Adult Income Analysis

Rahul Saran

14/06/2019

Introduction

About the dataset

The dataset used for this project is titled 'Adult Census Income'. Information about the dataset is publicly available at the below link. https://www.kaggle.com/uciml/adult-census-income. This data was extracted from the 1994 Census bureau database in the US. It contains various demographic, relationship, educational, and occupational information about ~32000 individuals in the US, along with their income level - <=$50K or >$50K a year. The prediction task is to use the remaining data to determine whether a given person makes over $50K a year. (The actual dataset used is a different version of the dataset provided in the above link - hence, do not use the dataset in the link, as it may give slightly different result values).

Goal of the Project

The goal of the project is to identify the impact of various factors on income, and to build the best model that determines a person's income level. Various machine learning models including logistic regression, k-nearest neighbours, and random forests are built. Various factors, including overall accuracy, sensitivity, specificity, and F1 score are used to determine the best model.

Key steps that were performed

1) Data exploration and visualization to understand the variables and data provided.
2) Data cleaning to account for missing observations.
3) Exploratory analysis to understand the impact of various factors on income.
4) Modelling - machine learning algorithms are applied and the best model & its performance determined.
5) Results from both exploratory analysis & modelling are noted & conclusions identified.

Analysis

This section has 3 major parts as follows

1) Setting up - loading the required packages & data
2) Exploratory Data Analysis & Visualization
3) Modelling

Important Note

This code was written using R 3.6.0

If you are using a different version of R, please replace set.seed(1, sample.kind = "Rounding") with set.seed(1) throughout the code, otherwise it may lead to errors.

Setting Up

Let's load the required packages.

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

```r
if(!require(ggthemes))
  install.packages("ggthemes", repos = "http://cran.us.r-project.org")
if(!require(caret))
  install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(purrr))
  install.packages("purrr", repos = "http://cran.us.r-project.org")
if(!require(randomForest))
  install.packages("randomForest", repos = "http://cran.us.r-project.org")
if(!require(Rborist))
  install.packages("Rborist", repos = "http://cran.us.r-project.org")
if(!require(DescTools))
  install.packages("DescTools", repos = "http://cran.us.r-project.org")
if(!require(GoodmanKruskal))
  install.packages("GoodmanKruskal", repos = "http://cran.us.r-project.org")
if(!require(corrplot))
  install.packages("corrplot", repos = "http://cran.us.r-project.org")
if(!require(ROCR))
  install.packages("ROCR", repos = "http://cran.us.r-project.org")

#Load the required packages
library(tidyverse)
library(ggthemes)
library(caret)
library(purrr)
library(randomForest)
library(Rborist)
library(DescTools)
library(GoodmanKruskal)
library(corrplot)
library(ROCR)
```

Let's load the dataset.

There are 3 options to download the dataset -

```r
#Option 1 - The code below downloads and then loads the data into the current working directory.
if(!file.exists("./adult.data")){

  fileUrl <- "http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data"

  download.file(fileUrl, destfile = "./adult.data")

}

#Option 2 - You may go to this link and manually download the file
#and save it in your current working directory
#"http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data".

#Option 3 - You may download from the github repository for this project
#and save it in your current working directory
# "http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data".
```

Once you have downloaded the file /adult.data in your current working directory, please run the below code to load it into the variable called census.

```r
census <- read.table("adult.data", sep = ",", header = FALSE)
colnames(census) <- c("age", "workclass", "fnlwgt",
                      "education", "education.num",
                      "marital.status", "occupation",
                      "relationship", "race", "sex",
                      "capital.gain", "capital.loss",
                      "hours.per.week", "native.country", "income")
```

Exploratory Data Anlaysis & Visualization

The aims of this major part are - 1) Understanding the structure of the dataset 2) Understanding each of the variables provided 3) Understanding the relationship of the different variables with income. In common literature, it is often mentioned that education, age, or sex, for example, are associated with income, and it will be worthwhile to examine which of these claims hold true in our data.

Overall Dataset

Let's understand the dataset.

```r
#Top-level look at data
head(census)
```

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

```r
str(census)
```

```
## 'data.frame':    32561 obs. of  15 variables:
##  $ age           : int  39 50 38 53 28 37 49 52 31 42 ...
##  $ workclass     : Factor w/ 9 levels " ?"," Federal-gov",..: 8 7 5 5 5 5 5 7 5 5 ...
##  $ 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/ 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 ...
##  $ hours.per.week: int  40 13 40 40 40 40 16 45 50 40 ...
##  $ native.country: Factor w/ 42 levels " ?"," Cambodia",..: 40 40 40 40 6 40 24 40 40 40 ...
##  $ income        : Factor w/ 2 levels " <=50K"," >50K": 1 1 1 1 1 1 1 2 2 2 ...
```

```r
dim(census)
```

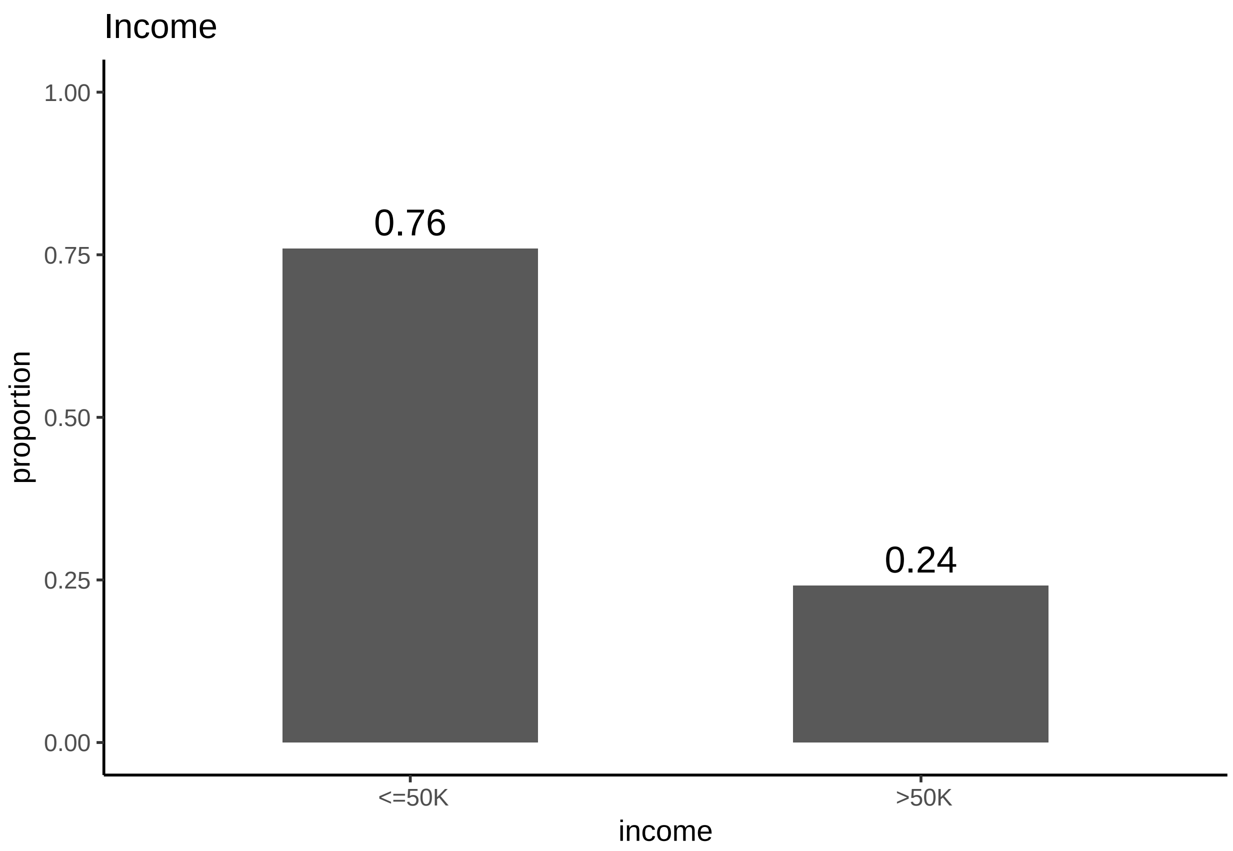
```
## [1] 32561    15
```

The dataset has 32561 observations, each observation corresponding to data for one individual. There are 14 feature variables and a label called ‘income’ which categorizes each individual into 1 of 2 buckets - income <=50K or income >50K.

Label variable - Income

Let’s now take a closer look at the ‘income’ variable.

```r
#calculate proportion of observations by income, or distribution of income across observations
incomec <- census %>% group_by(income) %>% summarize(n = n(), proportion = n/nrow(census))

#the below code gives a visual representation of this
incomec %>% ggplot(aes(income, proportion)) +
  geom_bar(stat = "identity", width = 0.5) +
  ylim(c(0,1)) +
  geom_text(label = round(incomec$proportion, 2),
            vjust = -0.5, color = "black", size = 5) +
  theme_classic() +
  ggtitle("Income")
```



income is a binary factor variable, and 24% of the individuals have income >50K.

income is converted to - 1) a factor variable incomeLevel with values 0 & 1, with 1 representing income >50K. 2) a numerical variable incomeNumeric with numeric values 0 & 1, with 1 representing income >50K. These are useful for later working.

```
#Convert income to a binary factor variable with classes "0" & "1"
census <- mutate(census, incomeLevel = ifelse(income == " <=50K", 0, 1))

#Convert income to a binary numeric variable with classes 0 & 1
census$incomeLevel <- census$incomeLevel %>% factor(levels = c("0", "1"))
census <- mutate(census, incomeNumeric = as.numeric(incomeLevel)-1)
```

Feature Variables

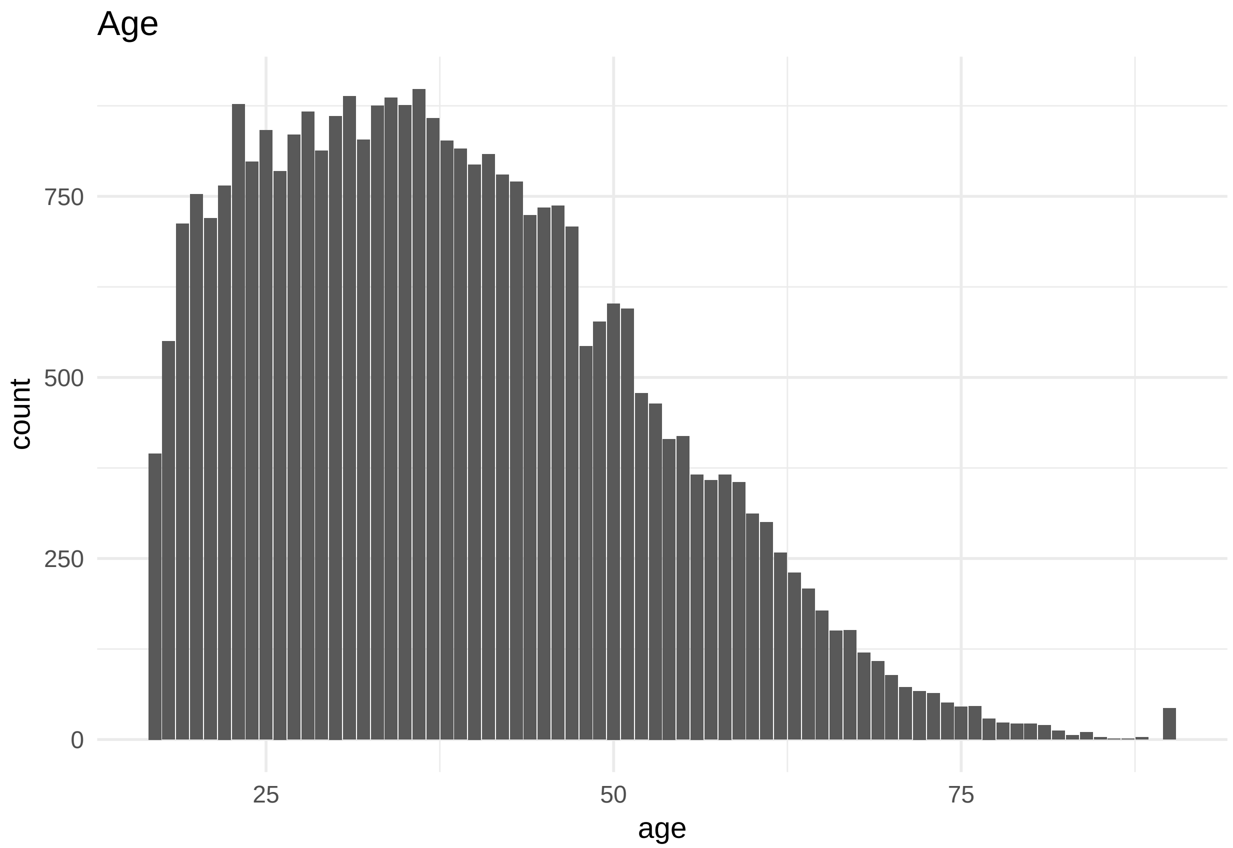
Let's understand each of the variables provided & their relationship with the income variable.

Age

```
#calculate proportion of observations by age, or distribution of age across observations
summary(census$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   17.00   28.00   37.00   38.58   48.00   90.00
```

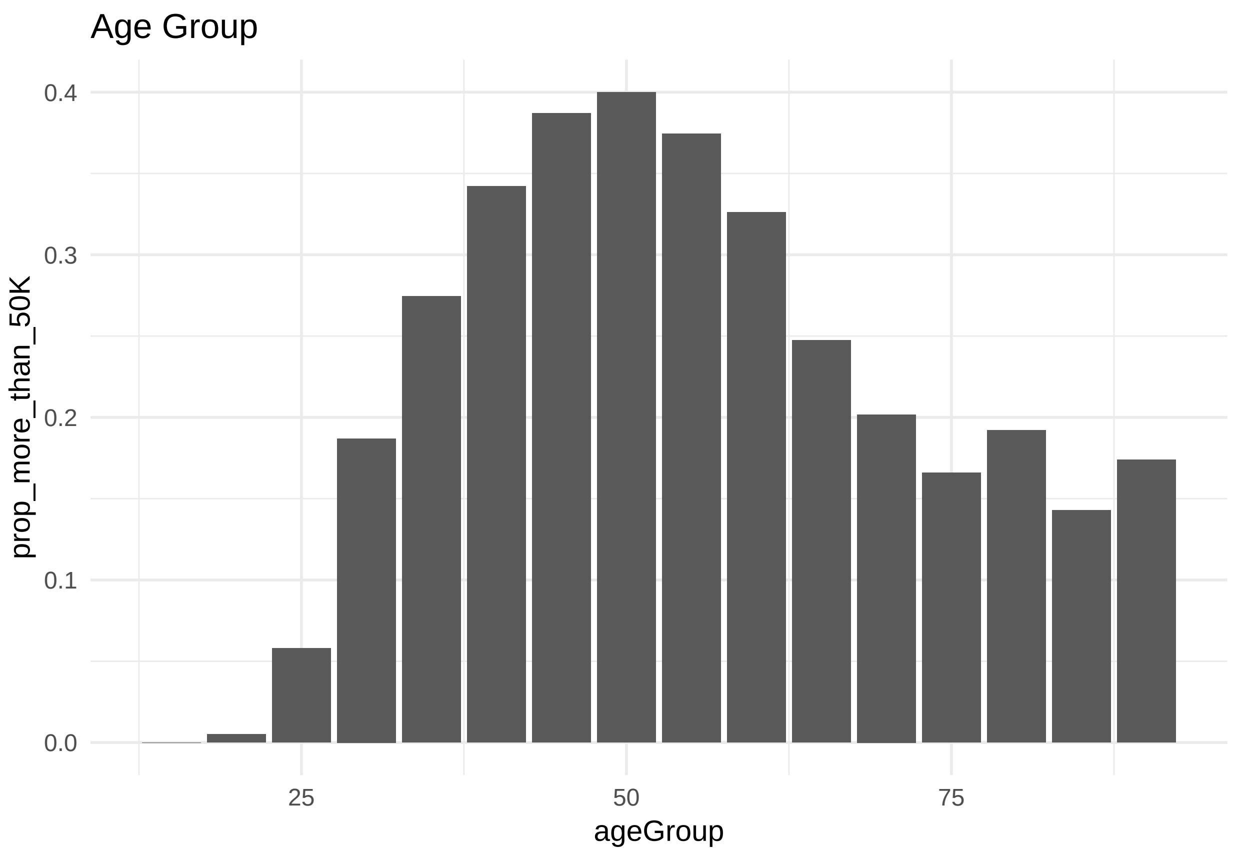
```
census %>% ggplot() +
  geom_bar(aes(x = age)) +
  theme_minimal() +
  ggtitle("Age")
```



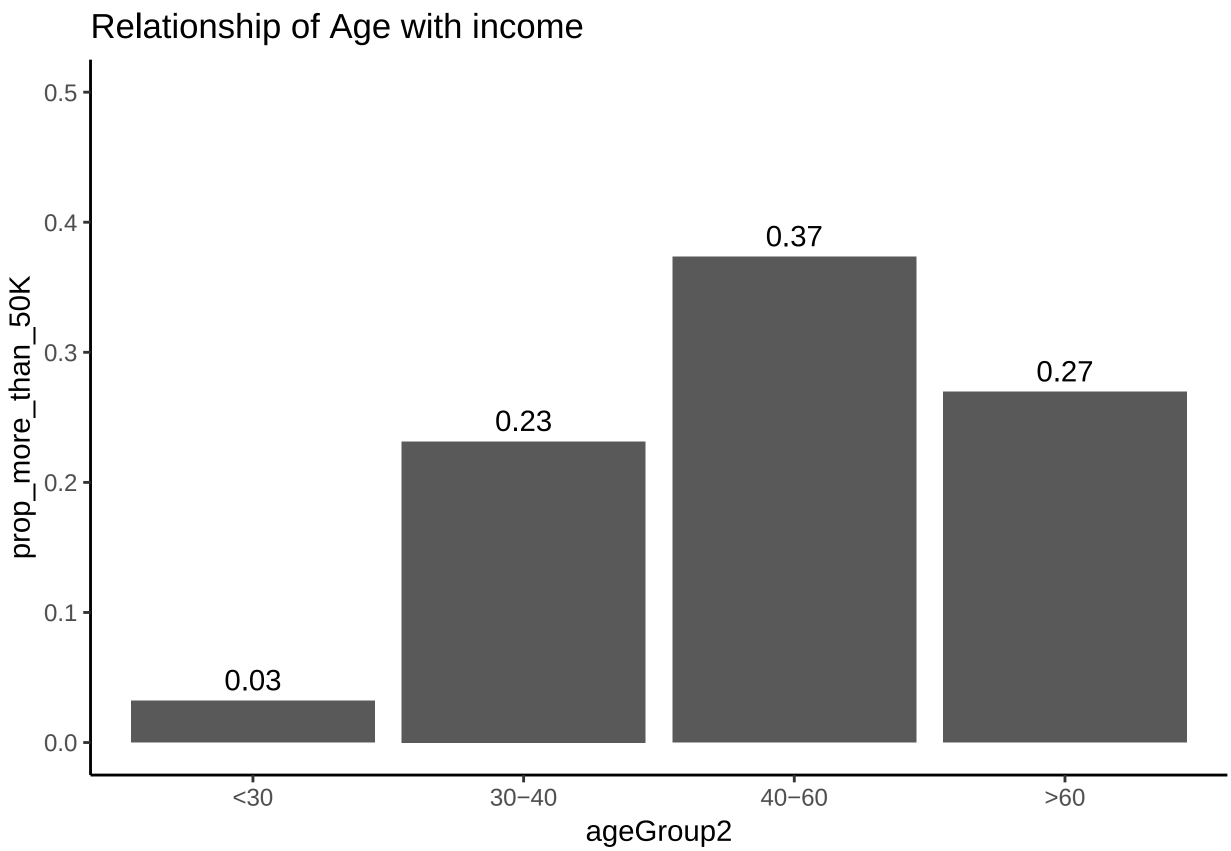
Age is integer, ranging from 17 to 90, and skewed towards the lower end (median of 37). This is confirmed from the plot, which shows most of the population appearing between the age of 17 and 55.

To assess the relationship of age with income, it is useful to bucket age into groups.

```
#first we do a preliminary bucketing of age into groups of 5 years
census <- mutate(census, ageGroup = round(age/5)*5)

#and then observe the average proportion with income >50K for these age groups
census %>%
  group_by(ageGroup) %>%
  summarize(prop_more_than_50K = mean(incomeNumeric)) %>%
  ggplot(aes(ageGroup, prop_more_than_50K)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  ggtitle("Age Group")
```



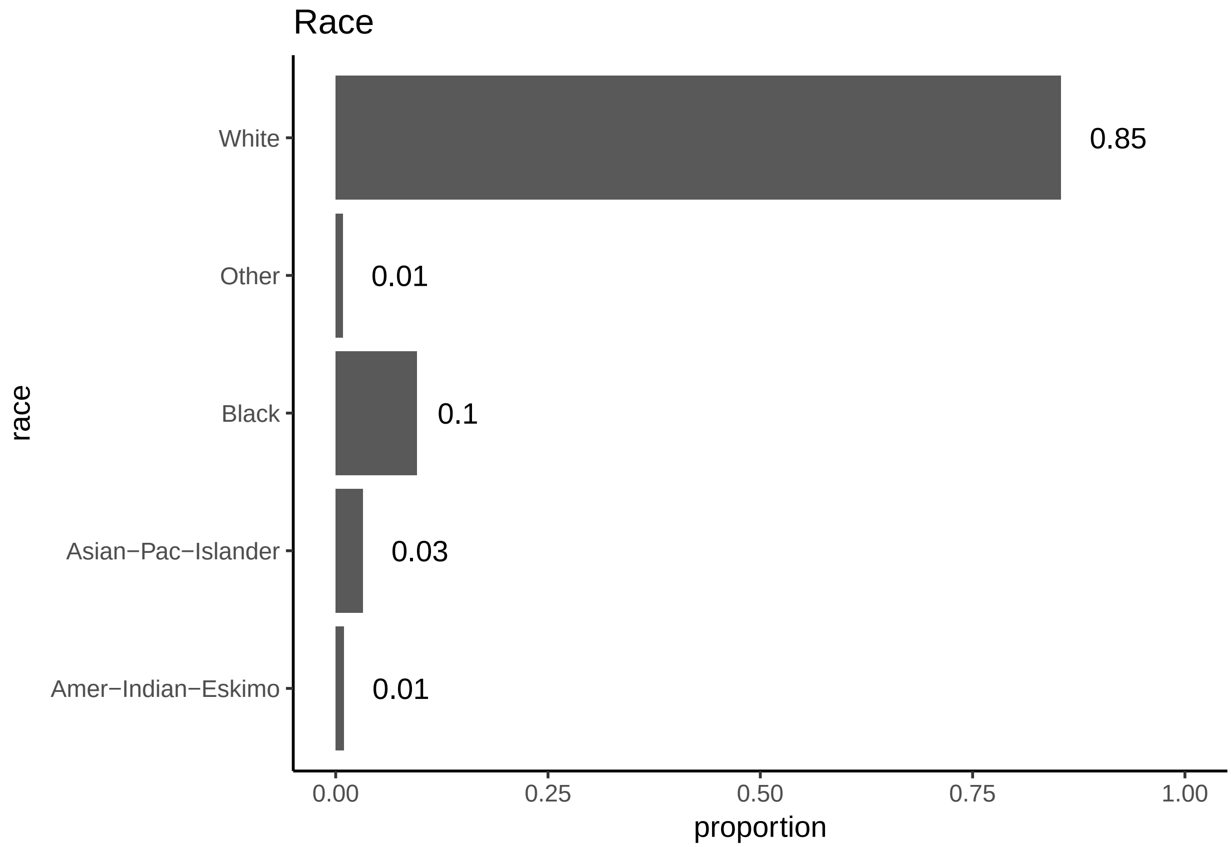
```
#Based on the plot, we will group age further into 4 groups - <25, 25-35, 40-60, >60
#which have very distinct proportions of people with income >50K
census <- mutate(census, ageGroup2 =
                   ifelse(ageGroup < 30, "<30",
                          ifelse(ageGroup < 40, "30-40",
                                 ifelse(ageGroup < 60, "40-60", ">60"))))

#Now we calculate the proportion with income >50K for these age groups
agec <- census %>%
  group_by(ageGroup2) %>%
  summarize(prop_more_than_50K = mean(incomeNumeric))

#We see this visually with the below code
agec %>%
  ggplot(aes(ageGroup2, prop_more_than_50K)) +
  ylim(c(0,0.5)) +
  geom_text(label = round(agec$prop_more_than_50K, 2),
            vjust = -0.5, color = "black", size = 4) +
  geom_bar(stat = "identity") +
  scale_x_discrete(limits = c("<30","30-40","40-60",">60")) +
  theme_classic() +
  ggtitle("Relationship of Age with income")
```



Thus, this chart shows a clear relationship of approximate age group with income level, in line with common knowledge. At age <30, it is very unlikely to have income >50K. In the age groups of 30-40 and >60, there is a 20% chance of income >50K. The highest chance of income >50K is in the age group of 40-60 (nearly 40%).

Race

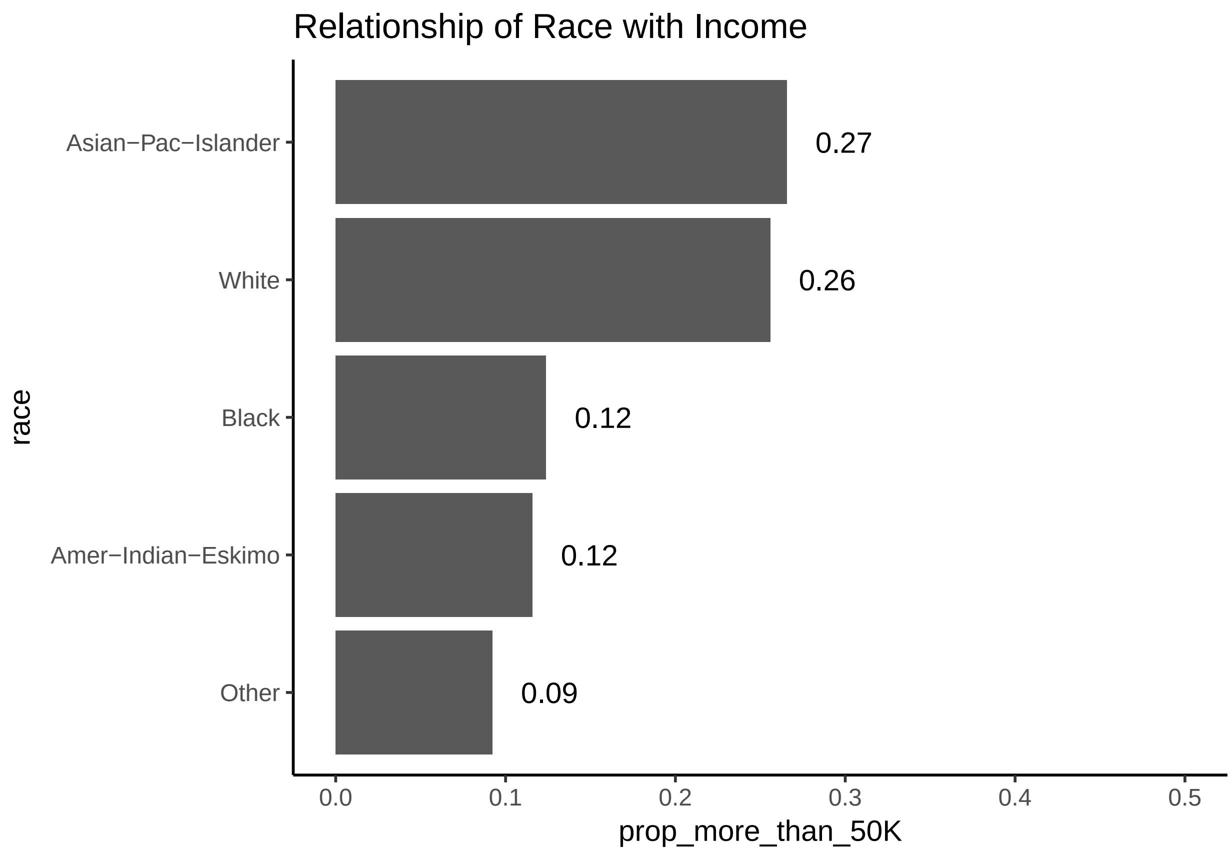
```
#calculate proportion of observations by race, or distribution of race across observations
racec <- census %>% group_by(race) %>% summarize(n = n(), proportion = n/nrow(census))

#see this visually
racec %>% ggplot(aes(race, proportion)) +
  geom_bar(stat = "identity") +
  ylim(c(0,1)) +
  coord_flip() +
  geom_text(label = round(racec$proportion, 2),
            hjust = -0.5, color = "black", size = 4) +
  theme_classic() +
  ggtitle("Race")
```



Race is factor, with 5 levels. 85% of persons are 'White'.

Let's see the relationship of race with income.

```
#calculate proportion with income >50K for each race
racec2 <- census %>%
  group_by(race) %>%
  summarize(prop_more_than_50K = mean(incomeNumeric))

#see this visually
racec2 %>%
  ggplot(aes(race, prop_more_than_50K)) +
  geom_bar(stat = "identity") +
  ylim(c(0,0.5)) +
  coord_flip() +
  geom_text(label = round(racec2$prop_more_than_50K, 2),
            hjust = -0.5, color = "black", size = 4) +
  scale_x_discrete(limits = racec2$race[order(racec2$prop_more_than_50K)]) +
  theme_classic() +
  ggtitle("Relationship of Race with Income")
```



This chart shows a difference in income level for different races. 25-30% of 'White' or 'Asian-Pac-Islander' individuals have income >50K, while only 9-12% of other racial backgrounds do.

Since racial discrimination is a topic of interest, let us examine whether some other factors could be causing this difference.

```
#calculate average educational level for each race
racec3 <- census %>%
  group_by(race) %>%
  summarize(prop_more_than_50K = mean(incomeNumeric), education = mean(education.num))
racec3
```

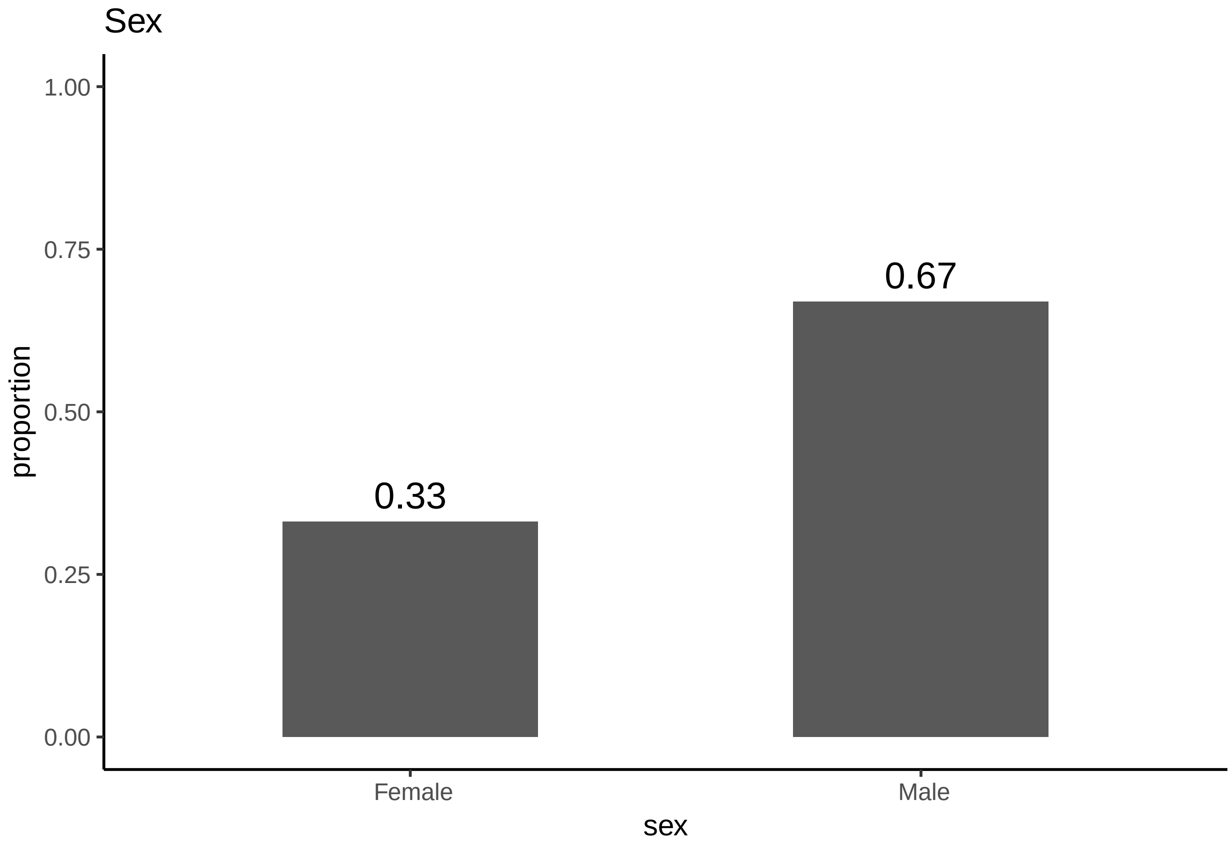
```
## # A tibble: 5 x 3
##   race                    prop_more_than_50K education
##   <fct>                                <dbl>     <dbl>
## 1 " Amer-Indian-Eskimo"               0.116       9.31
## 2 " Asian-Pac-Islander"               0.266      11.0
## 3 " Black"                            0.124       9.49
## 4 " Other"                            0.0923      8.84
## 5 " White"                            0.256      10.1
```

We see that 'White' and 'Asian-Pac-Islander' have higher education levels as well - hence, the difference in proportion with incomes >50K may not be due to racial background alone.

Sex

```
#calculate proportion of observations by sex, or distribution of sex across observations
sexc <- census %>% group_by(sex) %>% summarize(n = n(), proportion = n/nrow(census))
```

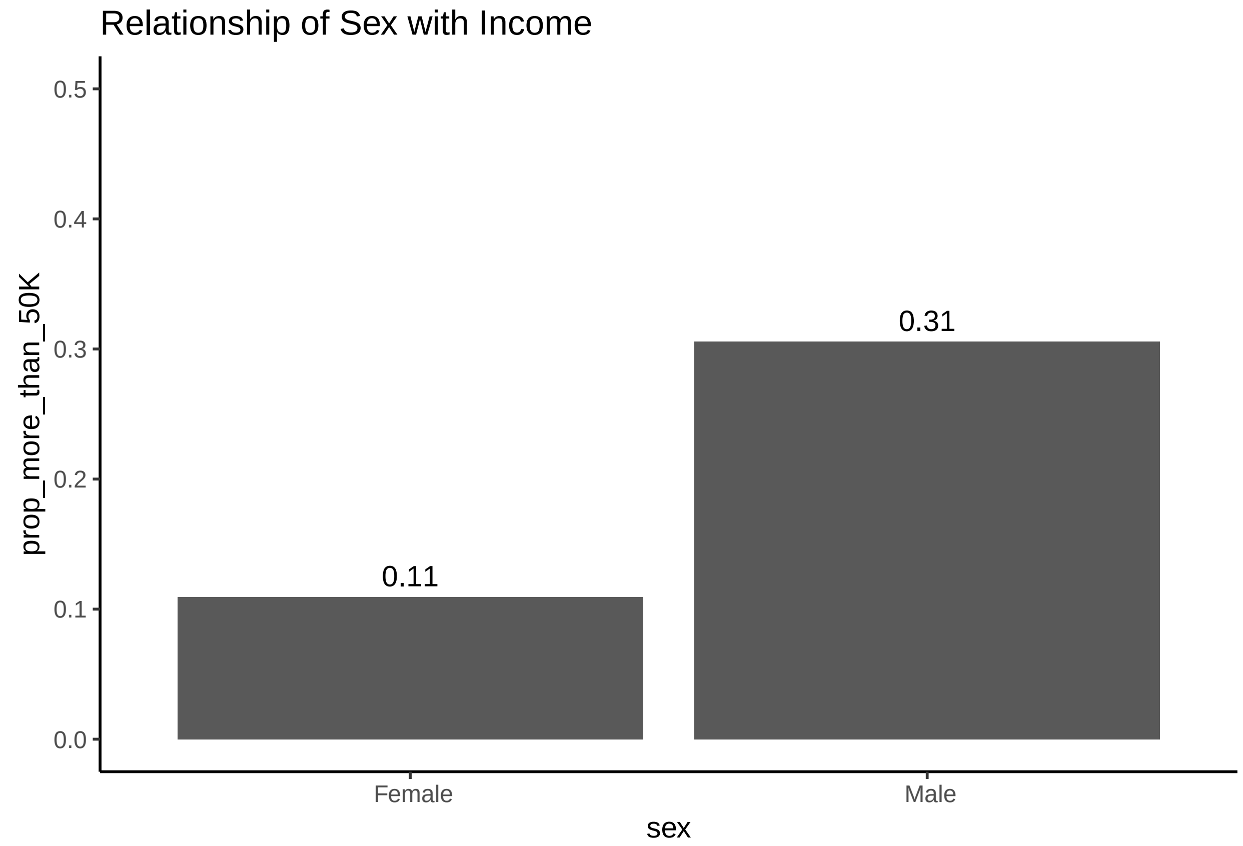
```r
#see this visually
sexc %>% ggplot(aes(sex, proportion)) +
  geom_bar(stat = "identity", width = 0.5) +
  ylim(c(0,1)) +
  geom_text(label = round(sexc$proportion, 2),
            vjust = -0.5, color = "black", size = 5) +
  theme_classic() +
  ggtitle("Sex")
```



Sex is factor, with 2 levels. 'Male' is 2/3rd of the observations.

We see if there is any income gap based on sex -

```r
#calculate proportion with income >50K for each sex
sexc2 <- census %>%
  group_by(sex) %>%
  summarize(prop_more_than_50K = mean(incomeNumeric))

#see this visually
sexc2 %>%
  ggplot(aes(sex, prop_more_than_50K)) +
  geom_bar(stat = "identity") +
  ylim(c(0,0.5)) +
  geom_text(label = round(sexc2$prop_more_than_50K, 2),
            vjust = -0.5, color = "black", size = 4) +
  theme_classic() +
  ggtitle("Relationship of Sex with Income")
```



This chart shows a considerable gap. While 31% of males have income >50K, only 11% of females have income >50K, which indicates a ’gender wage gap.

Native Country

```
summary(census$native.country)
```

```
##                           ?                    Cambodia
##                         583                          19
##                      Canada                       China
##                         121                          75
##                    Columbia                        Cuba
##                          59                          95
##          Dominican-Republic                     Ecuador
##                          70                          28
##                 El-Salvador                     England
##                         106                          90
##                      France                     Germany
##                          29                         137
##                      Greece                   Guatemala
##                          29                          64
##                       Haiti          Holand-Netherlands
##                          44                           1
##                    Honduras                        Hong
##                          13                          20
##                     Hungary                       India
##                          13                         100
##                        Iran                     Ireland
##                          43                          24
##                       Italy                     Jamaica
##                          73                          81
##                       Japan                        Laos
##                          62                          18
##                      Mexico                   Nicaragua
##                         643                          34
##  Outlying-US(Guam-USVI-etc)                        Peru
##                          14                          31
##                 Philippines                      Poland
##                         198                          60
##                    Portugal                 Puerto-Rico
##                          37                         114
##                    Scotland                       South
##                          12                          80
##                      Taiwan                    Thailand
##                          51                          18
##             Trinadad&Tobago               United-States
##                          19                       29170
##                     Vietnam                  Yugoslavia
##                          67                          16
```

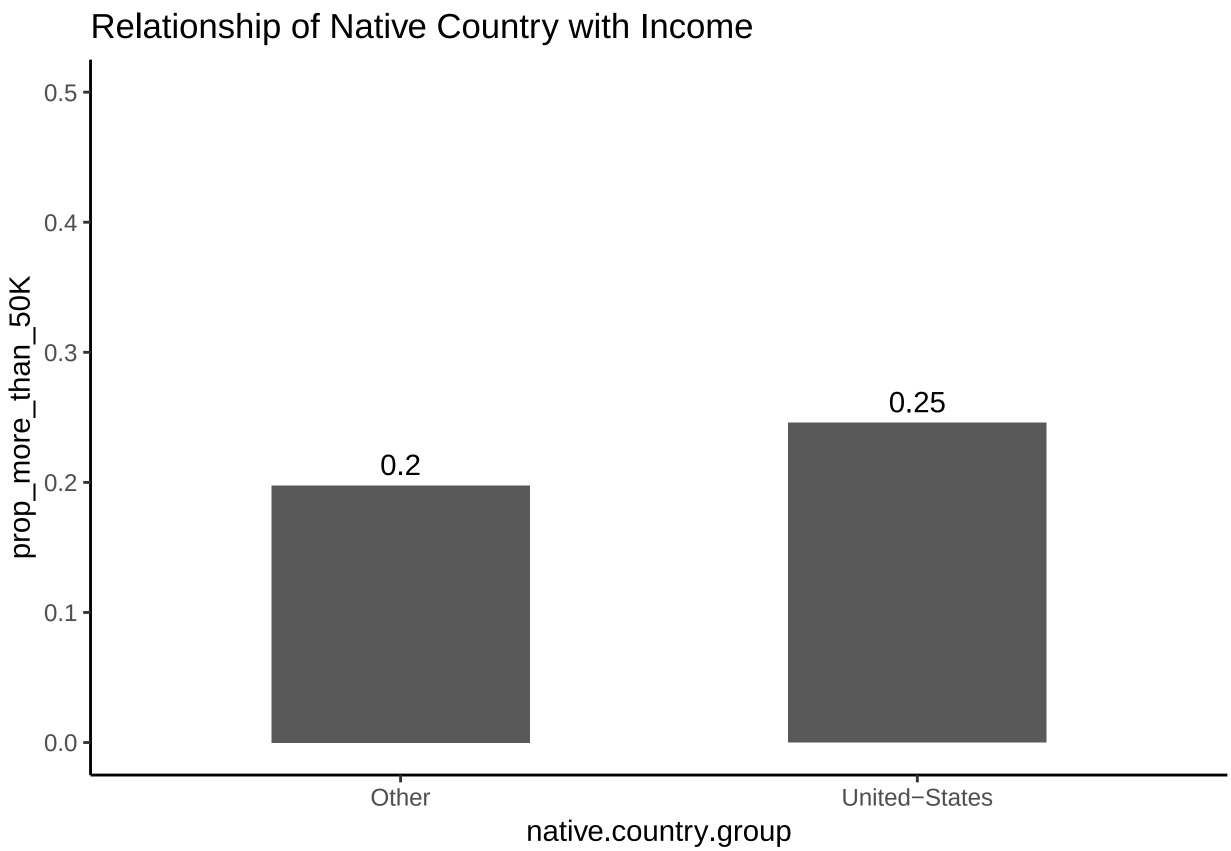
```r
mean(census$native.country == " United-States")
```

```
## [1] 0.895857
```

native.country is a factor variable with 42 levels.United-States with 90% observations is the most common. native.country is thus very skewed, with many other countries having very few observations. Using this variable may lead to very small sample sizes affecting our analysis.

Let's take a top-level look at whether there is a gap between individuals whose native country is United States, vs others.

```r
#group all observations into 2 buckets of native country - "United-States" & "Other"
census <- mutate(census, native.country.group =
                   ifelse(native.country == " United-States", "United-States", "Other"))

#calculate proportion with income >50K for both these buckets
nativec <- census %>%
  group_by(native.country.group) %>%
  summarize(prop_more_than_50K = mean(incomeNumeric))

#see this visually
nativec %>%
  ggplot(aes(native.country.group, prop_more_than_50K)) +
  geom_bar(stat = "identity", width = 0.5) +
  ylim(c(0,0.5)) +
  geom_text(label = round(nativec$prop_more_than_50K, 2),
            vjust = -0.5, color = "black", size = 4) +
  theme_classic() +
  ggtitle("Relationship of Native Country with Income")
```

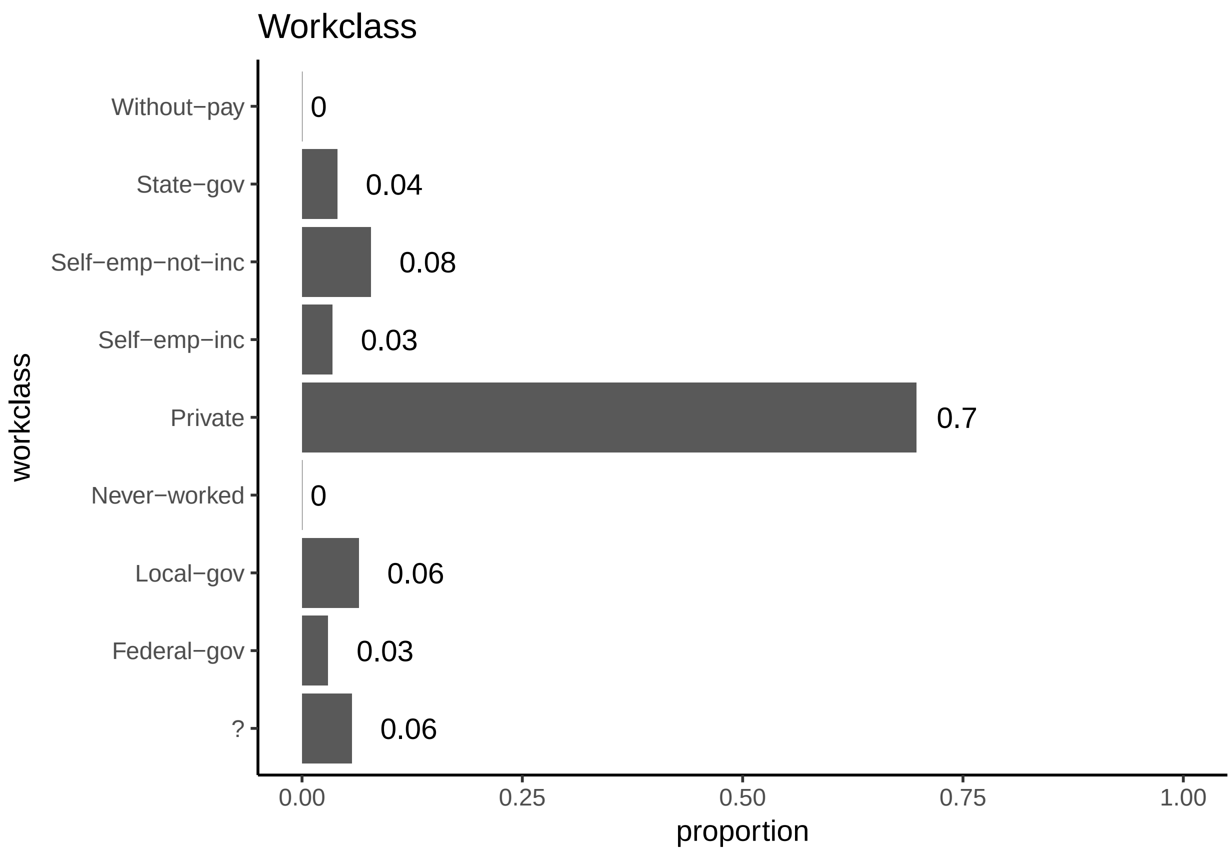


There is not a significant gap between proportion of individuals with income >50K depending on whether their native country is United States or Other.

Given the skew of the native.country variable and its apparent lack of significant impact, this variable is excluded from the modelling analysis.

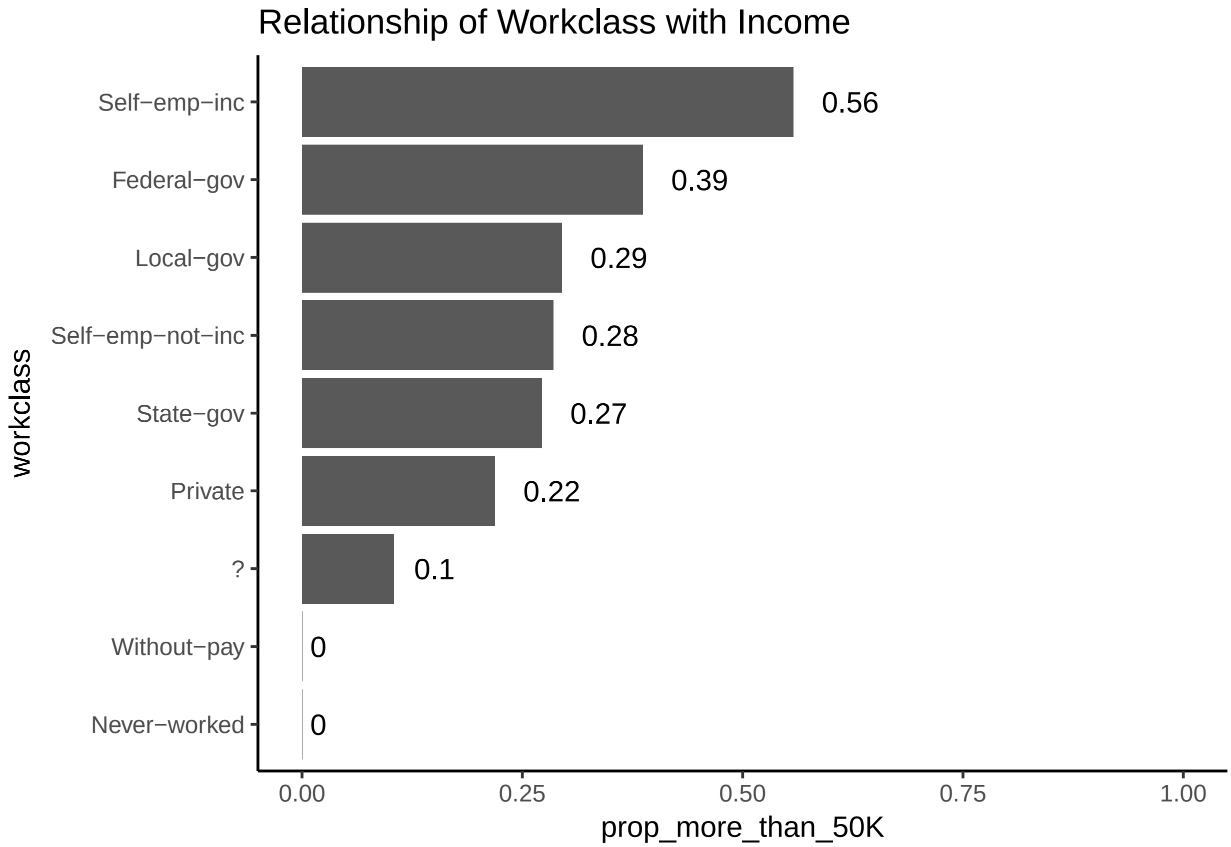
Workclass

```
#calculate proportion of observations by workclass,
#or distribution of workclass across observations
workc <- census %>% group_by(workclass) %>% summarize(n = n(), proportion = n/nrow(census))

#see this visually
workc %>% ggplot(aes(workclass, proportion)) +
  geom_bar(stat = "identity") +
  ylim(c(0,1)) +
  coord_flip() +
  geom_text(label = round(workc$proportion, 2),
            hjust = -0.5, color = "black", size = 4) +
  theme_classic() +
  ggtitle("Workclass")
```



Work class is a factor variable, with 9 classes. The majority - 70% - are in 'private' workclass.

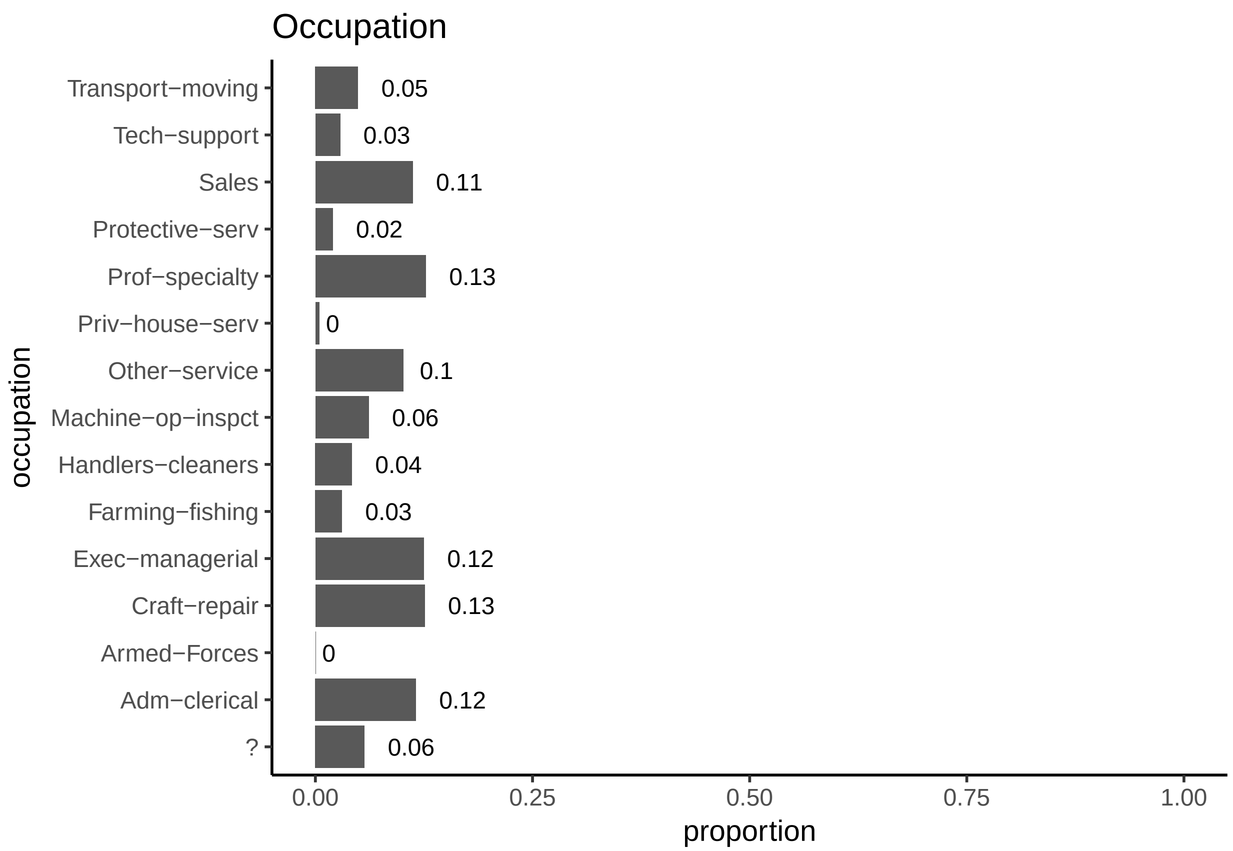
```
#calculate proportion with income >50K for each workclass
workc2 <- census %>%
  group_by(workclass) %>%
  summarize(prop_more_than_50K = mean(incomeNumeric))

#see this visually
workc2 %>%
  ggplot(aes(workclass, prop_more_than_50K)) +
  geom_bar(stat = "identity") +
  ylim(c(0,1)) +
  coord_flip() +
  geom_text(label = round(workc2$prop_more_than_50K, 2),
            hjust = -0.5, color = "black", size = 4) +
  scale_x_discrete(limits = workc2$workclass[order(workc2$prop_more_than_50K)]) +
  theme_classic() +
  ggtitle("Relationship of Workclass with Income")
```



This chart shows that 'self-emp-inc' workclass - which indicates incorporated self-employment - gives the best chance of having income >50K, more than 50%. Individuals Working for federal government also have a higher-than-average chance of income >50K (37%). Those who have never worked or working without pay, as expected, do not have income >50K. There is little variation among the other 4 workclasses. The proportion of individuals with income >50K is between 20-30% for all these workclasses (around the average of 24%).

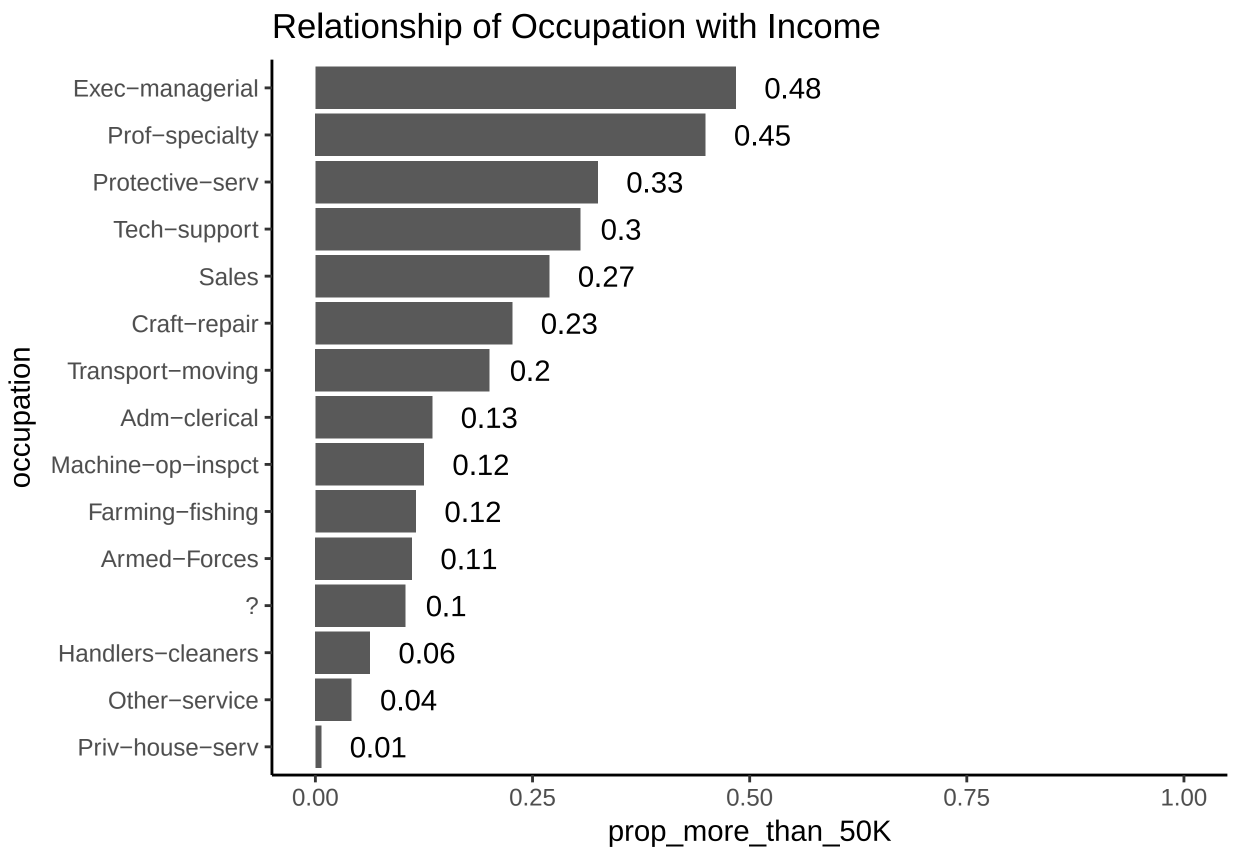
Occupation

```
#calculate proportion of observations by occupation,
#or distribution of occupation across observations
occupationc <- census %>% group_by(occupation) %>%
  summarize(n = n(), proportion = n/nrow(census))

#see this visually
occupationc %>% ggplot(aes(occupation, proportion)) +
  geom_bar(stat = "identity") +
  ylim(c(0,1)) +
  coord_flip() +
  geom_text(label = round(occupationc$proportion, 2),
            hjust = -0.5, color = "black", size = 3) +
  theme_classic() +
  ggtitle("Occupation")
```



Occupation is a factor variables, with 15 classes. Persons are distributed across occupations.

```
#calculate proportion with income >50K for each occupation
occupationc2 <- census %>%
  group_by(occupation) %>%
  summarize(prop_more_than_50K = mean(incomeNumeric))

#see this visually
occupationc2 %>%
  ggplot(aes(occupation, prop_more_than_50K)) +
  geom_bar(stat = "identity") +
  ylim(c(0,1)) +
  coord_flip() +
  geom_text(label = round(occupationc2$prop_more_than_50K, 2),
            hjust = -0.5, color = "black", size = 4) +
  scale_x_discrete(limits = occupationc2$occupation[order(occupationc2$prop_more_than_50K)]) +
  theme_classic() +
  ggtitle("Relationship of Occupation with Income")
```



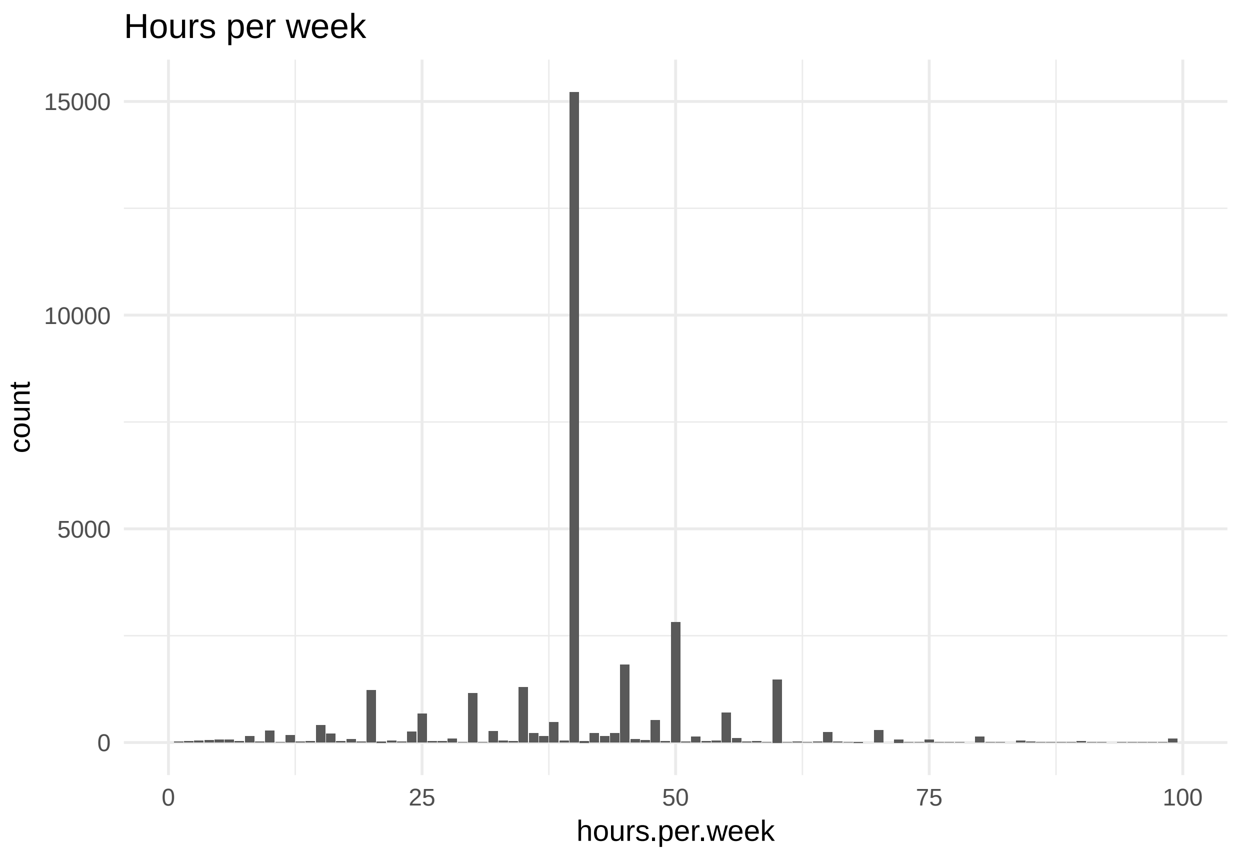
This chart shows the occupations with highest & lowest proportion of people with income >50K.

Hours.per.week

```
summary(census$hours.per.week)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    1.00   40.00   40.00   40.44   45.00   99.00
```

```
census %>% ggplot() +
  geom_bar(aes(x = hours.per.week)) +
  theme_minimal() +
  ggtitle("Hours per week")
```

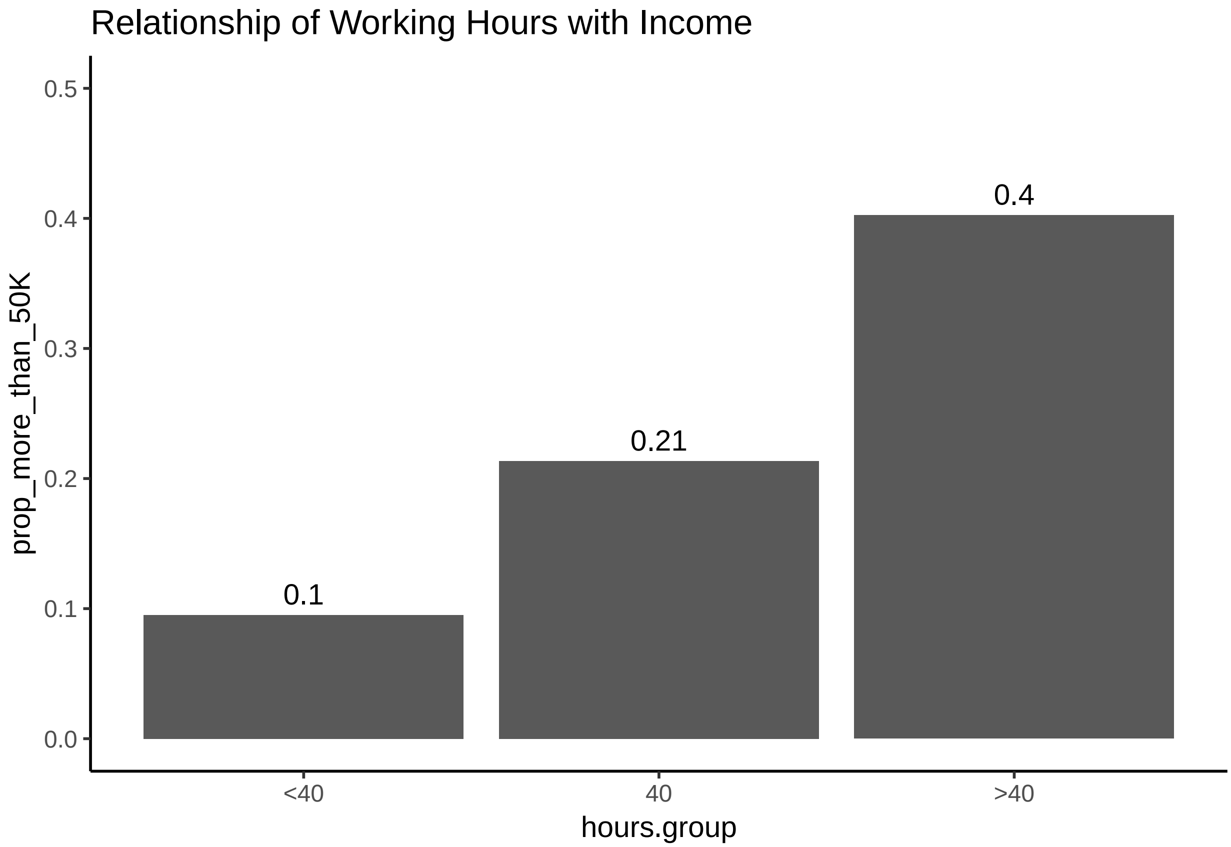


```
mean(census$hours.per.week == 40)
```

```
## [1] 0.4673382
```

This is integer, ranging from 1 to 99. Median hours are 40 hours per week, with 47% observations having this value.

```
#Since this is integer, let's first convert it into groups to assess the impact on income.
#There are 3 clear groups that stand out -
#those who work 40 hours, those who work less, and those who work more

#convert all values of hours.per.week to one of 3 categories - <40, 40, >40
census <- mutate(census, hours.group =
                   ifelse(hours.per.week < 40, "<40",
                          ifelse(hours.per.week == "40", "40", ">40")))

#calculate proportion with income >50K for each of these categories
hoursc <- census %>%
  group_by(hours.group) %>%
  summarize(prop_more_than_50K = mean(incomeNumeric))

#see this visually
hoursc %>%
  ggplot(aes(hours.group, prop_more_than_50K)) +
  geom_bar(stat = "identity") +
  ylim(c(0,0.5)) +
  geom_text(label = round(hoursc$prop_more_than_50K, 2),
          vjust = -0.5, color = "black", size = 4) +
  scale_x_discrete(limits = c("<40","40",">40")) +
  theme_classic() +
  ggtitle("Relationship of Working Hours with Income")
```



There is a strong relationship of hours worked per week with income level. Those who work 40 hours per week - the median value - also have a proportion with income >50K that is close to the median (21%). However, working <40 hours halves that probability to 10%, while working >40 hours doubles that probability to 40%.

Education & education.num

```
summary(census$education)
```

```
##         10th         11th         12th      1st-4th      5th-6th
##          933         1175          433          168          333
##      7th-8th          9th   Assoc-acdm    Assoc-voc    Bachelors
##          646          514         1067         1382         5355
##    Doctorate      HS-grad      Masters    Preschool  Prof-school
##          413        10501         1723           51          576
## Some-college
##         7291
```

Education is a factor variable with 16 classes. There is a similar variable education.num, let's also explore that variable

```
summary(census$education.num)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    9.00   10.00   10.08   12.00   16.00
```

education.num is an integer variable with 16 values from 1 to 16. This indicates that education.num is a variable assigned based on the ‘education’ variable, in a one-to-one mapping. The below code confirms this mapping.

```r
#compare education with education.num
table(census$education, census$education.num)
```

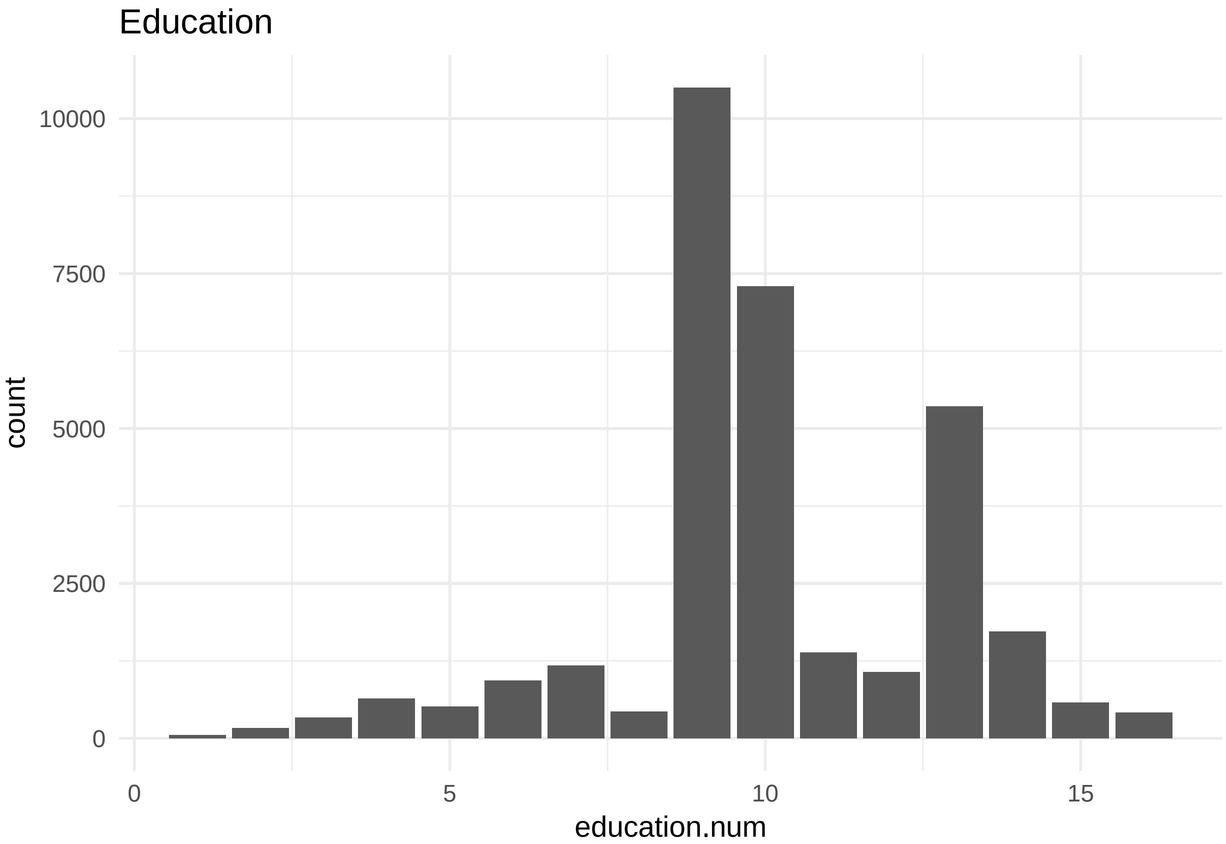
```
## 
##                      1     2     3     4     5     6     7     8     9
##    10th              0     0     0     0     0   933     0     0     0
##    11th              0     0     0     0     0     0  1175     0     0
##    12th              0     0     0     0     0     0     0   433     0
##    1st-4th           0   168     0     0     0     0     0     0     0
##    5th-6th           0     0   333     0     0     0     0     0     0
##    7th-8th           0     0     0   646     0     0     0     0     0
##    9th               0     0     0     0   514     0     0     0     0
##    Assoc-acdm        0     0     0     0     0     0     0     0     0
##    Assoc-voc         0     0     0     0     0     0     0     0     0
##    Bachelors         0     0     0     0     0     0     0     0     0
##    Doctorate         0     0     0     0     0     0     0     0     0
##    HS-grad           0     0     0     0     0     0     0     0 10501
##    Masters           0     0     0     0     0     0     0     0     0
##    Preschool        51     0     0     0     0     0     0     0     0
##    Prof-school       0     0     0     0     0     0     0     0     0
##    Some-college      0     0     0     0     0     0     0     0     0
## 
##                     10    11    12    13    14    15    16
##    10th              0     0     0     0     0     0     0
##    11th              0     0     0     0     0     0     0
##    12th              0     0     0     0     0     0     0
##    1st-4th           0     0     0     0     0     0     0
##    5th-6th           0     0     0     0     0     0     0
##    7th-8th           0     0     0     0     0     0     0
##    9th               0     0     0     0     0     0     0
##    Assoc-acdm        0     0  1067     0     0     0     0
##    Assoc-voc         0  1382     0     0     0     0     0
##    Bachelors         0     0     0  5355     0     0     0
##    Doctorate         0     0     0     0     0     0   413
##    HS-grad           0     0     0     0     0     0     0
##    Masters           0     0     0     0  1723     0     0
##    Preschool         0     0     0     0     0     0     0
##    Prof-school       0     0     0     0     0   576     0
##    Some-college   7291     0     0     0     0     0     0
```

```r
census %>% group_by(education.num) %>% summarize(education = first(education))
```

```
## # A tibble: 16 x 2
##    education.num education
##            <int> <fct>
##  1             1 " Preschool"
##  2             2 " 1st-4th"
##  3             3 " 5th-6th"
##  4             4 " 7th-8th"
##  5             5 " 9th"
##  6              6 " 10th"
##  7              7 " 11th"
##  8              8 " 12th"
##  9              9 " HS-grad"
## 10             10 " Some-college"
## 11             11 " Assoc-voc"
## 12             12 " Assoc-acdm"
## 13             13 " Bachelors"
## 14             14 " Masters"
## 15             15 " Prof-school"
## 16             16 " Doctorate"
```

Thus, education.num is a numerical variable wherein a higher number represents a qualitatively higher level of education. education.num therefore provides a useful numerical summary of the educational qualifications of the individual. education as a variable therefore does not have additional value and is not required for the analysis.

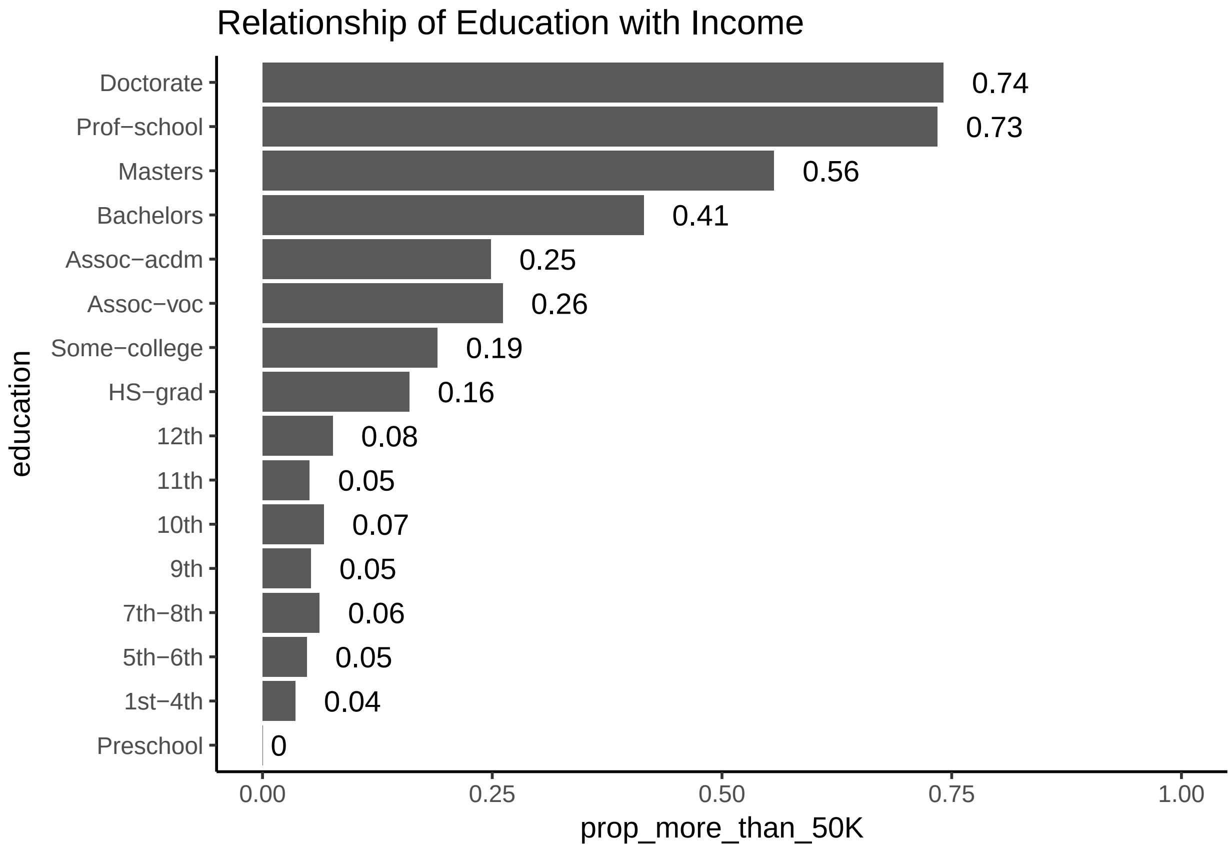
```
#calculate proportion of observations by education.num,
#or distribution of education.num across observations
census %>% ggplot() +
  geom_bar(aes(x = education.num)) +
  theme_minimal() +
  ggtitle("Education")
```



Most common educational values are 9, 10, and 13, though there are people across educational levels in the sample.

Now, let's see the relationship between education & income.

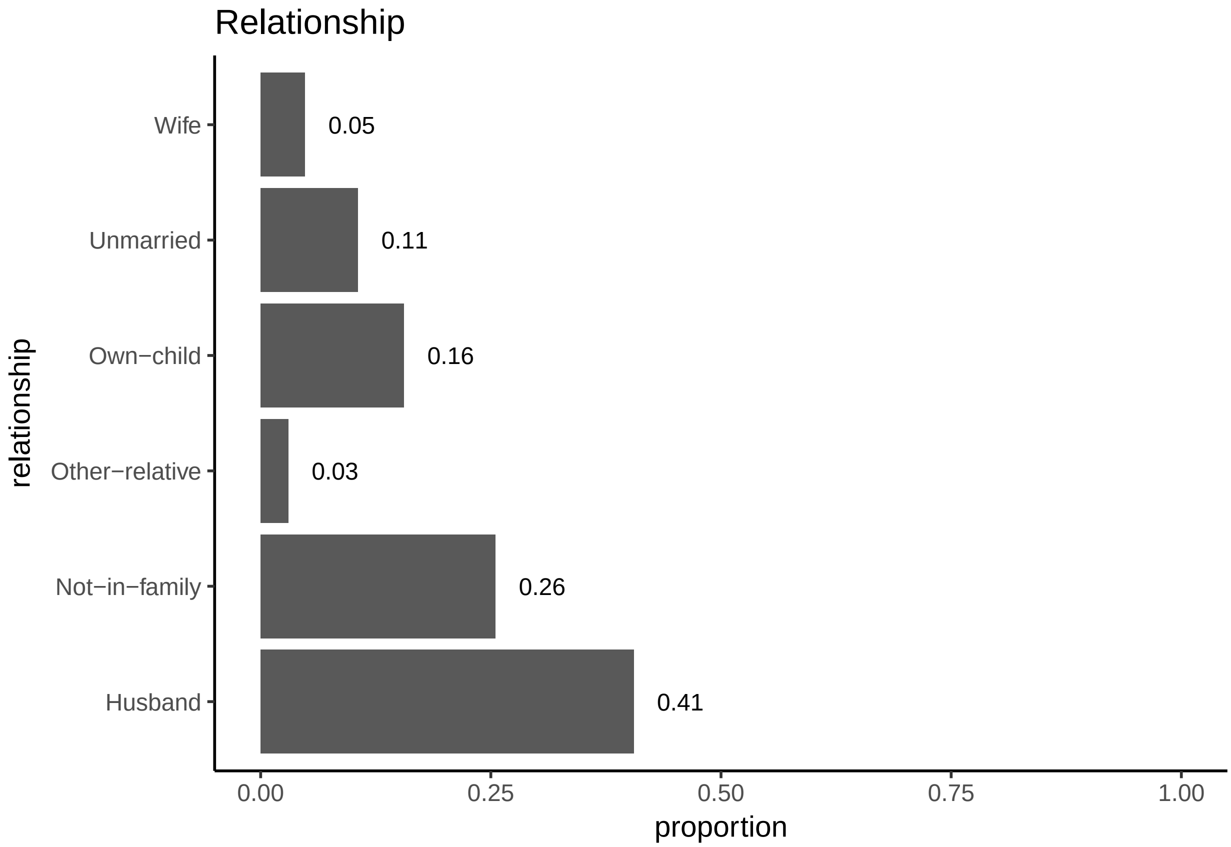
```
#calculate proportion with income >50K for each value of education.num
educationc <- census %>%
  group_by(education) %>%
  summarize(education.num = first(education.num),prop_more_than_50K = mean(incomeNumeric))

#see this visually
educationc %>%
  ggplot(aes(education, prop_more_than_50K)) +
  geom_bar(stat = "identity") +
  ylim(c(0,1)) +
  coord_flip() +
  geom_text(label = round(educationc$prop_more_than_50K, 2),
            hjust = -0.5, color = "black", size = 4) +
  scale_x_discrete(limits = educationc$education[order(educationc$education.num)]) +
  theme_classic() +
  ggtitle("Relationship of Education with Income")
```



This chart is ordered by education.num - essentially by one's education level. However, it almost appears ordered by proportion with income >50K. Thus, this chart shows a strong relationship between education & income. We can clearly see some groups here - those with some level of schooling have <10% proportion with income <50K, and this does not vary significantly by the grade up to which one has studied. Those who are high school graduates and have done some college/vocational education but dont have a college degree, have an average proportion (16-25%) with income <50K. Post this, each additional degree obtained - bachelors, masters, and doctorate - appears to increase the probability of having income >50K by approximately 15% each.

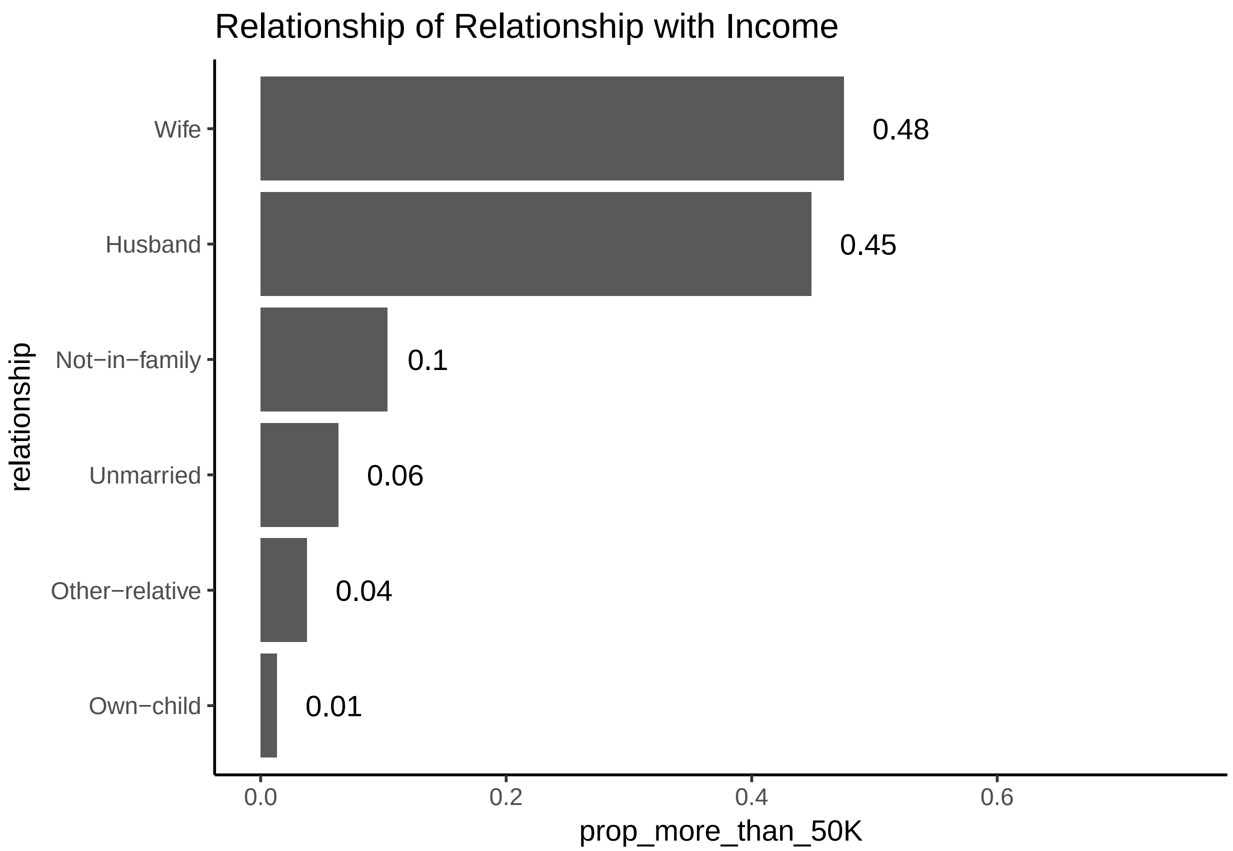
Relationship

```
#calculate proportion of observations by relationship,
#or distribution of relationship across observations
relationshipc <- census %>% group_by(relationship) %>%
  summarize(n = n(), proportion = n/nrow(census))

#see this visually
relationshipc %>% ggplot(aes(relationship, proportion)) +
  geom_bar(stat = "identity") +
  ylim(c(0,1)) +
  coord_flip() +
  geom_text(label = round(relationshipc$proportion, 2),
            hjust = -0.5, color = "black", size = 3) +
  theme_classic() +
  ggtitle("Relationship")
```



Relationship is a variable with 6 classes, with values distributed across the classes.

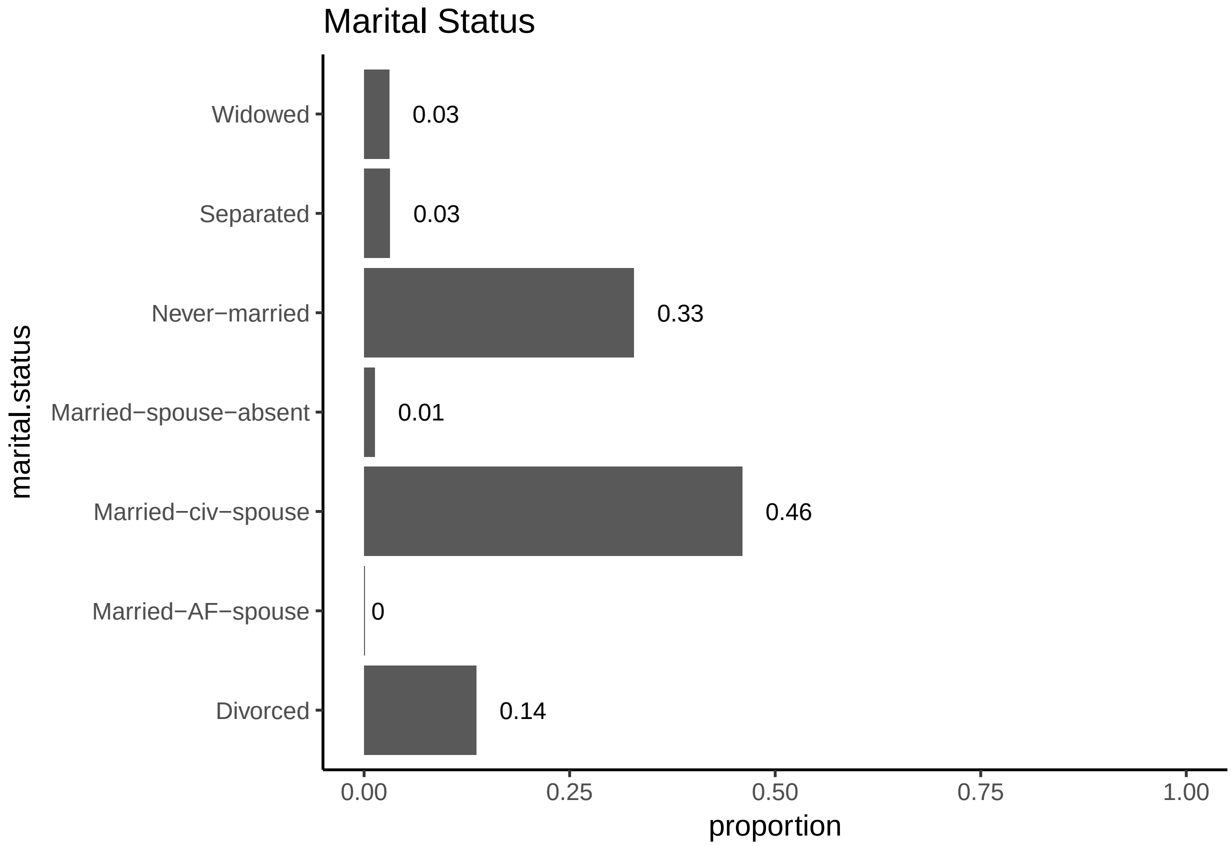
```
#calculate proportion with income >50K for each value of relationship
relationshipc2 <- census %>%
  group_by(relationship) %>%
  summarize(prop_more_than_50K = mean(incomeNumeric))

#see this visually
relationshipc2 %>%
  ggplot(aes(relationship, prop_more_than_50K)) +
  geom_bar(stat = "identity") +
  ylim(c(0,0.75)) +
  coord_flip() +
  geom_text(label = round(relationshipc2$prop_more_than_50K, 2),
            hjust = -0.5, color = "black", size = 4) +
  scale_x_discrete(limits = relationshipc2$relationship[order(relationshipc2$prop_more_than_50K)]) +
  theme_classic() +
  ggtitle("Relationship of Relationship with Income")
```



This chart clearly shows that being in a marital relationship (having a husband/wife) has a strong correlation with higher earning, with 45-50% of these individuals having income >50K. In contrast, 10% or lower of others have income > 50K.

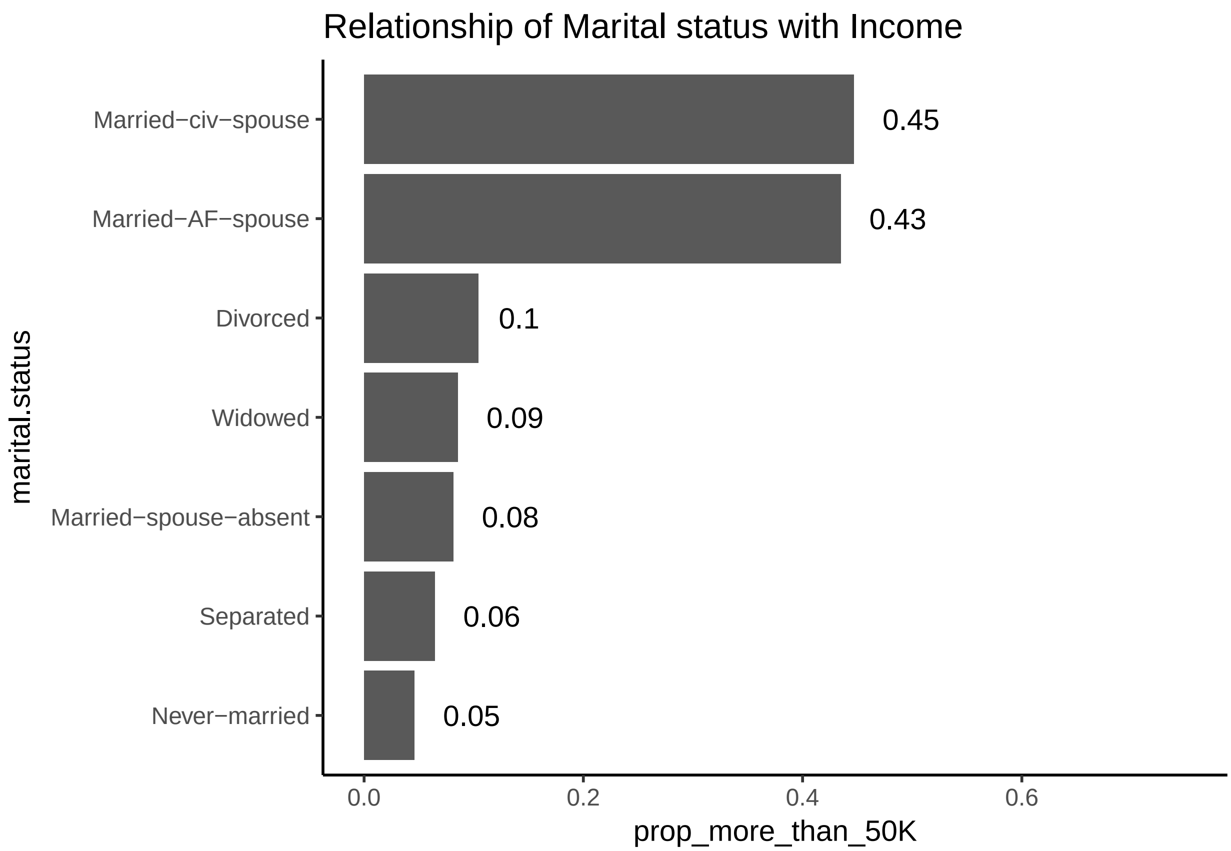
Marital-status

```
#calculate proportion of observations by marital status,
#or distribution of marital status across observations
maritalc <- census %>% group_by(marital.status) %>%
  summarize(n = n(), proportion = n/nrow(census))

#see this visually
maritalc %>% ggplot(aes(marital.status, proportion)) +
  geom_bar(stat = "identity") +
  ylim(c(0,1)) +
  coord_flip() +
  geom_text(label = round(maritalc$proportion, 2),
            hjust = -0.5, color = "black", size = 3) +
  theme_classic() +
  ggtitle("Marital Status")
```



Marital status is a factor variable with 7 classes, with 2 classes (Married-civ-spouse and Never-Married) having the majority of observations.

```
#calculate proportion with income >50K for each value of marital.status
maritalc2 <- census %>%
  group_by(marital.status) %>%
  summarize(prop_more_than_50K = mean(incomeNumeric))

#see this visually
maritalc2 %>%
  ggplot(aes(marital.status, prop_more_than_50K)) +
  geom_bar(stat = "identity") +
  ylim(c(0,0.75)) +
  coord_flip() +
  geom_text(label = round(maritalc2$prop_more_than_50K, 2),
            hjust = -0.5, color = "black", size = 4) +
  scale_x_discrete(limits = maritalc2$marital.status[order(maritalc2$prop_more_than_50K)]) +
  theme_classic() +
  ggtitle("Relationship of Marital status with Income")
```



The conclusions of this chart are similar to those seen earlier - 40-50% of married individuals have income >50K, while <10% of others do.

While these 2 variables are related, they are not related one-to-one as seen in this table.

```
#compare relationship vs marital.status
table(census$relationship, census$marital.status)
```

```
##
##                   Divorced  Married-AF-spouse  Married-civ-spouse
##    Husband               0                  9               13184
##    Not-in-family      2404                  0                  17
##    Other-relative      110                  1                 124
##    Own-child           328                  1                  95
##    Unmarried          1601                  0                   0
##    Wife                  0                 12                1556
##
##                   Married-spouse-absent  Never-married  Separated
##    Husband                            0              0          0
##    Not-in-family                    211           4706        420
##    Other-relative                    32            611         55
##    Own-child                         45           4485         99
##    Unmarried                        130            881        451
##    Wife                               0              0          0
##
##                   Widowed
##    Husband              0
##    Not-in-family       547
##    Other-relative       48
##    Own-child            15
##    Unmarried           383
##    Wife                  0
```

Hence both these variables are retained for further analysis.

Capital variables - capital.gain, capital.loss

```
summary(census$capital.gain)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0       0    1078       0   99999
```

```
#see proportion with positive capital gain
capGain <- census %>% filter(capital.gain > 0) %>% summarize(n = n())
capGain/nrow(census)
```

```
##            n
## 1 0.08328983
```

8% of the population has a positive capital.gain.

```
summary(census$capital.loss)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     0.0     0.0    87.3     0.0  4356.0
```

```
#see proportion with positive capital loss
capLoss <- census %>% filter(capital.loss > 0) %>% summarize(n = n())
capLoss/nrow(census)
```

```
##           n
## 1 0.0466509
```

4.6% of the population has a positive capital.loss.

```
census %>% filter(capital.gain >0) %>% summarize(sum(capital.loss))
```

```
##   sum(capital.loss)
## 1                 0
```

```
census %>% filter(capital.loss >0) %>% summarize(sum(capital.gain))
```

```
##   sum(capital.gain)
## 1                 0
```

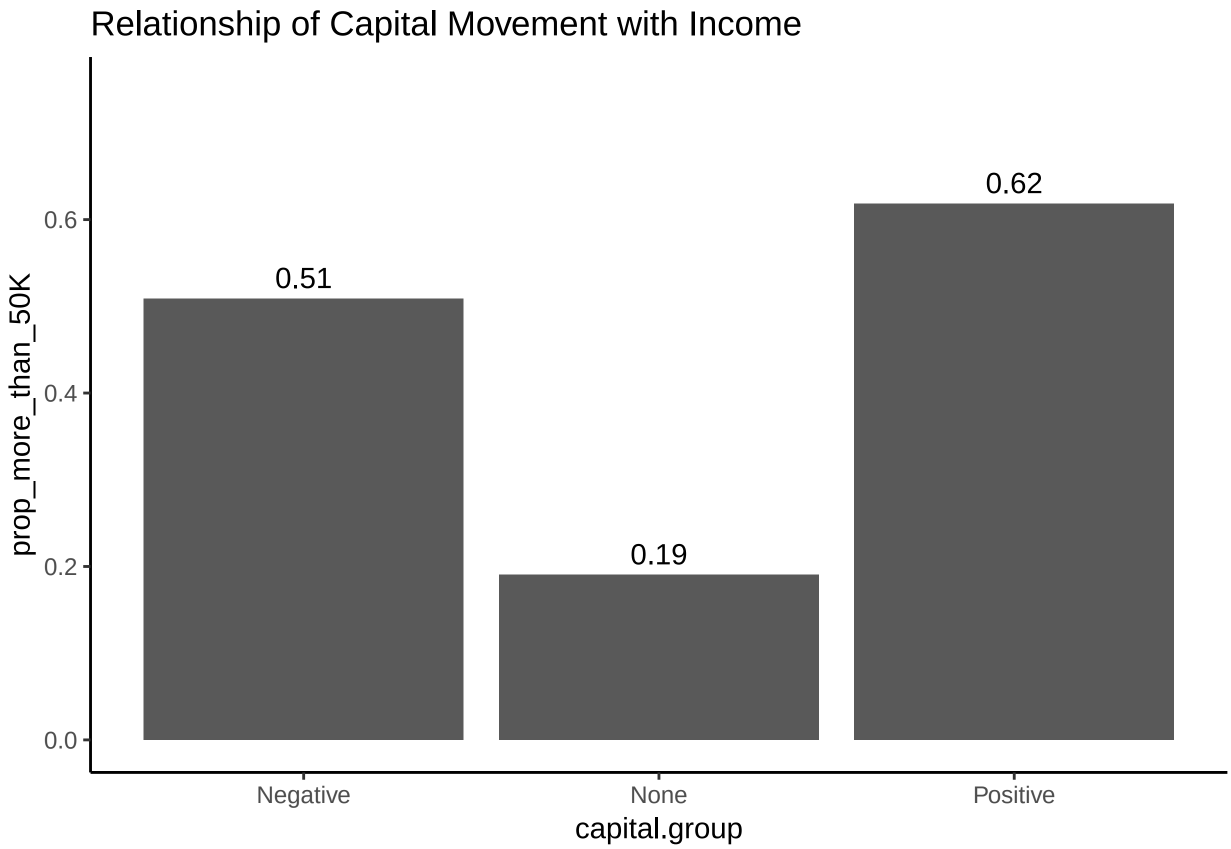
It is also confirmed that capital gain & loss are never positive together. Thus, these 2 columns are combined into a net capital movement column.

```
#create new capitalMovement variable
census <- mutate(census, capitalMovement =
                   ifelse(capital.gain > 0, capital.gain,
                          ifelse(capital.loss > 0, -capital.loss, 0)))
summary(census$capitalMovement)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -4356.0     0.0     0.0   990.4     0.0 99999.0
```

Let's now see the relationship of capital movement with income level.

```
#Since capitalMovement is integer, let's first convert it into groups
#to assess the impact on income. There are 3 clear groups that stand out -
#those with positive capital movement, those with negative, and those with none.

census <- mutate(census, capital.group =
                   ifelse(capitalMovement < 0, "Negative",
                          ifelse(capitalMovement == 0, "None", "Positive")))

#now calculate proportion with income >50K for each of these groups
capitalc <- census %>%
  group_by(capital.group) %>%
  summarize(prop_more_than_50K = mean(incomeNumeric))

#see this visually
capitalc %>%
  ggplot(aes(capital.group, prop_more_than_50K)) +
  geom_bar(stat = "identity") +
  ylim(c(0,0.75)) +
  geom_text(label = round(capitalc$prop_more_than_50K, 2),
            vjust = -0.5, color = "black", size = 4) +
  scale_x_discrete(limits = c("Negative","None","Positive")) +
  theme_classic() +
  ggtitle("Relationship of Capital Movement with Income")
```



The results of this chart are somewhat ambiguous. It seems to indicate that a significant percentage (50-60%) of those with positive or negative capital movement have income >50K, while only 20% of those with no capital movement have income >50K. The likely interpretation here is that it is income level causing capital movement, rather than the other way round. Those with incomes > 50K are more likely to invest and have capital movement, while those with incomes <50K do not engage in investing.

fnlwgt

A description of fnlwgt is provided along with the dataset, which indicates that this variable is a weight of the observation based on demographics. It is thus not used in any analysis.

Data Cleaning

3 variables - workclass, occupation, and native.country - have several "?" observations. These can be interpreted as missing data. It is now attempted to replace this missing data with meaningful data.

Replacing "?" in workclass & occupation

```
table(census$occupation, census$workclass)
```

```
##
##                         ?  Federal-gov  Local-gov  Never-worked  Private
##    ?                 1836            0          0             7        0
##    Adm-clerical         0          317        283             0     2833
##    Armed-Forces         0            9          0             0        0
##    Craft-repair         0           64        146             0     3195
##    Exec-managerial      0          180        214             0     2691
##    Farming-fishing      0            8         29             0      455
##    Handlers-cleaners    0           23         47             0     1273
##    Machine-op-inspct    0           14         12             0     1913
##    Other-service        0           35        193             0     2740
##    Priv-house-serv      0            0          0             0      149
##    Prof-specialty       0          175        705             0     2313
##    Protective-serv      0           28        304             0      190
##    Sales                0           14          7             0     2942
##    Tech-support         0           68         38             0      736
##    Transport-moving     0           25        115             0     1266
##
##                      Self-emp-inc  Self-emp-not-inc  State-gov
##    ?                            0                 0          0
##    Adm-clerical                31                50        253
##    Armed-Forces                 0                 0          0
##    Craft-repair               106               531         56
##    Exec-managerial            400               392        189
##    Farming-fishing             51               430         15
##    Handlers-cleaners            2                15          9
##    Machine-op-inspct           13                36         13
##    Other-service               27               175        124
##    Priv-house-serv              0                 0          0
##    Prof-specialty             160               373        414
##    Protective-serv              5                 6        116
##    Sales                      291               385         11
##    Tech-support                 3                26         57
##    Transport-moving            27               122         41
##
##                      Without-pay
##    ?                            0
##    Adm-clerical                 3
##    Armed-Forces                 0
##    Craft-repair                 1
##    Exec-managerial              0
##    Farming-fishing              6
##    Handlers-cleaners            1
##    Machine-op-inspct            1
##    Other-service                1
##    Priv-house-serv              0
##    Prof-specialty               0
##    Protective-serv              0
##    Sales                        0
##    Tech-support                 0
##    Transport-moving             1
```

Some workclasses & occupations have a one-to-many or many-to-one relationship. Thus, workclass & occupation are related, with some workclasses occuring only with some occupations. All possible combinations of workclass & occupation together are identified - combinations which already have some nonzero observations in them. Then, the "?" in both these variables together are filled such that these observations are distributed among the other possible combinations in a proportionate manner.

```
#create data frame of all combinations of non-"?" values
#of occupation & workclass
workclass_vs_occupation <- as.data.frame(table(census$workclass, census$occupation)[2:9,2:15])
colnames(workclass_vs_occupation)<-c("Workclass","Occupation","Count")

#calculate total number of such combinations
total<-sum(workclass_vs_occupation$Count)

#for each combination, what proportion of observations have that combination
workclass_vs_occupation$proportion<-workclass_vs_occupation$Count/total

#cumulative proportion till a particular point. This will be used in the next section
workclass_vs_occupation$cumulate<-cumsum(workclass_vs_occupation$proportion)
```

There is now a table with all possible combinations and the proportion of observations in each combination.

```
set.seed(1, sample.kind = "Rounding")

#create random numbers between 0 & 1, one for every instance of "?"
randomNumbers <- runif(sum(census$occupation=="?"))

#calculates the number of entries till that random number crosses the cumulative proportion.
#essentially, we have to replace the missing "?" values with a combination of
#occupation & workclass that is present. So, the missing 1843 values have to be replaced by
#one of these 112 combinations in a proportionate manner. The below code calculates that
#proportionate division.
ind <- rowSums(vapply(workclass_vs_occupation$cumulate,
                function(x) x<=randomNumbers, logical(length(randomNumbers))))+1
a<-as.character(workclass_vs_occupation$Workclass)[ind]
```

There are 1836 "?" entries in workclass & 1843 "?" entries in occupation, as seen above. All 1836 "?" entries in workclass are included in the 1843 "?" entries in occupation.

```r
#get the entries which have missing "?" values
missing_occupation <-which(census$occupation=="?")
missing_workclass <- which(census$workclass=="?")
missing_only_occ <- setdiff(missing_occupation,missing_workclass)

#replace each of the missing "?" values with an existing value
census$workclass[missing_workclass]= as.character(workclass_vs_occupation$Workclass)[ind][1:1836]
census$occupation[missing_workclass]= as.character(workclass_vs_occupation$Occupation)[ind][1:1836]
census$occupation[missing_only_occ] <- as.character(workclass_vs_occupation$Occupation)[ind][1837:1843]
table(census$occupation, census$workclass)
```

```
##
##                          ?  Federal-gov  Local-gov  Never-worked  Private
##    ?                 1836            0          0             7        0
##    Adm-clerical         0          317        283             0     2833
##    Armed-Forces         0            9          0             0        0
##    Craft-repair         0           64        146             0     3195
##    Exec-managerial      0          180        214             0     2691
##    Farming-fishing      0            8         29             0      455
##    Handlers-cleaners    0           23         47             0     1273
##    Machine-op-inspct    0           14         12             0     1913
##    Other-service        0           35        193             0     2740
##    Priv-house-serv      0            0          0             0      149
##    Prof-specialty       0          175        705             0     2313
##    Protective-serv      0           28        304             0      190
##    Sales                0           14          7             0     2942
##    Tech-support         0           68         38             0      736
##    Transport-moving     0           25        115             0     1266
##
##                       Self-emp-inc  Self-emp-not-inc  State-gov
##    ?                             0                 0          0
##    Adm-clerical                 31                50        253
##    Armed-Forces                  0                 0          0
##    Craft-repair                106               531         56
##    Exec-managerial             400               392        189
##    Farming-fishing              51               430         15
##    Handlers-cleaners             2                15          9
##    Machine-op-inspct            13                36         13
##    Other-service                27               175        124
##    Priv-house-serv               0                 0          0
##    Prof-specialty              160               373        414
##    Protective-serv               5                 6        116
##    Sales                       291               385         11
##    Tech-support                  3                26         57
##    Transport-moving             27               122         41
##
##                       Without-pay
##    ?                            0
##    Adm-clerical                 3
##    Armed-Forces                 0
##    Craft-repair                 1
##    Exec-managerial              0
##    Farming-fishing              6
##    Handlers-cleaners            1
##    Machine-op-inspct            1
##    Other-service                1
##    Priv-house-serv              0
##    Prof-specialty               0
##    Protective-serv              0
##    Sales                        0
##    Tech-support                 0
##    Transport-moving             1
```

This code replaces all the "?" entries in workclass & occupation with a value that already exists.

Replacing "?" in native.country

Not relevant given it is not used in analysis going forward and there is no logical way to deduce the value of the missing native.country.

Modelling

The aim of this major part is to develop models that will allow us to predict income level from the other feature variables.

Creating training & validation data

First train & test sets are created.

```
#create train & test sets
y <- census$incomeLevel
set.seed(1, sample.kind = "Rounding")
index <- createDataPartition(y, times = 1, p = 0.2, list = FALSE)
train_set <- census[-index, ]
test_set <- census[index, ]
```

Now that the data is ready, let's start modelling.

First, let's create a table to hold and compare our results from different models. The following 4 metrics are calculated and the models evaluated across these metrics to identify the best ones. 1) Overall accuracy 2) F1 score 3) Sensitivity 4) Specificity

```
#create table of results
results<-data.frame(Model=character(),
                    Accuracy=numeric(),
                    F1_score=numeric(),
                    Sensitivity=numeric(),
                    Specificity=numeric(),
                    stringsAsFactors = FALSE)
```

Logistic Regression

Given that the task is to predict a categorical variable using several other variables, logistic regression is a good model to build.

Model 1

The first model will try to predict income level using all factor variables, excluding native.country.

```
#run glm model on all factor variables
glm_fit1 <- train_set %>%
  glm(incomeLevel ~ age + race + sex + capitalMovement + hours.per.week +
        education.num + marital.status + relationship + workclass +
        occupation, data = ., family = "binomial")

#predict the cutoff probability value
p_hat_logit1 <- predict(glm_fit1, train_set, type = "response")

summary(p_hat_logit1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0164  0.1094  0.2408  0.3958  1.0000
```

The median 50% range for p_hat_logit1 is from 0.01 to 0.39.

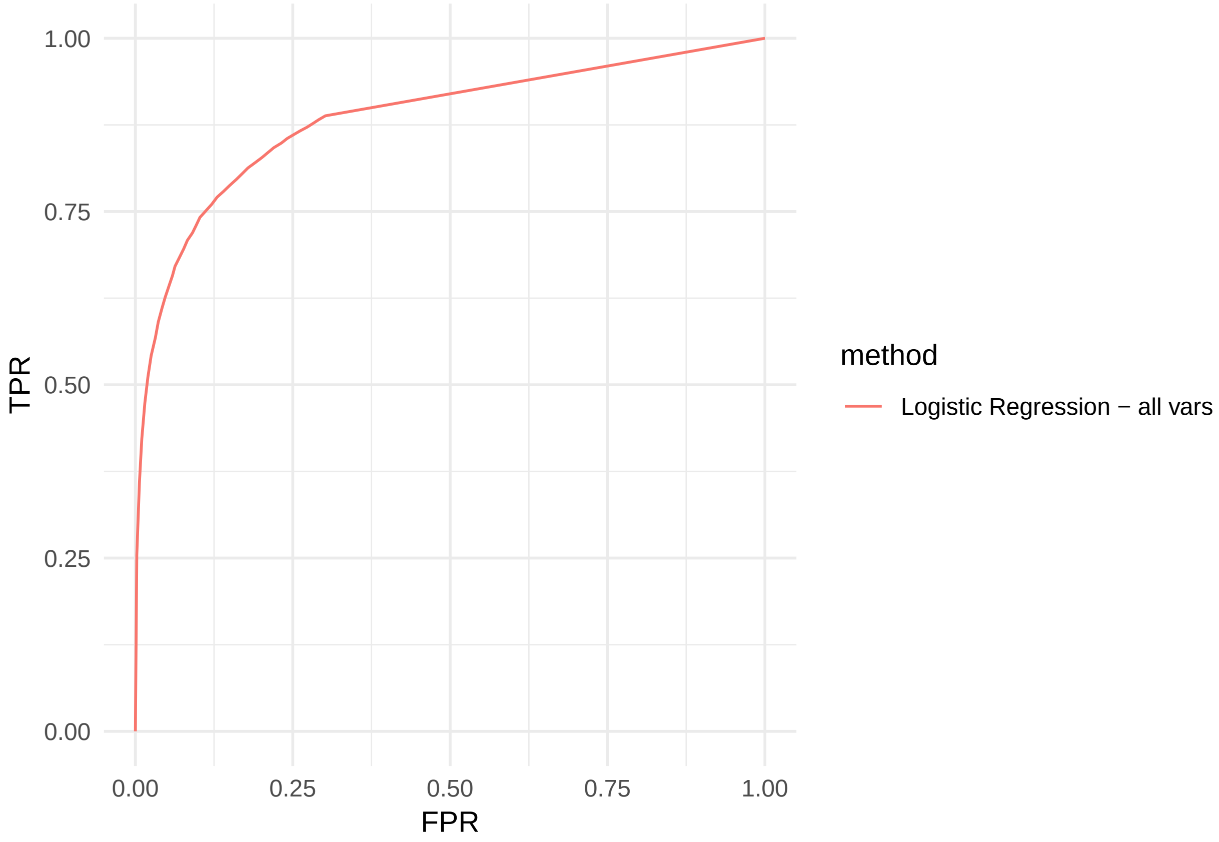
The model is now tuned to obtain the best value of p from the above range to make the prediction y_hat.

```
cutoffs <- c(0, seq(0.01,0.39,0.01), 1)

#tuning-calculate the FPR & TPR for each value of p
prob_cutoff1 <- map_df(cutoffs, function(x){
  y_hat_logit1 <- ifelse(p_hat_logit1 < x, "0", "1") %>% factor(levels = c("0","1"))
  list(method = "Logistic Regression - all vars",
       p = x,
       FPR = 1 - specificity(y_hat_logit1, train_set$incomeLevel),
       TPR = sensitivity(y_hat_logit1, train_set$incomeLevel))
})

# calculate distance of our model at each point from the ideal (0,1) point.
#the (0,1) point on the ROC curve indicates sensitivity & specificity = 1.
prob_cutoff1 <- mutate(prob_cutoff1, distance = sqrt((FPR-0)^2 + (TPR-1)^2))

#the best value of p is that for which this distance is the lowest.
best_prob1 <- prob_cutoff1$p[which.min(prob_cutoff1$distance)]
best_prob1
```

```
## [1] 0.27
```

p = 0.27 has the best performance of the model. Sensitivity & specificity are taken as equally important.

```
#generate ROC curve
prob_cutoff1 %>% ggplot(aes(FPR, TPR)) + geom_line(aes(col = method)) + theme_minimal()
```



Here is the ROC curve, which shows a reasonable fit.

```
#get prediction y_hat for y
y_hat_logit1 <- ifelse(p_hat_logit1 < best_prob1, "0", "1") %>% factor
```

This is the model prediction, let's now note the results.

```
#calculate various result metrics for this model based on the predicted value of y_hat
results[1, "Model"] <- "Logistic - All Vars"
results[1,"Accuracy"] <- confusionMatrix(y_hat_logit1, train_set$incomeLevel)$overall["Accuracy"]
results[1,"F1_score"] <- F_meas(y_hat_logit1, train_set$incomeLevel)
results[1, "Sensitivity"] <- sensitivity(y_hat_logit1, train_set$incomeLevel)
results[1, "Specificity"] <- specificity(y_hat_logit1, train_set$incomeLevel)
results %>% knitr::kable()
```

Model	Accuracy	F1_score	Sensitivity	Specificity
Logistic - All Vars	0.8149186	0.8696076	0.8129045	0.8212691

The model gives overall accuracy of 0.81 with sensitivity 0.81 and specificity 0.82. The model seems balanced overall, as reflected in the F1 score of 0.86.

Model 2

```
summary(glm_fit1)
```

```
##
## Call:
```

```
## glm(formula = incomeLevel ~ age + race + sex + capitalMovement + 
##     hours.per.week + education.num + marital.status + relationship + 
##     workclass + occupation, family = "binomial", data = .)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.9727  -0.5274  -0.1899  -0.0272   3.8260  
## 
## Coefficients: (1 not defined because of singularities)
##                                        Estimate Std. Error z value
## (Intercept)                          -9.785e+00  4.532e-01 -21.591
## age                                   2.629e-02  1.778e-03  14.785
## race Asian-Pac-Islander               3.680e-01  2.617e-01   1.406
## race Black                            2.556e-01  2.506e-01   1.020
## race Other                           -2.705e-01  3.760e-01  -0.719
## race White                            4.666e-01  2.389e-01   1.953
## sex Male                              8.791e-01  8.582e-02  10.243
## capitalMovement                       2.456e-04  9.469e-06  25.932
## hours.per.week                        3.090e-02  1.779e-03  17.372
## education.num                         2.895e-01  1.003e-02  28.877
## marital.status Married-AF-spouse      2.308e+00  6.193e-01   3.727
## marital.status Married-civ-spouse     1.820e+00  3.182e-01   5.721
## marital.status Married-spouse-absent  1.897e-01  2.412e-01   0.786
## marital.status Never-married         -3.616e-01  9.482e-02  -3.813
## marital.status Separated             -1.289e-01  1.802e-01  -0.716
## marital.status Widowed                9.729e-02  1.704e-01   0.571
## relationship Not-in-family            1.681e-01  3.149e-01   0.534
## relationship Other-relative          -9.414e-01  2.922e-01  -3.222
## relationship Own-child               -1.186e+00  3.208e-01  -3.696
## relationship Unmarried                3.029e-02  3.306e-01   0.092
## relationship Wife                     1.338e+00  1.119e-01  11.962
## workclass Federal-gov                 1.030e+00  1.690e-01   6.095
## workclass Local-gov                   3.531e-01  1.536e-01   2.298
## workclass Never-worked               -1.153e+01  4.461e+02  -0.026
## workclass Private                     5.004e-01  1.373e-01   3.643
## workclass Self-emp-inc                6.374e-01  1.642e-01   3.883
## workclass Self-emp-not-inc            6.746e-02  1.498e-01   0.450
## workclass State-gov                   1.572e-01  1.661e-01   0.946
## workclass Without-pay                -1.342e+01  3.698e+02  -0.036
## occupation Adm-clerical               1.121e-01  1.101e-01   1.018
## occupation Armed-Forces              -1.313e+01  5.111e+02  -0.026
## occupation Craft-repair               2.447e-01  9.477e-02   2.583
## occupation Exec-managerial            9.556e-01  9.677e-02   9.875
## occupation Farming-fishing           -9.503e-01  1.599e-01  -5.944
## occupation Handlers-cleaners         -6.125e-01  1.650e-01  -3.712
## occupation Machine-op-inspct         -1.574e-01  1.180e-01  -1.333
## occupation Other-service             -6.936e-01  1.366e-01  -5.078
## occupation Priv-house-serv           -1.218e+01  1.124e+02  -0.108
## occupation Prof-specialty             6.883e-01  1.030e-01   6.685
## occupation Protective-serv            6.534e-01  1.442e-01   4.530
## occupation Sales                      4.300e-01  1.002e-01   4.291
## occupation Tech-support               8.091e-01  1.320e-01   6.130
## occupation Transport-moving                  NA         NA      NA
##                                      Pr(>|z|)
## (Intercept)                          < 2e-16 ***
## age                                  < 2e-16 ***
## race Asian-Pac-Islander             0.159676    
## race Black                          0.307671    
## race Other                          0.472000    
## race White                          0.050832 .  
## sex Male                             < 2e-16 ***
## capitalMovement                      < 2e-16 ***
## hours.per.week                       < 2e-16 ***
## education.num                        < 2e-16 ***
## marital.status Married-AF-spouse     0.000194 ***
## marital.status Married-civ-spouse    1.06e-08 ***
## marital.status Married-spouse-absent 0.431641    
## marital.status Never-married         0.000137 ***
## marital.status Separated             0.474270    
## marital.status Widowed               0.568116    
## relationship Not-in-family           0.593578    
## relationship Other-relative          0.001274 ** 
## relationship Own-child               0.000219 ***
## relationship Unmarried               0.926990    
## relationship Wife                     < 2e-16 ***
## workclass Federal-gov                1.10e-09 ***
## workclass Local-gov                  0.021562 *  
## workclass Never-worked               0.979380    
## workclass Private                    0.000269 ***
## workclass Self-emp-inc               0.000103 ***
## workclass Self-emp-not-inc           0.652394    
## workclass State-gov                  0.343928    
## workclass Without-pay                0.971051    
## occupation Adm-clerical              0.308637    
## occupation Armed-Forces              0.979498    
## occupation Craft-repair              0.009806 ** 
## occupation Exec-managerial            < 2e-16 ***
## occupation Farming-fishing           2.78e-09 ***
## occupation Handlers-cleaners         0.000206 ***
## occupation Machine-op-inspct         0.182375    
## occupation Other-service             3.81e-07 ***
## occupation Priv-house-serv           0.913711    
## occupation Prof-specialty            2.31e-11 ***
## occupation Protective-serv           5.89e-06 ***
## occupation Sales                     1.78e-05 ***
## occupation Tech-support              8.79e-10 ***
## occupation Transport-moving                NA    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 28756  on 26047  degrees of freedom
## Residual deviance: 17047  on 26006  degrees of freedom
## AIC: 17131
## 
## Number of Fisher Scoring iterations: 14
```

race, education.nam are not significant variables since p-value for all the levels is >0.05. The next model can attempt to exclude these variables.

```r
#run glm on selected factor variables excluding those with high p-values
glm_fit2 <- train_set %>%
  glm(incomeLevel ~ age + sex + capitalMovement + hours.per.week +
        marital.status + relationship + workclass + occupation,
      data = ., family = "binomial")
p_hat_logit2 <- predict(glm_fit2, train_set, type = "response")
summary(p_hat_logit2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.02068 0.12649 0.24079 0.39427 1.00000
```

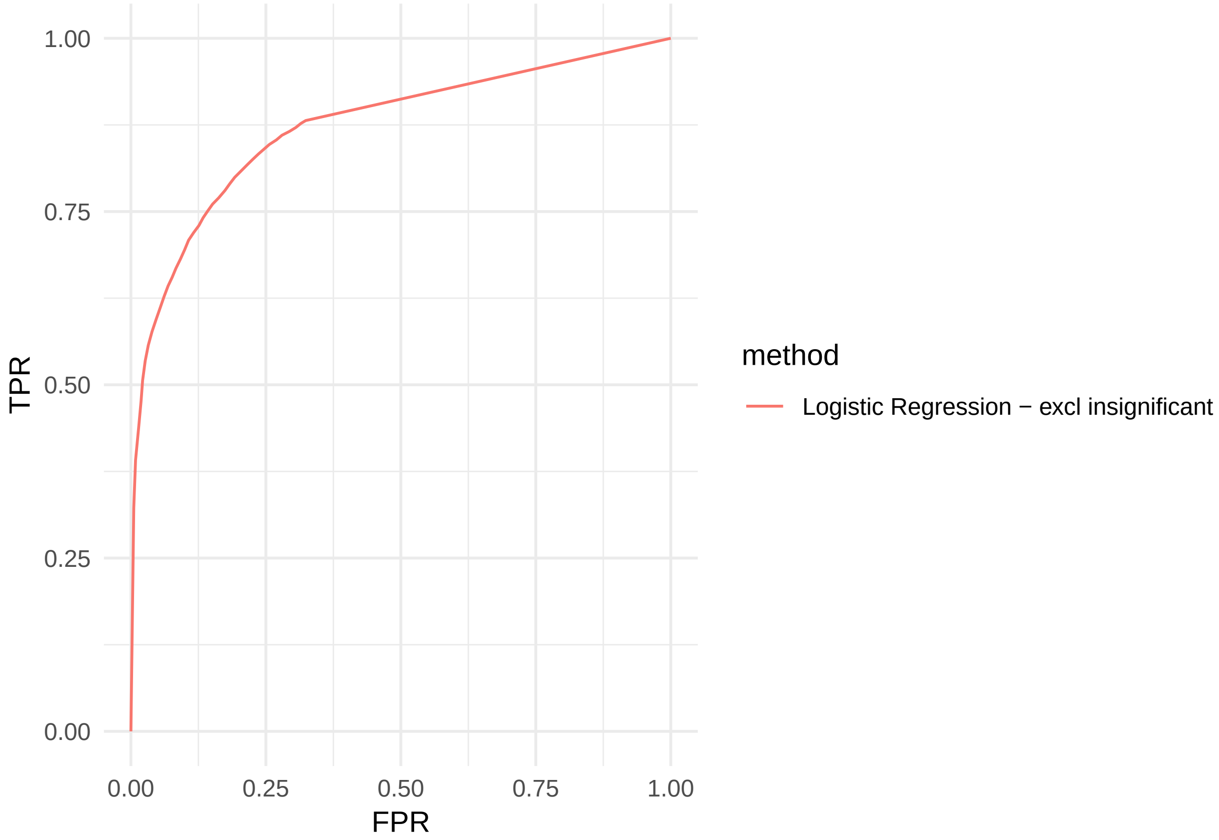
The median 50% range is from 0.02 to 0.39. The model is now tuned to obtain the best value of p from the above range to make the prediction y_hat.

```r
#tuning-calculate the FPR & TPR for each value of p
cutoffs <- c(0, seq(0.02,0.39,0.01), 1)
prob_cutoff2 <- map_df(cutoffs, function(x){
  y_hat_logit2 <- ifelse(p_hat_logit2 < x, "0", "1") %>% factor
  list(method = "Logistic Regression - excl insignificant",
       p = x,
       FPR = 1 - specificity(y_hat_logit2, train_set$incomeLevel),
       TPR = sensitivity(y_hat_logit2, train_set$incomeLevel))
})

#calculate distance from (0,1) and value of p which minimizes this
prob_cutoff2 <- mutate(prob_cutoff2, distance = sqrt((FPR-0)^2 + (TPR-1)^2))
best_prob2 <- prob_cutoff2$p[which.min(prob_cutoff2$distance)]
best_prob2
```

```
## [1] 0.27
```

p = 0.27 has the best performance of the model. Sensitivity & specificity are taken as equally important.

```r
#generate ROC curve
prob_cutoff2 %>% ggplot(aes(FPR, TPR)) + geom_line(aes(col = method)) + theme_minimal()
```



Here is the ROC curve, which shows a reasonable fit.

```
#get prediction
y_hat_logit2 <- ifelse(p_hat_logit2 < best_prob2, "0", "1") %>% factor
```

This is the prediction, let's now note the results.

```
#Note results
results[2, "Model"] <- "Logistic - excl insignificant"
results[2,"Accuracy"] <- confusionMatrix(y_hat_logit2, train_set$incomeLevel)$overall["Accuracy"]
results[2,"F1_score"] <- F_meas(y_hat_logit2, train_set$incomeLevel)
results[2, "Sensitivity"] <- sensitivity(y_hat_logit2, train_set$incomeLevel)
results[2, "Specificity"] <- specificity(y_hat_logit2, train_set$incomeLevel)
results %>% knitr::kable()
```

Model	Accuracy	F1_score	Sensitivity	Specificity
Logistic - All Vars	0.8149186	0.8696076	0.8129045	0.8212691
Logistic - excl insignificant	0.8013667	0.8593640	0.7993528	0.8077168

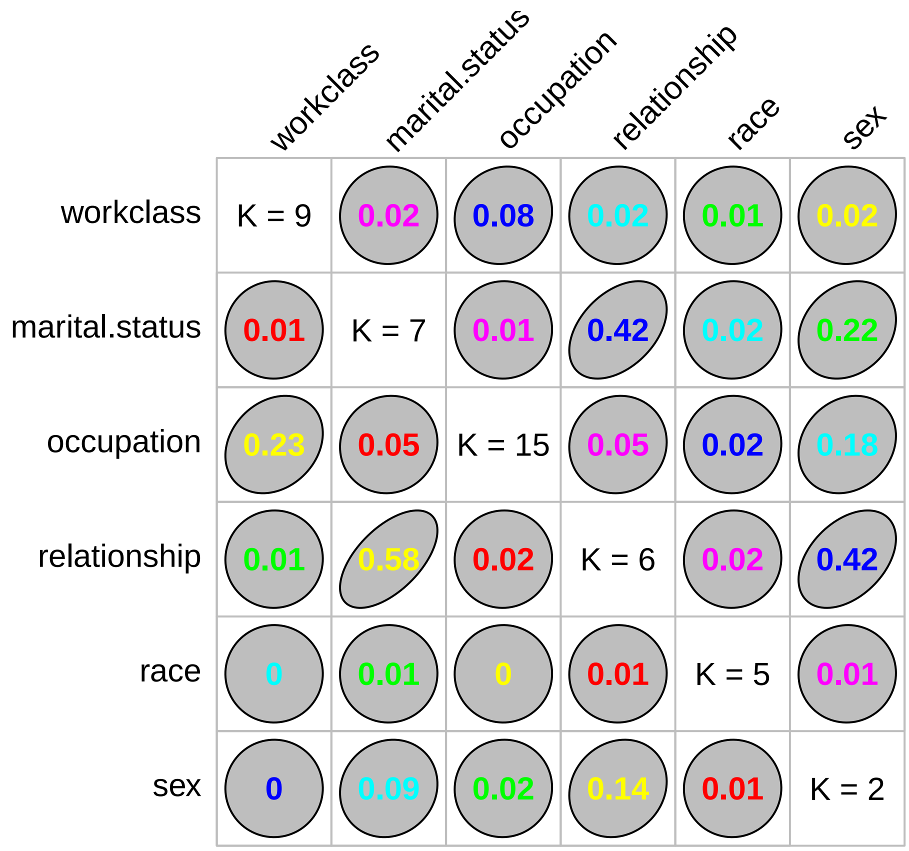
The results show that model 1 performs better across all 4 metrics. Thus, model 1 is taken as the base model for further analysis.

Model 3

Now, another attempt is made to improve the model by excluding highly correlated variables. There are both categorical & numerical variables, and hence different techniques will have to be employed.

First, let's find correlation between categorical variables using Goodman-Kruskal tau, which is a metric of the degree to which one variable can predict another variable.

```
#select categorical variables
train_set_categorical <- subset(train_set,select =
                                  c(workclass, marital.status, occupation,
                                    relationship, race, sex))

#calculate Goodman-Kruskal tau for correlation
plot(GKtauDataframe(train_set_categorical))
```



	workclass	marital.status	occupation	relationship	race	sex
workclass	K = 9	0.02	0.08	0.02	0.01	0.02
marital.status	0.01	K = 7	0.01	0.42	0.02	0.22
occupation	0.23	0.05	K = 15	0.05	0.02	0.18
relationship	0.01	0.58	0.02	K = 6	0.02	0.42
race	0	0.01	0	0.01	K = 5	0.01
sex	0	0.09	0.02	0.14	0.01	K = 2

The chart shows that relationship can predict marital status (high value of 0.58), hence marital status may be excluded. No other correlation is significant.

Next let's find correlation between numerical variables.

```
#select numerical variables
train_set_numerical <- subset(train_set,
                              select = c(age, hours.per.week, education.num, capitalMovement))
#calculate correlation
cor(train_set_numerical)
```

```
##                        age hours.per.week education.num capitalMovement
## age             1.00000000     0.06945440    0.03556986      0.07329989
## hours.per.week  0.06945440     1.00000000    0.14331175      0.07666785
## education.num   0.03556986     0.14331175    1.00000000      0.11467852
## capitalMovement 0.07329989     0.07666785    0.11467852      1.00000000
```

No 2 numeric variables are significantly correlated.

Next, let's find correlation between categorical & numerical variables using the Kruskal test.

```
#create a data frame for all combinations of categorical vs numerical variables
cat_vs_num <- as.data.frame(matrix(NA,nrow = 6, ncol = 4))
rownames(cat_vs_num) <-
  c("workclass", "marital.status", "occupation", "relationship",
    "race", "sex")
colnames(cat_vs_num) <- c("age", "hours.per.week", "education.num", "capitalMovement")

#perform Kruskal test on each combination.
for(i in 1:6){
  for(j in 1:4){
    x_var<-train_set[colnames(cat_vs_num)[j]][[1]]
    y_var<-train_set[rownames(cat_vs_num)[i]][[1]]
    cat_vs_num[i,j] <- kruskal.test(x=x_var,g=y_var)$p.value
  }
}
cat_vs_num
```

```
##                          age hours.per.week education.num capitalMovement
## workclass      2.452076e-265  1.828792e-271 9.860168e-210    3.204647e-07
## marital.status  0.000000e+00   0.000000e+00  3.680924e-65    1.887192e-16
## occupation     1.267726e-229   0.000000e+00  0.000000e+00    8.104840e-12
## relationship    0.000000e+00   0.000000e+00 5.414729e-136    2.362659e-18
## race            6.682217e-09   7.290679e-43  6.264067e-72    2.732553e-01
## sex             1.123391e-59   0.000000e+00  6.767702e-01    9.758914e-06
```

```
#gives that sex & education.num are correlated
```

sex & education.num are correlated.

```
#see variation of income with sex
sex_table<-table(train_set$sex, train_set$incomeLevel)
sex_table
```

```
##
##              0     1
##   Female  7709   924
##   Male   12067  5348
```

```
#see variation of income with each value of education.num
educationnum_table<-table(train_set$education.num, train_set$incomeLevel)
educationnum_total<-rowSums(educationnum_table)
educationnum_proportion<-educationnum_table/educationnum_total
educationnum_proportion
```

```
##
##                0          1
##   1  1.00000000 0.00000000
##   2  0.96183206 0.03816794
##   3  0.94909091 0.05090909
##   4  0.93904762 0.06095238
##   5  0.94117647 0.05882353
##   6  0.92923899 0.07076101
##   7  0.94838710 0.05161290
##   8  0.92211838 0.07788162
##   9  0.84105802 0.15894198
##   10 0.81057495 0.18942505
##   11 0.75227687 0.24772313
##   12 0.74942263 0.25057737
##   13 0.58259656 0.41740344
##   14 0.44083694 0.55916306
##   15 0.26373626 0.73626374
##   16 0.27743902 0.72256098
```

```
#there is a clear increase in 1's with increase in education.num
```

However, this data shows that sex, education.num are both significant and hence retained.

Thus, only marital.status is to be excluded from the third model. Now let's build this model.

```
#build 3rd glm model, excluding only marital.status
glm_fit3 <- train_set %>%
  glm(incomeLevel ~ age + race + sex + capitalMovement +
        hours.per.week + education.num + relationship +
        workclass + occupation, data = ., family = "binomial")
p_hat_logit3 <- predict(glm_fit3, train_set, type = "response")
summary(p_hat_logit3)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.01747 0.10969 0.24079 0.39553 1.00000
```

The median 50% range for p_hat_logit1 is from 0.01 to 0.39.

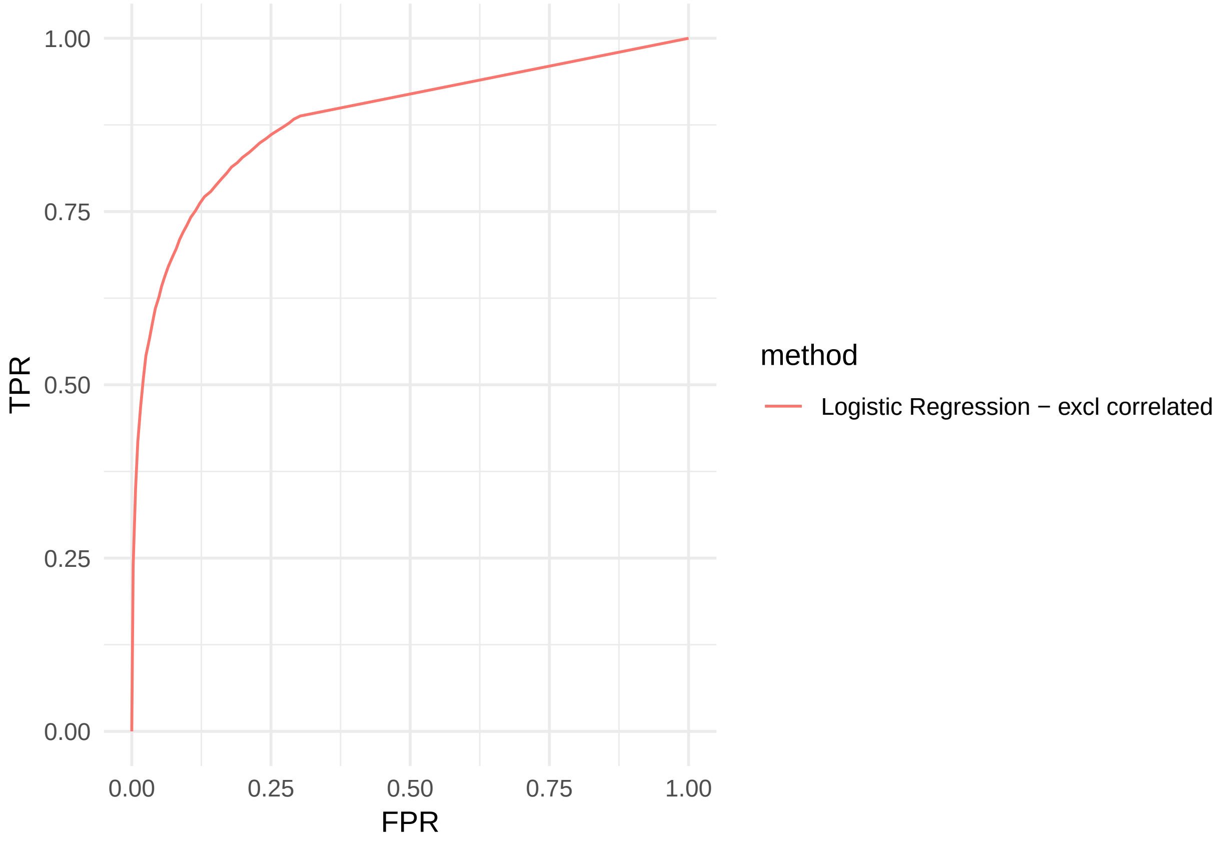
The model is now tuned to obtain the best value of p from the above range to make the prediction y_hat.

```
#tuning-calculate FPR & TPR, & p which gives lowest value of distance from (0,1)
cutoffs <- c(0, seq(0.01,0.39,0.01), 1)
prob_cutoff3 <- map_df(cutoffs, function(x){
  y_hat_logit3 <- ifelse(p_hat_logit3 < x, "0", "1") %>% factor
  list(method = "Logistic Regression - excl correlated",
       p = x,
       FPR = 1 - specificity(y_hat_logit3, train_set$incomeLevel),
       TPR = sensitivity(y_hat_logit3, train_set$incomeLevel))
})
prob_cutoff3 <- mutate(prob_cutoff3, distance = sqrt((FPR-0)^2 + (TPR-1)^2))
best_prob3 <- prob_cutoff3$p[which.min(prob_cutoff3$distance)]
best_prob3
```

```
## [1] 0.27
```

p = 0.27 has the best performance of the model. Sensitivity & specificity are taken as equally important.

```
#generate ROC curve
prob_cutoff3 %>% ggplot(aes(FPR, TPR)) + geom_line(aes(col = method)) + theme_minimal()
```



Here is the ROC curve, which shows a reasonable fit.

```
#get prediction
y_hat_logit3 <- ifelse(p_hat_logit3 < best_prob3, "0", "1") %>% factor
```

This is the prediction, let’s now note the results.

```
#note results
results[3, "Model"] <- "Logistic - excl correlated"
results[3,"Accuracy"] <- confusionMatrix(y_hat_logit3, train_set$incomeLevel)$overall["Accuracy"]
results[3,"F1_score"] <- F_meas(y_hat_logit3, train_set$incomeLevel)
results[3, "Sensitivity"] <- sensitivity(y_hat_logit3, train_set$incomeLevel)
results[3, "Specificity"] <- specificity(y_hat_logit3, train_set$incomeLevel)
results %>% knitr::kable()
```

Model	Accuracy	F1_score	Sensitivity	Specificity
Logistic - All Vars	0.8149186	0.8696076	0.8129045	0.8212691
Logistic - excl insignificant	0.8013667	0.8593640	0.7993528	0.8077168
Logistic - excl correlated	0.8157248	0.8702633	0.8140676	0.8209503

Thus, accuracy & F1 score is improved, while the model is more balanced between sensitivity & specificity. Hence, we will go ahead with model 3.

Final Logistic Regression Model

The final model is as below -

```r
#run the final glm model on the train set. This includes all factor variables except
#native.country & marital.status
glm_fit <- train_set %>%  glm(incomeLevel ~ age + race + sex + capitalMovement +
        hours.per.week + education.num + relationship +
        workclass + occupation, data = ., family = "binomial")

#Model applied on train set, and results noted.
p_hat_logit_train <- predict(glm_fit, train_set, type = "response")
y_hat_logit_train <- ifelse(p_hat_logit_train < 0.27, "0", "1") %>% factor
results[4, "Model"] <- "Logistic final - train set"
results[4,"Accuracy"] <- confusionMatrix(y_hat_logit_train, train_set$incomeLevel)$overall["Accuracy"]
results[4,"F1_score"] <- F_meas(y_hat_logit_train, train_set$incomeLevel)
results[4, "Sensitivity"] <- sensitivity(y_hat_logit_train, train_set$incomeLevel)
results[4, "Specificity"] <- specificity(y_hat_logit_train, train_set$incomeLevel)

#Model applied on test set, and results noted.
p_hat_logit_test <- predict(glm_fit, test_set, type = "response")
y_hat_logit_test <- ifelse(p_hat_logit_test < 0.27, "0", "1") %>% factor
results[5, "Model"] <- "Logistic final - test set"
results[5,"Accuracy"] <- confusionMatrix(y_hat_logit_test, test_set$incomeLevel)$overall["Accuracy"]
results[5,"F1_score"] <- F_meas(y_hat_logit_test, test_set$incomeLevel)
results[5, "Sensitivity"] <- sensitivity(y_hat_logit_test, test_set$incomeLevel)
results[5, "Specificity"] <- specificity(y_hat_logit_test, test_set$incomeLevel)

results %>% knitr::kable()
```

Model	Accuracy	F1_score	Sensitivity	Specificity
Logistic - All Vars	0.8149186	0.8696076	0.8129045	0.8212691
Logistic - excl insignificant	0.8013667	0.8593640	0.7993528	0.8077168
Logistic - excl correlated	0.8157248	0.8702633	0.8140676	0.8209503
Logistic final - train set	0.8157248	0.8702633	0.8140676	0.8209503
Logistic final - test set	0.8116076	0.8676233	0.8133091	0.8062460

We thus see that this model performs well, with sensitivity of 0.80, specificity of 0.81, overall accuracy of 0.80, and F1 score of 0.86. Also, the performance of the model is similar on train & test sets which indicates no overfitting.

KNN

Next, let's fit k-nearest neighbours to our data.

```r
set.seed(1, sample.kind = "Rounding")

#this fits knn to our data with all factor variables taken in the model
knn_fit <- knn3(incomeLevel ~ age + race + sex + occupation + workclass +
                  capitalMovement + hours.per.week + education.num +
                  marital.status + relationship, train_set, k = 5)

#calculate the prediction y_hat from this model
y_hat_knn <- predict(knn_fit, test_set, type = "class") %>% factor(levels = c("0", "1"))

#note the results from KNN
results[6, "Model"] <- "KNN"
```

```
results[6,"Accuracy"] <- confusionMatrix(y_hat_knn, test_set$incomeLevel)$overall["Accuracy"]
results[6,"F1_score"] <- F_meas(y_hat_knn, test_set$incomeLevel)
results[6, "Sensitivity"] <- sensitivity(y_hat_knn, test_set$incomeLevel)
results[6, "Specificity"] <- specificity(y_hat_knn, test_set$incomeLevel)
results %>% knitr::kable()
```

Model	Accuracy	F1_score	Sensitivity	Specificity
Logistic - All Vars	0.8149186	0.8696076	0.8129045	0.8212691
Logistic - excl insignificant	0.8013667	0.8593640	0.7993528	0.8077168
Logistic - excl correlated	0.8157248	0.8702633	0.8140676	0.8209503
Logistic final - train set	0.8157248	0.8702633	0.8140676	0.8209503
Logistic final - test set	0.8116076	0.8676233	0.8133091	0.8062460
KNN	0.8512206	0.9035149	0.9176780	0.6418101

Thus, the KNN model which is forecasting income level on all other factors gives 0.91 sensitivity & 0.64 specificity, with 0.85 overall accuracy and 0.90 F1 score.

Now, let's tune for k. However, we will require numeric variables to calculate distance. age, capitalMovement, hours.per.week are already numeric race, sex, relationship can be converted to numeric as below. occupation, workclass, marital.status have no logical way to convert to numeric & hence are dropped for tuning.

```
#convert to numeric variables
train_set <- mutate(train_set,
                    raceNum = ifelse(race == "White", 0, 1),
                    sexNum = ifelse(sex == "Female", 0, 1),
                    relationshipNum = ifelse(relationship %in% c("Husband", "Wife"), 1, 0))
col_index_knn <- which(colnames(train_set) %in%
                         c("age","capitalMovement","hours.per.week","education.num",
                           "raceNum", "sexNum", "relationshipNum"))
```

Now, let's tune on this data.

```
#create control for 10-fold cross-validation
control_knn <- trainControl(method = "cv",number = 10, p= 0.9)

#create index to sample 5000 observations with each run to make it faster
n <- 5000
index <- sample(nrow(train_set), n)
set.seed(1, sample.kind = "Rounding")

#default value of k is 5. Try values 3, 5, 7, 9 to see which works best.
knn_train <- train(train_set[index,col_index_knn],train_set$incomeLevel[index],
                   method = "knn", tuneGrid = data.frame(k = c(3,5,7,9)), trControl = control_knn)
knn_train$bestTune
```

```
##   k
## 4 9
```

Thus, k=9 gives the best fit. Let's now optimize the final model.

```
#build second knn model with k=9
knn_fit2 <- knn3(incomeLevel ~ age + race + sex + occupation + workclass +
                   capitalMovement + hours.per.week + education.num +
                   marital.status + relationship, train_set, k = 9)
```

```
#generate prediction for y_hat with this model
y_hat_knn2 <- predict(knn_fit2, test_set, type = "class") %>% factor(levels = c("0", "1"))

#note results from this model
results[7, "Model"] <- "KNN - tuned"
results[7,"Accuracy"] <- confusionMatrix(y_hat_knn2, test_set$incomeLevel)$overall["Accuracy"]
results[7,"F1_score"] <- F_meas(y_hat_knn2, test_set$incomeLevel)
results[7, "Sensitivity"] <- sensitivity(y_hat_knn2, test_set$incomeLevel)
results[7, "Specificity"] <- specificity(y_hat_knn2, test_set$incomeLevel)
results %>% knitr::kable()
```

Model	Accuracy	F1_score	Sensitivity	Specificity
Logistic - All Vars	0.8149186	0.8696076	0.8129045	0.8212691
Logistic - excl insignificant	0.8013667	0.8593640	0.7993528	0.8077168
Logistic - excl correlated	0.8157248	0.8702633	0.8140676	0.8209503
Logistic final - train set	0.8157248	0.8702633	0.8140676	0.8209503
Logistic final - test set	0.8116076	0.8676233	0.8133091	0.8062460
KNN	0.8512206	0.9035149	0.9176780	0.6418101
KNN - tuned	0.8556733	0.9066349	0.9231392	0.6430848

This KNN model gives 0.92 sensitivity and 0.64 specificity, with overall accuracy of 0.855 and F1 score of 0.906. The performance of KNN is thus marginally improved with tuning.

Random Forest

Now, let's fit random forest to our data.

```
set.seed(1, sample.kind = "Rounding")

#fit random forest to all variables
rf_fit <- randomForest(incomeLevel~age + race + sex + occupation +
                         workclass + capitalMovement + hours.per.week+education.num +
                         marital.status + relationship, data = train_set)

#calculate prediction y_hat from this model
y_hat_rf <- predict(rf_fit, test_set)

#note results from this model
results[8, "Model"] <- "Random Forest"
results[8,"Accuracy"] <- confusionMatrix(y_hat_rf, test_set$incomeLevel)$overall["Accuracy"]
results[8,"F1_score"] <- F_meas(y_hat_rf, test_set$incomeLevel)
results[8, "Sensitivity"] <- sensitivity(y_hat_rf, test_set$incomeLevel)
results[8, "Specificity"] <- specificity(y_hat_rf, test_set$incomeLevel)
results %>% knitr::kable()
```

Model	Accuracy	F1_score	Sensitivity	Specificity
Logistic - All Vars	0.8149186	0.8696076	0.8129045	0.8212691
Logistic - excl insignificant	0.8013667	0.8593640	0.7993528	0.8077168
Logistic - excl correlated	0.8157248	0.8702633	0.8140676	0.8209503
Logistic final - train set	0.8157248	0.8702633	0.8140676	0.8209503
Logistic final - test set	0.8116076	0.8676233	0.8133091	0.8062460
KNN	0.8512206	0.9035149	0.9176780	0.6418101

Model	Accuracy	F1_score	Sensitivity	Specificity
KNN - tuned	0.8556733	0.9066349	0.9231392	0.6430848
Random Forest	0.8687241	0.9154888	0.9366909	0.6545570

This random forest model gives a sensitivity of 0.936, specificity of 0.654, accuracy of 0.868, and F1 score of 0.915.

Let's also try Rborist which runs faster.

```
#run Rborist on the data
col_index <- which(colnames(train_set) %in%
                     c("age","race","sex","occupation","workclass","capitalMovement",
                       "hours.per.week","education.num", "relationship", "marital.status"))
set.seed(1, sample.kind = "Rounding")
rb_fit <- Rborist(train_set[, col_index], train_set$incomeLevel)

#note predictions from Rborist
y_hat_rb <- predict(rb_fit, test_set[, col_index])$yPred %>% factor(levels = c("0","1"))

#note results from this model
results[9, "Model"] <- "Rborist"
results[9,"Accuracy"] <- confusionMatrix(y_hat_rb, test_set$incomeLevel)$overall["Accuracy"]
results[9,"F1_score"] <- F_meas(y_hat_rb, test_set$incomeLevel)
results[9, "Sensitivity"] <- sensitivity(y_hat_rb, test_set$incomeLevel)
results[9, "Specificity"] <- specificity(y_hat_rb, test_set$incomeLevel)
results %>% knitr::kable()
```

Model	Accuracy	F1_score	Sensitivity	Specificity
Logistic - All Vars	0.8149186	0.8696076	0.8129045	0.8212691
Logistic - excl insignificant	0.8013667	0.8593640	0.7993528	0.8077168
Logistic - excl correlated	0.8157248	0.8702633	0.8140676	0.8209503
Logistic final - train set	0.8157248	0.8702633	0.8140676	0.8209503
Logistic final - test set	0.8116076	0.8676233	0.8133091	0.8062460
KNN	0.8512206	0.9035149	0.9176780	0.6418101
KNN - tuned	0.8556733	0.9066349	0.9231392	0.6430848
Random Forest	0.8687241	0.9154888	0.9366909	0.6545570
Rborist	0.8289575	0.8924087	0.9344660	0.4964946

However, this gives sensitivity of 0.934, specificity of 0.496, accuracy of 0.828, F1 score of 0.892. randomForest thus performs much better and we tune that model.

Let's setup the tuning parameters.

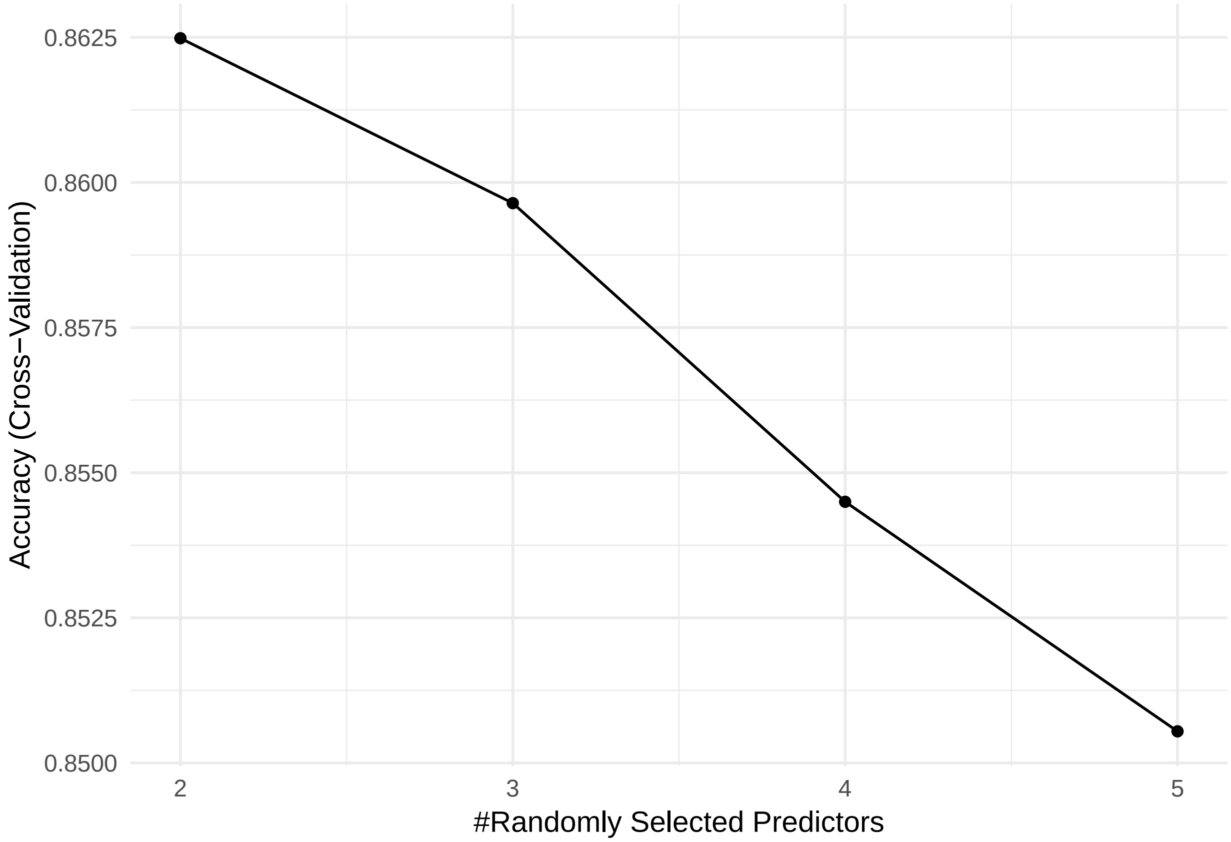
```
control_rf <- trainControl(method = "cv",number = 5, p= 0.8)
#we use 5-fold cross validation to reduce time taken
```

```
grid <- expand.grid(mtry = c(2, 3, 4, 5))
#mtry default is sqrt(variables) = ~3.2, hence testing 2, 3, 4, 5
```

Now let's tune the model.

```
#tune for different values of mtry, with 5000 random observations sampled, and 50 trees
set.seed(1, sample.kind = "Rounding")
rf_train <- train(train_set[, col_index], train_set$incomeLevel,
```

```r
                   method = "rf",
                   nTree = 50,
                   trControl = control_rf, tuneGrid = grid, nSamp = 5000)
ggplot(rf_train) + theme_minimal()
```



```r
rf_train$bestTune
```

```
##   mtry
## 1    2
```

Thus, the best value of mtry is 2. Let's now optimize the final model.

```r
set.seed(1, sample.kind = "Rounding")
#build final model with mtry =2
rf_fit2 <- randomForest(incomeLevel~age + race + sex + occupation +
                         workclass + capitalMovement + hours.per.week+education.num +
                         marital.status + relationship, mtry = 2, data = train_set)

#note predictions from this model
y_hat_rf2 <- predict(rf_fit2, test_set) %>% factor(levels = c("0", "1"))

#note results
results[10, "Model"] <- "Random Forest - tuned"
results[10,"Accuracy"] <- confusionMatrix(y_hat_rf2, test_set$incomeLevel)$overall["Accuracy"]
results[10,"F1_score"] <- F_meas(y_hat_rf2, test_set$incomeLevel)
results[10, "Sensitivity"] <- sensitivity(y_hat_rf2, test_set$incomeLevel)
results[10, "Specificity"] <- specificity(y_hat_rf2, test_set$incomeLevel)
```

```
results %>% knitr::kable()
```

Model	Accuracy	F1_score	Sensitivity	Specificity
Logistic - All Vars	0.8149186	0.8696076	0.8129045	0.8212691
Logistic - excl insignificant	0.8013667	0.8593640	0.7993528	0.8077168
Logistic - excl correlated	0.8157248	0.8702633	0.8140676	0.8209503
Logistic final - train set	0.8157248	0.8702633	0.8140676	0.8209503
Logistic final - test set	0.8116076	0.8676233	0.8133091	0.8062460
KNN	0.8512206	0.9035149	0.9176780	0.6418101
KNN - tuned	0.8556733	0.9066349	0.9231392	0.6430848
Random Forest	0.8687241	0.9154888	0.9366909	0.6545570
Rborist	0.8289575	0.8924087	0.9344660	0.4964946
Random Forest - tuned	0.8665745	0.9147789	0.9433657	0.6246017

We try increasing nTree to further improve the performance of the model, but on manually trying a few values we see that the performance doesn't improve. This value is what we get with 100 trees.

```
#the same exercise as above, with 100 trees specified
set.seed(1, sample.kind = "Rounding")
rf_fit3 <- randomForest(incomeLevel~age + race + sex + occupation +
                        workclass + capitalMovement + hours.per.week+education.num +
                        marital.status + relationship, mtry = 2, nTree = 100, data = train_set)
y_hat_rf3 <- predict(rf_fit3, test_set) %>% factor(levels = c("0", "1"))
results[11, "Model"] <- "Random Forest - final"
results[11,"Accuracy"] <- confusionMatrix(y_hat_rf3, test_set$incomeLevel)$overall["Accuracy"]
results[11,"F1_score"] <- F_meas(y_hat_rf3, test_set$incomeLevel)
results[11, "Sensitivity"] <- sensitivity(y_hat_rf3, test_set$incomeLevel)
results[11, "Specificity"] <- specificity(y_hat_rf3, test_set$incomeLevel)
results %>% knitr::kable()
```

Model	Accuracy	F1_score	Sensitivity	Specificity
Logistic - All Vars	0.8149186	0.8696076	0.8129045	0.8212691
Logistic - excl insignificant	0.8013667	0.8593640	0.7993528	0.8077168
Logistic - excl correlated	0.8157248	0.8702633	0.8140676	0.8209503
Logistic final - train set	0.8157248	0.8702633	0.8140676	0.8209503
Logistic final - test set	0.8116076	0.8676233	0.8133091	0.8062460
KNN	0.8512206	0.9035149	0.9176780	0.6418101
KNN - tuned	0.8556733	0.9066349	0.9231392	0.6430848
Random Forest	0.8687241	0.9154888	0.9366909	0.6545570
Rborist	0.8289575	0.8924087	0.9344660	0.4964946
Random Forest - tuned	0.8665745	0.9147789	0.9433657	0.6246017
Random Forest - final	0.8665745	0.9147789	0.9433657	0.6246017

The original model is thus taken as the final one.

Results

Let's now look at the final results produced by the 3 models to judge which is the best. These are 1) Logistic final - test set 2) KNN - tuned 3) Random Forest - final in rows 5, 7, 11 of the results data table.

```r
results[c(5,7,11), ] %>% knitr::kable()
```

	Model	Accuracy	F1_score	Sensitivity	Specificity
5	Logistic final - test set	0.8116076	0.8676233	0.8133091	0.8062460
7	KNN - tuned	0.8556733	0.9066349	0.9231392	0.6430848
11	Random Forest - final	0.8665745	0.9147789	0.9433657	0.6246017

On comparing KNN & Random Forest, it becomes clear that Random Forest is performing better. Random forest gives better sensitivity (0.943 vs 0.923) but worse specificity (0.624 vs 0.643). Overall accuracy is much improved (0.866 vs 0.855) and F1 score is better (0.914 vs 0.906).

The comparison between logistic regression & random forest is not as straightforward. Random forest has significantly better accuracy (0.866 vs 0.807) and F1 score (0.914 vs 0.864). However, this is coming at the cost of a gap between sensitivity & specificity. Logistic regression is balanced with sensitivity 0.805 and specificity 0.814. However, while Random forest's sensitivity is 0.943, it's specificity is poor at 0.624.

Let's examine the confusion matrix to see this better.

```r
confusionMatrix(y_hat_logit_test, test_set$incomeLevel)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 4021  304
##          1  923 1265
##
##                Accuracy : 0.8116
##                  95% CI : (0.8019, 0.821)
##     No Information Rate : 0.7591
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.546
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.8133
##             Specificity : 0.8062
##          Pos Pred Value : 0.9297
##          Neg Pred Value : 0.5782
##              Prevalence : 0.7591
##          Detection Rate : 0.6174
##    Detection Prevalence : 0.6641
##       Balanced Accuracy : 0.8098
##
##        'Positive' Class : 0
##
```

```r
confusionMatrix(y_hat_rf2, test_set$incomeLevel)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 4664  589
##          1  280  980
##                                           
##                Accuracy : 0.8666          
##                  95% CI : (0.8581, 0.8747)
##     No Information Rate : 0.7591          
##     P-Value [Acc > NIR] : < 2.2e-16       
##                                           
##                   Kappa : 0.6089          
##                                           
##  Mcnemar's Test P-Value : < 2.2e-16       
##                                           
##             Sensitivity : 0.9434          
##             Specificity : 0.6246          
##          Pos Pred Value : 0.8879          
##          Neg Pred Value : 0.7778          
##              Prevalence : 0.7591          
##          Detection Rate : 0.7161          
##    Detection Prevalence : 0.8065          
##       Balanced Accuracy : 0.7840          
##                                           
##        'Positive' Class : 0               
##
```

Prevalence is skewed in the data, with "1" having a prevalence of 24%. Sensitivity & specificity are taken as equally important, since the task is overall prediction.

Given that the task is to correctly predict whether an individual makes >50K, random forest with its higher overall accuracy is the best performing model. However, KNN performs reasonably well also, and logistic regression while having lower accuracy is more balanced between true positives and false negatives.

There were also several interesting results obtained from the exploratory data analysis, which are detailed there and hence not repeated in this section.

Conclusions

The adult income dataset offers an opportunity to understand the relationship of several factors specific to an individual to his/her income, as well as an opportunity to build a model using these factors that allows one to predict the income using these factors. The binary nature of the prediction variable income is a limiting factor, since we don't have information on the actual income level beyond it being lesser or more than $50K. At the same time, it simplifies the exercise and allows us to gain key observations.

The conclusions from the analysis are presented below.

1) The dataset contains census data for a set of 32000 individuals in the United States. Assuming that this sample set is random, 24% of individuals in the United States have income greater than $50,000. $50,000 may thus be interpreted as an income level demarcating the population into 2 parts, 'high income' and 'low income', in a 1:3 ratio - with high income being characterized as those with income >$50,000. The entire exercise may thus be interpreted as identifying factors that lead to individuals having high incomes and developing a model to predict whether an individual has a high or low income.

2) There are 11 relevant pieces of information provided about each individual, relating to his/her demographic background, education, employment, relationships, and investments. Each of them may have some bearing on his/her income.

i) Age
ii) Race

iii) Sex
iv) Native Country
v) Working class
vi) Occupation
vii) Education level
viii) Hours worked per week
ix) Relationship status
x) Marital status
xi) Capital gain/loss information

3) Age has a clear association with income level, with ages 40-60 having 40% 'high income' people. This is supported by conventional wisdom that the peak earning potential of an individual is when he/she is towards the later stages of his/her career, while retirement at some point post the age of 60 leads to depressed income.

4) Race has some association with income level, with Whites and Asians having a higher proportion of 'high income' people (26-27%) vs others (9-12%). This may or may not suggest racial discrimination; we saw, for example, that Whites & Asians are also the most educated.

5) Sex has a strong association with income level, with 30% of men but only 10% of women having incomes > $50,000. In general, it is assumed that men & women have similar capabilities. It is also assumed that the men & women in the data will have similar backgrounds. With these assumptions, the data appears to point to a clear 'gender wage gap' as often reported in public media.

6) Being in incorporated self-employment - i.e. having an established own business - significantly increases one's chances of having a high income to over 50%. This is supported by popular wisdom as well. Interestingly, working for federal government offers the next-best chances of having high income (37%). Working for the private sector - as done by 70% of the population - offers a slightly lower-than-average chance of having high income (21%).

7) Management executives and professors (interpreting prof-specialty as professors) have the highest income levels (40-50% are high income), while those engaged in agricultural, cleaning, housework, or armed forces have very few 'high income' individuals (<15%).

8) Education is strongly associated with income level, with the logical interpretation being that higher education is leading to higher income levels. There are several interesting conclusions here -

- individuals who are not high school graduates tend to have a very low chance of high income (<10%). The level of schooling completed does not appear to affect this.
- individuals who have completed high school but not a college degree have a 15-25% chance of high income
- post high-school, each additional degree - bachelors, masters, and doctorate - adds 15-20% to the chances that an individual is high income. This is very strong evidence for additional degrees leading to higher earning potential.

9) Hours worked per week is also strongly associated with, and likely causing, higher income levels. Working 40 hours a week gives one a median chance - 21% - of being high income. Working more doubles that probability to 40%, while working less halves the probability to 10%.

10) Being in a marital relationship is also strongly associated with, and likely causing, higher income levels. Individuals who are married have a 40-50% chance of high income while those who are not have a <10% chance.

11) Capital gains and losses are both associated with higher income. However, it is quite likely that higher incomes are causing capital gains & losses instead. It is likely that those with incomes >$50,000 engage more in investing and thus have capital gains or losses, while those with low incomes do not.

12) Logistic regression, k-nearest neighbours, and random forest all allow us to predict an individual's income level with >80% accuracy, with random forest giving the highest accuracy of over 86%.The

random forest model described here is therefore the best model to predict an individual's income level.

13) This modelling exercise indicates that the collection of an individual's basic information can be used to identify an individual's income level with a great degree of accuracy. This implies that -

- an individual's income level is influenced significantly by these ~10 factors
- an individual's income level may not be influenced to a significant extent by factors taken as conventionally important (e.g. each individual's relationship with his/her manager).

14) While some of the factors used are demographic and cannot be changed - e.g. race, sex, native country, age - several of these can be taken as within individual control - e.g. education, employment, hours worked, and relationship. The relationships with these factors imply that an individual can increase his chances of having a high income by seeking higher education, employment of a certain type, working more hours, and being in a stable relationship personally.

15) However, it is also important to note that several of these factors are decided at an early age - education, employment - and are difficult to change later. This indicates that an individual's earning potential may be decided to a significant extent by his/her decisions/actions till age 30.