Data Science in R - Adult UCI income Prediction

Hervé Wan

3 janvier 2019

1. Introduction

Background and Motivation

We identify problem as classification problem when independent variables are continuous in nature and dependent variable is in categorical form i.e. in classes like positive class and negative class. The real life example of classification example would be, to categorize the mail as spam or not spam, to categorize the tumor as malignant or benign and to categorize the transaction as fraudulent or genuine. All these problem's answers are in categorical form i.e. Yes or No. and that is why they are two class classification problems.

Used DataSet

For this project a dataset merge from UCI repository is used. this dataset includes adult data as education, region, marital status, employment, sex, age,etc.

Goal

The goal is to train a machine learning algorithm using the inputs of a provided subset to predict if they have an income of more than 50K US dollar.

Furthermore visualisations of the data is necessary (using ggplot2) in order to identify factors that could affect their income. We will try different models, where the overral accuracy (if the prediction is good or bad) are calculated to assess the quality of the models. Finally, we apply the best model to the provided validation set and submitt our predictions.

Read in of Data

```
#############################################################
# Getting Data
#############################################################

if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: tidyverse

## -- Attaching packages ----------------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.8
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.3.0     v forcats 0.3.0

## Warning: package 'ggplot2' was built under R version 3.5.1
```

```
## Warning: package 'tidyr' was built under R version 3.5.1

## Warning: package 'readr' was built under R version 3.5.1

## Warning: package 'dplyr' was built under R version 3.5.1

## -- Conflicts -------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
if(!require(caTools)) install.packages("caTools", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: caTools

## Warning: package 'caTools' was built under R version 3.5.1
```

```r
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: caret

## Warning: package 'caret' was built under R version 3.5.1

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```r
if(!require(Amelia)) install.packages("Amelia", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: Amelia

## Warning: package 'Amelia' was built under R version 3.5.1

## Loading required package: Rcpp

## Warning: package 'Rcpp' was built under R version 3.5.1

## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.5, built: 2018-05-07)
## ## Copyright (C) 2005-2019 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```r
#We Read in the adult_sal.csv file and set it to a data frame called adult.

adult <- read.csv('adult_sal.csv')
head(adult)
```

```
##   X age    type_employer fnlwgt education education_num            marital
## 1 1  39        State-gov  77516 Bachelors            13      Never-married
## 2 2  50 Self-emp-not-inc  83311 Bachelors            13 Married-civ-spouse
## 3 3  38          Private 215646   HS-grad             9           Divorced
## 4 4  53          Private 234721      11th             7 Married-civ-spouse
## 5 5  28          Private 338409 Bachelors            13 Married-civ-spouse
## 6 6  37          Private 284582   Masters            14 Married-civ-spouse
##          occupation  relationship  race    sex capital_gain capital_loss
## 1      Adm-clerical Not-in-family White   Male         2174            0
## 2   Exec-managerial       Husband White   Male            0            0
## 3 Handlers-cleaners Not-in-family White   Male            0            0
## 4 Handlers-cleaners       Husband Black   Male            0            0
## 5    Prof-specialty          Wife Black Female            0            0
## 6   Exec-managerial          Wife White Female            0            0
##   hr_per_week       country income
## 1          40 United-States  <=50K
## 2          13 United-States  <=50K
## 3          40 United-States  <=50K
## 4          40 United-States  <=50K
## 5          40          Cuba  <=50K
## 6          40 United-States  <=50K
```

2. Method/Analysis

Data Prepatation/Cleaning

```r
head(adult)
```

```
##   X age    type_employer fnlwgt education education_num            marital
## 1 1  39        State-gov  77516 Bachelors            13      Never-married
## 2 2  50 Self-emp-not-inc  83311 Bachelors            13 Married-civ-spouse
## 3 3  38          Private 215646   HS-grad             9           Divorced
## 4 4  53          Private 234721      11th             7 Married-civ-spouse
## 5 5  28          Private 338409 Bachelors            13 Married-civ-spouse
## 6 6  37          Private 284582   Masters            14 Married-civ-spouse
##          occupation  relationship  race    sex capital_gain capital_loss
## 1      Adm-clerical Not-in-family White   Male         2174            0
## 2   Exec-managerial       Husband White   Male            0            0
## 3 Handlers-cleaners Not-in-family White   Male            0            0
## 4 Handlers-cleaners       Husband Black   Male            0            0
## 5    Prof-specialty          Wife Black Female            0            0
## 6   Exec-managerial          Wife White Female            0            0
##   hr_per_week       country income
## 1          40 United-States  <=50K
## 2          13 United-States  <=50K
```

```
## 3          40 United-States  <=50K
## 4          40 United-States  <=50K
## 5          40          Cuba  <=50K
## 6          40 United-States  <=50K
```

```
#We can notice that the index has been repeated, we drop this column
head(adult)
```

```
##   X age     type_employer fnlwgt education education_num            marital
## 1 1  39         State-gov  77516 Bachelors            13      Never-married
## 2 2  50 Self-emp-not-inc  83311 Bachelors            13 Married-civ-spouse
## 3 3  38           Private 215646   HS-grad             9           Divorced
## 4 4  53           Private 234721      11th             7 Married-civ-spouse
## 5 5  28           Private 338409 Bachelors            13 Married-civ-spouse
## 6 6  37           Private 284582   Masters            14 Married-civ-spouse
##          occupation  relationship  race    sex capital_gain capital_loss
## 1      Adm-clerical Not-in-family White   Male         2174            0
## 2   Exec-managerial       Husband White   Male            0            0
## 3 Handlers-cleaners Not-in-family White   Male            0            0
## 4 Handlers-cleaners       Husband Black   Male            0            0
## 5    Prof-specialty          Wife Black Female            0            0
## 6   Exec-managerial          Wife White Female            0            0
##   hr_per_week       country income
## 1          40 United-States  <=50K
## 2          13 United-States  <=50K
## 3          40 United-States  <=50K
## 4          40 United-States  <=50K
## 5          40          Cuba  <=50K
## 6          40 United-States  <=50K
```

```
##############################################################
# Data Preparation
##############################################################


#####Grouping
#We can see a lot of colmuns that are categorical factors,
#and lot of them have too many factors than necessary.


#Type_employer
table(adult$type_employer)
```

```
##
##                ?      Federal-gov        Local-gov     Never-worked
##             1836              960             2093                7
##          Private     Self-emp-inc Self-emp-not-inc        State-gov
##            22696             1116             2541             1298
##      Without-pay
##               14
```

```r
#We see 1836 with a question mark.
# As well a s 2 small group of never-worked and without-pay.
#We will create a group "unemployed" and put them there.
# Same with local government job and State job. As well as Self-employed jobs
group_type <- function(job){
  job <- as.character(job)
  if (job=='Local-gov' | job=='State-gov'){
    return('SL-gov')
  }else if (job=='Self-emp-inc' | job=='Self-emp-not-inc'){
    return('self-emp')
  }else if(job=='Never-worked' | job=='Without-pay'){
    return('Unemployed')
  }else{
    return(job)
  }
}
adult$type_employer <- sapply(adult$type_employer,group_type)
table(adult$type_employer)
```

```
##
##           ? Federal-gov     Private    self-emp      SL-gov  Unemployed
##        1836         960       22696        3657        3391          21
```

```r
#Marital
table(adult$marital)
```

```
##
##              Divorced     Married-AF-spouse    Married-civ-spouse
##                  4443                    23                 14976
## Married-spouse-absent         Never-married             Separated
##                   418                 10683                  1025
##               Widowed
##                   993
```

```r
#We will regroup in 3 group: Married, Not Married and Never married

group_marital <- function(mar){
  mar <- as.character(mar)

  # Not-Married
  if (mar=='Separated' | mar=='Divorced' | mar=='Widowed'){
    return('Not-Married')

    # Never-Married
  }else if(mar=='Never-married'){
    return(mar)

    #Married
  }else{
    return('Married')
  }
}
```

```r
adult$marital <- sapply(adult$marital,group_marital)
table(adult$marital)
```

```
##
##       Married Never-married   Not-Married
##         15417         10683          6461
```

```r
levels(adult$country)
```

```
##  [1] "?"                          "Cambodia"
##  [3] "Canada"                     "China"
##  [5] "Columbia"                   "Cuba"
##  [7] "Dominican-Republic"         "Ecuador"
##  [9] "El-Salvador"                "England"
## [11] "France"                     "Germany"
## [13] "Greece"                     "Guatemala"
## [15] "Haiti"                      "Holand-Netherlands"
## [17] "Honduras"                   "Hong"
## [19] "Hungary"                    "India"
## [21] "Iran"                       "Ireland"
## [23] "Italy"                      "Jamaica"
## [25] "Japan"                      "Laos"
## [27] "Mexico"                     "Nicaragua"
## [29] "Outlying-US(Guam-USVI-etc)" "Peru"
## [31] "Philippines"                "Poland"
## [33] "Portugal"                   "Puerto-Rico"
## [35] "Scotland"                   "South"
## [37] "Taiwan"                     "Thailand"
## [39] "Trinadad&Tobago"            "United-States"
## [41] "Vietnam"                    "Yugoslavia"
```

```r
#Grouping countries by continent
#Creating continents as strings vectos
Asia <- c('China','Hong','India','Iran','Cambodia','Japan', 'Laos' ,
          'Philippines' ,'Vietnam' ,'Taiwan', 'Thailand')

North.America <- c('Canada','United-States','Puerto-Rico' )

Europe <- c('England' ,'France', 'Germany' ,'Greece','Holand-Netherlands','Hungary',
            'Ireland','Italy','Poland','Portugal','Scotland','Yugoslavia')

Latin.and.South.America <- c('Columbia','Cuba','Dominican-Republic','Ecuador',
                             'El-Salvador','Guatemala','Haiti','Honduras',
                             'Mexico','Nicaragua','Outlying-US(Guam-USVI-etc)','Peru',
                             'Jamaica','Trinadad&Tobago')
Other <- c('South')

group_country <- function(ctry){
  if (ctry %in% Asia){
    return('Asia')
  }else if (ctry %in% North.America){
    return('North.America')
```

```
  }else if (ctry %in% Europe){
    return('Europe')
  }else if (ctry %in% Latin.and.South.America){
    return('Latin.and.South.America')
  }else{
    return('Other')
  }
}

adult$country <- sapply(adult$country,group_country)

#Change country column to region
names(adult)[names(adult)=="country"] <- "region"
#Checking the table
table(adult$region)
```

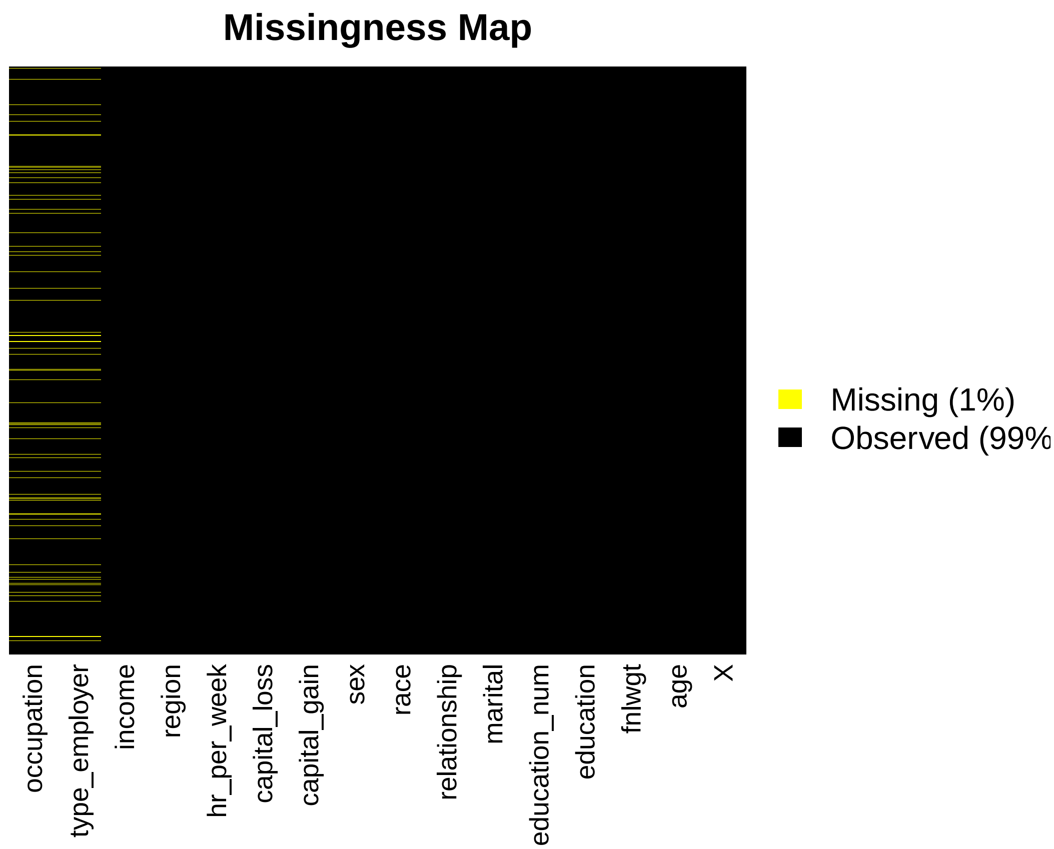
```
##
##                    Asia                  Europe Latin.and.South.America
##                     671                     521                    1301
##           North.America                   Other
##                   29405                     663
```

```
#Checking if categorical columns have factor levels and change if necessary
str(adult)
```

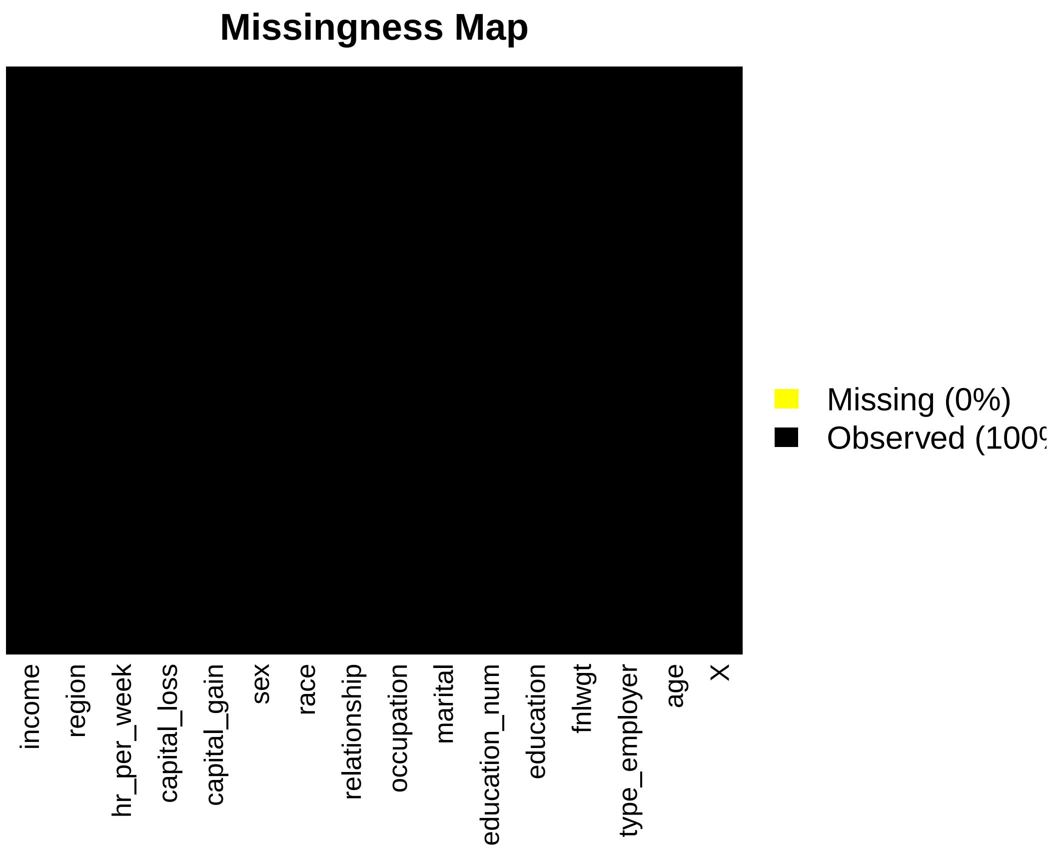
```
## 'data.frame':    32561 obs. of  16 variables:
##  $ X            : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ age          : int  39 50 38 53 28 37 49 52 31 42 ...
##  $ type_employer: chr  "SL-gov" "self-emp" "Private" "Private" ...
##  $ fnlwgt       : int  77516 83311 215646 234721 338409 284582 160187 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      : chr  "Never-married" "Married" "Not-Married" "Married" ...
##  $ occupation   : Factor w/ 15 levels "?","Adm-clerical",..: 2 5 7 7 11 5 9 5 11 5 ...
##  $ 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 ...
##  $ hr_per_week  : int  40 13 40 40 40 40 16 45 50 40 ...
##  $ region       : chr  "North.America" "North.America" "North.America" "North.America" ...
##  $ income       : Factor w/ 2 levels "<=50K",">50K": 1 1 1 1 1 1 1 2 2 2 ...
```

```
adult$type_employer <- sapply(adult$type_employer,factor)
adult$region <- sapply(adult$region,factor)
adult$marital <- sapply(adult$marital,factor)

#############Missing Data

#First any cell with a "?" value will be converted to a NA Value
adult[adult == '?'] <- NA
```

```r
#Plot MissMap
#This is a heatmap pointing out missing values (NA).
#This gives a quick glance at how much data is missing,
#in this case, not a whole lot (relatively speaking)
missmap(adult,y.at=c(1),y.labels = c(''),col=c('yellow','black'))
```

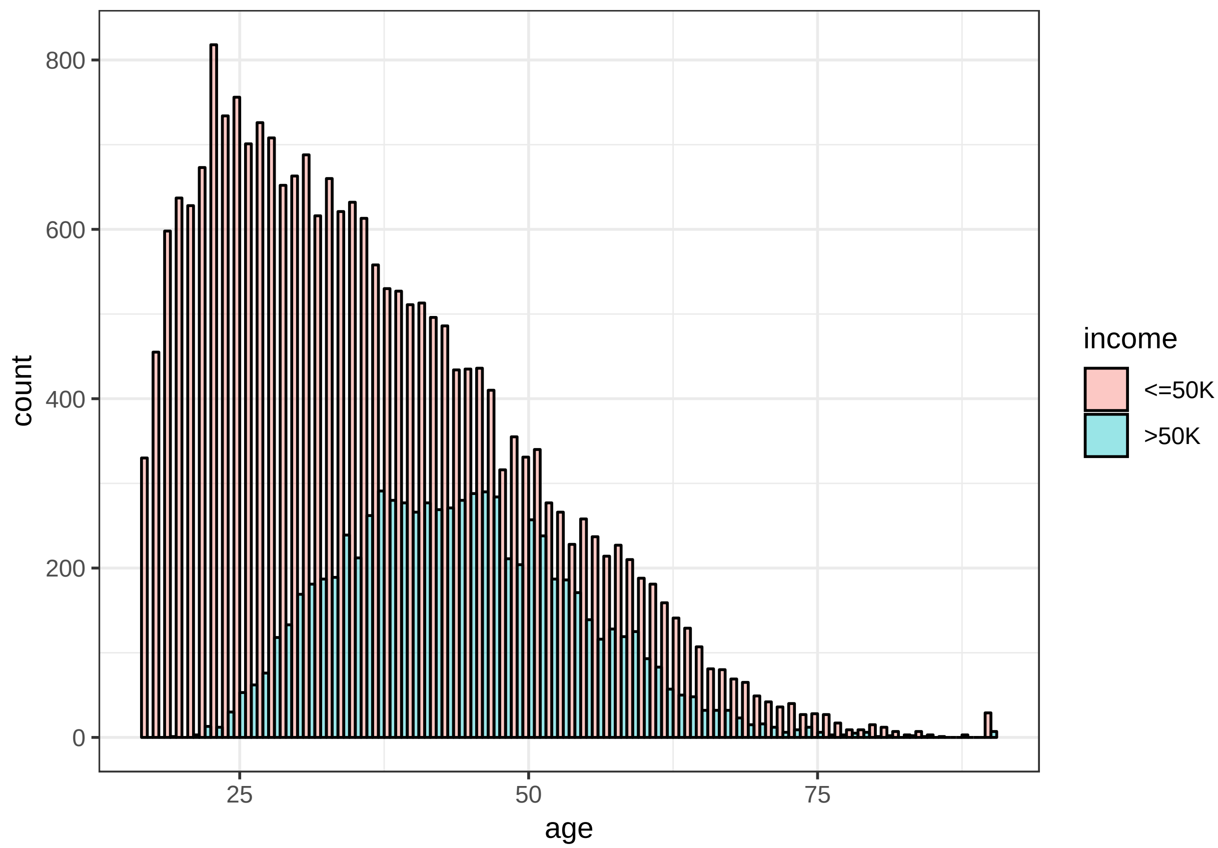



```r
#We will decide to drop them from the data Frame
# May take awhile
adult <- na.omit(adult)

#Check missmap again
missmap(adult,y.at=c(1),y.labels = c(''),col=c('yellow','black'))
```

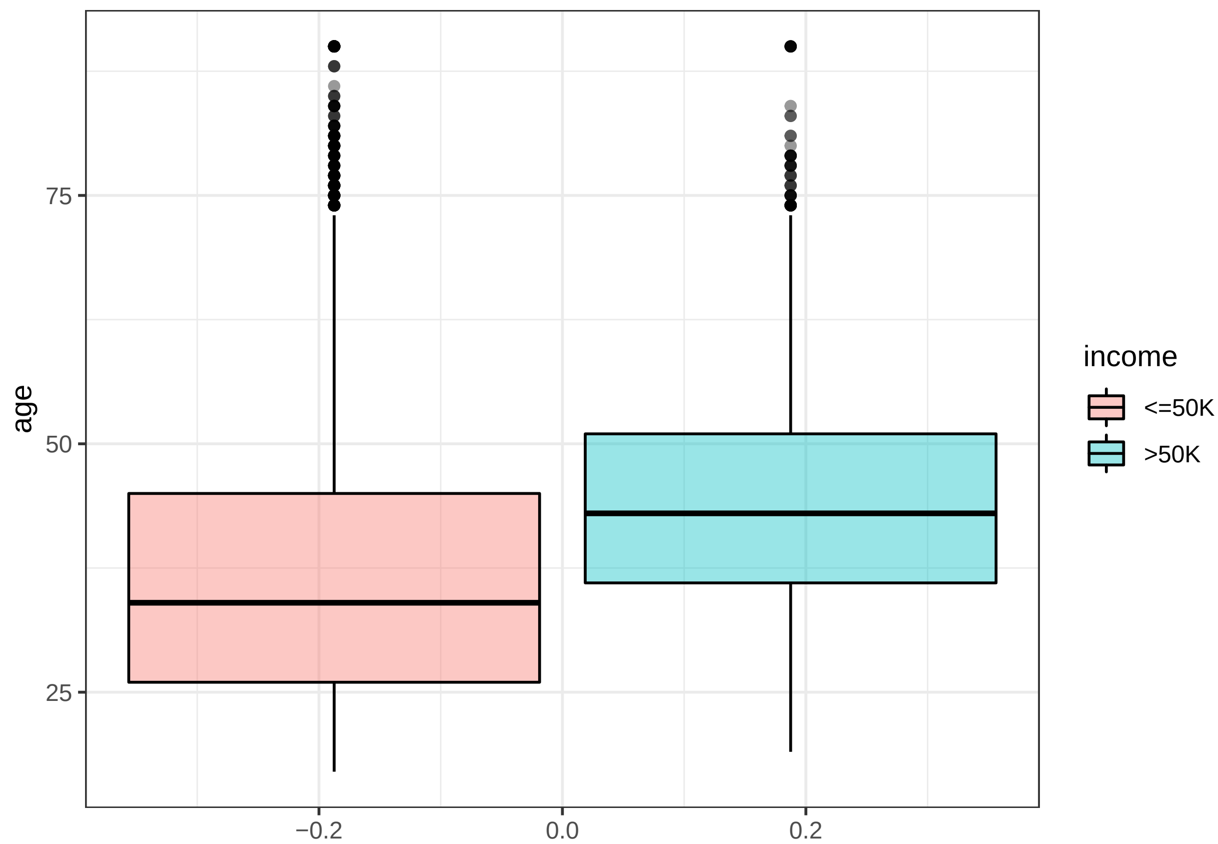


Data Analysis

```
##############################################
#Data Analysis
#############################################

#Effect of Age

#Plot histogram Income by Age
adult %>% ggplot(aes(age)) + geom_histogram(aes(fill=income),color='black',binwidth=1,alpha=0.4, positi
```




```
adult %>% ggplot() + geom_boxplot(aes(y=age,fill=income),color='black',alpha=0.4) + theme_bw()
```



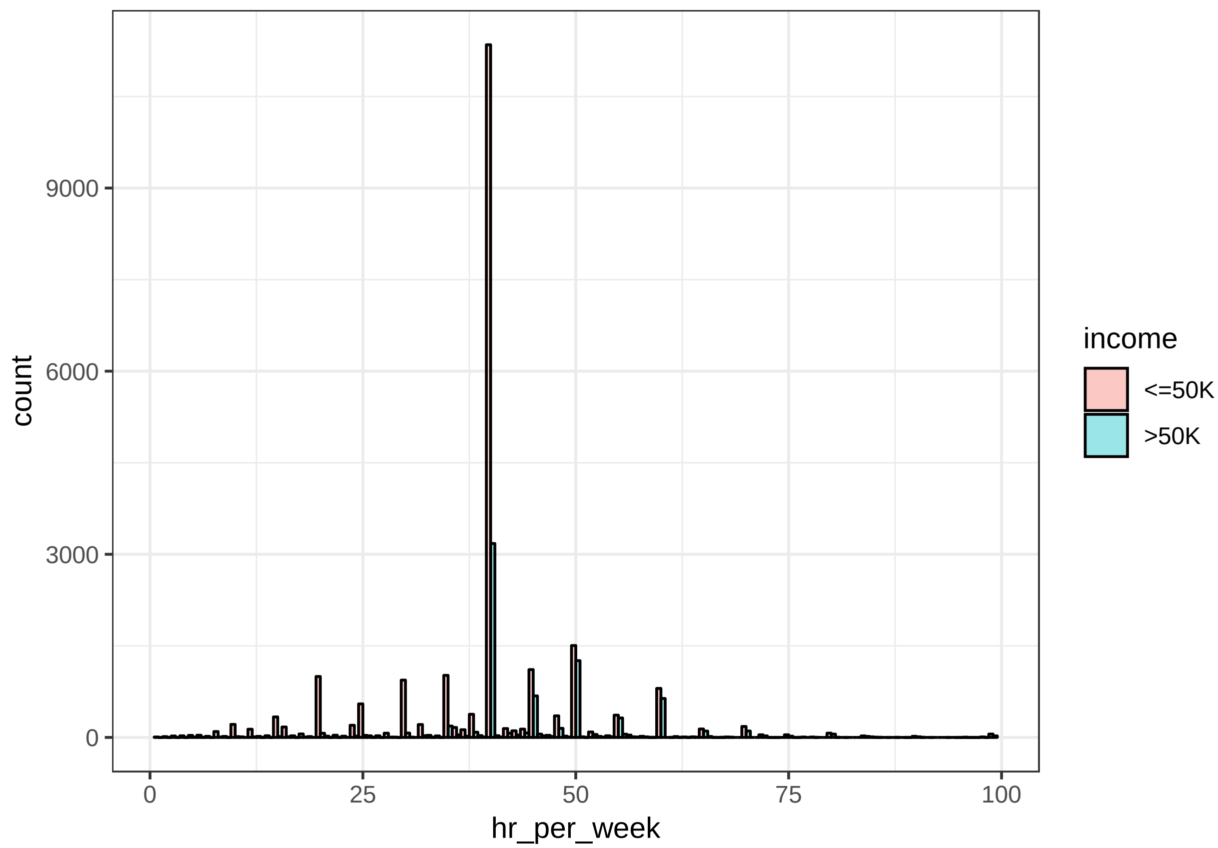

```
mean1 <-adult %>% select(age,income) %>%filter(income=="<=50K")
mean(mean1$age)
```

```
## [1] 36.61219
```

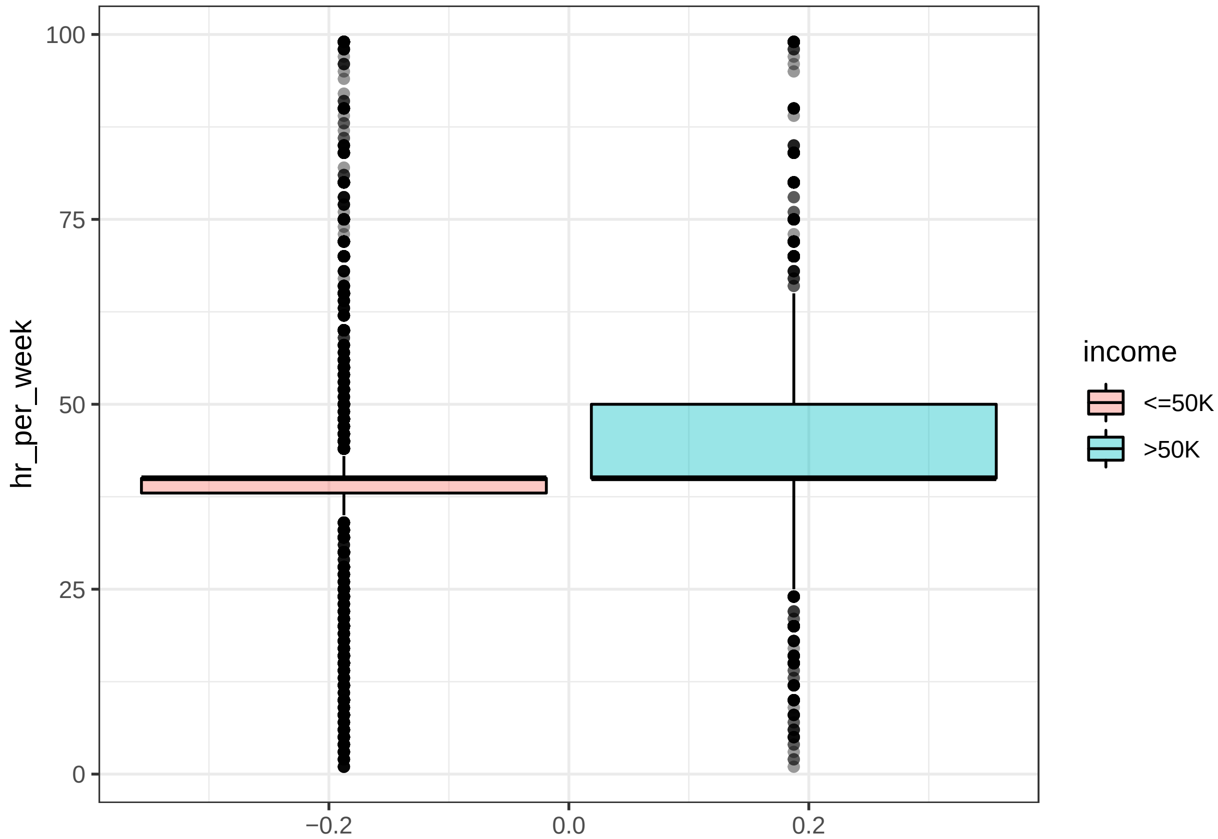
```
mean2 <-adult %>% select(age,income) %>%filter(income==">50K")
mean(mean2$age)
```

```
## [1] 43.96601
```

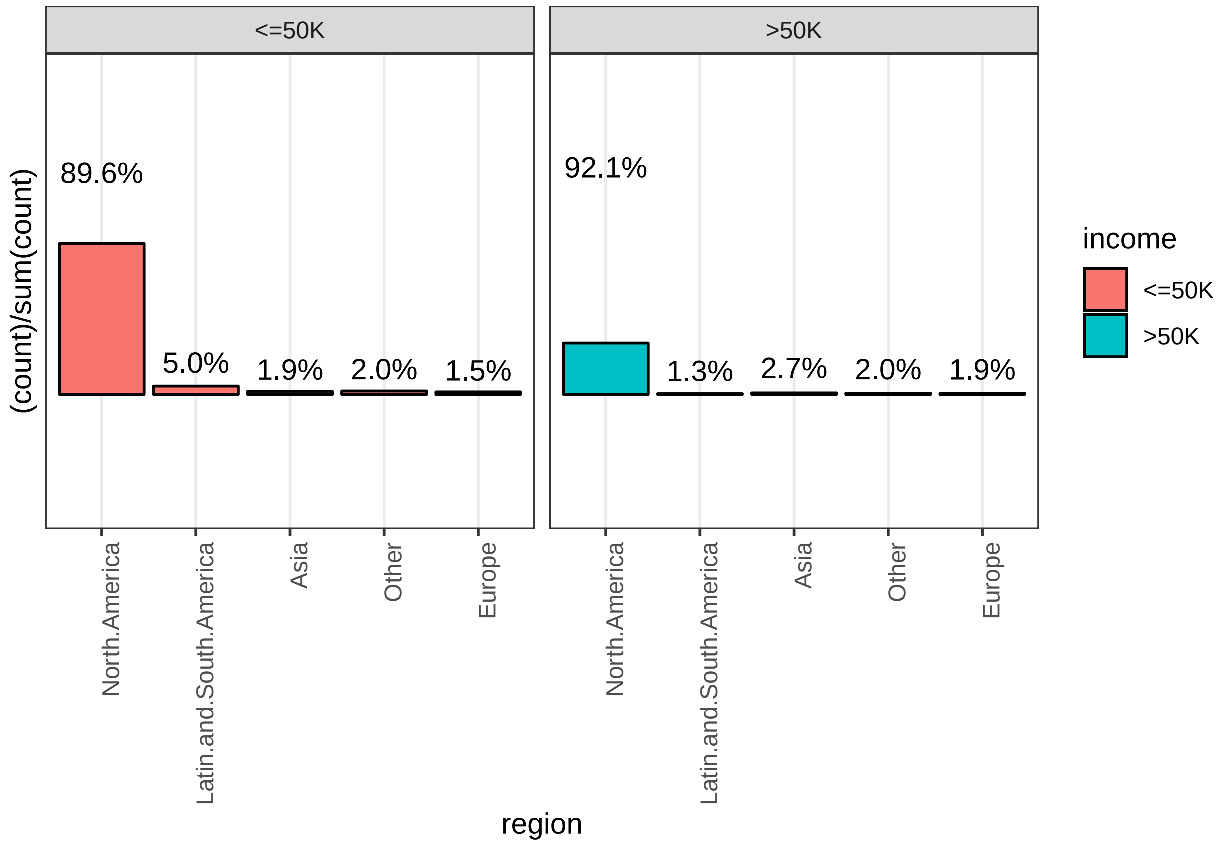
```
#We can see that the age has a big impact on the income through the distribution and box plot
#The average age of people earning more than 50K are 44years old against 36 and a half for those earnin

#Effect of hours worked per week
adult %>% ggplot(aes(hr_per_week)) + geom_histogram(aes(fill=income),color='black',binwidth=1,alpha=0.4
```



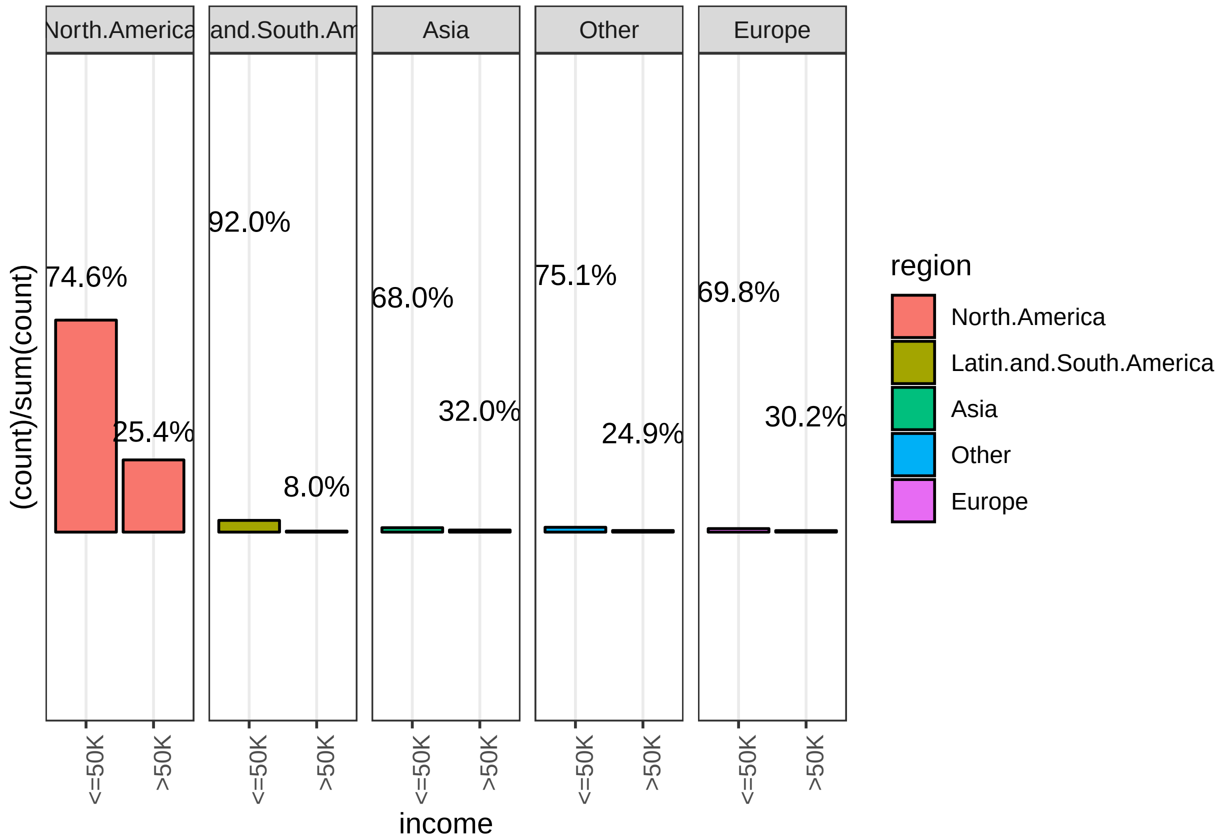

```
adult %>% ggplot() + geom_boxplot(aes(y=hr_per_week,fill=income),color='black',alpha=0.4) + theme_bw()
```



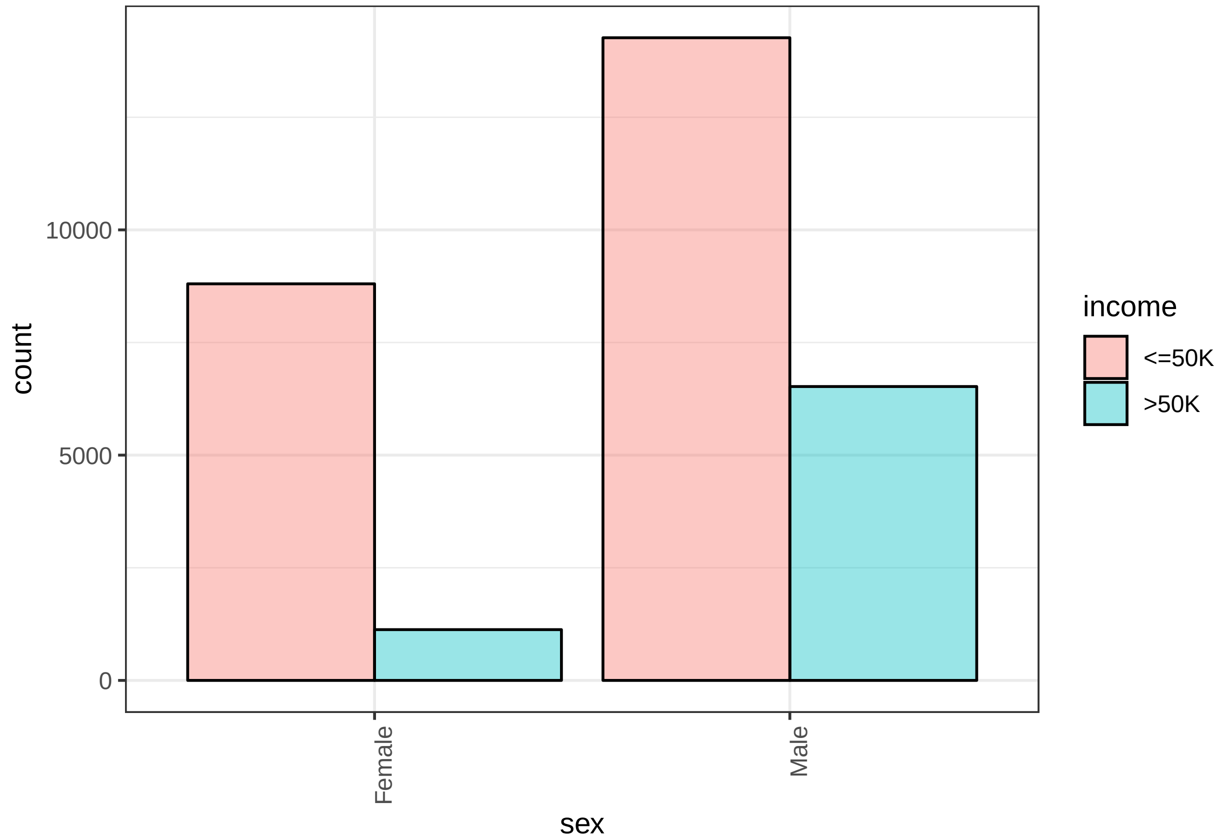

```
#Here, we can see that although the mean is the same, people earning more than 50K tend to work much mo

############Region effect
ggplot(adult,aes(region,group=income)) + geom_bar(aes(y=(..count..)/sum(..count..),fill=income),color='
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +  scale_y_discrete(labels = scales::percent)
```



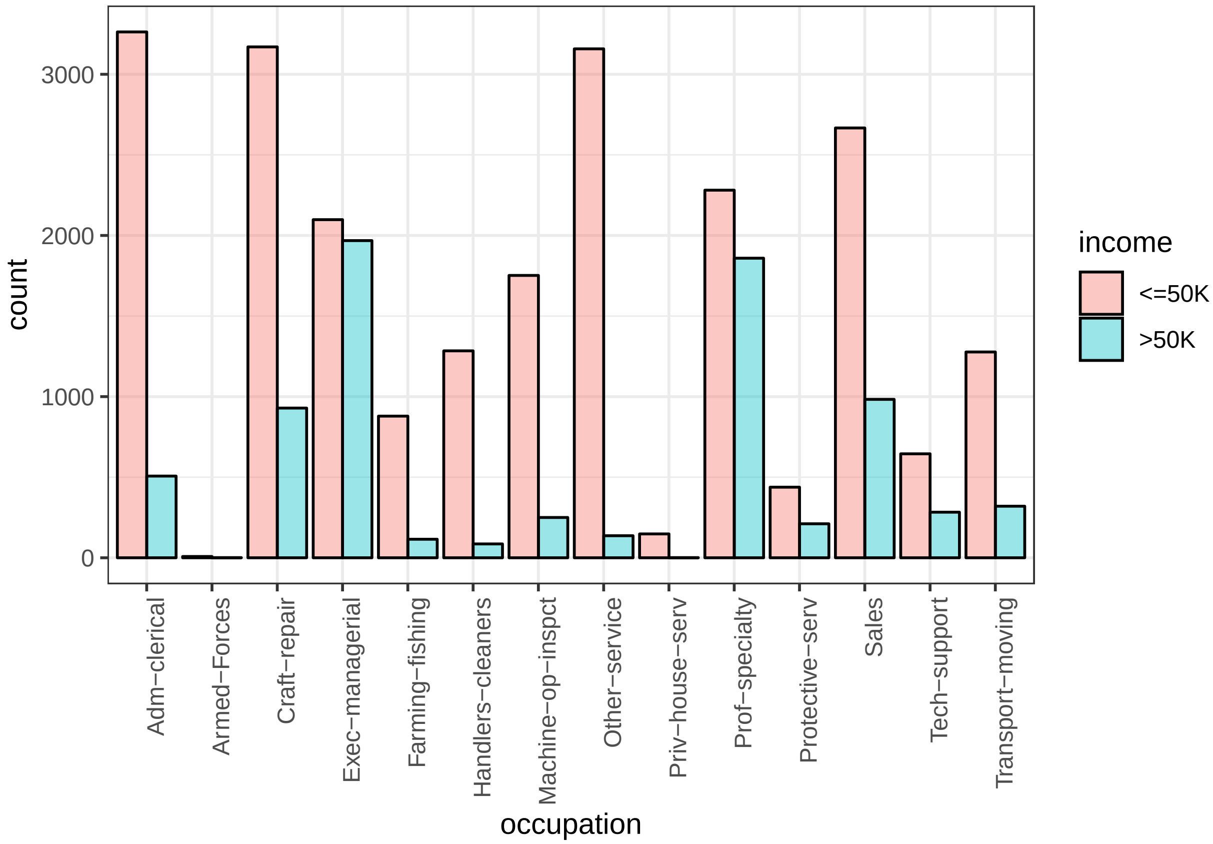

```
ggplot(adult,aes(income,group=region)) + geom_bar(aes(y=(..count..)/sum(..count..),fill=region),color='
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +  scale_y_discrete(labels = scales::percent
```



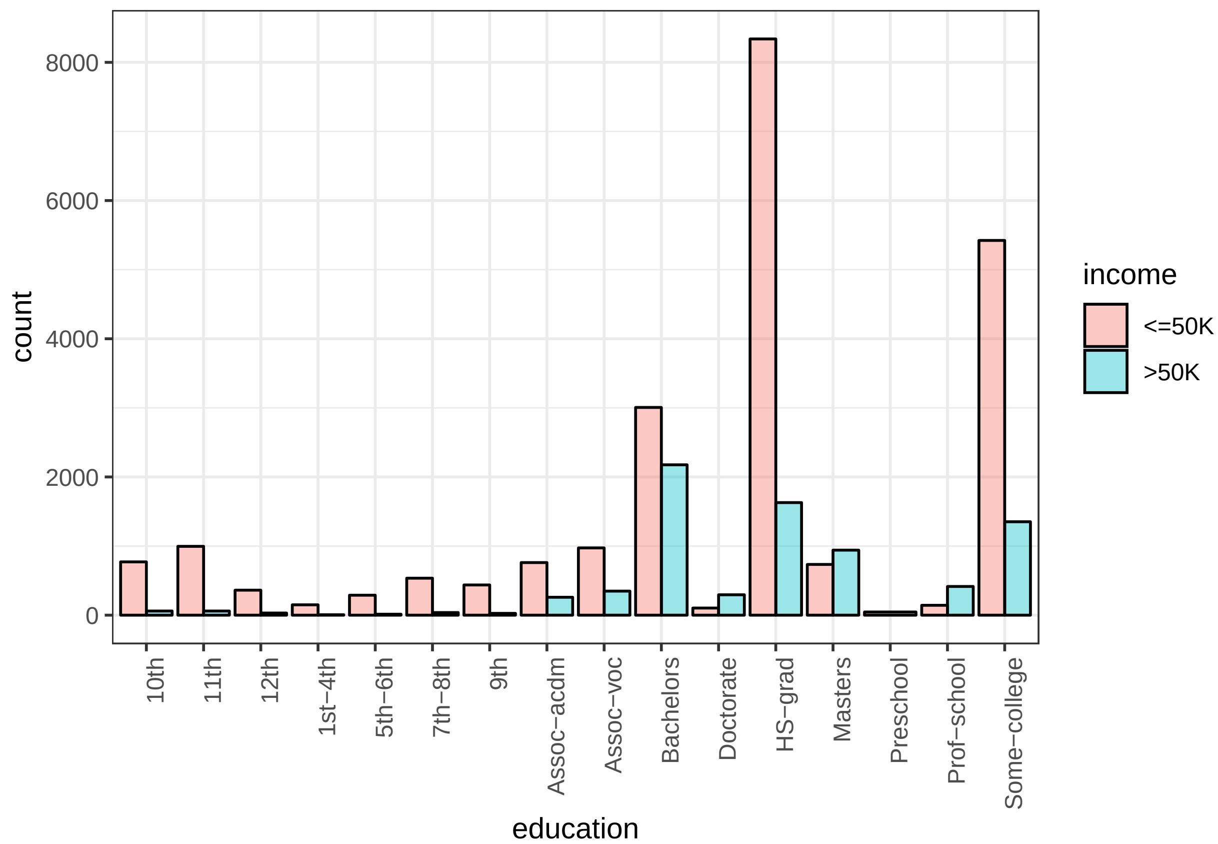

```r
#For the region effect, we can see that most of the data come from north america. But we can also see t
#the percentage of people earning more than 50K and those that earn less are similar in every region.

############Sex
ggplot(adult,aes(sex,group=income)) + geom_bar(aes(fill=income),color='black',alpha=0.4, position="dodge
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



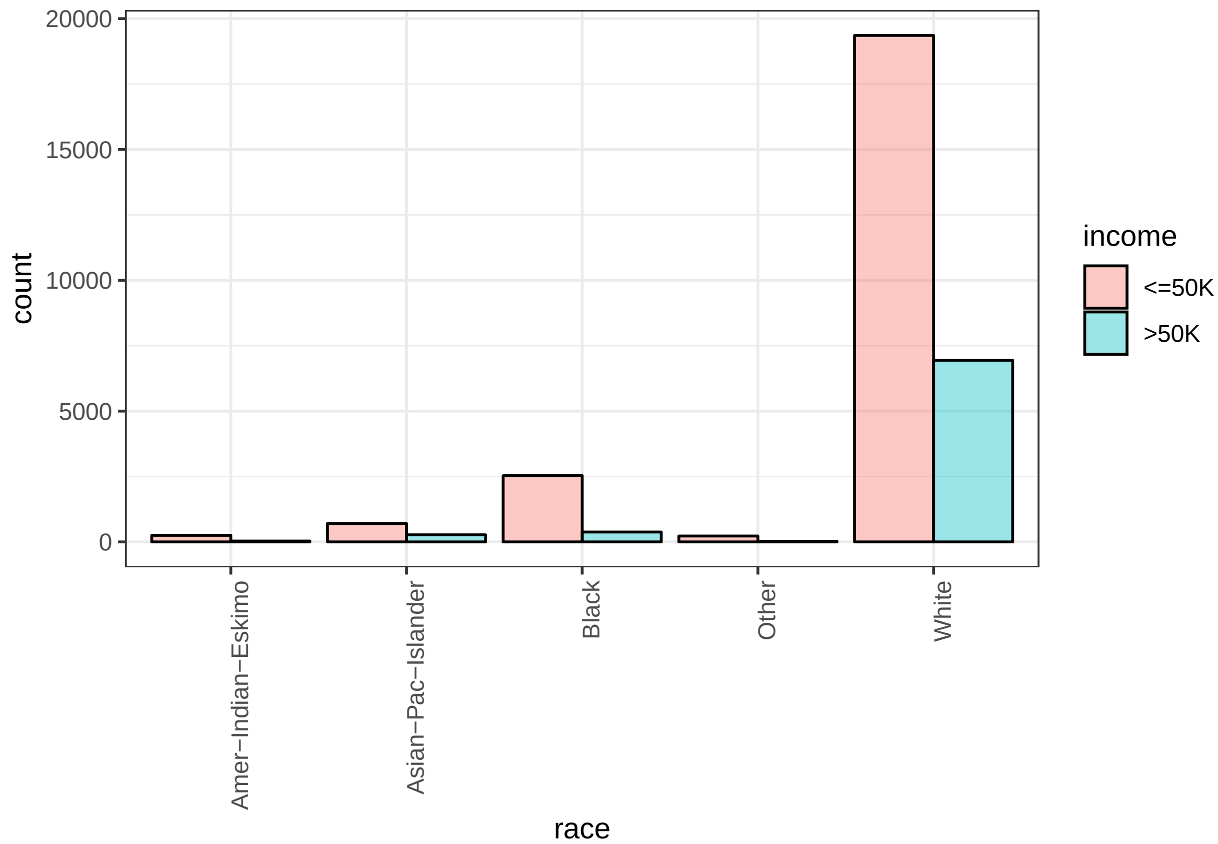

```
#We can see that according to sex, the income might be different. Women tend to earn less.

###########Occupation
ggplot(adult,aes(occupation,group=income)) + geom_bar(aes(fill=income),color='black',alpha=0.4, positio
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



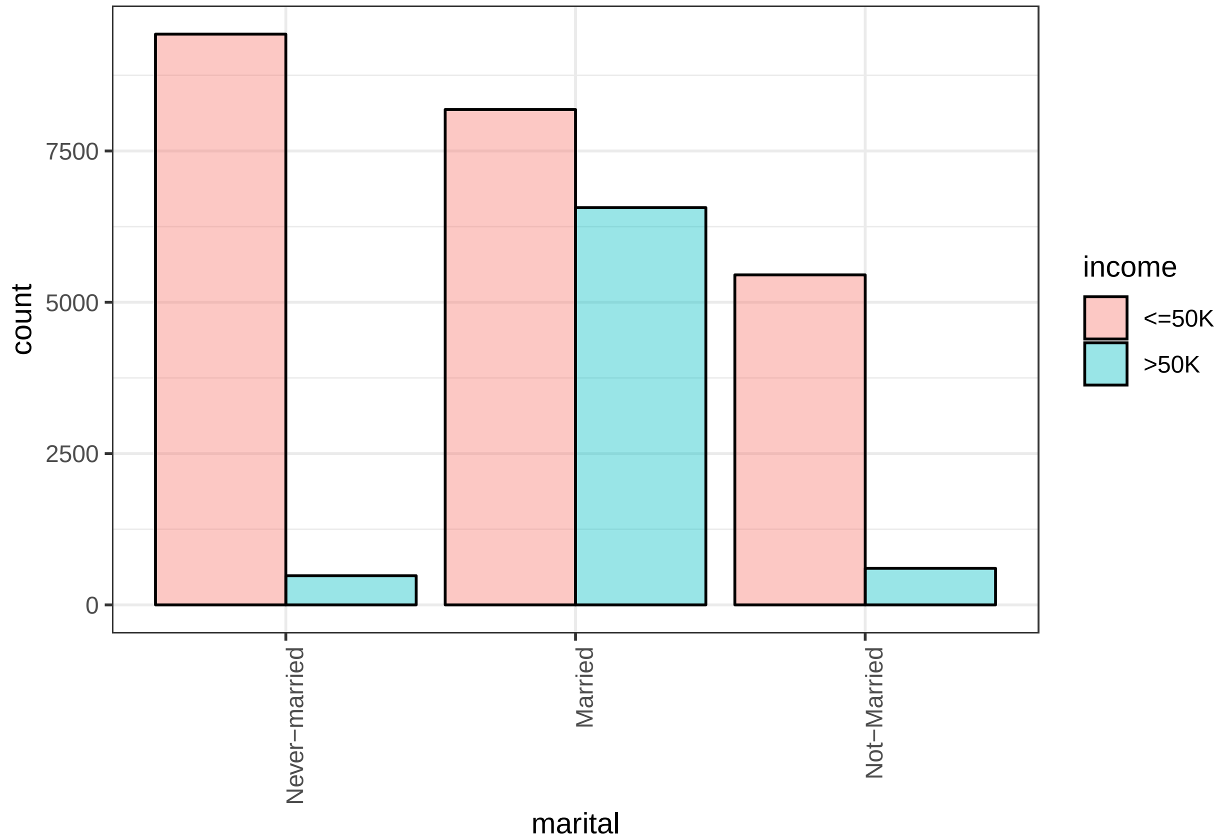

```r
#The occupation can also play a big impact as occupation such as Executive managers or professor ( spec

###########Education
ggplot(adult,aes(education,group=income)) + geom_bar(aes(fill=income),color='black',alpha=0.4, position=
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



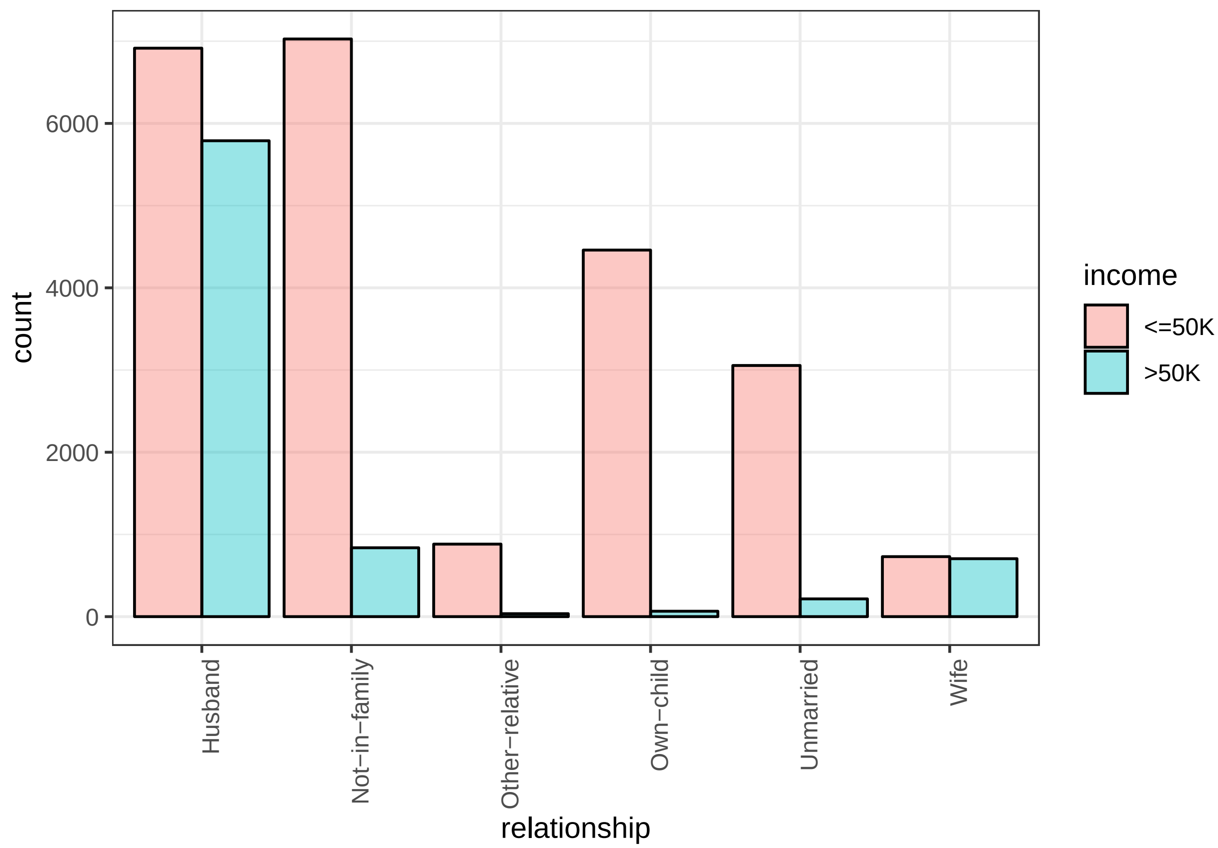

```
#Education also reflect a possible impact. As we can see that around half of those having a bachelor, m

##########Race
ggplot(adult,aes(race,group=income)) + geom_bar(aes(fill=income),color='black',alpha=0.4, position="dodg
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```




```
#Here, we can see that race could also provide some information. It looks like that Asian and white hav

##########Marital and relationship
ggplot(adult,aes(marital,group=income)) + geom_bar(aes(fill=income),color='black',alpha=0.4, position="
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```




```
ggplot(adult,aes(relationship,group=income)) + geom_bar(aes(fill=income),color='black',alpha=0.4, positi
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```




```
# Here again, Married people tend to earn more than the never married and not married counterpart.
```

Model Building

Now it's time to build a model to classify people into two groups: Above or Below 50k in Salary.

Logistic Regression is a type of classification model. In classification models, we attempt to predict the outcome of categorical dependent variables, using one or more independent variables. The independent variables can be either categorical or numerical.

Logistic regression is based on the logistic function,which always takes values between 0 and 1. Replacing the dependent variable of the logistic function with a linear combination of dependent variables we intend to use for regression, we arrive at the formula for logistic regression.

An algorithm will be build in order to predict if an adult earn more than 50K or not -the data set for building our algorithm -the data set for testing

```
# Split raw data set into train and test set: Validation set will be 10% of the Set
set.seed(101)
sample <- sample.split(adult$income, SplitRatio = 0.80)

# Training Data
train = subset(adult, sample == TRUE)

# Testing Data
test = subset(adult, sample == FALSE)
```

```
######Change income for accuracy
Change_income <- function(inc){
  inc <- as.character(inc)

  # More than 50K
  if (inc=='>50K'){
    return('1')
  }else{
    #Less than 50K
    return('0')
  }
}
test_ver <- test
test_ver$income <- sapply(test$income,Change_income)
```

Testing Models Using only one column (one variable)

```
########################Test With only age
Test_Age <- glm(formula = income ~ age, family = binomial(logit),
                data = train)
test$predicted.income = predict(Test_Age, newdata=test, type="response")

#Print Confusion Matrix
table(test$income, test$predicted.income > 0.5)
```

```
##
##          FALSE TRUE
##   <=50K  4493  121
##   >50K   1513   17
```

```
######### Print Overall Accuracy
fitted.probabilities <- test$predicted.income
fitted.results <- ifelse(fitted.probabilities > 0.5,1,0)
misClasificError <- mean(fitted.results != test_ver$income)
accuracy <-1-misClasificError
print(paste('Accuracy',format(round(accuracy, 3), nsmall = 2)))
```

```
## [1] "Accuracy 0.734"
```

```
Accuracy_results <- data_frame(method = "Using only age", Accuracy = paste('Accuracy =',format(round(ac
Accuracy_results %>% knitr::kable()
```

method	Accuracy
Using only age	Accuracy = 0.734

```
#########################
```

```
########################Test With only type of employer
Test_type_employer <- glm(formula = income ~ type_employer, family = binomial(logit),
                data = train)
test$predicted.income = predict(Test_type_employer, newdata=test, type="response")
#Print Confusion Matrix
table(test$income, test$predicted.income > 0.5)
```

```
##
##          FALSE
##   <=50K  4614
##   >50K   1530
```

```
######### Print Overall Accuracy
fitted.probabilities <- test$predicted.income
fitted.results <- ifelse(fitted.probabilities > 0.5,1,0)
misClasificError <- mean(fitted.results != test_ver$income)
accuracy <-1-misClasificError
Accuracy_results <- bind_rows(Accuracy_results,
                          data_frame(method="Using only type of employer",
                                     Accuracy = paste('Accuracy =',format(round(accuracy, 3), nsmall = 
Accuracy_results %>% knitr::kable()
```

method	Accuracy
Using only age	Accuracy = 0.734
Using only type of employer	Accuracy = 0.751

```
#########################

########################Test With only financial weight
Test_fnlwgt <- glm(formula = income ~ fnlwgt, family = binomial(logit),
                          data = train)
test$predicted.income = predict(Test_fnlwgt, newdata=test, type="response")
#Print Confusion Matrix
table(test$income, test$predicted.income > 0.5)
```

```
##
##          FALSE
##   <=50K  4614
##   >50K   1530
```

```
######### Print Overall Accuracy
fitted.probabilities <- test$predicted.income
fitted.results <- ifelse(fitted.probabilities > 0.5,1,0)
misClasificError <- mean(fitted.results != test_ver$income)
accuracy <-1-misClasificError
Accuracy_results <- bind_rows(Accuracy_results,
                              data_frame(method="Using only financial weight",
                                         Accuracy = paste('Accuracy =',format(round(accuracy, 3), nsmall
Accuracy_results %>% knitr::kable()
```

method	Accuracy
Using only age	Accuracy = 0.734
Using only type of employer	Accuracy = 0.751
Using only financial weight	Accuracy = 0.751

```
#########################

########################Test With only education
Test_education <- glm(formula = income ~ education, family = binomial(logit),
                   data = train)
test$predicted.income = predict(Test_education, newdata=test, type="response")
#Print Confusion Matrix
table(test$income, test$predicted.income > 0.5)
```

```
##
##          FALSE TRUE
##   <=50K  4408  206
##   >50K   1182  348
```

```
######### Print Overall Accuracy
fitted.probabilities <- test$predicted.income
fitted.results <- ifelse(fitted.probabilities > 0.5,1,0)
misClasificError <- mean(fitted.results != test_ver$income)
accuracy <-1-misClasificError
Accuracy_results <- bind_rows(Accuracy_results,
                              data_frame(method="Using only education",
                                         Accuracy = paste('Accuracy =',format(round(accuracy, 3), nsmal
Accuracy_results %>% knitr::kable()
```

method	Accuracy
Using only age	Accuracy = 0.734
Using only type of employer	Accuracy = 0.751
Using only financial weight	Accuracy = 0.751
Using only education	Accuracy = 0.774

```
#########################


########################Test With only region
Test_region <- glm(formula = income ~ region, family = binomial(logit),
                      data = train)
test$predicted.income = predict(Test_region, newdata=test, type="response")
#Print Confusion Matrix
table(test$income, test$predicted.income > 0.5)
```

```
##
##          FALSE
##   <=50K  4614
##   >50K   1530
```

```r
######### Print Overall Accuracy
fitted.probabilities <- test$predicted.income
fitted.results <- ifelse(fitted.probabilities > 0.5,1,0)
misClasificError <- mean(fitted.results != test_ver$income)
accuracy <-1-misClasificError
Accuracy_results <- bind_rows(Accuracy_results,
                              data_frame(method="Using only region",
                                         Accuracy = paste('Accuracy =',format(round(accuracy, 3), nsmall
Accuracy_results %>% knitr::kable()
```

method	Accuracy
Using only age	Accuracy = 0.734
Using only type of employer	Accuracy = 0.751
Using only financial weight	Accuracy = 0.751
Using only education	Accuracy = 0.774
Using only region	Accuracy = 0.751

```r
#########################


########################Test With only sex
Test_sex <- glm(formula = income ~ sex, family = binomial(logit),
                   data = train)
test$predicted.income = predict(Test_sex, newdata=test, type="response")
#Print Confusion Matrix
table(test$income, test$predicted.income > 0.5)
```

```
##
##          FALSE
##    <=50K  4614
##    >50K   1530
```

```r
######### Print Overall Accuracy
fitted.probabilities <- test$predicted.income
fitted.results <- ifelse(fitted.probabilities > 0.5,1,0)
misClasificError <- mean(fitted.results != test_ver$income)
accuracy <-1-misClasificError
Accuracy_results <- bind_rows(Accuracy_results,
                              data_frame(method="Using only sex",
                                         Accuracy = paste('Accuracy =',format(round(accuracy, 3), nsmall
Accuracy_results %>% knitr::kable()
```

method	Accuracy
Using only age	Accuracy = 0.734
Using only type of employer	Accuracy = 0.751
Using only financial weight	Accuracy = 0.751
Using only education	Accuracy = 0.774
Using only region	Accuracy = 0.751
Using only sex	Accuracy = 0.751
We can see that by using those	variables independentely, we have an accuracy ranging around 75%

Model with two and three variables

```r
########################Test With age and education
Test_age_education <- glm(formula = income ~ age +education, family = binomial(logit),
                data = train)
test$predicted.income = predict(Test_age_education, newdata=test, type="response")
#Print Confusion Matrix
table(test$income, test$predicted.income > 0.5)
```

```
##
##         FALSE TRUE
##   <=50K  4310  304
##   >50K   1062  468
```

```r
######### Print Overall Accuracy
fitted.probabilities <- test$predicted.income
fitted.results <- ifelse(fitted.probabilities > 0.5,1,0)
misClasificError <- mean(fitted.results != test_ver$income)
accuracy <-1-misClasificError
Accuracy_results <- bind_rows(Accuracy_results,
                             data_frame(method="Using age and education",
                                        Accuracy = paste('Accuracy =',format(round(accuracy, 3), nsmal
Accuracy_results %>% knitr::kable()
```

method	Accuracy
Using only age	Accuracy = 0.734
Using only type of employer	Accuracy = 0.751
Using only financial weight	Accuracy = 0.751
Using only education	Accuracy = 0.774
Using only region	Accuracy = 0.751
Using only sex	Accuracy = 0.751
Using age and education	Accuracy = 0.778

```r
#########################


########################Test With region and type of employer
Test_region_typemployer <- glm(formula = income ~ region + type_employer, family = binomial(logit),
                          data = train)
test$predicted.income = predict(Test_region_typemployer, newdata=test, type="response")
#Print Confusion Matrix
table(test$income, test$predicted.income > 0.5)
```

```
##
##         FALSE
##   <=50K  4614
##   >50K   1530
```

```r
######### Print Overall Accuracy
fitted.probabilities <- test$predicted.income
fitted.results <- ifelse(fitted.probabilities > 0.5,1,0)
misClasificError <- mean(fitted.results != test_ver$income)
accuracy <-1-misClasificError
Accuracy_results <- bind_rows(Accuracy_results,
                              data_frame(method="Using region and type of employer",
                                         Accuracy = paste('Accuracy =',format(round(accuracy, 3), nsmal
Accuracy_results %>% knitr::kable()
```

method	Accuracy
Using only age	Accuracy = 0.734
Using only type of employer	Accuracy = 0.751
Using only financial weight	Accuracy = 0.751
Using only education	Accuracy = 0.774
Using only region	Accuracy = 0.751
Using only sex	Accuracy = 0.751
Using age and education	Accuracy = 0.778
Using region and type of employer	Accuracy = 0.751

```r
###########################
#######Test three variables
###########################

########################Test With age and education and sex
Test_age_education_sex <- glm(formula = income ~ age +education+sex, family = binomial(logit),
                              data = train)
test$predicted.income = predict(Test_age_education_sex, newdata=test, type="response")
#Print Confusion Matrix
table(test$income, test$predicted.income > 0.5)
```

```
##
##          FALSE TRUE
##    <=50K  4296  318
##    >50K    986  544
```

```r
######### Print Overall Accuracy
fitted.probabilities <- test$predicted.income
fitted.results <- ifelse(fitted.probabilities > 0.5,1,0)
misClasificError <- mean(fitted.results != test_ver$income)
accuracy <-1-misClasificError
Accuracy_results <- bind_rows(Accuracy_results,
                              data_frame(method="Using age, education and sex",
                                         Accuracy = paste('Accuracy =',format(round(accuracy, 3), nsmal
Accuracy_results %>% knitr::kable()
```

method	Accuracy
Using only age	Accuracy = 0.734
Using only type of employer	Accuracy = 0.751
Using only financial weight	Accuracy = 0.751

method	Accuracy
Using only education	Accuracy = 0.774
Using only region	Accuracy = 0.751
Using only sex	Accuracy = 0.751
Using age and education	Accuracy = 0.778
Using region and type of employer	Accuracy = 0.751
Using age, education and sex	Accuracy = 0.788

```
#########################



########################Test With age and financial weight and type of employer
Test_age_financial_employer <- glm(formula = income ~ age +fnlwgt+type_employer, family = binomial(logi
                              data = train)
test$predicted.income = predict(Test_age_financial_employer, newdata=test, type="response")
#Print Confusion Matrix
table(test$income, test$predicted.income > 0.5)
```

```
##
##          FALSE TRUE
##   <=50K  4453  161
##   >50K   1474   56
```

```
######### Print Overall Accuracy
fitted.probabilities <- test$predicted.income
fitted.results <- ifelse(fitted.probabilities > 0.5,1,0)
misClasificError <- mean(fitted.results != test_ver$income)
accuracy <-1-misClasificError
Accuracy_results <- bind_rows(Accuracy_results,
                              data_frame(method="Using age, financial weight and type of employer",
                                         Accuracy = paste('Accuracy =',format(round(accuracy, 3), nsmal
Accuracy_results %>% knitr::kable()
```

method	Accuracy
Using only age	Accuracy = 0.734
Using only type of employer	Accuracy = 0.751
Using only financial weight	Accuracy = 0.751
Using only education	Accuracy = 0.774
Using only region	Accuracy = 0.751
Using only sex	Accuracy = 0.751
Using age and education	Accuracy = 0.778
Using region and type of employer	Accuracy = 0.751
Using age, education and sex	Accuracy = 0.788
Using age, financial weight and type of employer	Accuracy = 0.734

We can see that by using two of those variables, we have a similar accuracy ranging around 75% Moreover we see that by using three of those variables, we have a similar accuracy that can go down 73,4%% or go up to 79%

Use of all variables

As the Data Set is quite small, we can use the whole range of variables for the logistic regression We will use all the features to train a glm() model on the training data set

```
#######################
#Model with all variables
#######################
model = glm(income ~ ., family = binomial(logit), data = train)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
test$predicted.income = predict(model, newdata=test, type="response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
#Print Confusion Matrix
table(test$income, test$predicted.income > 0.5)
```

```
##
##         FALSE TRUE
##   <=50K  4246  368
##   >50K    590  940
```

```
######### Print Overall Accuracy
fitted.probabilities <- test$predicted.income
fitted.results <- ifelse(fitted.probabilities > 0.5,1,0)
misClasificError <- mean(fitted.results != test_ver$income)
accuracy <-1-misClasificError
Accuracy_results <- bind_rows(Accuracy_results,
                              data_frame(method="Using every variables",
                                         Accuracy = paste('Accuracy =',format(round(accuracy, 3), nsmall
Accuracy_results %>% knitr::kable()
```

method	Accuracy
Using only age	Accuracy = 0.734
Using only type of employer	Accuracy = 0.751
Using only financial weight	Accuracy = 0.751
Using only education	Accuracy = 0.774
Using only region	Accuracy = 0.751
Using only sex	Accuracy = 0.751
Using age and education	Accuracy = 0.778
Using region and type of employer	Accuracy = 0.751
Using age, education and sex	Accuracy = 0.788
Using age, financial weight and type of employer	Accuracy = 0.734
Using every variables	Accuracy = 0.844

Use of stepWise Function. AIC algoritm

We have a range of variables at our disposal to include in the model or not. Can we have a similar accuracy by using less variables? Thus making the model more interpretable ? We will use the function called step(). The step() function iteratively tries to remove predictor, variables from the model in an attempt to delete variables that do not significantly add to the fit

```
new.step.model <- step(model)
```

```
## Start:  AIC=16096.81
## income ~ X + age + type_employer + fnlwgt + education + education_num +
##     marital + occupation + relationship + race + sex + capital_gain +
##     capital_loss + hr_per_week + region

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=16096.81
## income ~ X + age + type_employer + fnlwgt + education + marital +
##     occupation + relationship + race + sex + capital_gain + capital_loss +
##     hr_per_week + region

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                 Df Deviance   AIC
## - X              1    15988 16096
## <none>                15987 16097
## - race           4    16001 16103
## - fnlwgt         1    15996 16104
## - region         4    16013 16115
## - type_employer  4    16044 16146
## - marital        2    16046 16152
## - sex            1    16087 16195
## - age            1    16167 16275
## - capital_loss   1    16216 16324
## - hr_per_week    1    16235 16343
## - relationship   5    16296 16396
## - occupation    13    16465 16549
## - education     15    16836 16916
## - capital_gain   1    17441 17549

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=16095.77
## income ~ age + type_employer + fnlwgt + education + marital +
##     occupation + relationship + race + sex + capital_gain + capital_loss +
##     hr_per_week + region

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
##                 Df Deviance   AIC
## <none>                15988 16096
## - race           4    16002 16102
## - fnlwgt         1    15997 16103
## - region         4    16014 16114
## - type_employer  4    16045 16145
## - marital        2    16047 16151
## - sex            1    16088 16194
## - age            1    16168 16274
## - capital_loss   1    16217 16323
## - hr_per_week    1    16236 16342
## - relationship   5    16296 16395
## - occupation    13    16466 16548
## - education     15    16838 16916
## - capital_gain   1    17442 17548
```

```
test$predicted.income = predict(new.step.model, newdata=test, type="response")

#Print Summary of Model
summary(new.step.model)
```

```
##
## Call:
## glm(formula = income ~ age + type_employer + fnlwgt + education +
##     marital + occupation + relationship + race + sex + capital_gain +
##     capital_loss + hr_per_week + region, family = binomial(logit),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.1327  -0.5188  -0.1961   0.0000   3.6650
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)                  -7.299e+00  4.017e-01 -18.172  < 2e-16 ***
## age                           2.508e-02  1.876e-03  13.369  < 2e-16 ***
## type_employerself-emp        -1.585e-02  8.446e-02  -0.188 0.851112
## type_employerPrivate          2.452e-01  6.836e-02   3.587 0.000335 ***
## type_employerFederal-gov      6.949e-01  1.179e-01   5.894 3.77e-09 ***
## type_employerUnemployed      -1.238e+01  2.166e+02  -0.057 0.954430
## fnlwgt                        5.848e-07  1.962e-07   2.981 0.002874 **
## education11th                 1.693e-01  2.423e-01   0.699 0.484816
## education12th                 5.035e-01  3.126e-01   1.611 0.107264
## education1st-4th             -6.887e-01  5.943e-01  -1.159 0.246542
## education5th-6th             -2.319e-01  3.877e-01  -0.598 0.549744
## education7th-8th             -4.671e-01  2.687e-01  -1.738 0.082180 .
## education9th                 -6.354e-02  2.994e-01  -0.212 0.831955
## educationAssoc-acdm           1.354e+00  2.022e-01   6.697 2.13e-11 ***
## educationAssoc-voc            1.418e+00  1.945e-01   7.290 3.09e-13 ***
## educationBachelors            2.013e+00  1.814e-01  11.098  < 2e-16 ***
## educationDoctorate            3.115e+00  2.497e-01  12.478  < 2e-16 ***
## educationHS-grad              8.239e-01  1.768e-01   4.661 3.15e-06 ***
## educationMasters              2.318e+00  1.932e-01  11.998  < 2e-16 ***
## educationPreschool           -1.809e+01  1.141e+02  -0.159 0.873990
## educationProf-school          2.897e+00  2.318e-01  12.499  < 2e-16 ***
## educationSome-college         1.207e+00  1.792e-01   6.735 1.63e-11 ***
## maritalMarried                1.229e+00  1.876e-01   6.547 5.86e-11 ***
## maritalNot-Married            5.482e-01  9.321e-02   5.881 4.08e-09 ***
## occupationArmed-Forces       -7.122e-01  1.753e+00  -0.406 0.684512
## occupationCraft-repair        3.125e-02  8.890e-02   0.352 0.725208
## occupationExec-managerial     7.636e-01  8.561e-02   8.920  < 2e-16 ***
## occupationFarming-fishing    -1.089e+00  1.530e-01  -7.120 1.08e-12 ***
## occupationHandlers-cleaners  -7.314e-01  1.584e-01  -4.616 3.91e-06 ***
## occupationMachine-op-inspct  -2.493e-01  1.125e-01  -2.215 0.026752 *
## occupationOther-service      -8.116e-01  1.293e-01  -6.275 3.49e-10 ***
## occupationPriv-house-serv    -3.659e+00  1.951e+00  -1.876 0.060703 .
## occupationProf-specialty      4.782e-01  9.036e-02   5.292 1.21e-07 ***
## occupationProtective-serv     5.820e-01  1.397e-01   4.165 3.11e-05 ***
## occupationSales               2.687e-01  9.167e-02   2.932 0.003372 **
## occupationTech-support        6.228e-01  1.245e-01   5.002 5.68e-07 ***
## occupationTransport-moving   -1.249e-01  1.110e-01  -1.125 0.260486
## relationshipNot-in-family    -9.379e-01  1.841e-01  -5.095 3.49e-07 ***
## relationshipOther-relative   -1.199e+00  2.445e-01  -4.905 9.32e-07 ***
## relationshipOwn-child        -1.920e+00  2.280e-01  -8.421  < 2e-16 ***
## relationshipUnmarried        -1.105e+00  2.061e-01  -5.364 8.15e-08 ***
## relationshipWife              1.388e+00  1.154e-01  12.030  < 2e-16 ***
## raceAsian-Pac-Islander        6.520e-01  3.027e-01   2.154 0.031253 *
## raceBlack                     4.982e-01  2.693e-01   1.850 0.064264 .
## raceOther                     1.755e-01  3.989e-01   0.440 0.660064
## raceWhite                     6.735e-01  2.569e-01   2.621 0.008760 **
## sexMale                       8.669e-01  8.820e-02   9.829  < 2e-16 ***
## capital_gain                  3.218e-04  1.181e-05  27.245  < 2e-16 ***
## capital_loss                  6.367e-04  4.265e-05  14.927  < 2e-16 ***
## hr_per_week                   2.898e-02  1.860e-03  15.584  < 2e-16 ***
## regionLatin.and.South.America -6.006e-01  1.492e-01  -4.025 5.70e-05 ***
## regionAsia                   -3.857e-02  1.925e-01  -0.200 0.841180
## regionOther                  -4.470e-01  1.540e-01  -2.902 0.003707 **
## regionEurope                  8.924e-02  1.447e-01   0.617 0.537524
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 27586  on 24573  degrees of freedom
## Residual deviance: 15988  on 24520  degrees of freedom
## AIC: 16096
##
## Number of Fisher Scoring iterations: 13
```

```
#Print Confusion Matrix
table(test$income, test$predicted.income > 0.5)
```

```
##
##         FALSE TRUE
##   <=50K  4248  366
##   >50K    587  943
```

```
######### Print Overall Accuracy
fitted.probabilities <- test$predicted.income
fitted.results <- ifelse(fitted.probabilities > 0.5,1,0)
misClasificError <- mean(fitted.results != test_ver$income)
accuracy <-1-misClasificError
Accuracy_results <- bind_rows(Accuracy_results,
                              data_frame(method="Using Step algorithm",
                                         Accuracy = paste('Accuracy =',format(round(accuracy, 3), nsmal
Accuracy_results %>% knitr::kable()
```

method	Accuracy
Using only age	Accuracy = 0.734
Using only type of employer	Accuracy = 0.751
Using only financial weight	Accuracy = 0.751
Using only education	Accuracy = 0.774
Using only region	Accuracy = 0.751
Using only sex	Accuracy = 0.751
Using age and education	Accuracy = 0.778
Using region and type of employer	Accuracy = 0.751
Using age, education and sex	Accuracy = 0.788
Using age, financial weight and type of employer	Accuracy = 0.734
Using every variables	Accuracy = 0.844
Using Step algorithm	Accuracy = 0.845

With this, we can see that we still use the whole range of variables in order to get 84,5% accuracy. Our final model is thus: glm(formula = income ~ age + type_employer + fnlwgt + education + marital + occupation + relationship + race + sex + capital_gain +capital_loss + hr_per_week + region, family = binomial(logit), data = train)

Final Model

```
##############################################
#Final Model
###############################################
model =glm(formula = income ~ age + type_employer + fnlwgt + education +
  marital + occupation + relationship + race + sex + capital_gain +
    capital_loss + hr_per_week + region, family = binomial(logit),
  data = train)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
test$predicted.income = predict(model, newdata=test, type="response")

#Print Confusion Matrix
table(test$income, test$predicted.income > 0.5)
```

```
##
##         FALSE TRUE
##   <=50K  4248  366
##   >50K    587  943
```

```
######### Print Overall Accuracy
fitted.probabilities <- test$predicted.income
fitted.results <- ifelse(fitted.probabilities > 0.5,1,0)
misClasificError <- mean(fitted.results != test_ver$income)
accuracy <-1-misClasificError
Accuracy_results <- bind_rows(Accuracy_results,
                              data_frame(method="Final Model ( every variables except education_num)",
                                         Accuracy = paste('Accuracy =',format(round(accuracy, 3), nsmal
Accuracy_results %>% knitr::kable()
```

method	Accuracy
Using only age	Accuracy = 0.734
Using only type of employer	Accuracy = 0.751
Using only financial weight	Accuracy = 0.751
Using only education	Accuracy = 0.774
Using only region	Accuracy = 0.751
Using only sex	Accuracy = 0.751
Using age and education	Accuracy = 0.778
Using region and type of employer	Accuracy = 0.751
Using age, education and sex	Accuracy = 0.788
Using age, financial weight and type of employer	Accuracy = 0.734
Using every variables	Accuracy = 0.844
Using Step algorithm	Accuracy = 0.845
Final Model (every variables except education_num)	Accuracy = 0.845

```
#####Recall
print((4248)/(4248+366))
```

```
## [1] 0.9206762
```

```r
#####Precision
print((4248)/(4248+587))
```

```
## [1] 0.8785936
```

3.Result

Final result table

```r
Accuracy_results %>%knitr::kable()
```

method	Accuracy
Using only age	Accuracy = 0.734
Using only type of employer	Accuracy = 0.751
Using only financial weight	Accuracy = 0.751
Using only education	Accuracy = 0.774
Using only region	Accuracy = 0.751
Using only sex	Accuracy = 0.751
Using age and education	Accuracy = 0.778
Using region and type of employer	Accuracy = 0.751
Using age, education and sex	Accuracy = 0.788
Using age, financial weight and type of employer	Accuracy = 0.734
Using every variables	Accuracy = 0.844
Using Step algorithm	Accuracy = 0.845
Final Model (every variables except education_num)	Accuracy = 0.845

We have an accuracy of 85%, recall of 92% and precision of 88% with the final model.

4. Conclusion

Bading on the Accuracy values the best model with this submission project is the one with all the different variables except the education_num variable. The accuracy was rather high (85%). However, as with all model, the cost associated with the accuracy aginst the cost of recall or precision has to be asked beforehand in the problem statement.

But considering the accuracy value, this model gives fairly good result.