Hyatt Census Case Study

Drew Wilimitis

October 16th, 2018

Scenario

Hyatt's marketing team is working on an email campaign and wants to be able to target customers based on their income. The training data consists of demographic data as well as income levels for those individuals. For new customers in the test set, predict the income level for each customer.

Initial Setup

We start by loading the required libraries and initializing the RMarkdown default settings

```
# Importing libraries
library(MASS)
library(GGally)
library(openintro)
library(mosaic)
library(knitr)
library(tidyverse)
library(ggformula)
library(gridExtra)
library(broom)
library(readr)
library(lubridate)
library(dplyr)
library(stringr)
library(ggplot2)
library(plotly)
library(xtable)
library(readxl)

# RMarkdown settings
options(width=70, digits=4, scipen=8)

# Set the default for displaying code and warnings
opts_chunk$set(echo = TRUE)
opts_chunk$set(message = FALSE)
opts_chunk$set(warning = FALSE)
```

Next we read in the Census Data

```
# Clear the workspace
rm(list = ls())
gc()

# Loading data
train_data <- read_excel("C:/Users/Drew/Desktop/Hyatt/censusTrain.xlsx")
test_data <- read_excel("C:/Users/Drew/Desktop/Hyatt/censusTest.xlsx")

# Convert to data frame
train_data <- as.data.frame(train_data)
test_data <- as.data.frame(test_data)
```

Exploratory Analysis and Data Prep

```r
# View first few rows
head(train_data)
```

```
##   id age        work_class fnlwgt education education_num
## 1  1  39         State-gov  77516 Bachelors            13
## 2  2  50  Self-emp-not-inc  83311 Bachelors            13
## 3  3  38           Private 215646   HS-grad             9
## 4  4  53           Private 234721      11th             7
## 5  5  28           Private 338409 Bachelors            13
## 6  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
# View structure and data types
str(train_data)
```

```
## 'data.frame':    32561 obs. of  16 variables:
##  $ id            : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ age           : num  39 50 38 53 28 37 49 52 31 42 ...
##  $ work_class    : chr  "State-gov" "Self-emp-not-inc" "Private" "Private" ...
##  $ fnlwgt        : num  77516 83311 215646 234721 338409 ...
##  $ education     : chr  "Bachelors" "Bachelors" "HS-grad" "11th" ...
##  $ education_num : num  13 13 9 7 13 14 5 9 14 13 ...
##  $ marital_status: chr  "Never-married" "Married-civ-spouse" "Divorced" "Married-civ-spouse" ...
##  $ occupation    : chr  "Adm-clerical" "Exec-managerial" "Handlers-cleaners" "Handlers-cleaners" ...
##  $ relationship  : chr  "Not-in-family" "Husband" "Not-in-family" "Husband" ...
##  $ race          : chr  "White" "White" "White" "Black" ...
##  $ sex           : chr  "Male" "Male" "Male" "Male" ...
##  $ capital_gain  : num  2174 0 0 0 0 ...
##  $ capital_loss  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ hours_per_week: num  40 13 40 40 40 40 16 45 50 40 ...
##  $ native_country: chr  "United-States" "United-States" "United-States" "United-States" ...
##  $ income        : chr  "<=50K" "<=50K" "<=50K" "<=50K" ...
```

Data Quality

Check for NA values, duplicates, and delete unnecessary columns

We remove NA values in the training data to maintain an accurate prediction model

We impute for NA values in the test data so that we can provide predictions for every test observation

```
# Check for duplicates
anyDuplicated(train_data)
anyDuplicated(test_data)

# Drop unnecessary 'id' column
train_data <- train_data[, 2:16]
test_data <- test_data[, 2:15]
```

```
# Check for NA values
colnames(train_data)[colSums(is.na(train_data)) > 0]
```

```
## [1] "work_class"     "occupation"     "native_country"
```

```
colnames(test_data)[colSums(is.na(test_data)) > 0]
```

```
## [1] "work_class"     "occupation"     "native_country"
```

```
# remove NA values for training data
train_data <- na.omit(train_data)
```

Understanding Input and Target Variables

14 predictor variables and 1 response variable

predictor variables

1. **age** - Age of individual in years
2. **work_class** - Individual's working class (State-gov, Federal-gov, Private, etc.)
3. **fnlwgt** - Final sampling weight, corrects for under/over representation in sample
4. **education** - Education level as character (HS-grad, Bachelors, Masters, Doctorate, etc.)
5. **education_num** - Numerical value for number of education years
6. **marital_status** - Character such as Never-married, Divorced, Widowed, etc.
7. **occupation** - Character such as Sales, Tech-support, Exec-managerial, etc.
8. **relationship** - Relationship status (Husband, Wife, Unmarried, Own-child)
9. **race** - Race (White, Black, Asian-Pac-Islander, Amer-Indian-Eskimo, other)
10. **sex** - Male or Female
11. **capital_gain** - Capital gains as pos. number (profit from sale of property or investment)
12. **capital_loss** - Capital losses as pos. number (loss from sale of property or investment)
13. **hours_per_week** - Number of hours worked per week, numerical
14. **native_country** - Native country of individual (United-States, Mexico, China, etc.)

target/response variable

1. **income** - Annual income level as a character, either "<=50k" or '>50k'

Two income classes as the target variable -> Binary Classification

Response Variable - Income

Summary of income level proportions

```
prop.table(table(train_data$income))
```

```
##
##  <=50K   >50K
## 0.7511 0.2489
```

This shows that the % of people earning less than 50K is 75.1% and the % of people earning more than 50k is 24.9%

For this binary classification there is an imbalance between the two classes

age variable

Age summary statistics

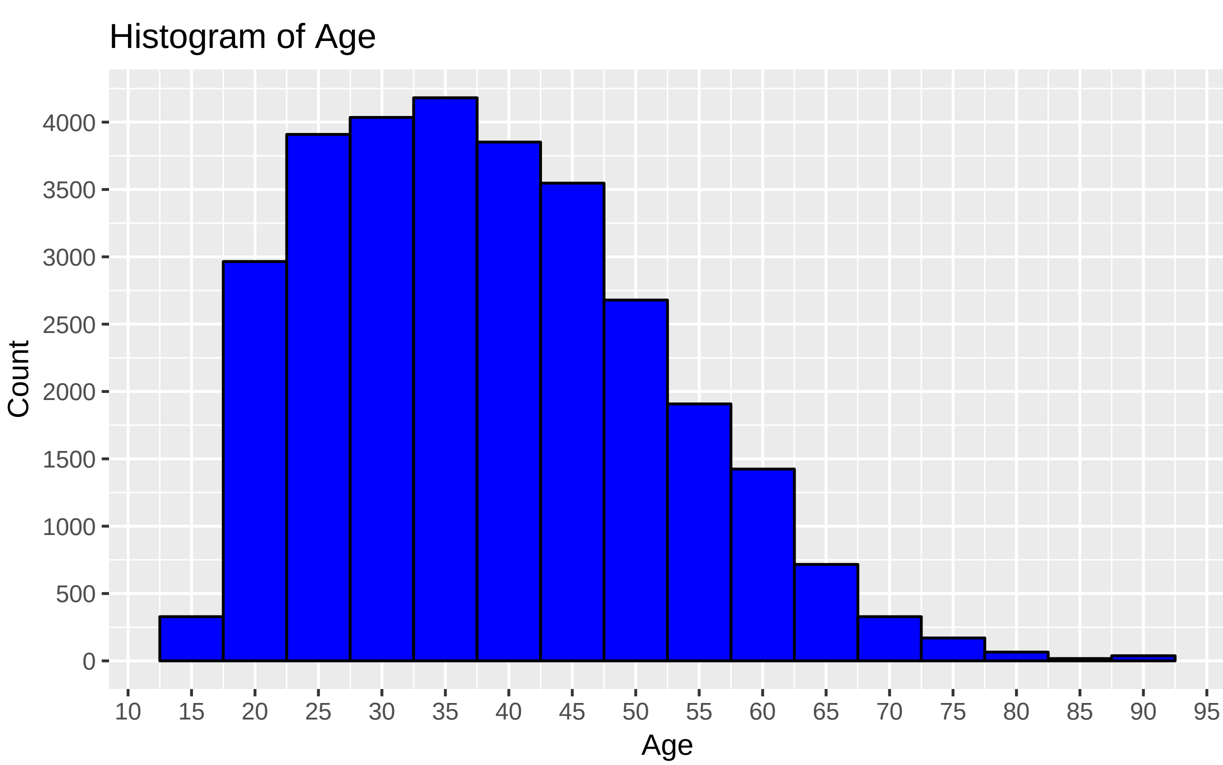
```
summary(train_data$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    17.0    28.0    37.0    38.4    47.0    90.0
```

This shows around 50% of the people are between age 28 and 47 years old.

Visualizing the distribution of age

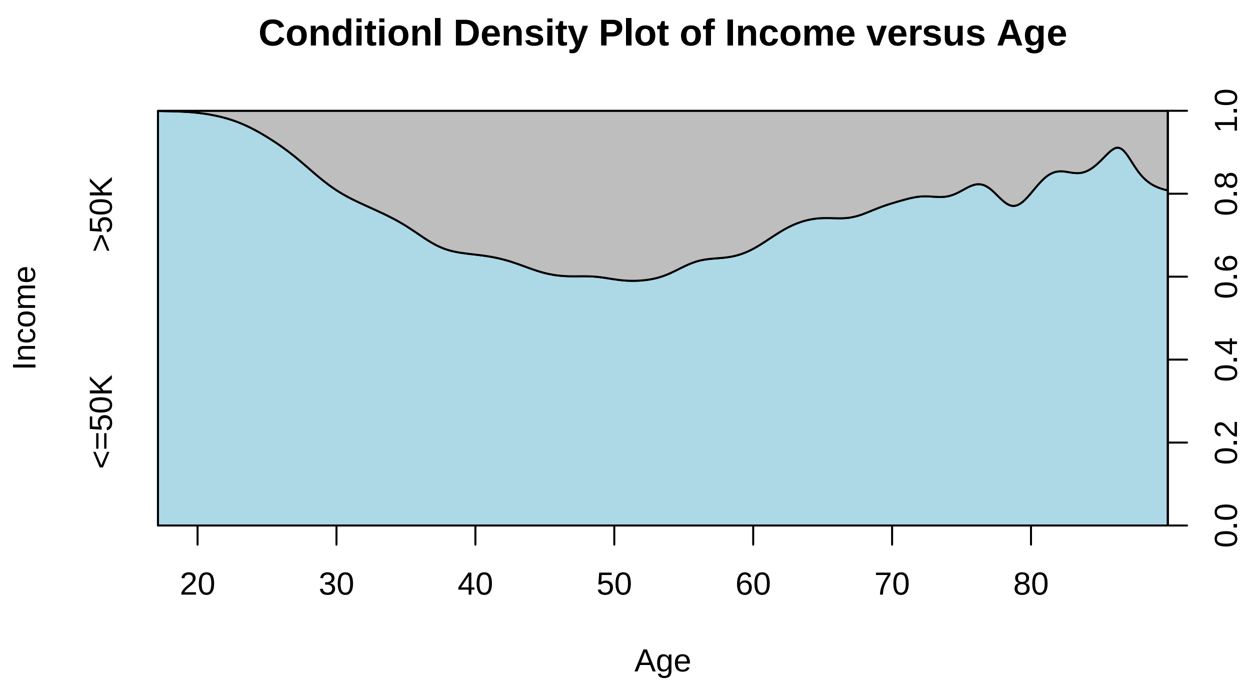
```
# histogram of ages
qplot(x = age,
      data = train_data,
      binwidth = 5,
      color = I('black'),
      fill = I('blue'),
      xlab = "Age",
      ylab = "Count",
      main = "Histogram of Age") +
      scale_x_continuous(breaks = seq(0, 100, 5)) + scale_y_continuous(breaks = seq(0, 5000, 500))
```



The histogram also shows that the most common ages are in groups between 25 - 45 years old

Now we explore the correlation between age and income level

```
# conditional density plot of income vs. age
cdplot(x = train_data$age,
       y = as.factor(train_data$income),
       col = c('light blue', 'gray'),
       border = 1,
       xlab = "Age",
       ylab = "Income",
       main = "Conditionl Density Plot of Income versus Age")
```



The conditional density plot shows that very young and very old age groups have the highest proportion of lower income, while the age group from 40-60 has the highest proportion of people with higher income.

This suggests a correlation where as age increases income tends to increase.

work_class variable

Summary statistics

```
# proportion of total people in each working class
kable(sort(prop.table(table(train_data$work_class)), decreasing = T),
      col.names = c('work_class', '% of total'))
```

work_class	% of total
Private	0.7389
Self-emp-not-inc	0.0829
Local-gov	0.0685
State-gov	0.0424
Self-emp-inc	0.0356
Federal-gov	0.0313
Without-pay	0.0005

We find proportion of people within each working class with income > 50K

```r
# temp df as copy of train_data, use indicator variables for easier computations for now
# after transforming features on temp df, check the accuracy against original training data
tmp <- train_data %>%
  mutate(income_ind = ifelse(income == "<=50K", 0, 1))

# percent high income by working class
work_class_income <- tmp %>%
  group_by(work_class) %>%
  mutate(pct_high_income = mean(income_ind),
         count = n()) %>%
  select(work_class, count, pct_high_income) %>%
  distinct()

kable(arrange(work_class_income, desc(pct_high_income)))
```

work_class	count	pct_high_income
Self-emp-inc	1074	0.5587
Federal-gov	943	0.3871
Local-gov	2067	0.2946
Self-emp-not-inc	2499	0.2857
State-gov	1279	0.2690
Private	22286	0.2188
Without-pay	14	0.0000

Now we create groups for the work_class feature to prep for analysis

```r
# creating groups for work_class
self_employed <- c('Self-emp-inc', 'Self-emp-not-inc')
gov <- c('Federal-gov', 'Local-gov', 'State-gov')

# transform data frames
tmp$work_class <- ifelse(tmp$work_class %in% self_employed, 'self',
                  ifelse(tmp$work_class %in% gov, 'gov', 'private'))
```

```r
# note we impute NA values in test data as 'private'
test_data$work_class <- ifelse(test_data$work_class %in% self_employed, 'self',
```

education and education_num

It is reasonable to assume that these variables are correlated, and so we will create a few education level groups to use in our model

Summary Statistics

```r
summary(train_data$education_num)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0     9.0    10.0    10.1    13.0    16.0
```

```r
kable(sort(prop.table(table(train_data$education)),
           decreasing = TRUE),
           col.names = c('education', '% of total'))
```

education	% of total
HS-grad	0.3262
Some-college	0.2214

education	% of total
Bachelors	0.1672
Masters	0.0539
Assoc-voc	0.0433
11th	0.0347
Assoc-acdm	0.0334
10th	0.0272
7th-8th	0.0185
Prof-school	0.0180
9th	0.0151
12th	0.0125
Doctorate	0.0124
5th-6th	0.0095
1st-4th	0.0050
Preschool	0.0015

Now we create five education levels

```
no_HS <- c('10th', '11th', '12th', '1st-4th', '5th-6th', '7th-8th', '9th')

HS_grad <- c('HS-grad')

assoc <- c('Assoc-acdm', 'Assoc-voc', 'Prof-school', 'Some-college')

college_grad <- c('Bachelors')

grad_school <- c('Masters', 'Doctorate')


tmp$education <- ifelse(tmp$education %in% grad_school, 'grad_school',
                 ifelse(tmp$education %in% college_grad, 'college_grad',
                 ifelse(tmp$education %in% assoc, 'assoc',
                 ifelse(tmp$education %in% HS_grad, 'hs_grad', 'no_hs'))))

test_data$education <- ifelse(test_data$education %in% grad_school, 'grad_school',
                       ifelse(test_data$education %in% college_grad, 'college_grad',
                       ifelse(test_data$education %in% assoc, 'assoc',
                       ifelse(test_data$education %in% HS_grad, 'hs_grad', 'no_hs'))))

prop.table(table(tmp$education, tmp$income), margin = 1)
```

```
##
##                  <=50K    >50K
##   assoc        0.75438 0.24562
##   college_grad 0.57851 0.42149
##   grad_school  0.40160 0.59840
##   hs_grad      0.83567 0.16433
##   no_hs        0.93986 0.06014
```

fnlwgt

We drop this feature as it is not relevant to this particular prediction model

marital_status and relationship

It is also reasonable to believe that marital_status is highly correlated with relationship status, and gender would almost completely determine relationship status as 'Husband', 'Wife', etc.

Before considering the utility of including these features in the model, we examine the distribution of the data

```r
# marital status summary
kable(sort(prop.table(table(tmp$marital_status)), decreasing = TRUE),
      col.names = c('marital_status', '% of total'))
```

marital_status	% of total
Married-civ-spouse	0.4663
Never-married	0.3225
Divorced	0.1397
Separated	0.0311
Widowed	0.0274
Married-spouse-absent	0.0123
Married-AF-spouse	0.0007

```r
prop.table(table(tmp$marital_status, tmp$income), margin = 1)
```

```
##
##                          <=50K    >50K
##   Divorced              0.89274 0.10726
##   Married-AF-spouse     0.52381 0.47619
##   Married-civ-spouse    0.54504 0.45496
##   Married-spouse-absent 0.91622 0.08378
##   Never-married         0.95168 0.04832
##   Separated             0.92971 0.07029
##   Widowed               0.90326 0.09674
```

```r
# relationship status summary
kable(sort(prop.table(table(tmp$relationship)), decreasing = TRUE),
           col.names = c('relationship', '% of total'))
```

relationship	% of total
Husband	0.4132
Not-in-family	0.2562
Own-child	0.1481
Unmarried	0.1065
Wife	0.0466
Other-relative	0.0295

```r
prop.table(table(tmp$relationship, tmp$income), margin = 1)
```

```
##
##                   <=50K    >50K
##   Husband        0.54433 0.45567
##   Not-in-family  0.89348 0.10652
##   Other-relative 0.96063 0.03937
##   Own-child      0.98567 0.01433
##   Unmarried      0.93369 0.06631
##   Wife           0.50640 0.49360
```

After examining the relationship between marital status, relationship, and income, we transform the marital status feature to show groups for 'Married' and 'Single'

We drop the relationship feature from our dataset

```r
married <- c('Married-civ-spouse', 'Married-AF-spouse')
not_married <- c('Divorced', 'Separated', 'Widowed', 'Never-Married', 'Married-spouse-absent')
```

```r
tmp$marital_status <- ifelse(tmp$marital_status %in% married, 'married', 'not_married')

test_data$marital_status <- ifelse(test_data$marital_status %in% married, 'married',
                                   'not_married')
```

occupation

As before, we investigate summary statistics and relation with income variable

```r
# top 5 occupations
kable(sort(prop.table(table(train_data$occupation)), decreasing = T),
      col.names = c('occupation', '% of total'))
```

occupation	% of total
Prof-specialty	0.1339
Craft-repair	0.1336
Exec-managerial	0.1324
Adm-clerical	0.1234
Sales	0.1188
Other-service	0.1065
Machine-op-inspct	0.0652
Transport-moving	0.0521
Handlers-cleaners	0.0448
Farming-fishing	0.0328
Tech-support	0.0302
Protective-serv	0.0214
Priv-house-serv	0.0047
Armed-Forces	0.0003

```r
# find proportion of high income level within each occupation
job_incomes <- tmp %>%
  group_by(occupation) %>%
  mutate(pct_high_income = mean(income_ind),
         count = n()) %>%
  select(occupation, count, pct_high_income) %>%
  distinct()

kable(arrange(job_incomes, desc(pct_high_income)))
```

occupation	count	pct_high_income
Exec-managerial	3992	0.4852
Prof-specialty	4038	0.4485
Protective-serv	644	0.3261
Tech-support	912	0.3048
Sales	3584	0.2706
Craft-repair	4030	0.2253
Transport-moving	1572	0.2029
Adm-clerical	3721	0.1338
Machine-op-inspct	1966	0.1246
Farming-fishing	989	0.1163
Armed-Forces	9	0.1111
Handlers-cleaners	1350	0.0615
Other-service	3212	0.0411
Priv-house-serv	143	0.0070

The table above shows a significant disparity in proportional income level

Now we create groups based on occupations with similar percentage of high income individuals

```
upper_class_job <- c('Exec-managerial',
                     'Prof-specialty',
                     'Protective-serv',
                     'Tech-support',
                     'Sales')

middle_class_job <- c('Craft-repair',
                      'Transport-moving',
                      'Adm-clerical')

low_class_job <- c('Handlers-cleaners',
                   'Other-service',
                   'Priv-house-serv',
                   'Armed-Forces',
                   'Farming-fishing',
                   'Machine-op-inspct')

tmp$occupation <- ifelse(tmp$occupation %in% upper_class_job, 'upper_class',
                  ifelse(tmp$occupation %in% middle_class_job, 'middle_class', 'lower_class'))

# note we impute NA values in test data as 'middle_class' - the most frequent class
test_data$occupation <- ifelse(test_data$occupation %in% upper_class_job, 'upper_class',
                        ifelse(test_data$occupation %in% low_class_job,
                               'lower_class',
                               'middle_class'))
```

race

Distribution of race in the same and race vs. income

```
# percent of each racial group
kable(sort(prop.table(table(train_data$race)), decreasing = T),
      col.names = c('race', '% of total'))
```

race	% of total
White	0.8598
Black	0.0934
Asian-Pac-Islander	0.0297
Amer-Indian-Eskimo	0.0095
Other	0.0077

```
# differences in income proportions by race
prop.table(table(train_data$race, train_data$income), margin = 1)
```

```
##
##                          <=50K    >50K
##   Amer-Indian-Eskimo 0.88112 0.11888
##   Asian-Pac-Islander 0.72291 0.27709
##   Black              0.87007 0.12993
##   Other              0.90909 0.09091
##   White              0.73628 0.26372
```

Since we have low counts for the minority groups, we separate the race feature into White and Non-White groups

```
tmp$race <- ifelse(tmp$race == 'White', 'white', 'non_white')
test_data$race <- ifelse(test_data$race == 'White', 'white', 'non_white')

prop.table(table(tmp$race, tmp$income), margin = 1)
```

```
##
##              <=50K   >50K
##   non_white 0.8418 0.1582
##   white     0.7363 0.2637
```

Among Non-Whites 15.8% have income >50k, and among Whites 26.4% have income >50k

sex

We find the proportion of male and female individuals in the data

```
sort(prop.table(table(train_data$sex)), decreasing = T)
```

```
##
##   Male Female
## 0.6757 0.3243
```

Then we show the relationship between gender and income class

```
# proportion of income class for each gender
prop.table(table(train_data$sex, train_data$income), margin = 1)
```

```
##
##           <=50K   >50K
##   Female 0.8863 0.1137
##   Male   0.6862 0.3138
```

11% of Females have income >50k while 31.4% of Males have income >50k

Significant difference in income level proportion for males vs. females, but the correlation between gender and other features needs to be investigated

capital_gain and capital_loss

The relationship with capital_gains and income level is much less intuitive than some of the other features like education level, age, occupation, etc.

I would guess that individuals with any capital loss or capital gain would be higher income, as they have the wealth to own investments or other assets

We will look at the distribution of capital_gains and capital_losses and consider different ways to engineer an explanatory feature

```
# summary statistics
summary(train_data$capital_gain)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0       0    1092       0   99999
```

```
summary(train_data$capital_loss)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0       0      88       0    4356
```

```
# new column indicating 1 if capital_gain or capital_loss is non-zero, 0 otherwise
non_zero_gain <- tmp$capital_gain != 0
non_zero_loss <- tmp$capital_loss != 0

non_zero_gain_test <- test_data$capital_gain != 0
```

```r
non_zero_loss_test <- test_data$capital_loss != 0

tmp$non_zero_cap <- as.numeric(non_zero_gain | non_zero_loss)
test_data$non_zero_cap <- as.numeric(non_zero_gain_test | non_zero_loss_test)

# new column as capital gain - capital loss
tmp <- tmp %>%
  mutate(capital_profit = ifelse(capital_gain - capital_loss > 0, 'positive',
                          ifelse(capital_gain - capital_loss < 0, 'negative', 'zero')))

# percent with positive/negative/zero profit
prop.table(table(tmp$capital_profit))
```

```
##
## negative positive     zero
##  0.04731  0.08415  0.86854
```

```r
# counts with positive/negative/zero profit
table(as.factor(tmp$capital_profit))
```

```
##
## negative positive     zero
##     1427     2538    26197
```

```r
# income level based on positive or negative capital profit
prop.table(table(tmp$capital_profit, tmp$income), margin = 1)
```

```
##
##             <=50K   >50K
##   negative 0.4835 0.5165
##   positive 0.3716 0.6284
##   zero     0.8024 0.1976
```

```r
# percent with some nonzero capital gain or loss
prop.table(table(tmp$non_zero_cap))
```

```
##
##      0      1
## 0.8685 0.1315
```

```r
# proportion of income levels among individuals with nonzero capital gain or loss
prop.table(table(tmp$non_zero_cap, tmp$income), margin = 1)
```

```
##
##      <=50K   >50K
##   0 0.8024 0.1976
##   1 0.4119 0.5881
```

58% of individuals with nonzero capital gain or nonzero capital loss have income >50K 19% of individuals with 0 capital gain and 0 capital loss have income >50K

Therefore, we only use the new feature non_zero_cap as an indicator variable where the value is 1 for nonzero capital gain or loss and 0 otherwise

hours_per_week

Distribution of hours worked per week

```r
summary(train_data$hours_per_week)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0    40.0    40.0    40.9    45.0    99.0
```

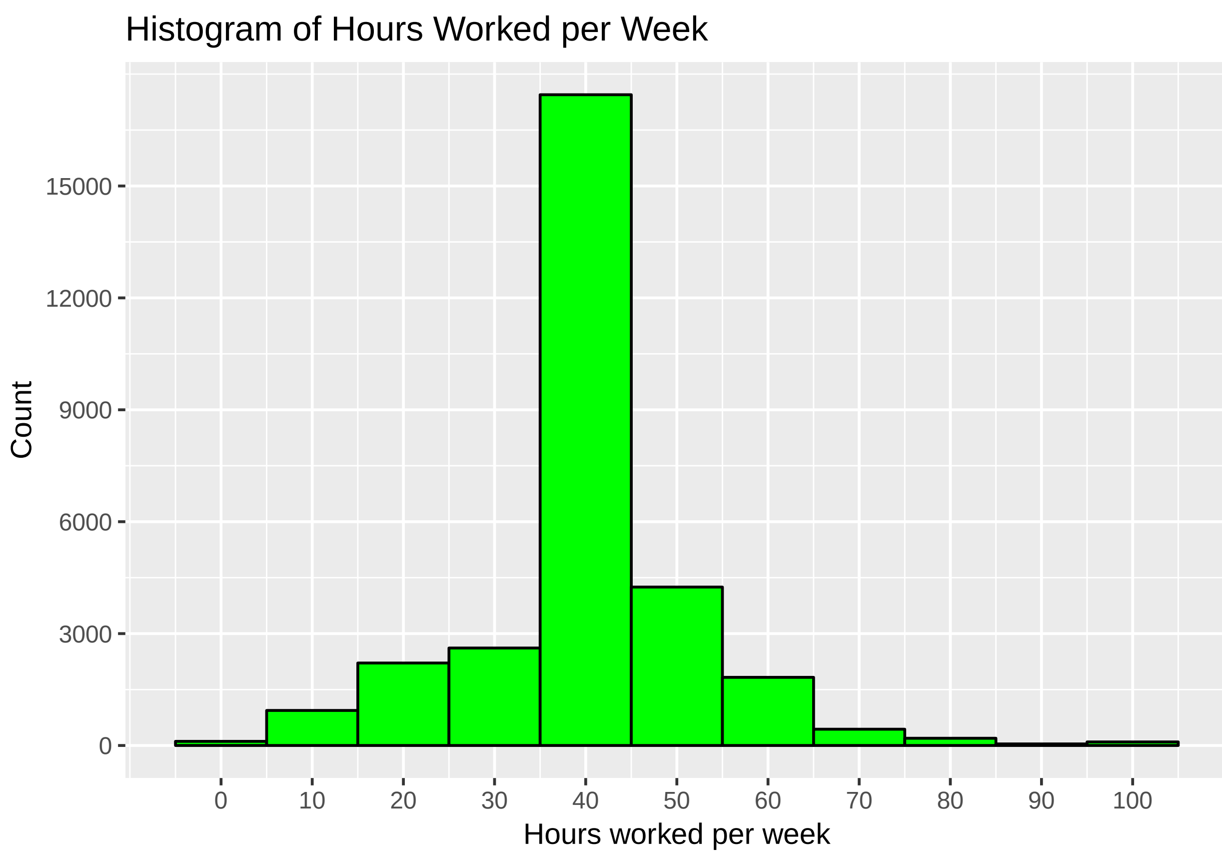
```r
kable(sort((prop.table(table(train_data$hours_per_week))), decreasing = T)[1:5],
      col.names = c('hours_per_week', '% of total'))
```

hours_per_week	% of total
40	0.4725
50	0.0901
45	0.0581
60	0.0466
35	0.0393

Based on the 1st and 3rd quartile, around 50% of individuals work between 40 and 45 hours per week.

Visualizing hours worked per week

```r
# histogram of hours worked per week
qplot(x = hours_per_week,
      data = train_data,
      binwidth = 10,
      color = I('black'),
      fill = I('green'),
      xlab = "Hours worked per week",
      ylab = "Count",
      main = "Histogram of Hours Worked per Week") +
  scale_x_continuous(breaks = seq(0, 100, 10)) +
  scale_y_continuous(breaks = seq(0, 15000, 3000))
```



I would hypothesize that the percentage of high income earners is greater among those who work more hours, so we group the observations to investigate the correlation with income

```r
# creating four levels for number of hours worked per week
tmp <- tmp %>%
  mutate(hours_worked = ifelse(hours_per_week < 40, '0-39',
                        ifelse(hours_per_week >= 40 & hours_per_week <= 45, '40-45',
                        ifelse(hours_per_week > 45 & hours_per_week <= 60, '46-60', '60+'))))

test_data <- test_data %>%
  mutate(hours_worked = ifelse(hours_per_week < 40, '0-39',
                        ifelse(hours_per_week >= 40 & hours_per_week <= 45, '40-45',
                        ifelse(hours_per_week > 45 & hours_per_week <= 60, '46-60', '60+'))))

# hours worked vs. income
kable(sort(prop.table(table(tmp$hours_worked)), decreasing = TRUE),
           col.names = c('hours_worked', '% of total'))
```

hours_worked	% of total
40-45	0.5506
0-39	0.2226
46-60	0.1920
60+	0.0349

```r
prop.table(table(tmp$hours_worked, tmp$income), margin = 1)
```

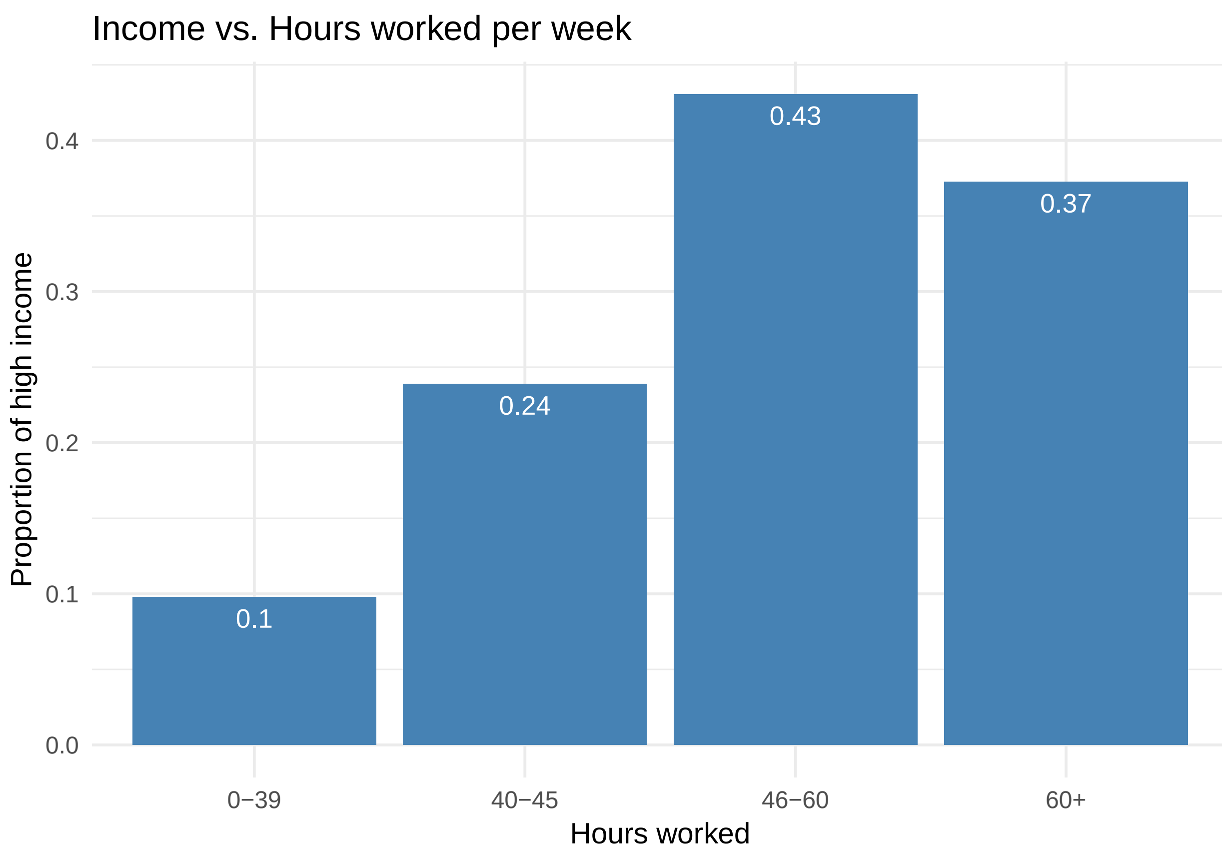
```
##
##           <=50K    >50K
##   0-39  0.90214 0.09786
##   40-45 0.76117 0.23883
##   46-60 0.56943 0.43057
##   60+   0.62738 0.37262
```

Barplot of hours worked vs. income

```r
hours_vs_income <- tmp %>%
  group_by(hours_worked) %>%
  mutate(prop_high_income = mean(income_ind)) %>%
  select(hours_worked, prop_high_income) %>%
  distinct()

hour_plot <- ggplot(data = hours_vs_income,
                    aes(x = hours_worked, y = prop_high_income)) +
                    geom_bar(stat ="identity", fill = "steelblue") +
                    geom_text(aes(label = round(prop_high_income, 2)),
                              vjust=1.6, color="white", size=3.5) +
                    labs(title="Income vs. Hours worked per week",
                         x="Hours worked",
                         y = "Proportion of high income" ) +
                    theme_minimal()

hour_plot
```



native_country

There are 42 unique countries listed, with 91% as United States.

Here we also have many different ways to potentially prepare and segment the data.

```
# top 5 countries represented in the sample
kable((sort(prop.table(table(train_data$native_country)), decreasing = T))[1:5],
      col.names = c('native_country', '% of total'))
```

native_country	% of total
United-States	0.9119
Mexico	0.0202
Philippines	0.0062
Germany	0.0042
Puerto-Rico	0.0036

We consider different ways to group countries together: by region, by national wealth, etc.

```
# Percent high income grouped by country
country_incomes <- tmp %>%
  group_by(native_country) %>%
  mutate(avg_country_income = mean(income_ind),
         count = n()) %>%
  select(native_country, count, avg_country_income) %>%
  distinct()

kable(arrange(country_incomes, desc(avg_country_income)))
```

native_country	count	avg_country_income
Taiwan	42	0.4524
France	27	0.4444
Iran	42	0.4286
India	100	0.4000
Japan	59	0.3898
Cambodia	18	0.3889
Yugoslavia	16	0.3750
Italy	68	0.3529
England	86	0.3488
Germany	128	0.3438
Canada	107	0.3364
Philippines	188	0.3191
Hong	19	0.3158
China	68	0.2941
Greece	29	0.2759
Cuba	92	0.2717
United-States	27504	0.2543
Hungary	13	0.2308
Ireland	24	0.2083
South	71	0.1972
Poland	56	0.1964
Scotland	11	0.1818
Thailand	17	0.1765
Ecuador	27	0.1481
Jamaica	80	0.1250
Laos	17	0.1176
Portugal	34	0.1176
Trinadad&Tobago	18	0.1111
Puerto-Rico	109	0.1101
Haiti	42	0.0952
El-Salvador	100	0.0900
Honduras	12	0.0833
Vietnam	64	0.0781
Peru	30	0.0667
Nicaragua	33	0.0606
Mexico	610	0.0541
Guatemala	63	0.0476
Columbia	56	0.0357
Dominican-Republic	67	0.0299
Outlying-US(Guam-USVI-etc)	14	0.0000
Holand-Netherlands	1	0.0000

```
# South/Central America vs. Non-South American
# 1335, or only around 4% of total are in this group
south_america <- c('Dominican-Republic',
                   "Peru",
                   "Columbia",
                   "Ecuador",
                   "Guatemala",
                   "Nicaragua",
                   "Outlying-US(Guam-USVI-etc)",
                   "Mexico",
                   "Honduras",
                   "El-Salvador",
```

```r
                   "Haiti",
                   "Puerto-Rico",
                   "Jamaica",
                   "Cuba")

# Developed vs. Non-developed countries
dev_countries <- c('United-States',
                   'England',
                   'Germany',
                   'France',
                   'Italy',
                   'Canada',
                   'China',
                   'Japan',
                   'India',
                   'Taiwan',
                   'Philippines')

# 1785 are in non_developed with 11.5% high income
# 28377 are in developed with 25.7% high income
dev_income <- mean(tmp[tmp$native_country %in% dev_countries, ]$income_ind)
non_dev_income <- mean(tmp[!tmp$native_country %in% dev_countries, ]$income_ind)
```

Therefore, we transform the native country feature to an indicator representing whether the country is among the developed countries or not

```r
# transforming native_country column
tmp$native_country <- ifelse(tmp$native_country %in% dev_countries,
                             'developed', 'under_developed')

# imputing NA values with most frequent value
test_data[is.na(test_data$native_country), ]$native_country <- 'United-States'
test_data$native_country <- ifelse(test_data$native_country %in% dev_countries,
                                   'developed', 'under_developed')

# proportion of high income by country
prop.table(table(tmp$native_country, tmp$income), margin = 1)
```

```
##
##                    <=50K   >50K
##   developed       0.7426 0.2574
##   under_developed 0.8852 0.1148
```

Now we can drop unnecessary columns and finalize our processed data

```r
# Deleting unnecessary columns
tmp <- tmp[ , !names(tmp) %in% c('id',
                                 'fnlwgt',
                                 'education_num',
                                 'relationship',
                                 'capital_gain',
                                 'capital_loss',
                                 'hours_per_week',
                                 'capital_profit',
                                 'income')]

test_data <- test_data[ , !names(test_data) %in% c('id',
                                                   'fnlwgt',
                                                   'education_num',
```

```r
                                                    'relationship',
                                                    'capital_gain',
                                                    'capital_loss',
                                                    'hours_per_week',
                                                    'capital_profit',
                                                    'income')]
```

```r
# Examine our processed training data
train_data <- tmp
head(train_data)
```

```
##   age work_class    education marital_status   occupation      race
## 1  39        gov college_grad    not_married middle_class     white
## 2  50       self college_grad        married  upper_class     white
## 3  38    private      hs_grad    not_married  lower_class     white
## 4  53    private        no_hs        married  lower_class non_white
## 5  28    private college_grad        married  upper_class non_white
## 6  37    private  grad_school        married  upper_class     white
##      sex  native_country income_ind non_zero_cap hours_worked
## 1   Male       developed          0            1        40-45
## 2   Male       developed          0            0         0-39
## 3   Male       developed          0            0        40-45
## 4   Male       developed          0            0        40-45
## 5 Female under_developed          0            0        40-45
## 6 Female       developed          0            0        40-45
```

```r
str(train_data)
```

```
## 'data.frame':    30162 obs. of  11 variables:
##  $ age           : num  39 50 38 53 28 37 49 52 31 42 ...
##  $ work_class    : chr  "gov" "self" "private" "private" ...
##  $ education     : chr  "college_grad" "college_grad" "hs_grad" "no_hs" ...
##  $ marital_status: chr  "not_married" "married" "not_married" "married" ...
##  $ occupation    : chr  "middle_class" "upper_class" "lower_class" "lower_class" ...
##  $ race          : chr  "white" "white" "white" "non_white" ...
##  $ sex           : chr  "Male" "Male" "Male" "Male" ...
##  $ native_country: chr  "developed" "developed" "developed" "developed" ...
##  $ income_ind    : num  0 0 0 0 0 0 0 1 1 1 ...
##  $ non_zero_cap  : num  1 0 0 0 0 0 0 0 1 1 ...
##  $ hours_worked  : chr  "40-45" "0-39" "40-45" "40-45" ...
```

Logistic Regression Model

Since this prediction problem is a binary classification, we will use logistic regression.

We fit the logistic model to our training data to calculate values of the coefficients for our predictor variables, which defines the model and allows us to make predictions on new data

We can then use the predict function, which will give probabilities between 0 and 1. We have to set the decision threshhold and classify probabilities greater than the cutoff as 1 (income > 50k) and probabilities less than the cutoff as 0 (income <= 50k)

With methods like forward/back selection, p-values, and AIC we attempt to optimize our model

```r
# Logistic Regression model

# consider models with fewer features vs. the full model
```

```r
mylogit1 <- glm(income_ind ~ age + education + occupation + marital_status,
               data = train_data, family = "binomial")

summary(mylogit1)
```

```
##
## Call:
## glm(formula = income_ind ~ age + education + occupation + marital_status,
##     family = "binomial", data = train_data)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.302  -0.576  -0.279  -0.079   3.381
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -2.2877     0.0794   -28.8   <2e-16 ***
## age                           0.0289     0.0014    20.7   <2e-16 ***
## educationcollege_grad         0.6270     0.0459    13.7   <2e-16 ***
## educationgrad_school          1.1285     0.0638    17.7   <2e-16 ***
## educationhs_grad             -0.5096     0.0425   -12.0   <2e-16 ***
## educationno_hs               -1.5511     0.0802   -19.3   <2e-16 ***
## occupationmiddle_class        0.7336     0.0556    13.2   <2e-16 ***
## occupationupper_class         1.4189     0.0544    26.1   <2e-16 ***
## marital_statusnot_married    -2.4241     0.0393   -61.7   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 33851  on 30161  degrees of freedom
## Residual deviance: 22978  on 30153  degrees of freedom
## AIC: 22996
##
## Number of Fisher Scoring iterations: 6
```

```r
mylogit2 <- glm(income_ind ~ age + education + occupation + marital_status + hours_worked,
               data = train_data, family = "binomial")

summary(mylogit2)
```

```
##
## Call:
## glm(formula = income_ind ~ age + education + occupation + marital_status +
##     hours_worked, family = "binomial", data = train_data)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.450  -0.573  -0.256  -0.053   3.340
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -3.28940    0.09826   -33.5   <2e-16 ***
## age                          0.03250    0.00145    22.5   <2e-16 ***
## educationcollege_grad        0.60665    0.04662    13.0   <2e-16 ***
## educationgrad_school         1.09276    0.06451    16.9   <2e-16 ***
## educationhs_grad            -0.51995    0.04309   -12.1   <2e-16 ***
```

```
## educationno_hs           -1.52602    0.08083   -18.9   <2e-16 ***
## occupationmiddle_class     0.71181    0.05639    12.6   <2e-16 ***
## occupationupper_class      1.36168    0.05528    24.6   <2e-16 ***
## marital_statusnot_married -2.31980    0.03983   -58.2   <2e-16 ***
## hours_worked40-45          0.84553    0.05439    15.6   <2e-16 ***
## hours_worked46-60          1.34495    0.05947    22.6   <2e-16 ***
## hours_worked60+            1.22899    0.09195    13.4   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 33851  on 30161  degrees of freedom
## Residual deviance: 22404  on 30150  degrees of freedom
## AIC: 22428
##
## Number of Fisher Scoring iterations: 6
```

```
mylogit_full <- glm(income_ind ~ ., data = train_data, family = "binomial")
drop1(mylogit_full,test="Chisq")
```

```
## Single term deletions
##
## Model:
## income_ind ~ age + work_class + education + marital_status +
##     occupation + race + sex + native_country + non_zero_cap +
##     hours_worked
##                Df Deviance   AIC  LRT    Pr(>Chi)
## <none>             21247 21283
## age             1    21644 21678  397     < 2e-16 ***
## work_class      2    21256 21288    9       0.014 *
## education       4    22305 22333 1058     < 2e-16 ***
## marital_status  1    24439 24473 3191     < 2e-16 ***
## occupation      2    21875 21907  628     < 2e-16 ***
## race            1    21250 21284    3       0.098 .
## sex             1    21252 21286    5       0.024 *
## native_country  1    21277 21311   30 0.000000051 ***
## non_zero_cap    1    22351 22385 1104     < 2e-16 ***
## hours_worked    3    21717 21747  470     < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This output shows that the Deviance is lowest under the full model, and the AIC is the lowest under the full model, and this implies that the full model minimizes the estimated loss of information and provides a better fit compared to the other potential models

All the input variables have small p-values < 0.05, and even though the p-value for the race coefficient is 0.098 we leave it in the model for now as the tolerated p-values are slightly higher for this test than typical hypothesis tests

Next we view the weights given to the coefficients

```
# view model summary
tidy(mylogit_full)
```

```
## # A tibble: 18 x 5
##    term                       estimate std.error statistic   p.value
##    <chr>                         <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)                 -3.56      0.124     -28.7   4.98e-181
##  2 age                          0.0301    0.00152    19.8   2.04e- 87
##  3 work_classprivate            0.0468    0.0486      0.965 3.35e-  1
```

```
##  4 work_classself              -0.0998   0.0635     -1.57  1.16e-  1
##  5 educationcollege_grad        0.598    0.0481     12.4   1.74e- 35
##  6 educationgrad_school         1.05     0.0672     15.6   1.20e- 54
##  7 educationhs_grad            -0.508    0.0445    -11.4   3.11e- 30
##  8 educationno_hs              -1.46     0.0834    -17.5   1.40e- 68
##  9 marital_statusnot_married   -2.29     0.0453    -50.6   0.
## 10 occupationmiddle_class       0.680    0.0582     11.7   1.56e- 31
## 11 occupationupper_class        1.32     0.0572     23.1   5.48e-118
## 12 racewhite                    0.0944   0.0572      1.65  9.87e-  2
## 13 sexMale                      0.110    0.0487      2.25  2.43e-  2
## 14 native_countryunder_develo~ -0.503    0.0949     -5.29  1.19e-  7
## 15 non_zero_cap                 1.49     0.0457     32.5   5.77e-232
## 16 hours_worked40-45            0.818    0.0570     14.4   9.57e- 47
## 17 hours_worked46-60            1.29     0.0626     20.5   8.01e- 94
## 18 hours_worked60+              1.20     0.0965     12.5   8.56e- 36
```

Since this is a logistic regression model, this is easier to interpret by outputing the odds ratios

```r
# Odds ratios
tidy(exp(coef(mylogit_full)))
```

```
## # A tibble: 18 x 2
##    names                            x
##    <chr>                        <dbl>
##  1 (Intercept)                 0.0285
##  2 age                         1.03
##  3 work_classprivate           1.05
##  4 work_classself              0.905
##  5 educationcollege_grad       1.82
##  6 educationgrad_school        2.85
##  7 educationhs_grad            0.602
##  8 educationno_hs              0.232
##  9 marital_statusnot_married   0.101
## 10 occupationmiddle_class      1.97
## 11 occupationupper_class       3.75
## 12 racewhite                   1.10
## 13 sexMale                     1.12
## 14 native_countryunder_developed 0.605
## 15 non_zero_cap                4.42
## 16 hours_worked40-45           2.27
## 17 hours_worked46-60           3.62
## 18 hours_worked60+             3.34
```

This suggests that having capital gains or losses significantly increases the odds of an individual being in the high income class by around 4x the odds for an individual without any capital gains or losses.

If an individual is in an upper class occupation vs. a lower class occupation, the odds of being in the higher income class is about 3.75x higher.

Now we use this model to predict the income classes for new observations in the testing or validation dataset

```r
# Predictions

# we set the prediction cutoff probability at 0.35
# since there is a class imbalance, this should give
# better accuracy predicting incomes that are greater than 50k

# create new column with predicted income values
test_data$pred_income <- 0
```

```r
# create copy of test data to experiment with different probability cutoffs
new_data <- test_data
new_data$pred_income <- predict(mylogit_full, newdata = new_data, type = "response")
new_data$pred_income <- ifelse(new_data$pred_income < 0.35, '<=50K', '>50K')


# read in censusTest file
censusTest <- read_excel("C:/Users/Drew/Desktop/Hyatt/censusTest.xlsx")
censusTest <- as.data.frame(censusTest[, 2:15])

# add income level predictions
censusTest$income <- new_data$pred_income

# output file as csv
write.csv(censusTest, file = "censusTest_Final.csv")
```

Conclusion

Remarks, limitations, further analyses, and business applications

Given the time constraints and the difficulty in not being able to truly assess the accuracy of the model predictions on the test/validation dataset, I chose to use a logistic regression model.

I am most familiar with logistic regression and I believe logistic regression would perform sufficiently well compared to other possible classification approaches (SVM, Random Forests, KNN, etc.). Logistic regression should be fairly well suited for the number of observations and the number of input variables, and won't be too computationally expensive.

This analysis could be extended by implementing alternative classifiers, engineering new features and further analyzing feature importance and correlation among the input variables, using other validation methods, and determining how the cutoff probability level can be tuned to give different accuracy levels

Finally, since the income class is unbalanced the overall accuracy could be improved by setting the probability cutoff value closer to 0.5. However, this would lead to less accurate predictions of high-income individuals, and for the purpose of a marketing campaign it seems that correctly predicting more higher income individuals would be preferrable