitted Values",
       y = "Residuals") +
  theme_minimal()
```

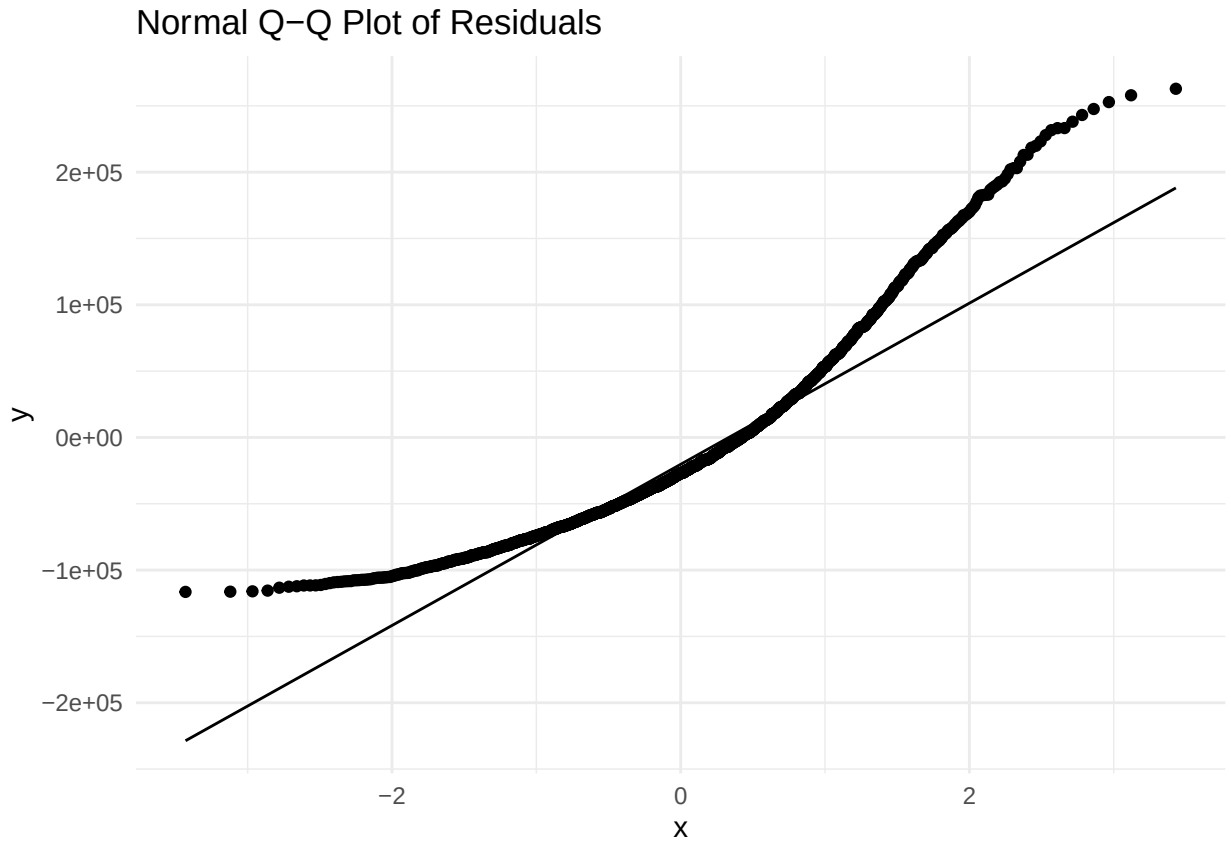


We can see from the residual plot that our data is distributed in all directions and doesn't bare any shape; however, looking at the data set, you could see that our majority of the data points are coming below the residual. That means our data has good underestimation ability and the model does have the tendency for larger over estimation compared to under estimations. It over estimates individual's household income based on the age.

In summary, This clustering of residuals below the zero line reflects systematic overestimation of earnings of individuals, particularly at earlier ages

###Normal Q-Q Plot

```
ggplot(test_set, aes(sample = residuals)) +
  stat_qq() +
  stat_qq_line() +
  labs(title = "Normal Q-Q Plot of Residuals") +
  theme_minimal()
```



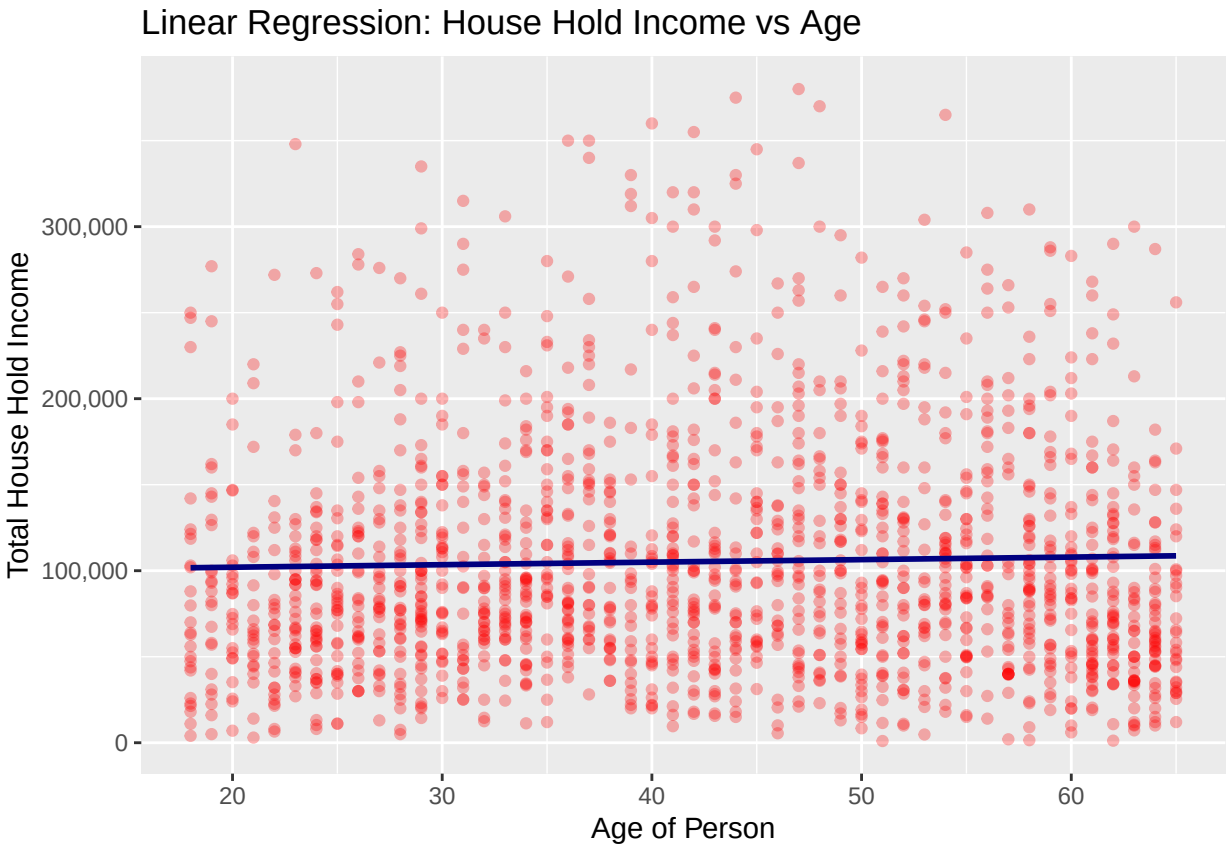
The Q-Q plot tells us about the normality of our regression model residuals in forecasting an individual's household income in the United States based on age. The points are meant to fall as close as possible to the diagonal line; however, our plot is S-shaped, indicating high skewness in the tails, i.e., income data is highly variable. This gap is likely due to the fact that the data does not suit the linear regression model very well. The points of interest on the plot indicate that household income is not distributed normally, and this may be due to outliers or a non-linear relationship between age and income.

Linear Regression with Testing Data

```
ggplot(test_set, aes(x = AGE, y = HHINCOME)) +
  geom_point(alpha = 0.3, color = "red") +  # Scatter plot of data points
  geom_smooth(method = "lm", color = "Navy", se = FALSE) +  # Regression line
  labs(title = "Linear Regression: House Hold Income vs Age",
       x = "Age of Person",
```

```
       y = "Total House Hold Income") +
  scale_y_continuous(labels = scales::comma)
```

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




```
  theme_minimal()
```

```
## List of 136
##  $ line                            :List of 6
##   ..$ colour       : chr "black"
##   ..$ linewidth    : num 0.5
##   ..$ linetype     : num 1
##   ..$ lineend      : chr "butt"
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ rect                            :List of 5
##   ..$ fill         : chr "white"
##   ..$ colour       : chr "black"
##   ..$ linewidth    : num 0.5
##   ..$ linetype     : num 1
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ text                            :List of 11
```

```
##   ..$ family       : chr ""
##   ..$ face         : chr "plain"
##   ..$ colour       : chr "black"
##   ..$ size         : num 11
##   ..$ hjust        : num 0.5
##   ..$ vjust        : num 0.5
##   ..$ angle        : num 0
##   ..$ lineheight   : num 0.9
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ title                          : NULL
##  $ aspect.ratio                   : NULL
##  $ axis.title                     : NULL
##  $ axis.title.x                   :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 2.75points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.top               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 2.75points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.bottom            : NULL
##  $ axis.title.y                   :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
```

```
##   ..$ margin       : 'margin' num [1:4] 0points 2.75points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.y.left               : NULL
##  $ axis.title.y.right              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : num -90
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 2.75points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text                       :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "grey30"
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x                     :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 2.2points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.top                 :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
```

```
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 2.2points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.bottom              : NULL
##  $ axis.text.y                     :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 1
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 2.2points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.y.left                : NULL
##  $ axis.text.y.right               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 2.2points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.theta                 : NULL
##  $ axis.text.r                     :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0.5
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 2.2points 0points 2.2points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.ticks                      : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
```

```
##  $ axis.ticks.x                   : NULL
##  $ axis.ticks.x.top               : NULL
##  $ axis.ticks.x.bottom            : NULL
##  $ axis.ticks.y                   : NULL
##  $ axis.ticks.y.left              : NULL
##  $ axis.ticks.y.right             : NULL
##  $ axis.ticks.theta               : NULL
##  $ axis.ticks.r                   : NULL
##  $ axis.minor.ticks.x.top         : NULL
##  $ axis.minor.ticks.x.bottom      : NULL
##  $ axis.minor.ticks.y.left        : NULL
##  $ axis.minor.ticks.y.right       : NULL
##  $ axis.minor.ticks.theta         : NULL
##  $ axis.minor.ticks.r             : NULL
##  $ axis.ticks.length              : 'simpleUnit' num 2.75points
##   ..- attr(*, "unit")= int 8
##  $ axis.ticks.length.x            : NULL
##  $ axis.ticks.length.x.top        : NULL
##  $ axis.ticks.length.x.bottom     : NULL
##  $ axis.ticks.length.y            : NULL
##  $ axis.ticks.length.y.left       : NULL
##  $ axis.ticks.length.y.right      : NULL
##  $ axis.ticks.length.theta        : NULL
##  $ axis.ticks.length.r            : NULL
##  $ axis.minor.ticks.length        : 'rel' num 0.75
##  $ axis.minor.ticks.length.x      : NULL
##  $ axis.minor.ticks.length.x.top  : NULL
##  $ axis.minor.ticks.length.x.bottom: NULL
##  $ axis.minor.ticks.length.y      : NULL
##  $ axis.minor.ticks.length.y.left  : NULL
##  $ axis.minor.ticks.length.y.right : NULL
##  $ axis.minor.ticks.length.theta   : NULL
##  $ axis.minor.ticks.length.r      : NULL
##  $ axis.line                      : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.line.x                    : NULL
##  $ axis.line.x.top                : NULL
##  $ axis.line.x.bottom             : NULL
##  $ axis.line.y                    : NULL
##  $ axis.line.y.left               : NULL
##  $ axis.line.y.right              : NULL
##  $ axis.line.theta                : NULL
##  $ axis.line.r                    : NULL
##  $ legend.background              : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.margin                  : 'margin' num [1:4] 5.5points 5.5points 5.5points 5.5points
##   ..- attr(*, "unit")= int 8
##  $ legend.spacing                 : 'simpleUnit' num 11points
##   ..- attr(*, "unit")= int 8
##  $ legend.spacing.x               : NULL
##  $ legend.spacing.y               : NULL
##  $ legend.key                     : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.key.size                : 'simpleUnit' num 1.2lines
```

```
##   ..- attr(*, "unit")= int 3
##  $ legend.key.height               : NULL
##  $ legend.key.width                : NULL
##  $ legend.key.spacing              : 'simpleUnit' num 5.5points
##   ..- attr(*, "unit")= int 8
##  $ legend.key.spacing.x            : NULL
##  $ legend.key.spacing.y            : NULL
##  $ legend.frame                    : NULL
##  $ legend.ticks                    : NULL
##  $ legend.ticks.length             : 'rel' num 0.2
##  $ legend.axis.line                : NULL
##  $ legend.text                     :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.text.position            : NULL
##  $ legend.title                    :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.title.position           : NULL
##  $ legend.position                 : chr "right"
##  $ legend.position.inside          : NULL
##  $ legend.direction                : NULL
##  $ legend.byrow                    : NULL
##  $ legend.justification            : chr "center"
##  $ legend.justification.top        : NULL
##  $ legend.justification.bottom     : NULL
##  $ legend.justification.left       : NULL
##  $ legend.justification.right      : NULL
##  $ legend.justification.inside     : NULL
##  $ legend.location                 : NULL
##  $ legend.box                      : NULL
##  $ legend.box.just                 : NULL
##  $ legend.box.margin               : 'margin' num [1:4] 0cm 0cm 0cm 0cm
##   ..- attr(*, "unit")= int 1
```

```
##  $ legend.box.background           : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.box.spacing              : 'simpleUnit' num 11points
##   ..- attr(*, "unit")= int 8
##   [list output truncated]
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi TRUE
##  - attr(*, "validate")= logi TRUE
```

From our model we can see that the linear regression line is an extremely poor fit and infact actually resembles the set up of the residual vs fitted value graphs. This output further suggest that a linear regression model is not appropriate for the data, as the residuals show the lower cluster patterns, and the speratic data points for the testing points appear to fail to encompass the model with any sort of trends.

Summary of the Simple Linear Regression Model

```
summary(linear_model)
```

```
##
## Call:
## lm(formula = HHINCOME ~ AGE, data = train_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -117716  -58313  -16213   41987  271020
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 115580.53     787.18 146.829   <2e-16 ***
## AGE             33.32      17.47   1.907   0.0565 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 78150 on 103721 degrees of freedom
## Multiple R-squared:  3.506e-05,  Adjusted R-squared:  2.542e-05
## F-statistic: 3.636 on 1 and 103721 DF,  p-value: 0.05653
```

From the Residual ranges we can see that the linear model has some very large error when it comes to underestimating and overestimating a US citizens house hold income. Our coefficient for the Age of a person is 33.32 meaning that the model predicts that for every additional year of age, household income increases bu 33.32 dollars on average. this is an extremely small change which can only mean the age alone is not a very strong depiction of income in our linear model.

Model Evaluation and Prediction

Conclusion and Summary

Reference