Project_1_456

Anthony Yasan, Preston O'Connor

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

Information about the Data set

Table of Data

```r
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
```

```r
#View(data)
# here Ther Code struggles to run the data set with 2million 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

```r
# 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))

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

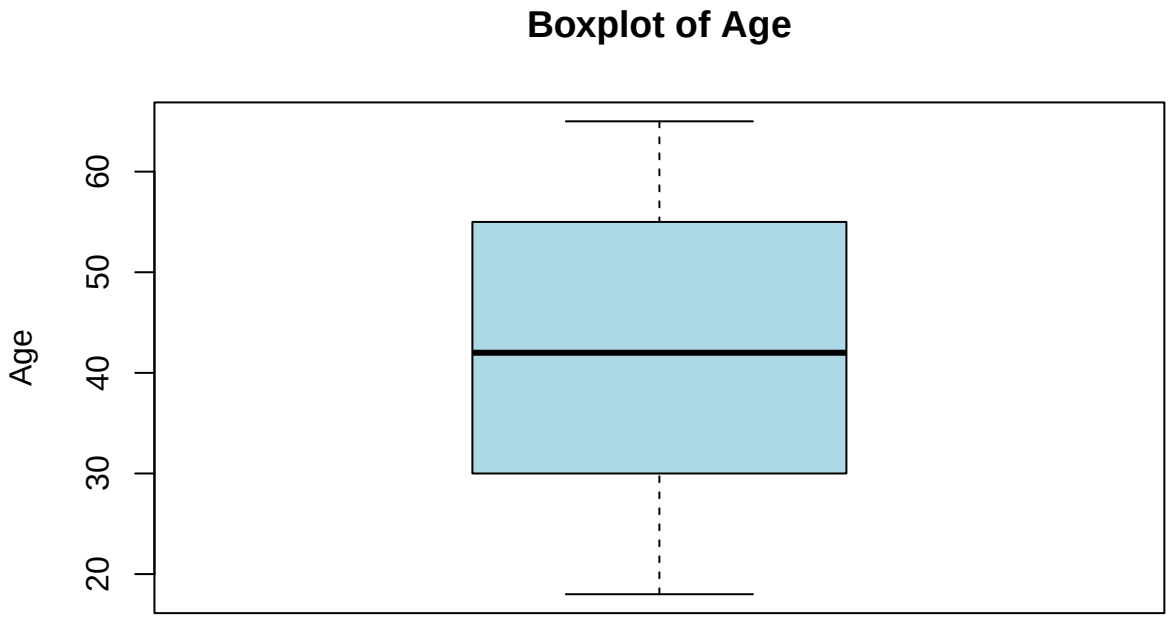
```
## [1] 118817      2
```

```r
IQR_of_AGE <- IQR(data$AGE)
IQR_of_HHINCOME <- IQR(data$HHINCOME)

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

HHINCOME_lower <- quantile(data$HHINCOME, 0.25) - 1.5 * IQR_of_HHINCOME
HHINCOME_upper <- quantile(data$HHINCOME, 0.75) + 1.5 * IQR_of_HHINCOME

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

Orignal Box Plots

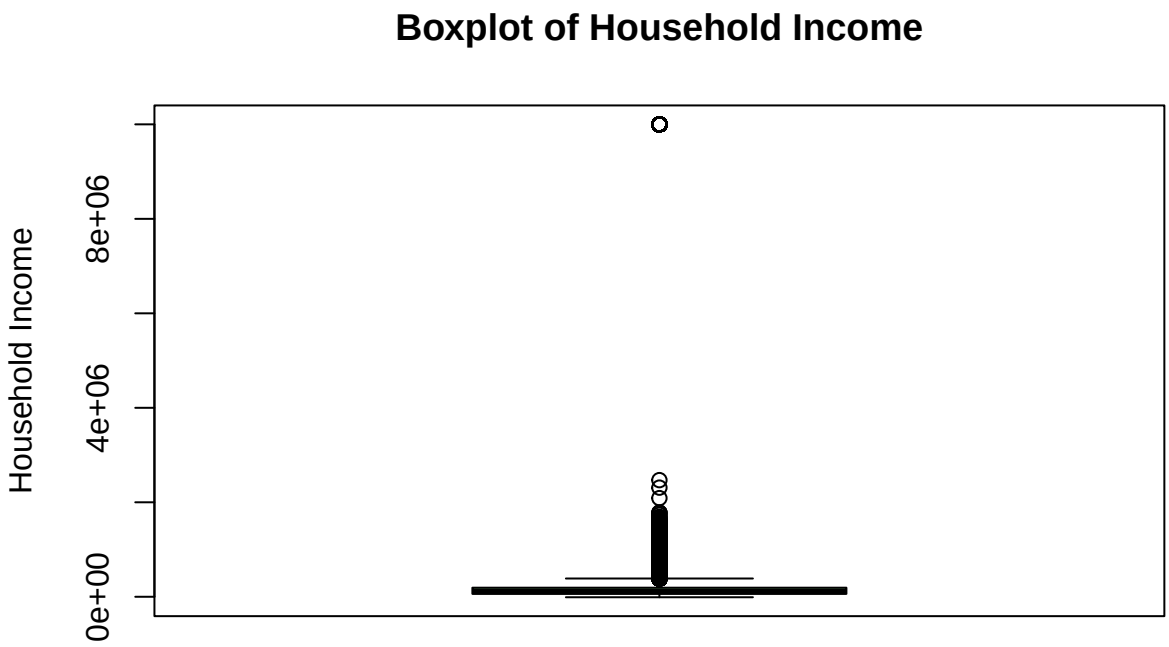
AGE

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



Total House Hold Income

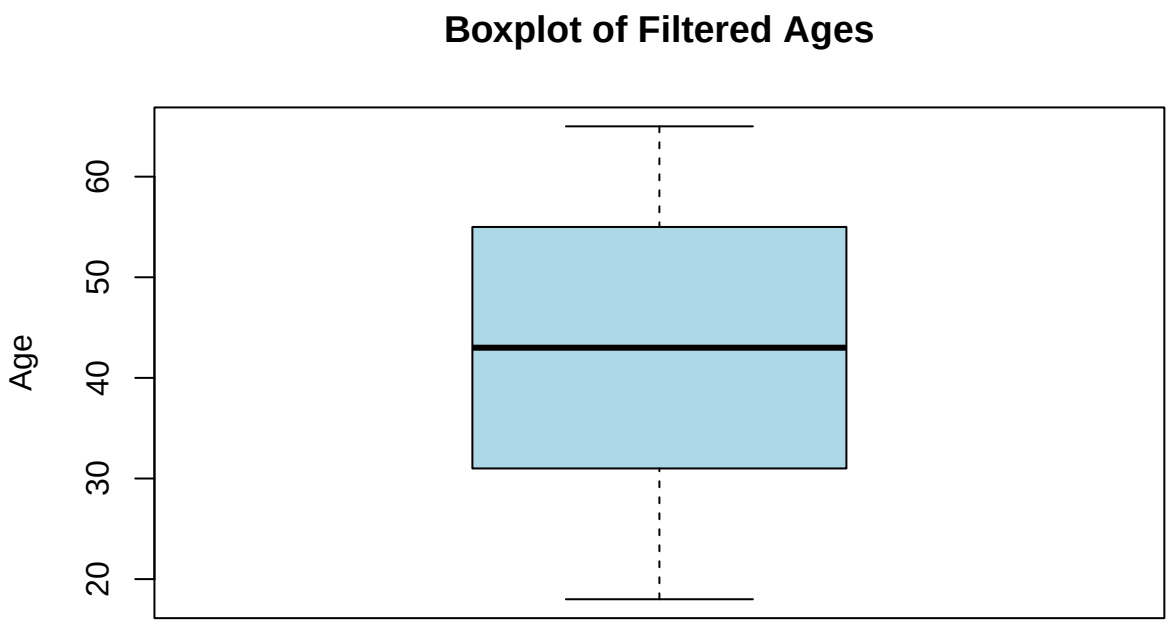
```
boxplot(data$HHINCOME, main = "Boxplot of Household Income", col = "lightgreen", ylab = "Household Incor
```



filtered Box Plots

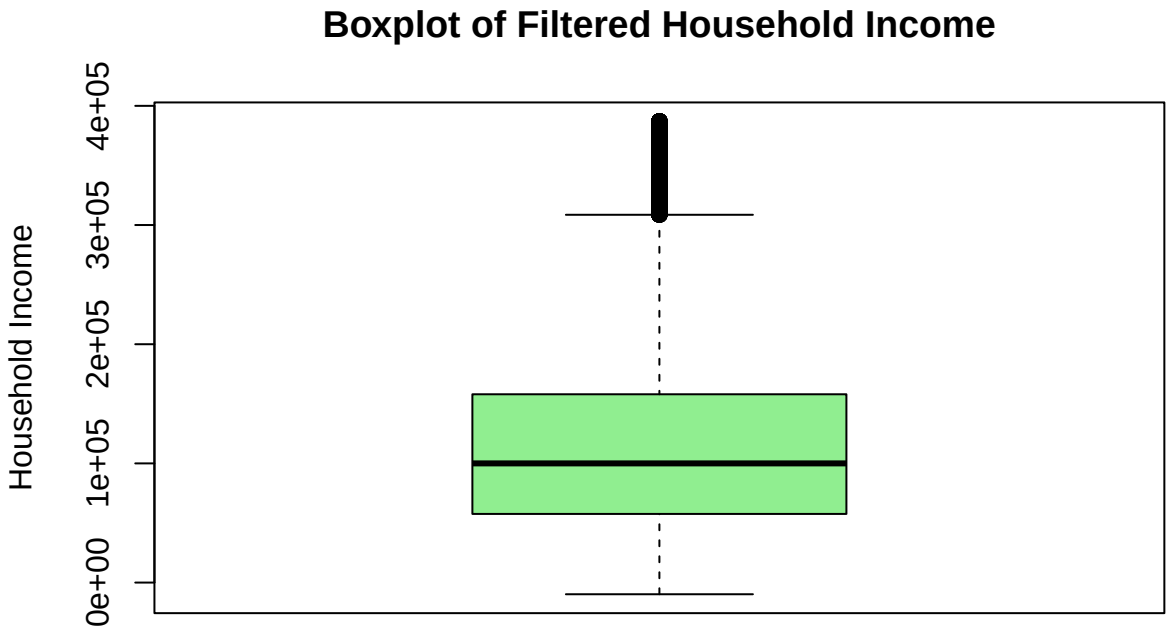
Filtered and Cleaned AGE

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



Filter and Cleaned Total House Hold Income

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



#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] 104541      2
```

```
dim(test_set)
```

```
## [1] 1676    2
```

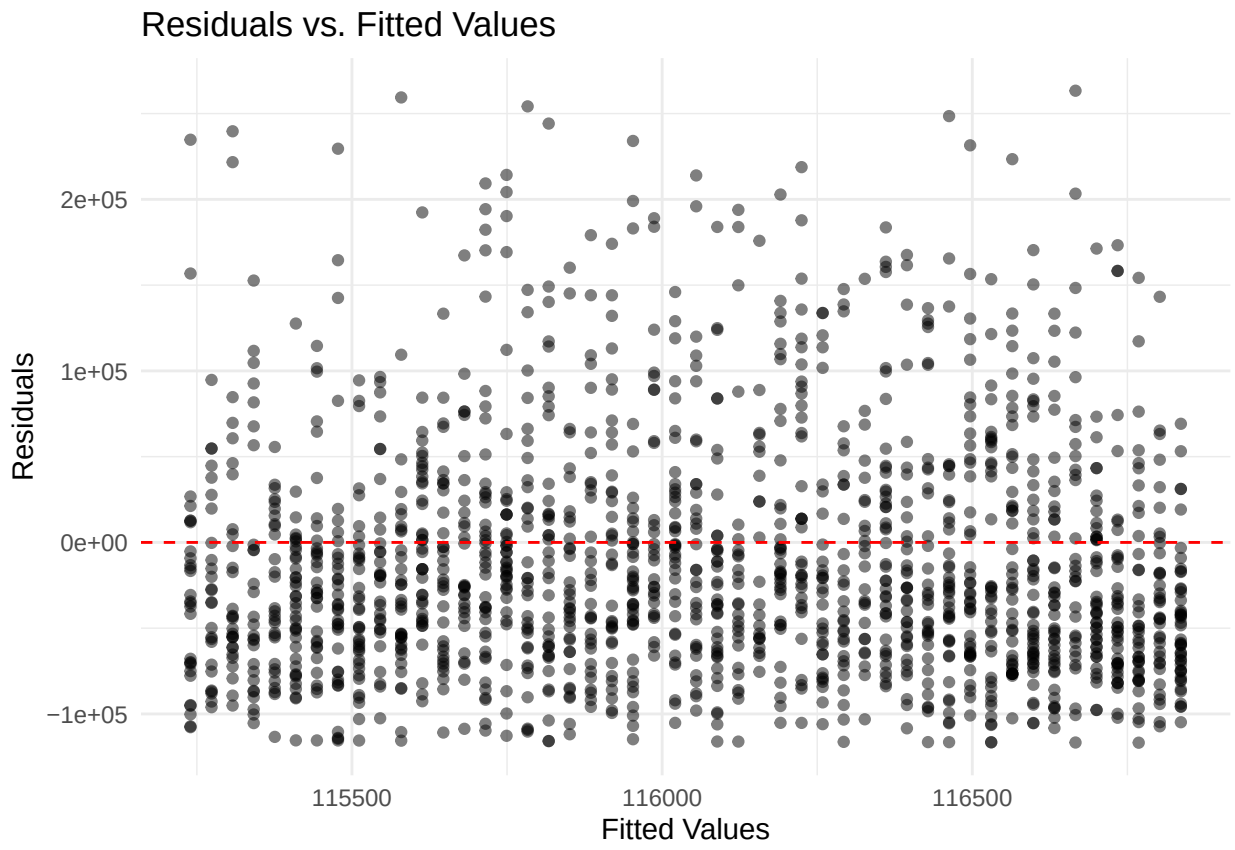
```
#model from the training data
linear_model <- lm(HHINCOME ~ AGE, data = train_set)

# 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

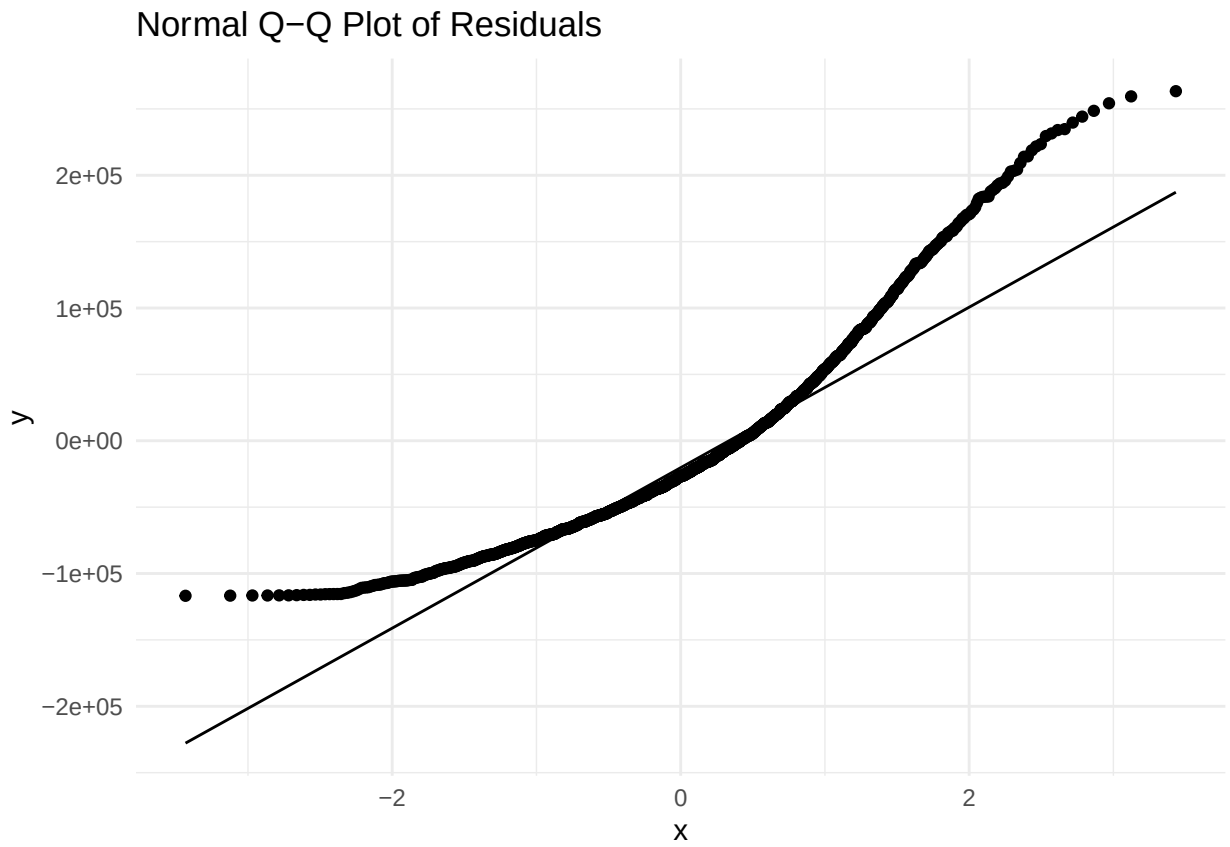
```r
ggplot(test_set, aes(x = predicted_HHI, y = residuals)) +
  geom_point(alpha = 0.5, color = 'black') +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Residuals vs. Fitted Values",
       x = "Fitted Values",
       y = "Residuals") +
  theme_minimal()
```



###Normal Q-Q Plot

```r
ggplot(test_set, aes(sample = residuals)) +
  stat_qq() +
```

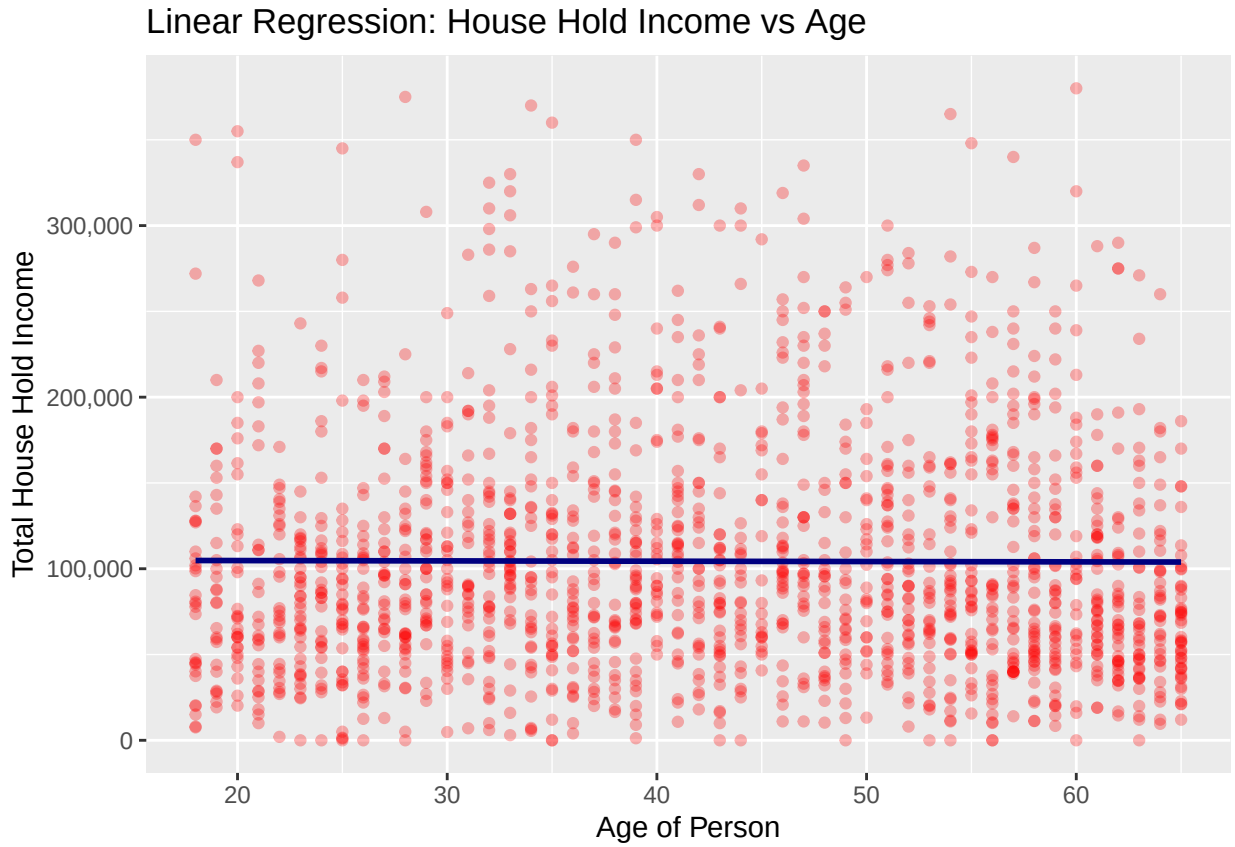
```
  stat_qq_line() +
  labs(title = "Normal Q-Q Plot of Residuals") +
  theme_minimal()
```



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
```

Summary of the Simple Linear Regression Model

```
summary(linear_model)
```

```
##
## Call:
## lm(formula = HHINCOME ~ AGE, data = train_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -125821  -58545  -15715   42164  271658
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 114628.32     787.72 145.520   <2e-16 ***
## AGE             33.97      17.48   1.943    0.052 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 78520 on 104539 degrees of freedom
## Multiple R-squared:  3.611e-05,  Adjusted R-squared:  2.654e-05
## F-statistic: 3.775 on 1 and 104539 DF,  p-value: 0.05203
```

Model Evaluation and Prediction

Conclusion and Summary

Reference