Predicting Income with Census Data

I. Introduction

This project aims to predict an individual's income level with machine learning models using the 14 attributes provided by the census income dataset. Specifically, I implement a classification tree and random forest algorithm to predict whether a person's annual income is more than $50,000.

The "Census Income" dataset, also known as the "Adult" dataset, was extracted by Barry Becker from the 1994 US Census database for a prediction task on income level. The dataset consists of 48,842 observations and 15 variables, six numerical and nine categorical, and is archived at the UC Irvine Machine Learning Repository. The following is a list of variable names by their types.

Categorical Variable	Numerical Variable
income	age
workclass	fnlwgt
education	education.num
occupation	capital.gain
marital.status	capital.loss
relationship	hours.per.week
race	
sex	
native.country	

The `income` variable identifies whether a person makes over $50,000 annually, and is used as the dependent variable. The dataset is divided into a `train` set with 32,561 observations, and a `test` set with 16,281 observations. After removing the missing values denoted by question marks, there are 45,222 rows left in the dataset with 30,162 rows in the `train` set and 15,060 rows in the `test` set.

The variable names are mostly self-explanatory, with a few exceptions. The variable `education` is a categorical representation of the numerical `education.num`, which stands for the number of years of education. The variable `relationship` indicates the respondent's role in the family. The variable `fnlwgt` specifies the final weight, which is the number of people the census believes the entry represents. Gain and loss from investments are stored in `capital.gain` and `capital.loss`. The variable `hours.per.week` indicates the number of working hours per week.

```
# Download the Adult Dataset (Census Income Dataset)

# Census Income Data Set:
# https://archive.ics.uci.edu/ml/datasets/Adult

url_train <- 'http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data'
url_test <- 'http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.test'
```

```r
url_names <- 'http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.names'

train <- read.table(url_train, sep = ',', stringsAsFactors = FALSE)
test <- readLines(url_test)[-1]
test <- read.table(textConnection(test), sep = ',', stringsAsFactors = FALSE)

names <- readLines(url_names)[97:110]
names <- as.character(lapply(strsplit(names,':'), function(x) x[1]))
names <- c(names, 'income')
colnames(train) <- names
colnames(test) <- names

# Remove the missing values (denoted by question marks)

no.question.mark <- apply(train, 1, function(r) !any(r %in% ' ?'))
train <- train[no.question.mark,]
no.question.mark <- apply(test, 1, function(r) !any(r %in% ' ?'))
test <- test[no.question.mark,]

train <- as.data.frame(unclass(train),stringsAsFactors = T)
test <- as.data.frame(unclass(test),stringsAsFactors = T)

# Create adult set (combine train, test set)

adult <- rbind(train, test)

# Remove the "." from "<=50K." and ">50K." in adult set

adult$income <- gsub(".", "", as.character(adult$income), fixed = TRUE)

str(adult)
```

```
## 'data.frame':    45222 obs. of  15 variables:
##  $ age           : int  39 50 38 53 28 37 49 52 31 42 ...
##  $ workclass     : Factor w/ 7 levels " Federal-gov",..: 6 5 3 3 3 3 3 5 3 3 ...
##  $ fnlwgt        : int  77516 83311 215646 234721 338409 284582 160187 209642 45781 159449 ...
##  $ education     : Factor w/ 16 levels " 10th"," 11th",..: 10 10 12 2 10 13 7 12 13 10 ...
##  $ education.num : int  13 13 9 7 13 14 5 9 14 13 ...
##  $ marital.status: Factor w/ 7 levels " Divorced"," Married-AF-spouse",..: 5 3 1 3 3 3 4 3 5 3 ...
##  $ occupation    : Factor w/ 14 levels " Adm-clerical",..: 1 4 6 6 10 4 8 4 10 4 ...
##  $ relationship  : Factor w/ 6 levels " Husband"," Not-in-family",..: 2 1 2 1 6 6 2 1 2 1 ...
##  $ race          : Factor w/ 5 levels " Amer-Indian-Eskimo",..: 5 5 5 3 3 5 3 5 5 5 ...
##  $ sex           : Factor w/ 2 levels " Female"," Male": 2 2 2 2 1 1 1 2 1 2 ...
##  $ capital.gain  : int  2174 0 0 0 0 0 0 0 14084 5178 ...
##  $ capital.loss  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hours.per.week: int  40 13 40 40 40 40 16 45 50 40 ...
##  $ native.country: Factor w/ 41 levels " Cambodia"," Canada",..: 39 39 39 39 5 39 23 39 39 39 ...
##  $ income        : chr  " <=50K" " <=50K" " <=50K" " <=50K" ...
```

```r
dim(train)
```

```
## [1] 30162    15
```

```r
dim(test)
```

```
## [1] 15060    15
```

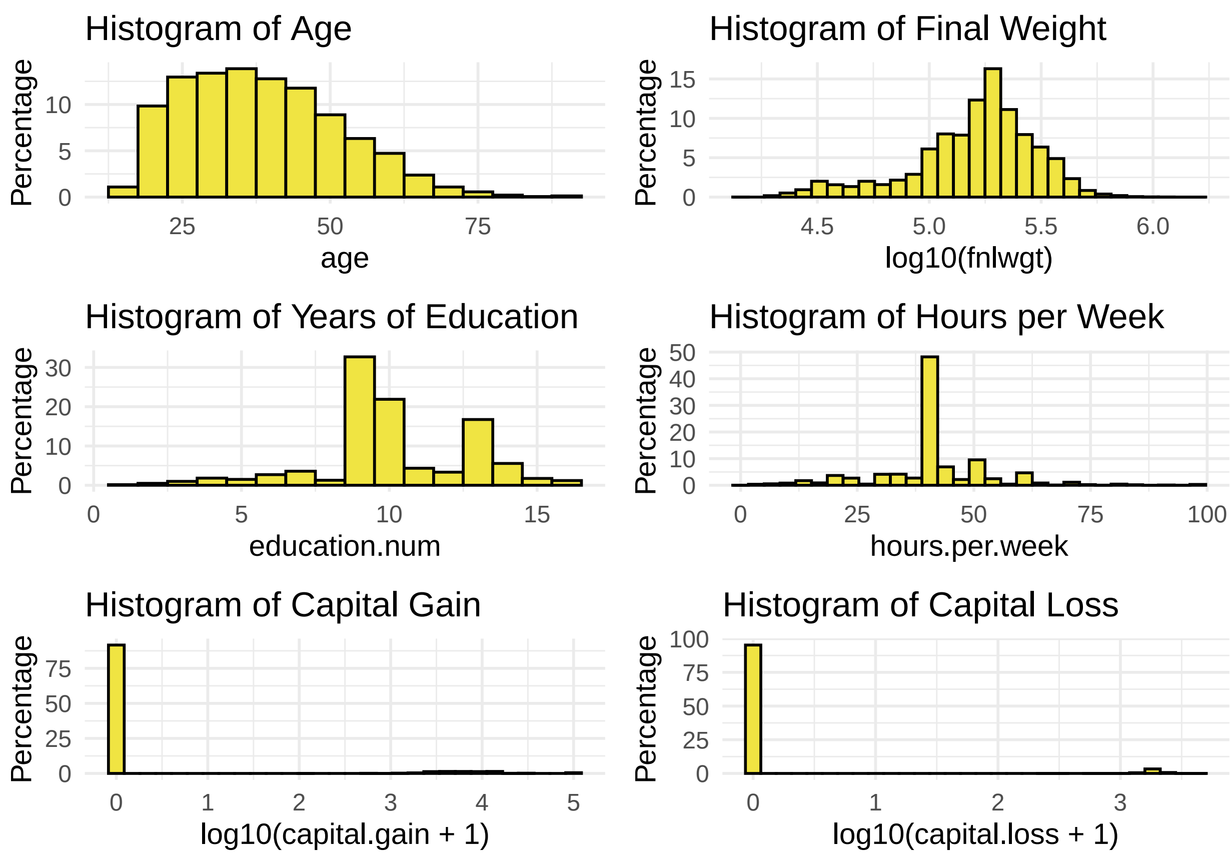
II. Analysis

2.1 Data Exploration

From the histograms of the numerical variables, I find that the distributions of `capital.gain` and `capital.loss` are very narrow and concentrated at zero. The zeros constitute 91.61912% of `capital.gain` entries and 95.26779% of `capital.loss` entries. In the modeling section, I combine these two columns into a single variable called `capital.change` for more efficient representation.

```r
# Visualize numerical variables

# Histogram of Age

p1 <- ggplot(train, aes(x = age)) +
  ggtitle("Histogram of Age") +
  geom_histogram(
    aes(y = 100*(..count..)/sum(..count..)),
    binwidth = 5, colour = "black", fill = "#F0E442"
    ) +
  ylab("Percentage") +
  theme_minimal()

# Histogram of Final Weight

p2 <- ggplot(adult, aes(x = log10(fnlwgt))) +
  ggtitle("Histogram of Final Weight") +
  geom_histogram(
    aes(y = 100*(..count..)/sum(..count..)),
    colour = "black", fill = "#F0E442"
    ) +
  ylab("Percentage") +
  theme_minimal()

# Histogram of Years of Education

p3 <- ggplot(adult, aes(x = education.num)) +
  ggtitle("Histogram of Years of Education") +
  geom_histogram(
    aes(y = 100*(..count..)/sum(..count..)),
    binwidth = 1, colour = "black", fill = "#F0E442"
    ) +
  ylab("Percentage") +
  theme_minimal()

# Histogram of Hours per Week

p4 <- ggplot(adult, aes(x = hours.per.week)) +
  ggtitle("Histogram of Hours per Week") +
```

```r
  geom_histogram(
    aes(y = 100*(..count..)/sum(..count..)),
    colour = "black", fill = "#F0E442"
    ) +
  ylab("Percentage") +
  theme_minimal()

# Histogram of Capital Gain

p5 <- ggplot(adult, aes(x = log10(capital.gain+1))) +
  ggtitle("Histogram of Capital Gain") +
  geom_histogram(
    aes(y = 100*(..count..)/sum(..count..)),
    colour = "black", fill = "#F0E442"
    ) +
  ylab("Percentage") +
  theme_minimal()

# Histogram of Capital Loss

p6 <- ggplot(adult, aes(x = log10(capital.loss+1))) +
  ggtitle("Histogram of Capital Loss") +
  geom_histogram(
    aes(y = 100*(..count..)/sum(..count..)),
    colour = "black", fill = "#F0E442"
    ) +
  ylab("Percentage") +
  theme_minimal()

grid.arrange(p1,p2,p3,p4,p5,p6)
```




```r
# Percentage of data with zero capital gain

sum(adult$capital.gain==0)/length(adult$capital.gain)*100
```

```
## [1] 91.61912
```

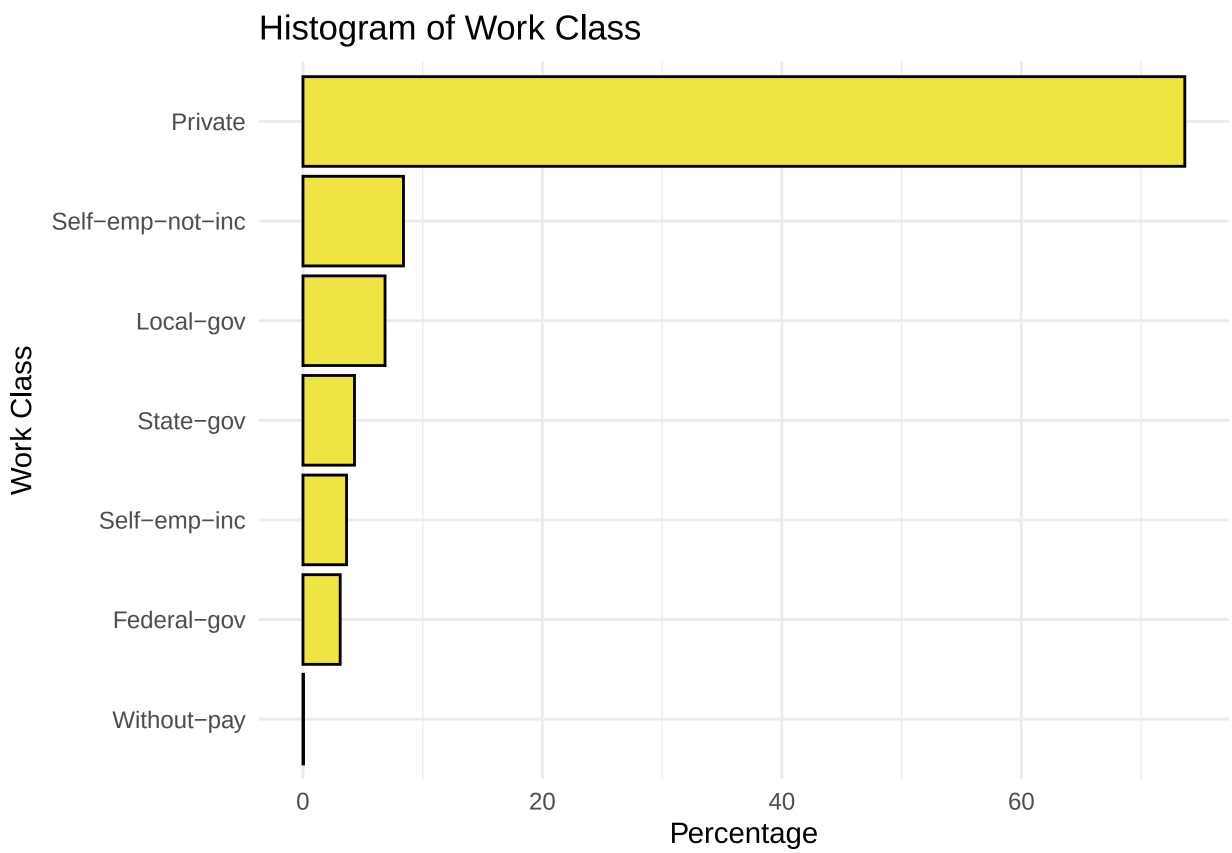
```r
# Percentage of data with zero capital loss

sum(adult$capital.loss==0)/length(adult$capital.loss)*100
```

```
## [1] 95.26779
```

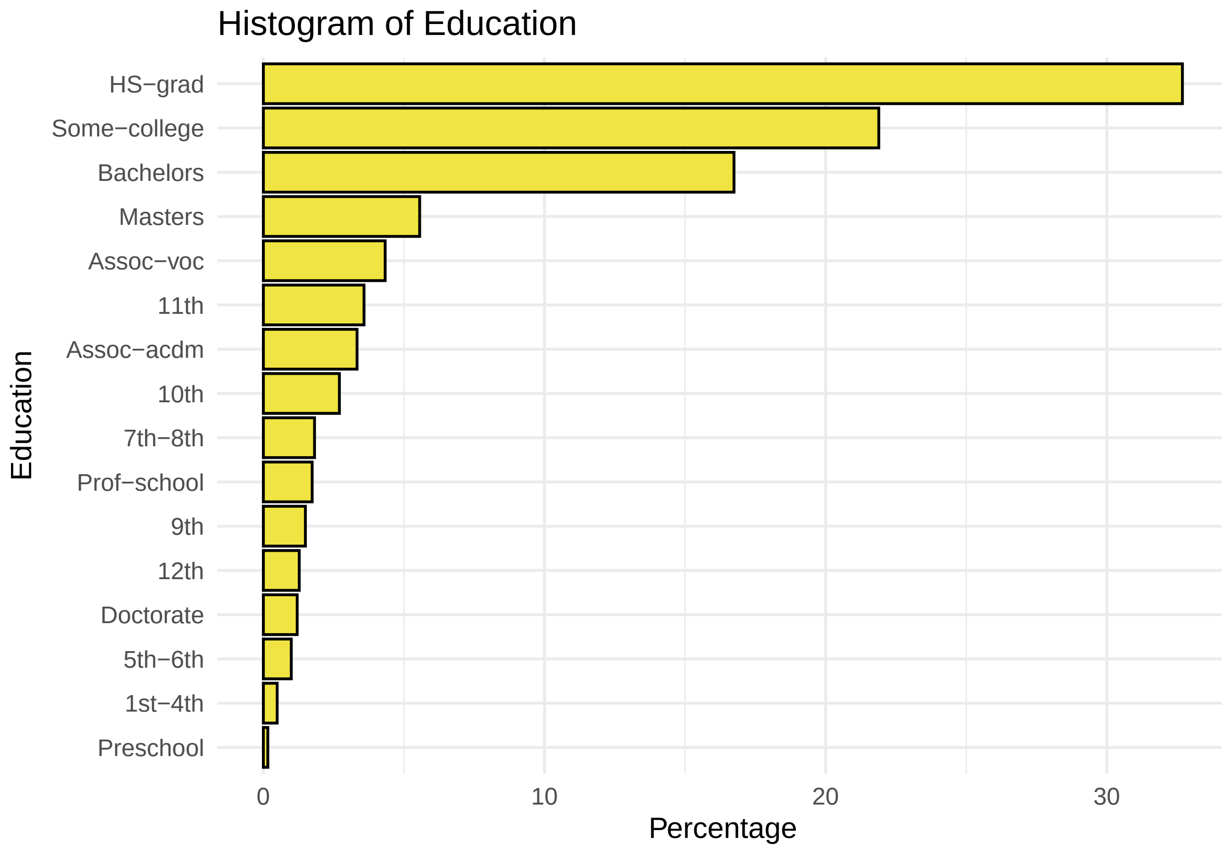
The histograms of the categorical variables are plotted as follows.

```r
# Sort categorical variables in descending order

sort.categ <- function(x){reorder(x,x,function(y){length(y)})}
var.categ <- which(sapply(adult, is.factor))
for (c in var.categ){adult[,c] <- sort.categ(adult[,c])}
attach(adult)

# Histogram of Work Class

ggplot(adult, aes(y = workclass)) +
```

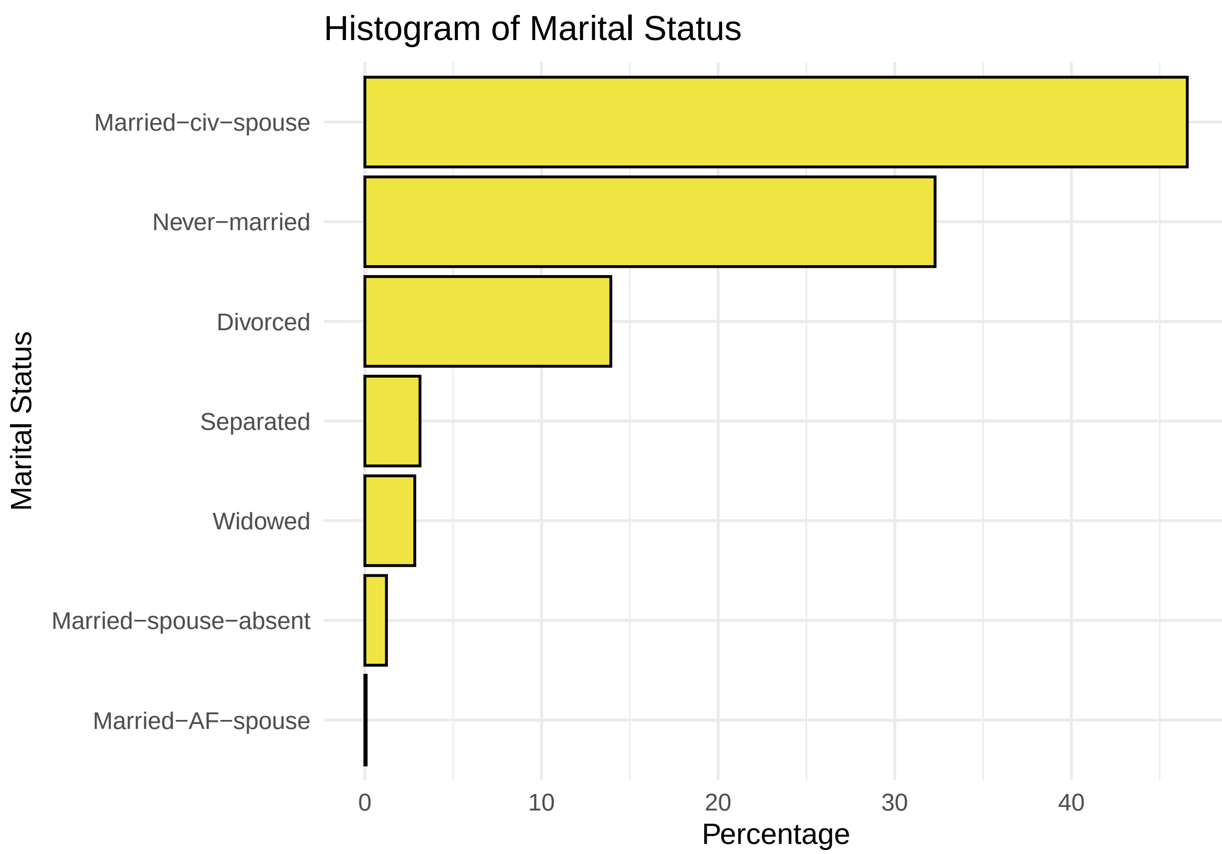
```
  ggtitle("Histogram of Work Class") +
  geom_bar(aes(x = 100*(..count..)/sum(..count..)), colour = "black", fill = "#F0E442") +
  scale_y_discrete(limits = levels(workclass)) +
  xlab("Percentage") +
  ylab("Work Class") +
  theme_minimal()
```



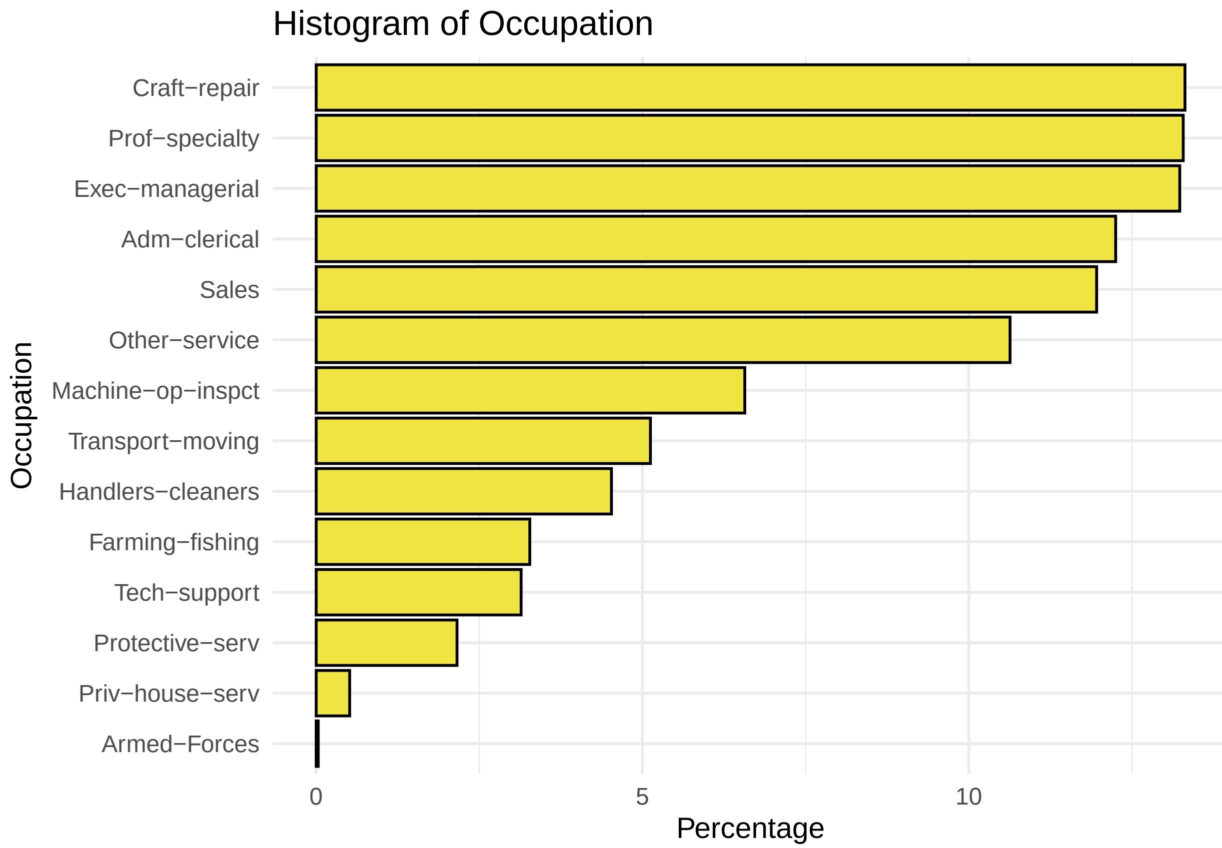

```
# Histogram of Education

ggplot(adult, aes(y = education)) +
  ggtitle("Histogram of Education") +
  geom_bar(aes(x = 100*(..count..)/sum(..count..)), colour = "black", fill = "#F0E442") +
  scale_y_discrete(limits = levels(education)) +
  xlab("Percentage") +
  ylab("Education") +
  theme_minimal()
```



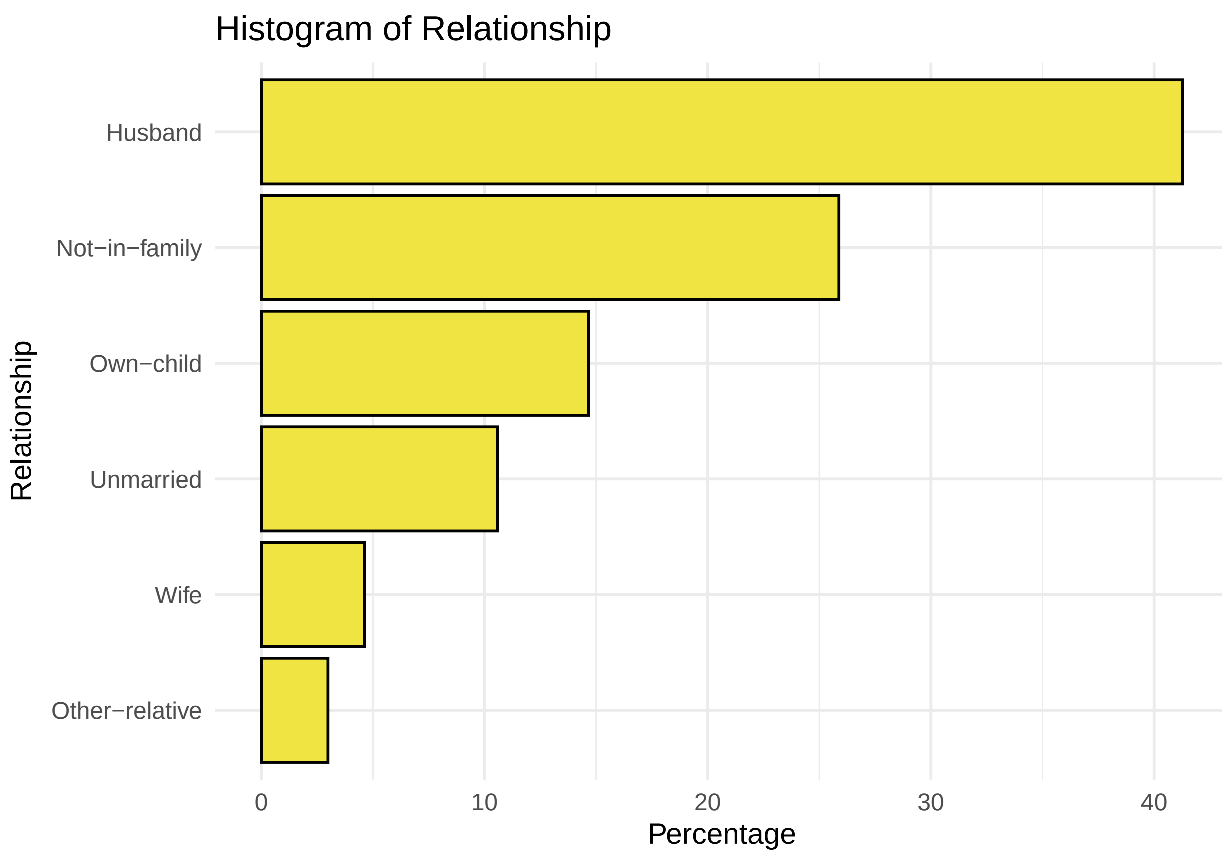

```
# Histogram of Marital Status

ggplot(adult, aes(y = marital.status)) +
  ggtitle("Histogram of Marital Status") +
  geom_bar(aes(x = 100*(..count..)/sum(..count..)), colour = "black", fill = "#F0E442") +
  scale_y_discrete(limits = levels(marital.status)) +
  xlab("Percentage") +
  ylab("Marital Status") +
  theme_minimal()
```



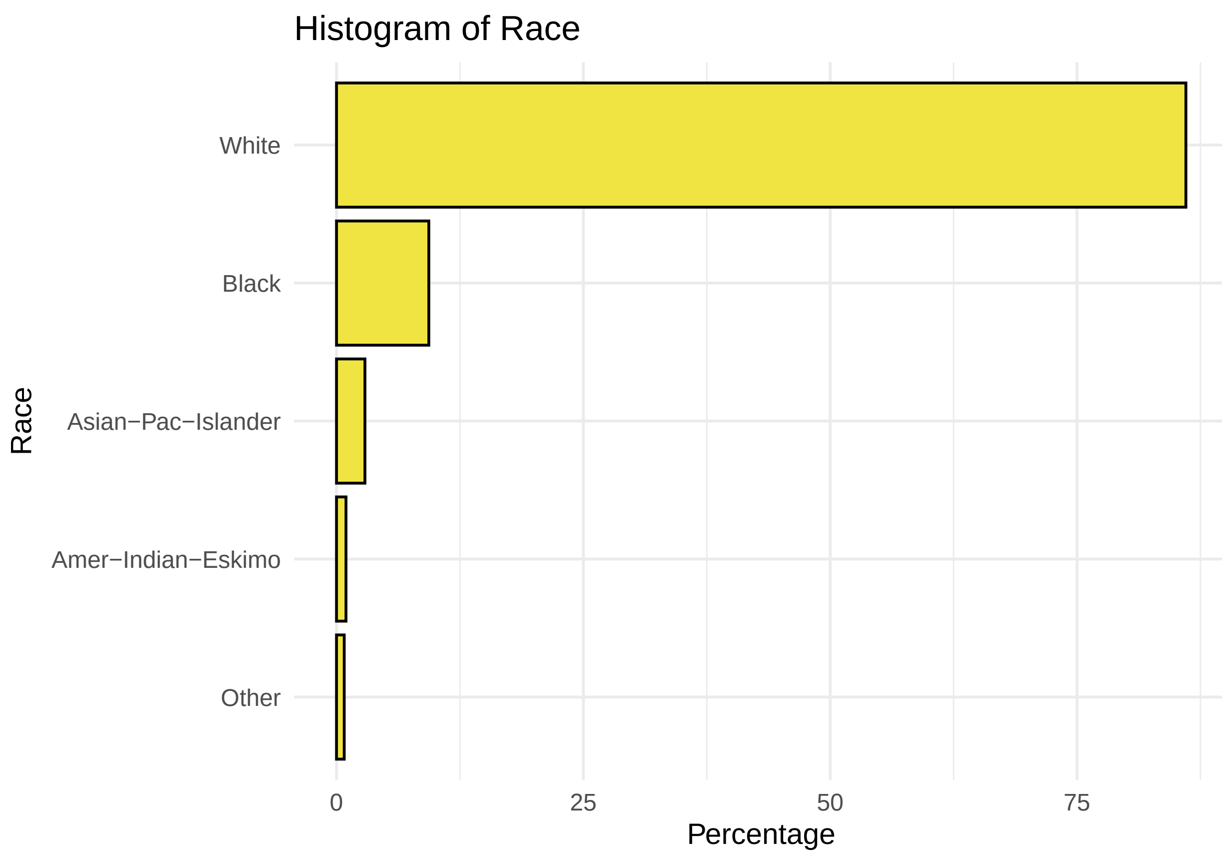

```r
# Histogram of Occupation

ggplot(adult, aes(y = occupation)) +
  ggtitle("Histogram of Occupation") +
  geom_bar(aes(x = 100*(..count..)/sum(..count..)), colour = "black", fill = "#F0E442") +
  scale_y_discrete(limits = levels(occupation)) +
  xlab("Percentage") +
  ylab("Occupation") +
  theme_minimal()
```



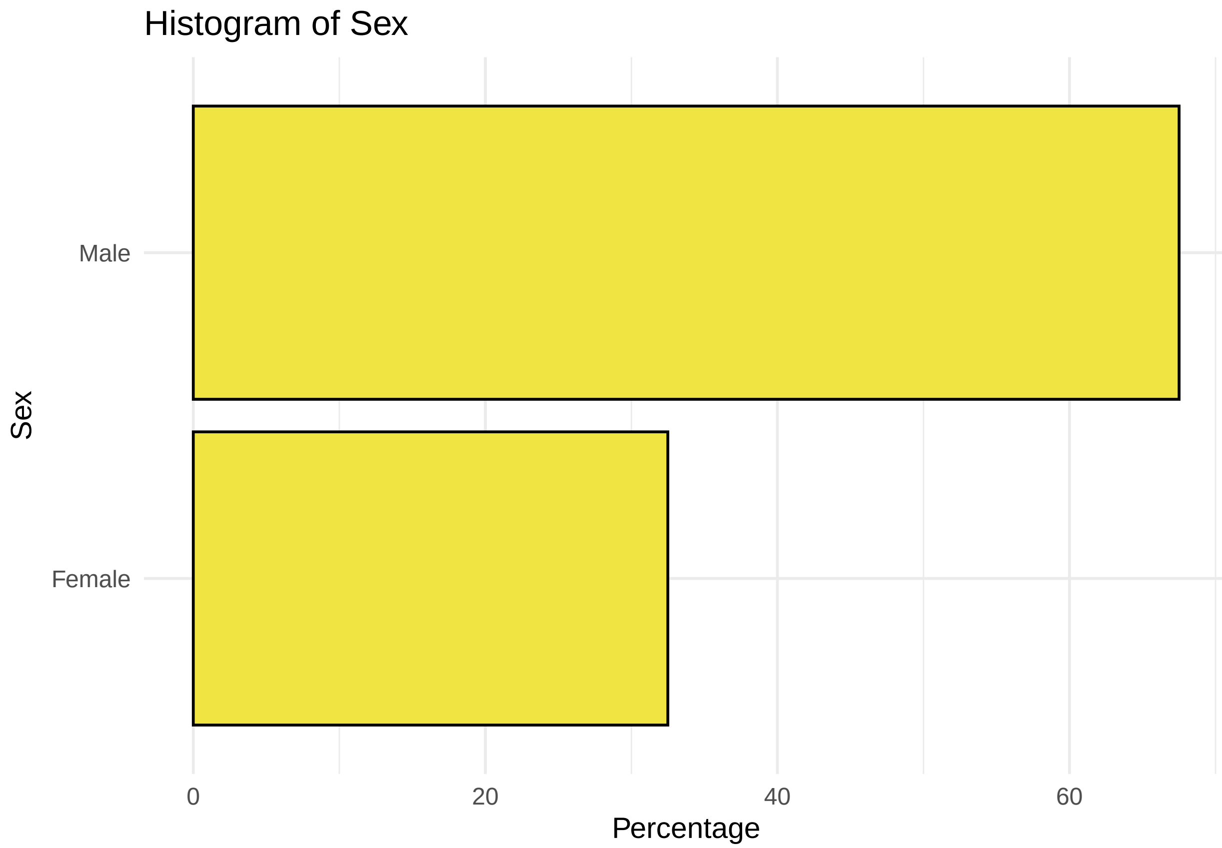

```
# Histogram of Relationship

ggplot(adult, aes(y = relationship)) +
  ggtitle("Histogram of Relationship") +
  geom_bar(aes(x = 100*(..count..)/sum(..count..)), colour = "black", fill = "#F0E442") +
  scale_y_discrete(limits = levels(relationship)) +
  xlab("Percentage") +
  ylab("Relationship") +
  theme_minimal()
```



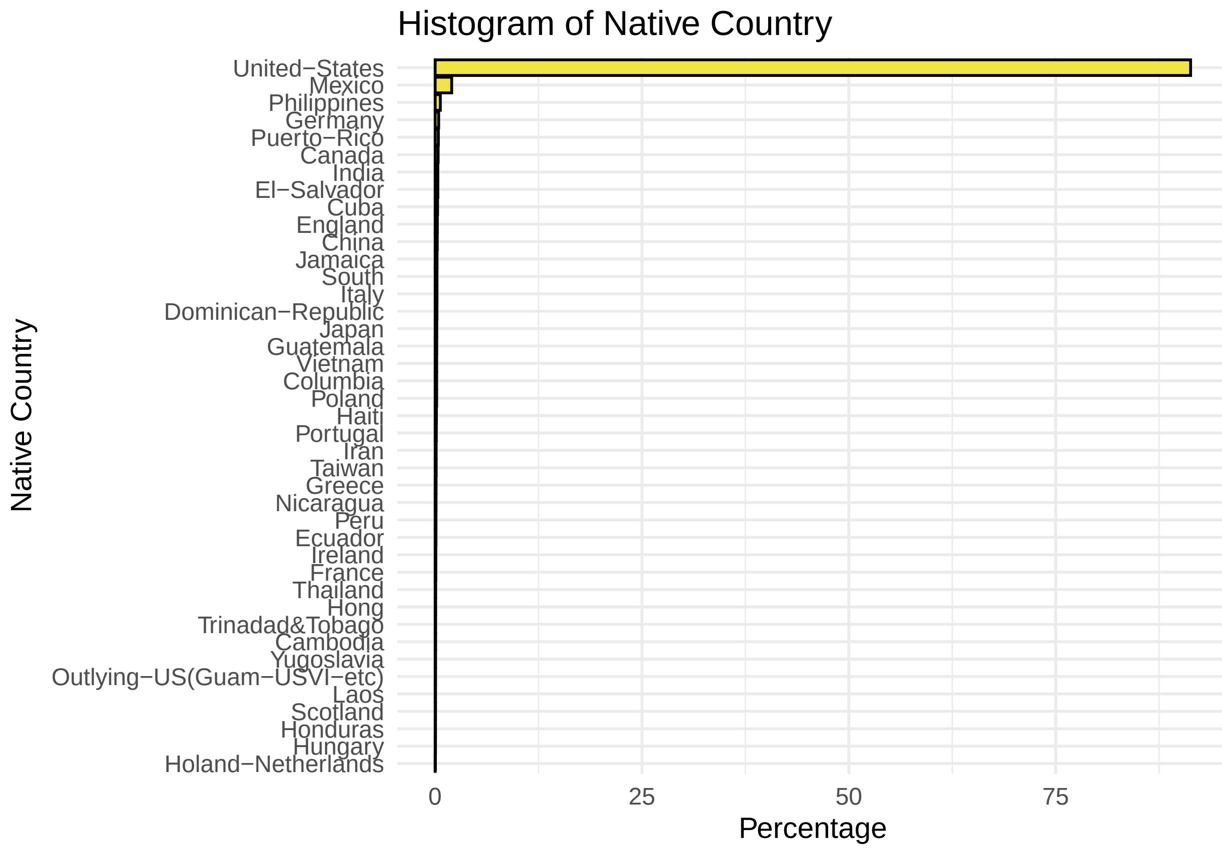

```
# Histogram of Race

ggplot(adult, aes(y = race)) +
  ggtitle("Histogram of Race") +
  geom_bar(aes(x = 100*(..count..)/sum(..count..)), colour = "black", fill = "#F0E442") +
  scale_y_discrete(limits = levels(race)) +
  xlab("Percentage") +
  ylab("Race") +
  theme_minimal()
```




```r
# Histogram of Sex

ggplot(adult, aes(y = sex)) +
  ggtitle("Histogram of Sex") +
  geom_bar(aes(x = 100*(..count..)/sum(..count..)), colour = "black", fill = "#F0E442") +
  scale_y_discrete(limits = levels(sex)) +
  xlab("Percentage") +
  ylab("Sex") +
  theme_minimal()
```




```
# Histogram of Native Country

ggplot(adult, aes(y = native.country)) +
  ggtitle("Histogram of Native Country") +
  geom_bar(aes(x = 100*(..count..)/sum(..count..)), colour = "black", fill = "#F0E442") +
  scale_y_discrete(limits = levels(native.country)) +
  xlab("Percentage") +
  ylab("Native Country") +
  theme_minimal()
```



The variable `native.country` shows a narrow distribution with 91.30954% of the respondents coming from the United States.

```r
summary(adult$native.country)
```

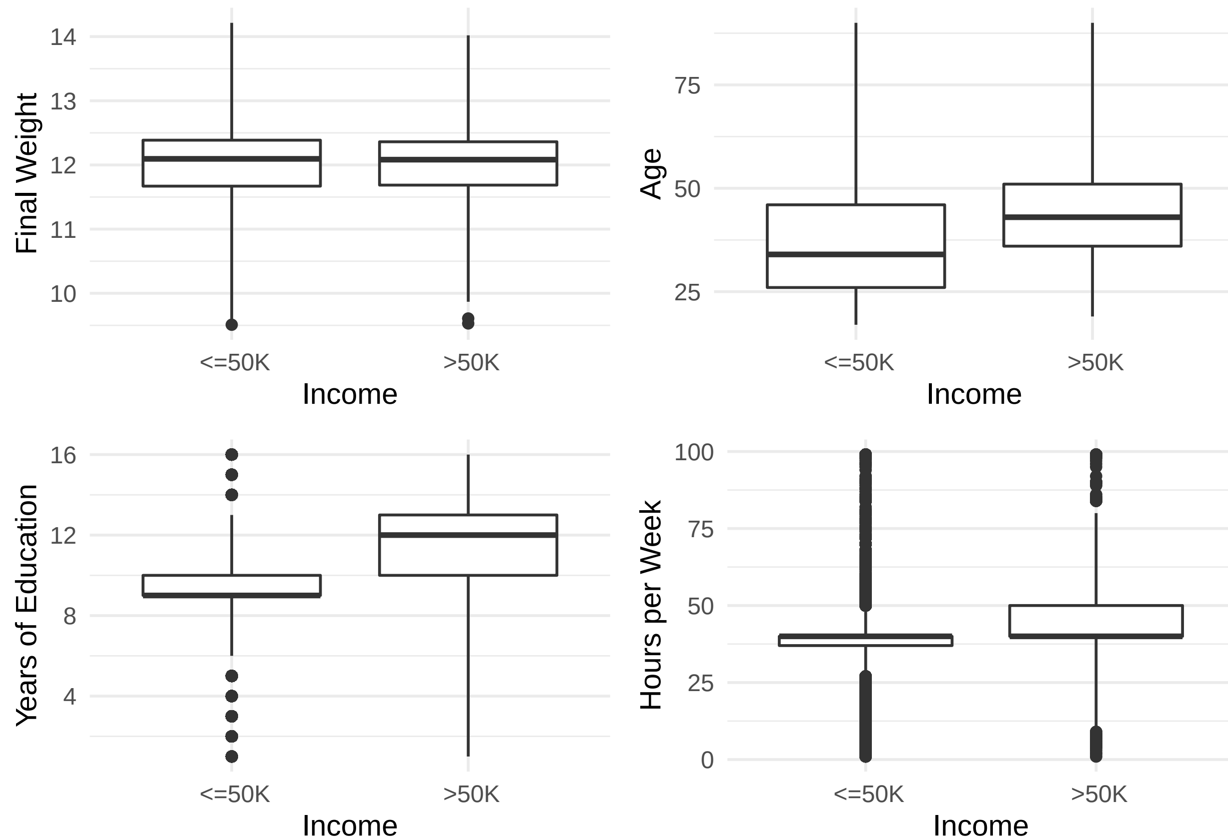
```
##         Holand-Netherlands                    Hungary
##                          1                         18
##                   Honduras                   Scotland
##                         19                         20
##                       Laos Outlying-US(Guam-USVI-etc)
##                         21                         22
##                 Yugoslavia                   Cambodia
##                         23                         26
##            Trinadad&Tobago                       Hong
##                         26                         28
##                   Thailand                     France
##                         29                         36
##                    Ireland                    Ecuador
##                         36                         43
##                       Peru                  Nicaragua
##                         45                         48
##                     Greece                     Taiwan
##                         49                         55
##                       Iran                   Portugal
##                         56                         62
##                      Haiti                     Poland
```

```
##                         69                            81
##                   Columbia                       Vietnam
##                         82                            83
##                  Guatemala                         Japan
##                         86                            89
##         Dominican-Republic                         Italy
##                         97                           100
##                      South                       Jamaica
##                        101                           103
##                      China                       England
##                        113                           119
##                       Cuba                   El-Salvador
##                        133                           147
##                      India                        Canada
##                        147                           163
##                Puerto-Rico                       Germany
##                        175                           193
##                Philippines                        Mexico
##                        283                           903
##              United-States
##                      41292
```

To identify the potential variable that impacts the income level, I plot the attributes against income to examine correlations. The boxplots show that the numeric variables `age`, `education`, and `hours.per.week` are correlated with income.

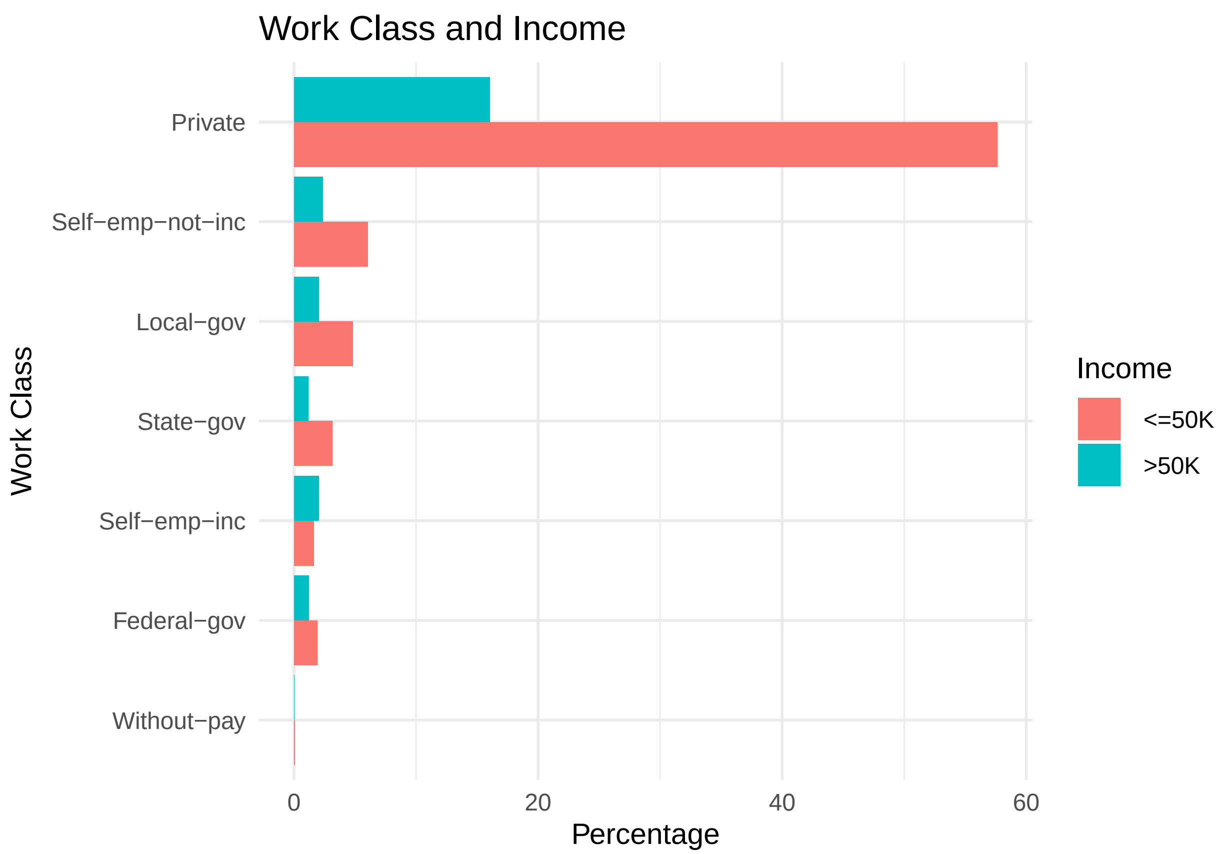
```r
# Final weight and income

b1 <- ggplot(adult, aes(income, log(fnlwgt))) +
  geom_boxplot(coef=3) +
  xlab("Income") +
  ylab("Final Weight") +
  theme_minimal()

# Age and income

b2 <- ggplot(adult, aes(income, age)) +
  geom_boxplot(coef=3) +
  xlab("Income") +
  ylab("Age") +
  theme_minimal()

# Education and income

b3 <- ggplot(adult, aes(income, education.num)) +
  geom_boxplot(coef=3) +
  xlab("Income") +
  ylab("Years of Education") +
  theme_minimal()

# Hours per week and income

b4 <- ggplot(adult, aes(income, hours.per.week)) +
  geom_boxplot(coef=3) +
```

```r
  xlab("Income") +
  ylab("Hours per Week") +
  theme_minimal()

grid.arrange(b1, b2, b3, b4)
```

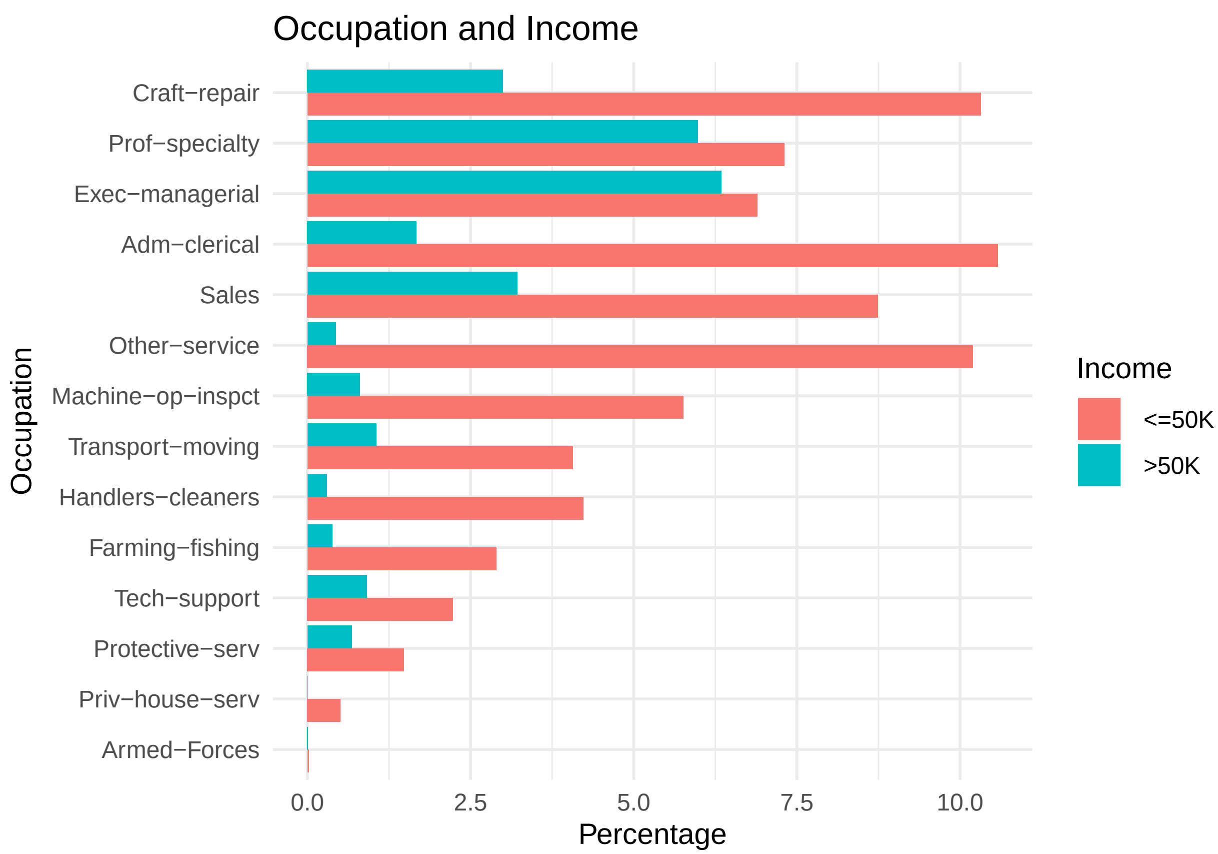


I use bar charts to show the correlation between categorical variables and income.

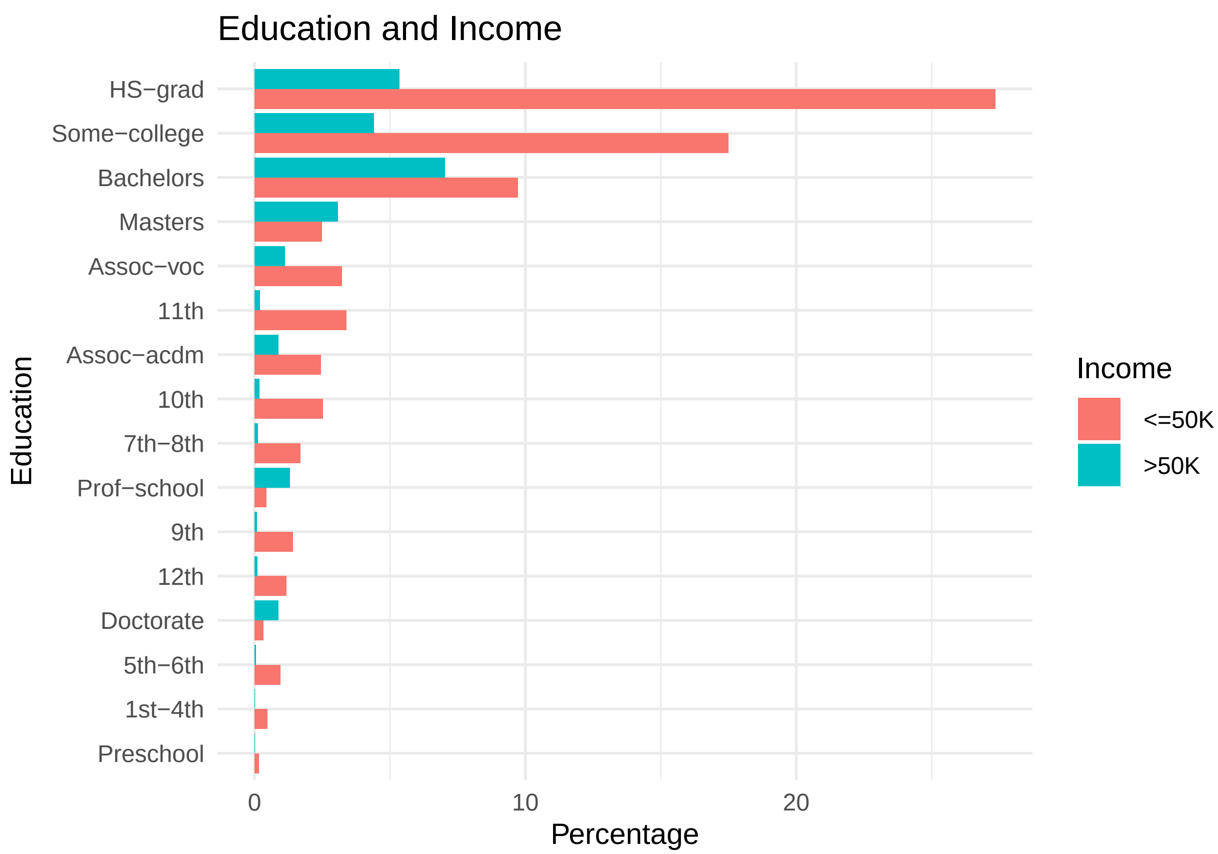
```r
# Work class and income

ggplot(adult, aes(y = workclass, fill = income)) +
  geom_bar(aes(x = 100*(..count..)/sum(..count..)), position=’dodge’) +
  ggtitle("Work Class and Income") +
  labs(fill = "Income") +
  xlab("Percentage") +
  ylab("Work Class") +
  theme_minimal()
```



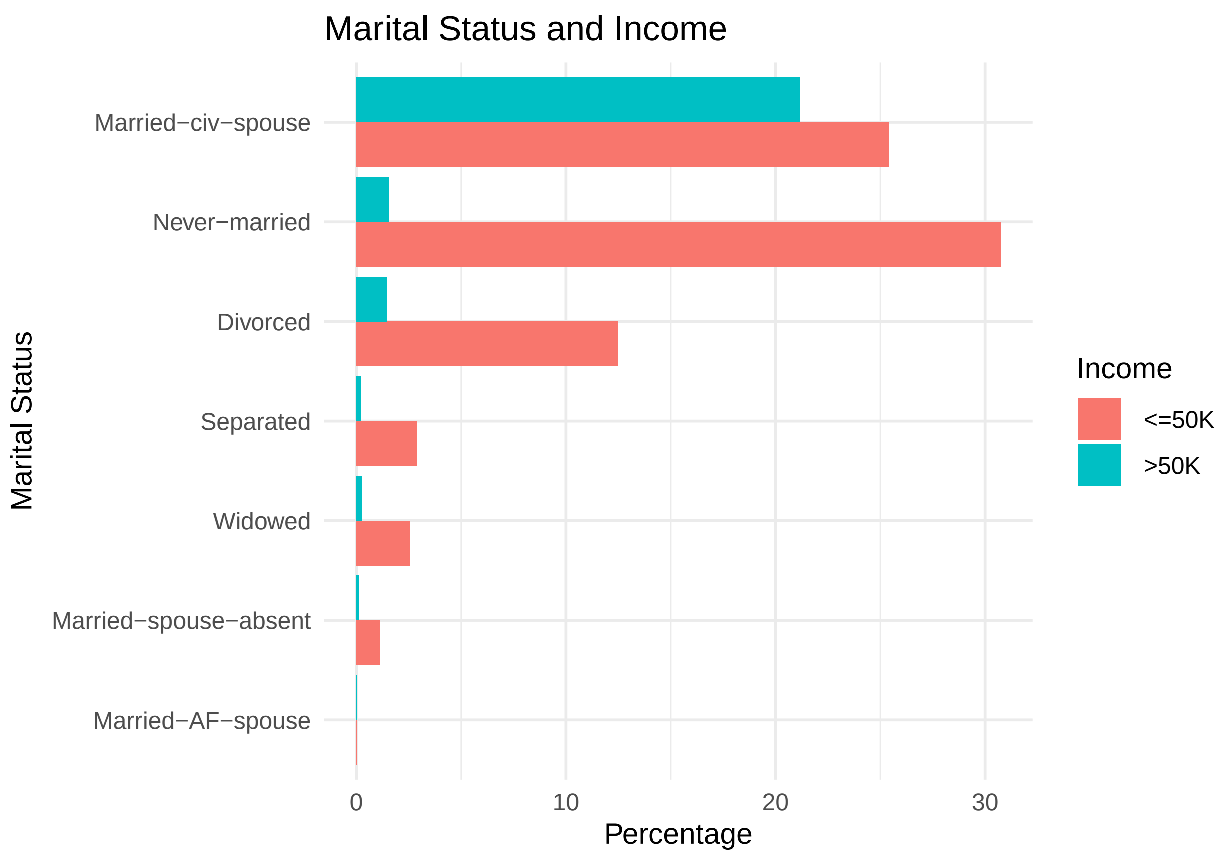

```
# Occupation and income

ggplot(adult, aes(y = occupation, fill = income)) +
  geom_bar(aes(x = 100*(..count..)/sum(..count..)), position='dodge') +
  ggtitle("Occupation and Income") +
  labs(fill = "Income") +
  xlab("Percentage") +
  ylab("Occupation") +
  theme_minimal()
```



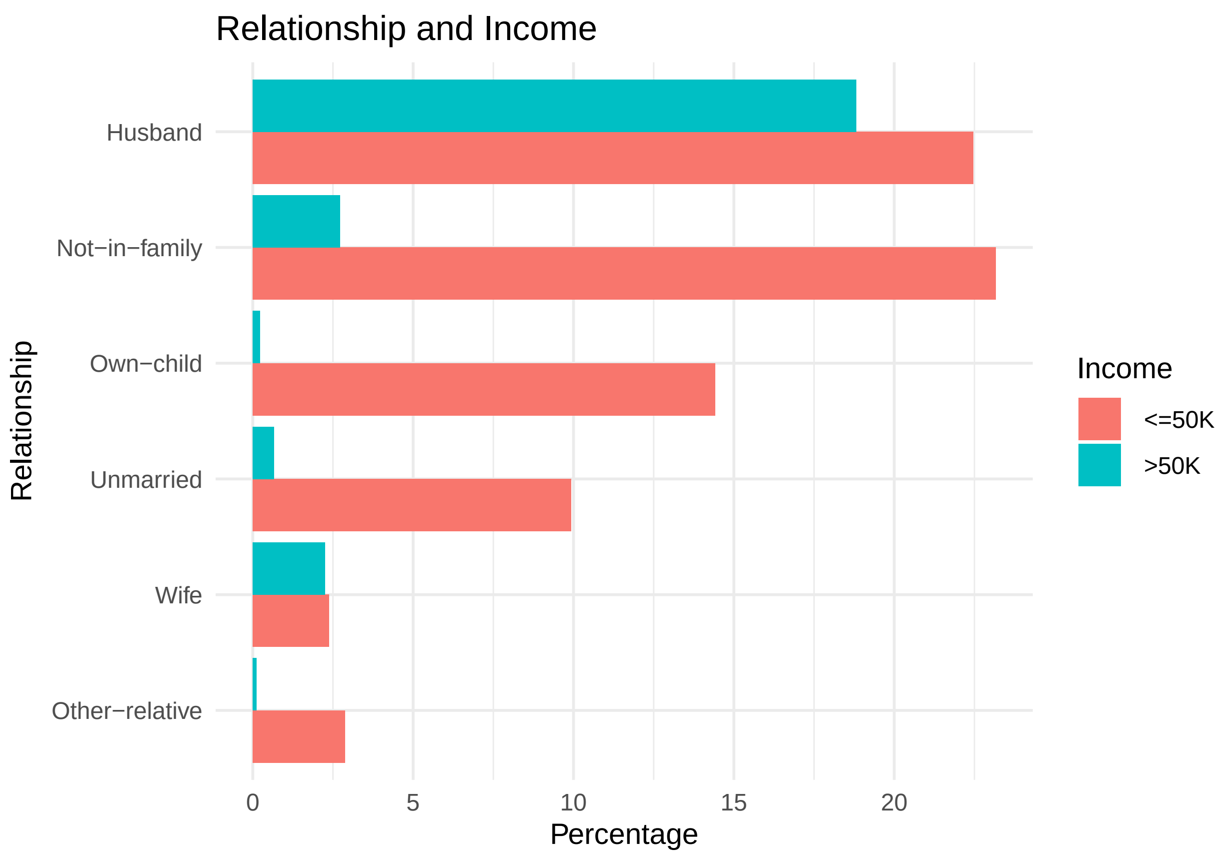

```r
# Education and income

ggplot(adult, aes(y = education, fill = income)) +
  geom_bar(aes(x = 100*(..count..)/sum(..count..)), position='dodge') +
  ggtitle("Education and Income") +
  labs(fill = "Income") +
  xlab("Percentage") +
  ylab("Education") +
  theme_minimal()
```



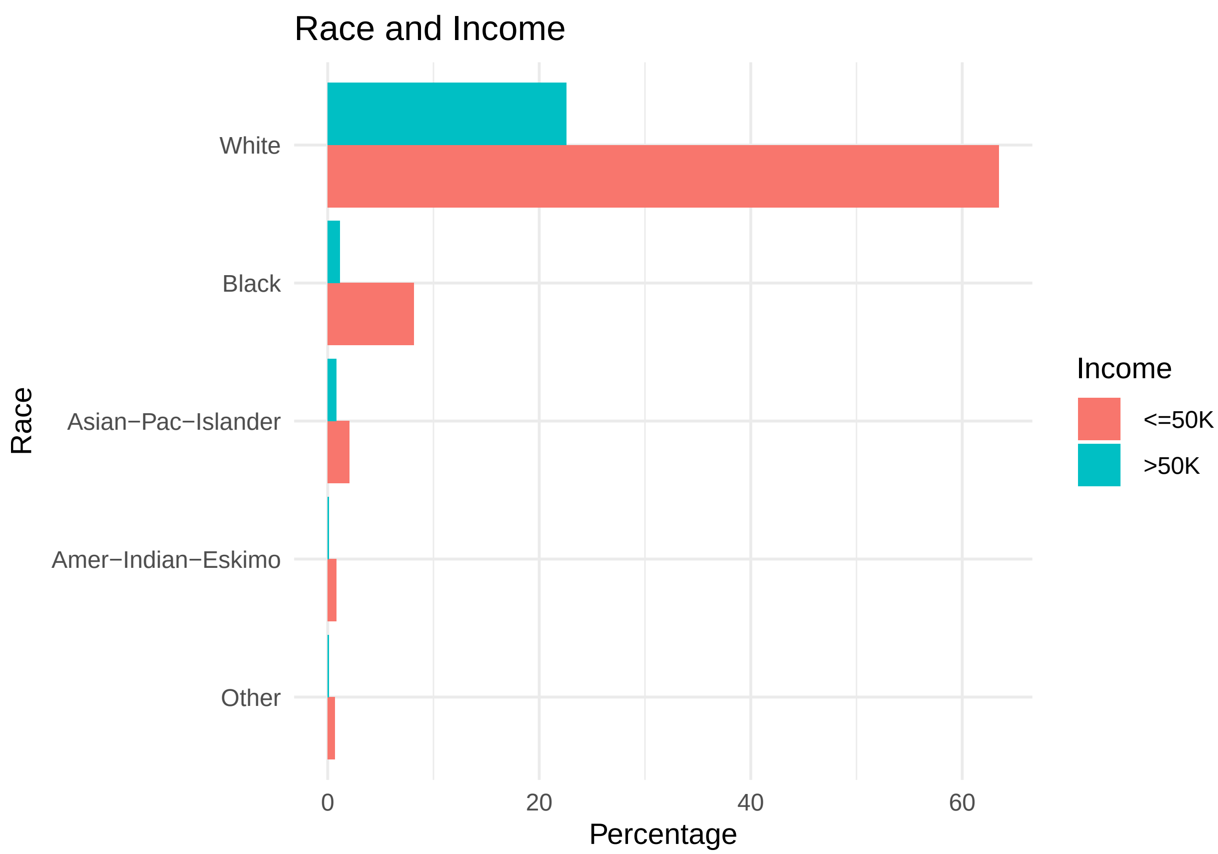

```r
# Marital status and income

ggplot(adult, aes(y = marital.status, fill = income)) +
  geom_bar(aes(x = 100*(..count..)/sum(..count..)), position='dodge') +
  ggtitle("Marital Status and Income") +
  labs(fill = "Income") +
  xlab("Percentage") +
  ylab("Marital Status") +
  theme_minimal()
```



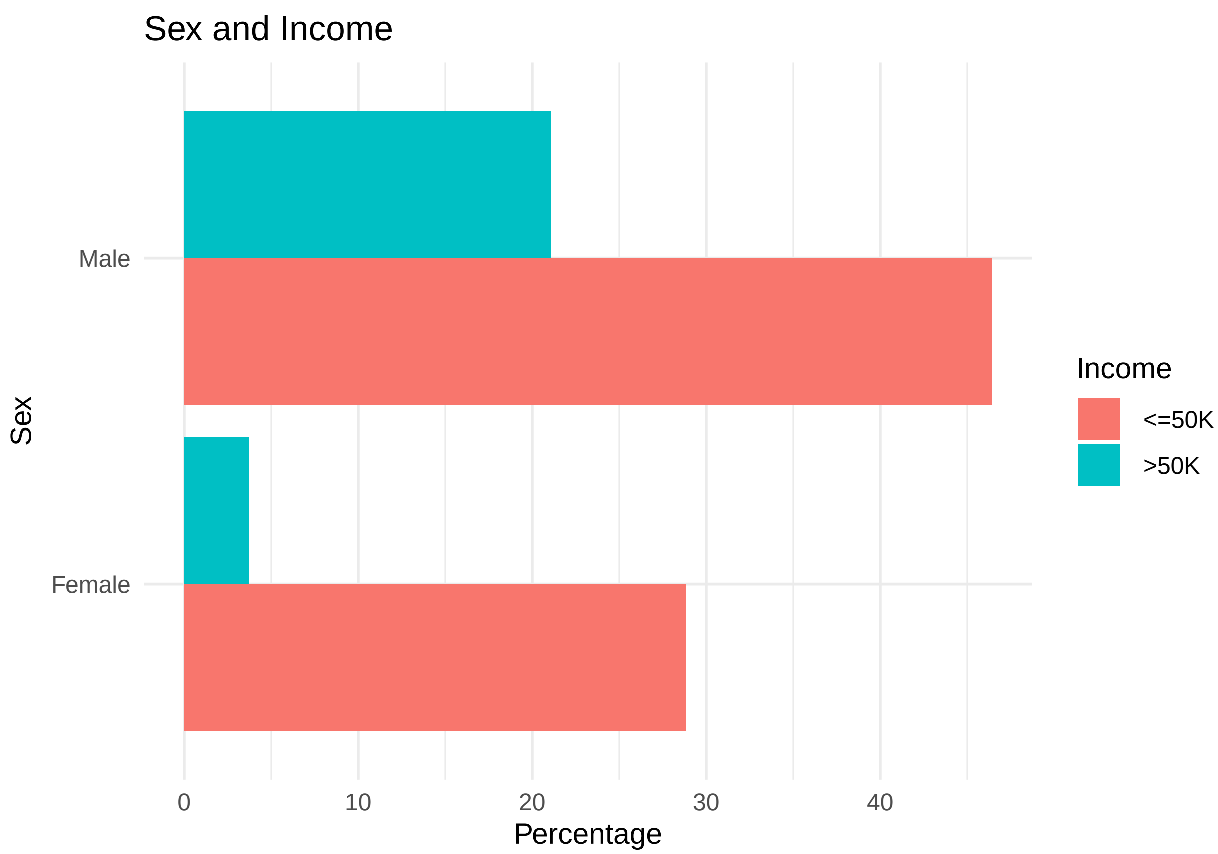

```
# Relationship and income

ggplot(adult, aes(y = relationship, fill = income)) +
  geom_bar(aes(x = 100*(..count..)/sum(..count..)), position='dodge') +
  ggtitle("Relationship and Income") +
  labs(fill = "Income") +
  xlab("Percentage") +
  ylab("Relationship") +
  theme_minimal()
```




```
# Race and income

ggplot(adult, aes(y = race, fill = income)) +
  geom_bar(aes(x = 100*(..count..)/sum(..count..)), position='dodge') +
  ggtitle("Race and Income") +
  labs(fill = "Income") +
  xlab("Percentage") +
  ylab("Race") +
  theme_minimal()
```




```
# Sex and income

ggplot(adult, aes(y = sex, fill = income)) +
  geom_bar(aes(x = 100*(..count..)/sum(..count..)), position='dodge') +
  ggtitle("Sex and Income") +
  labs(fill = "Income") +
  xlab("Percentage") +
  ylab("Sex") +
  theme_minimal()
```



As we see from the above graphs, all categorical variables show an influence on income level.

2.2 Data Cleaning

This section presents the data cleaning process for later modeling.

```
# Combine capital.gain and capital.loss into capital.change

train$capital.change <- train$capital.gain - train$capital.loss
test$capital.change <- test$capital.gain - test$capital.loss

train$capital.gain <- NULL
train$capital.loss <- NULL
test$capital.gain <- NULL
test$capital.loss<-NULL

# Switch income and capital.change columns (let income be the last column)

train[c(11,12)] <- train[c(12,11)]
colnames(train)[11:12] <- colnames(train)[12:11]
test[c(11,12)] <- test[c(12,11)]
colnames(test)[11:12] <- colnames(test)[12:11]


# Delete education variable in train and test set
```

```r
train$education <- NULL
test$education <- NULL

# Delete native.country variable in train and test set

train$native.country <- NULL
test$native.country <- NULL

# Convert income to dummy variable

train$income <- as.factor(ifelse(train$income == ' <=50K', 0, 1))
test$income <- as.factor(ifelse(test$income == ' <=50K.', 0, 1))
```

2.3 Classification Tree

In the classification tree model, we achieve a prediction accuracy of 83.89774%.

```r
set.seed(1, sample.kind = "Rounding")
tree <- rpart(income ~ ., data = train, method = 'class')
tree.hat <- predict(tree, newdata = test, type = 'class')
confusionMatrix(tree.hat, test$income)$overall["Accuracy"]
```

```
##  Accuracy
## 0.8389774
```

2.3 Random Forest

We achieve the OOB error rate of 13.92% and prediction accuracy of 85.73705% with the random forest model. The top five variables that appear to be most important are `capital.change`, `fnlwgt`, `age`, `relationship`, and `education.num`, as shown in the variable importance plot.

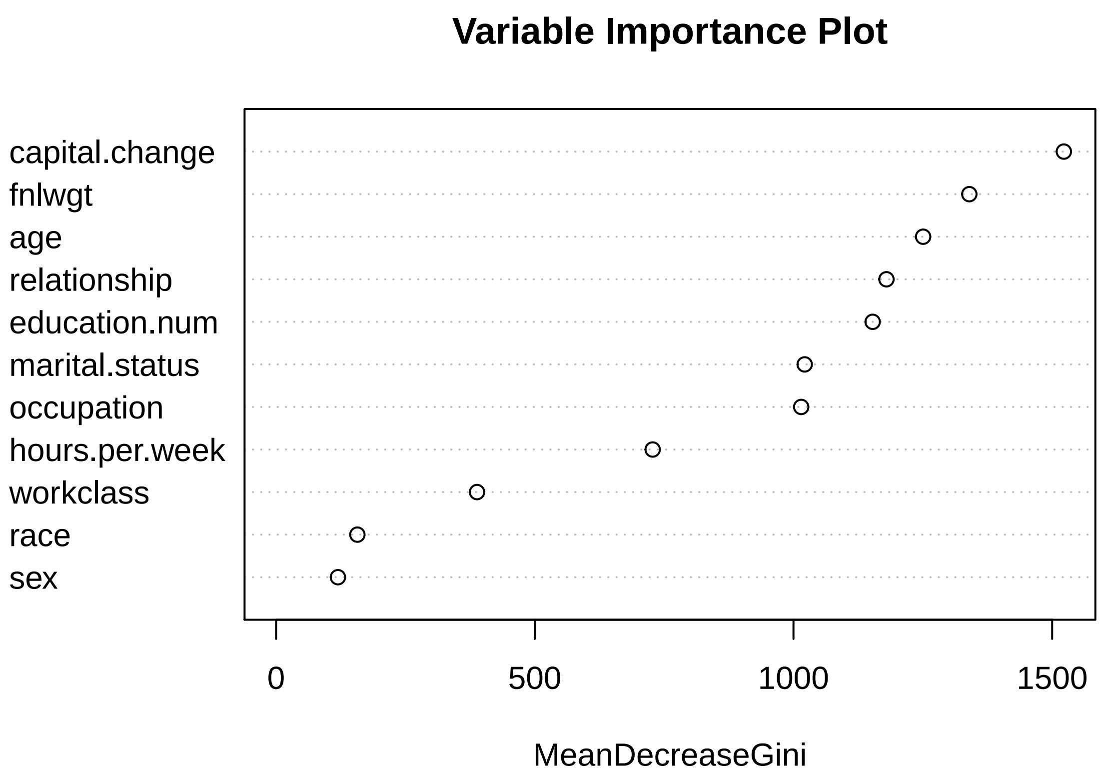
```r
set.seed(1, sample.kind="Rounding")
forest <- randomForest(train$income ~ ., data = train, mtry = sqrt(10), importance = TRUE)
forest
```

```
##
## Call:
##  randomForest(formula = train$income ~ ., data = train, mtry = sqrt(10),      importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 13.92%
## Confusion matrix:
##       0    1 class.error
## 0 21140 1514  0.06683146
## 1  2684 4824  0.35748535
```

```r
forest.hat <- predict(forest, newdata = test, type = "class")
confusionMatrix(forest.hat, test$income)$overall["Accuracy"]
```

```
##  Accuracy
## 0.8573705
```

```r
varImpPlot(forest, type = 2, main = "Variable Importance Plot")
```



III. Conclusion

In this project, I built a classification tree and random forest models to predict a person's income level with the Census Income dataset. The random forest model achieved an accuracy of 85.73705%, higher than the 83.89774% provided by the initial classification tree model. A significant portion of the income variability is generated by the investment gain and loss, the final weight assigned by the census, age, role in the family, and education level. The future work would focus on building other machine learning models such as logistic regression, KNN, Naive Bayes, and neural network to increase the accuracy and predictive power.