ChooseYourOwnCapstone

Daniel Constable

12/9/2020

- 0.1 Preface
- 0.2 Introduction and Project Overview
  - 0.2.1 The Adult Census Income Dataset
- 0.3 Process and Workflow
- 0.4 Exploratory Data Analysis
  - 0.4.1 What The Data Looks Like
  - 0.4.2 Exploring Working Class
  - 0.4.3 Exploring Education Level
  - 0.4.4 Exploring Marital Status
  - 0.4.5 Exploring Occupation
  - 0.4.6 Exploring Relationships
  - 0.4.7 Exploring Race
  - 0.4.8 Exploring Sex
  - 0.4.9 Exploring Hours Per Week
  - 0.4.10 Exploring Income
- 0.5 Partitioning the Data
- 0.6 Logistic Regression
- 0.7 Decision Tree
- 0.8 Random Forest
- 0.9 Results
- 0.10 Conclusion

0.1 Preface

This project is the choose your own capstone project for HarvardX's Professional Certificate in Data Science. This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents.

0.2 Introduction and Project Overview

Machine learning is one of the most important data science methodologies, and its use has led to a range of discoveries, inventions, and improvements to our lives. In short, machine learning is an algorithm (or set of algorithms) that improve automatically through experience.

Some common instances in which you may have interacted with machine learning would be:

- Spam filters in your email service
- Netflix movie recommendations
- Friend and page recommendations on social media
- Fraud detection through credit card companies

Although some of the most common machine learning use cases today are from private companies who have an interest in increasing time on a platform or revenue, there are also use cases in the public sector.

In this project, I will use a dataset from one such sector.

My goal in this project is to determine the effects of various socioeconomic factors in predicting income level. I will predict whether income exceeds $50K/year based on census data.

0.2.1 The Adult Census Income Dataset

This data was extracted from the 1994 Census bureau database by Ronny Kohavi and Barry Becker (Data Mining and Visualization, Silicon Graphics).

I'll download the dataset from my GitHub account.

0.3 Process and Workflow

I'll go through the following data science project steps to work toward that goal:

1. Data preparation

2. Data exploration
3. Data cleaning
4. Data analysis
5. Results communication

There will be two subsets of data for training and validation. The training subset is called 'test_set' and the validation subset is called 'training_set.'

0.4 Exploratory Data Analysis

Before I get into analyzing the data, it's important to explore the data to see how it's structured and what it looks like.

0.4.1 What The Data Looks Like

I use the str() function to oberve the structure of the data.

```
str(income_data)
```

```
## 'data.frame':    32561 obs. of  15 variables:
##  $ age           : int  90 82 66 54 41 34 38 74 68 41 ...
##  $ workclass     : Factor w/ 9 levels "?","Federal-gov",..: 1 5 1 5 5 5 5 8 2 5 ...
##  $ fnlwgt        : int  77053 132870 186061 140359 264663 216864 150601 88638 422013 70037 ...
##  $ education     : Factor w/ 16 levels "10th","11th",..: 12 12 16 6 16 12 1 11 12 16 ...
##  $ education.num : int  9 9 10 4 10 9 6 16 9 10 ...
##  $ marital.status: Factor w/ 7 levels "Divorced","Married-AF-spouse",..: 7 7 7 1 6 1 6 5 1 5 ...
##  $ occupation    : Factor w/ 15 levels "?","Adm-clerical",..: 1 5 1 8 11 9 2 11 11 4 ...
##  $ relationship  : Factor w/ 6 levels "Husband","Not-in-family",..: 2 2 5 5 4 5 5 3 2 5 ...
##  $ race          : Factor w/ 5 levels "Amer-Indian-Eskimo",..: 5 5 3 5 5 5 5 5 5 5 ...
##  $ sex           : Factor w/ 2 levels "Female","Male": 1 1 1 1 1 1 2 1 1 2 ...
##  $ capital.gain  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ capital.loss  : int  4356 4356 4356 3900 3900 3770 3770 3683 3683 3004 ...
##  $ hours.per.week: int  40 18 40 40 40 45 40 20 40 60 ...
##  $ native.country: Factor w/ 42 levels "?","Cambodia",..: 40 40 40 40 40 40 40 40 40 1 ...
##  $ income        : Factor w/ 2 levels "<=50K",">50K": 1 1 1 1 1 1 1 2 1 2 ...
```

We see that there are 32,561 observations (rows) and 15 variables (columns).

The 15 columns are:

1. age (integer)
2. workclass (factor with 9 levels)
3. fnlwgt (integer)
4. education (factor with 16 levels)
5. education.num (integer)
6. marital.status (factor with 7 levels)
7. occupation (factor with 15 levels)
8. relationship (factor with 6 levels)
9. race (factor with 5 levels)
10. sex (factor with 2 levels)
11. capital.gain (integer)
12. capital.loss (integer)
13. hours.per.week (integer)
14. native.country (factor with 42 levels)
15. income (factor with 2 levels)

To see the dimensions of the data, you can also use the dim() function.

```
dim(income_data)
```

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

We can then use the head() function to check the head of the dataset.

```
head(income_data)
```

```
##   age workclass fnlwgt    education education.num marital.status
## 1  90         ?  77053      HS-grad             9        Widowed
## 2  82   Private 132870      HS-grad             9        Widowed
## 3  66         ? 186061 Some-college            10        Widowed
## 4  54   Private 140359      7th-8th             4       Divorced
## 5  41   Private 264663 Some-college            10      Separated
## 6  34   Private 216864      HS-grad             9       Divorced
##          occupation  relationship  race    sex capital.gain capital.loss
## 1                 ? Not-in-family White Female            0         4356
## 2   Exec-managerial Not-in-family White Female            0         4356
## 3                 ?     Unmarried Black Female            0         4356
## 4 Machine-op-inspct     Unmarried White Female            0         3900
## 5    Prof-specialty     Own-child White Female            0         3900
## 6     Other-service     Unmarried White Female            0         3770
##   hours.per.week native.country income
## 1             40  United-States  <=50K
## 2             18  United-States  <=50K
## 3             40  United-States  <=50K
## 4             40  United-States  <=50K
## 5             40  United-States  <=50K
## 6             45  United-States  <=50K
```

The dataset is in tidy format. Tidy format means that each variable is a column and each observation is a row.

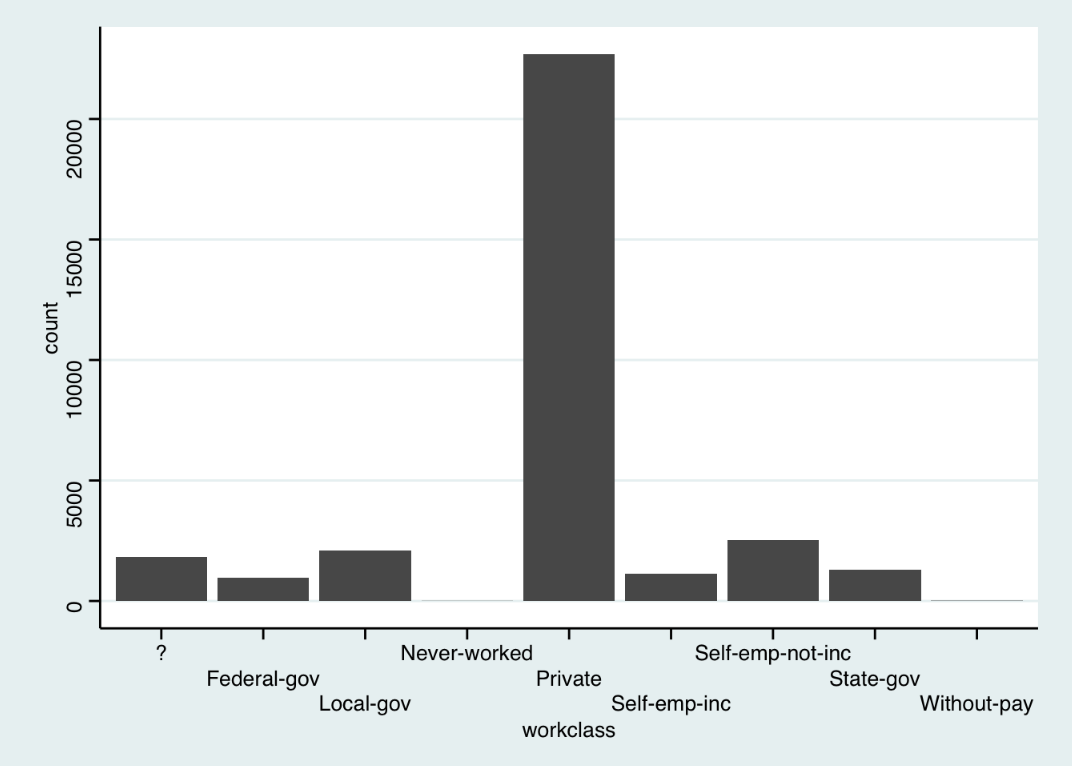
0.4.2 Exploring Working Class

We can check who the people are working for in this dataset.

```
income_data %>% group_by(workclass) %>%
  summarize(n=n()) %>% head()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 6 x 2
##   workclass        n
##   <fct>        <int>
## 1 ?             1836
## 2 Federal-gov    960
## 3 Local-gov     2093
## 4 Never-worked     7
## 5 Private      22696
## 6 Self-emp-inc  1116
```



We see the most common employer is in the private sector, while Self-emp-not-inc comes in second, and local government comes in second. We also note that a significant portion fall under the '?' category.

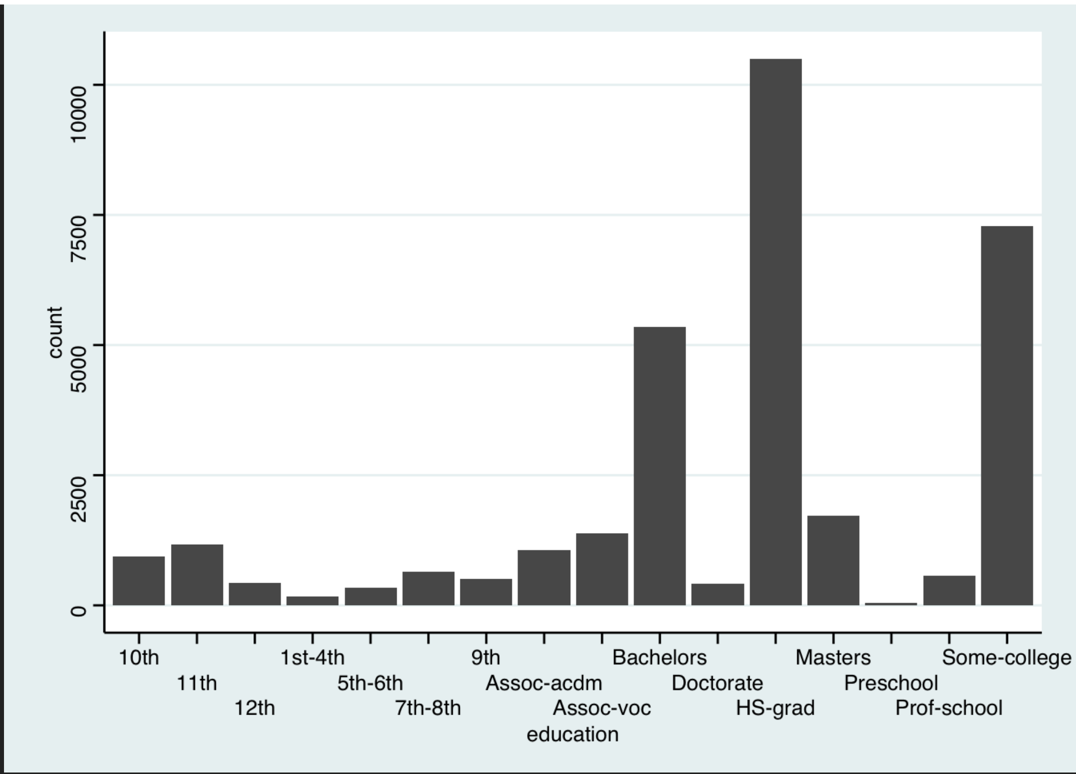
0.4.3 Exploring Education Level

We can check the education level of the people in the dataset.

```
income_data %>% group_by(education) %>%
  summarize(n=n()) %>% head()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 6 x 2
##   education     n
##   <fct>     <int>
## 1 10th        933
## 2 11th       1175
## 3 12th        433
## 4 1st-4th     168
## 5 5th-6th     333
## 6 7th-8th     646
```



We see that the most common level of education to have finished is high school grad, while the second most common is some college.

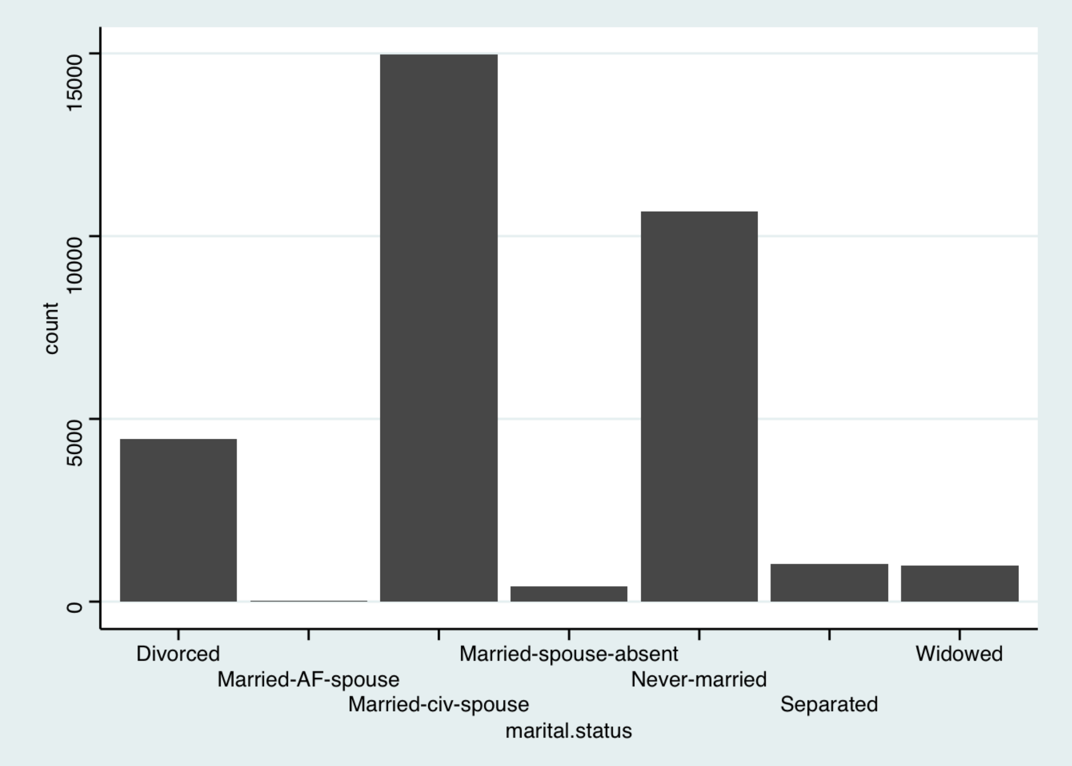
0.4.4 Exploring Marital Status

We can break down the marital status in our dataset.

```
income_data %>% group_by(marital.status) %>%
  summarize(n=n()) %>% head()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 6 x 2
##   marital.status            n
##   <fct>                 <int>
## 1 Divorced               4443
## 2 Married-AF-spouse        23
## 3 Married-civ-spouse    14976
## 4 Married-spouse-absent   418
## 5 Never-married         10683
## 6 Separated              1025
```



The most common status is married with a civilian spouse, while the second most common is never married.

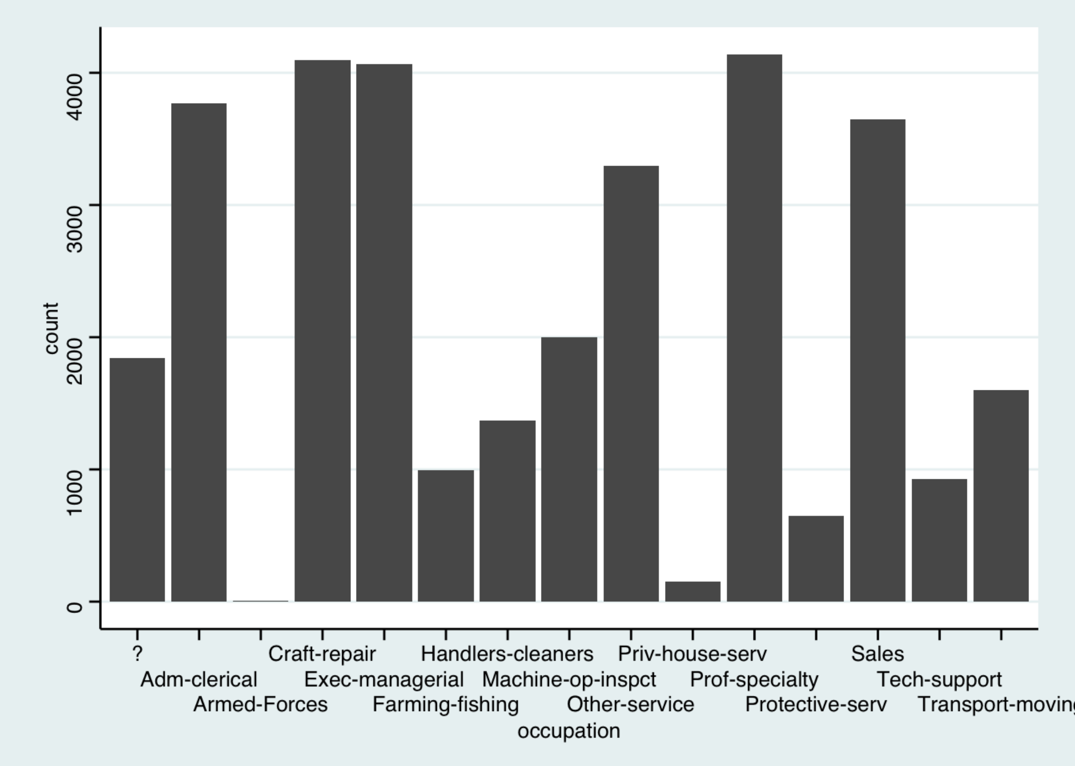
0.4.5 Exploring Occupation

We can break down the people in the dataset by most common occupations.

```
income_data %>% group_by(occupation) %>%
  summarize(n=n()) %>% head()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 6 x 2
##   occupation          n
##   <fct>           <int>
## 1 ?                1843
## 2 Adm-clerical     3770
## 3 Armed-Forces        9
## 4 Craft-repair     4099
## 5 Exec-managerial  4066
## 6 Farming-fishing   994
```



The most common occupations we see are professional specialites and craft/repair. Again, a significant portion falls under the '?' category.

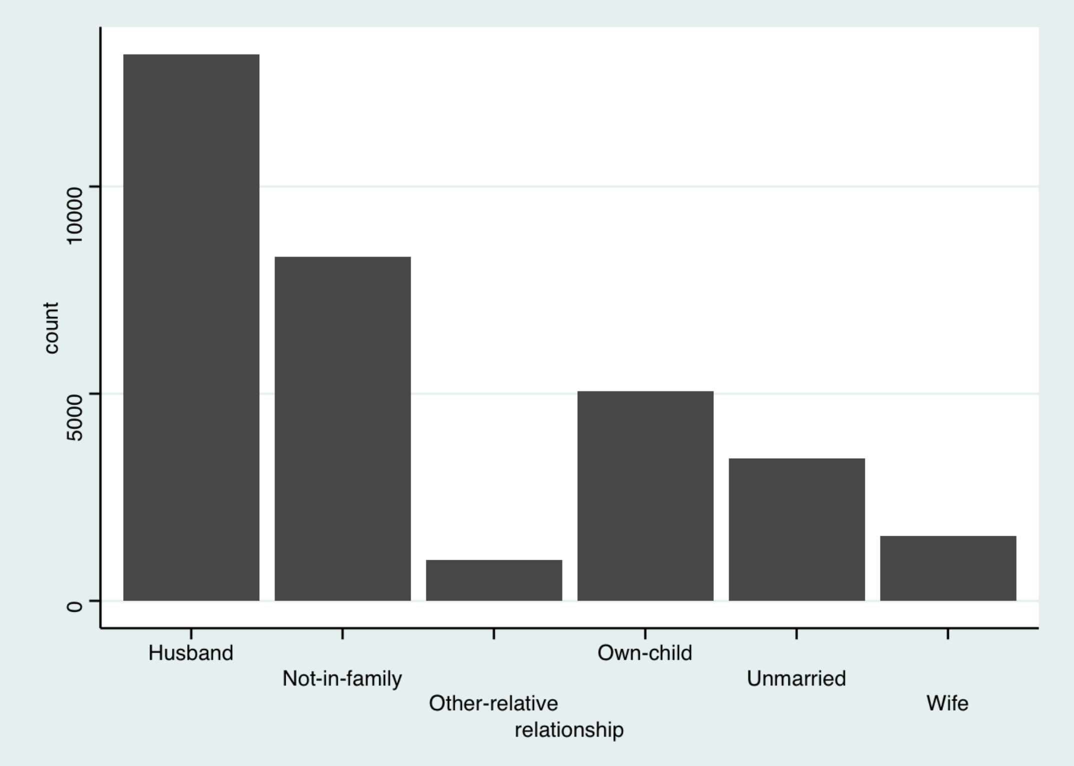
0.4.6 Exploring Relationships

We can check the status of relationships in the dataset.

```
income_data %>% group_by(relationship) %>%
  summarize(n=n()) %>% head()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 6 x 2
##   relationship       n
##   <fct>          <int>
## 1 Husband        13193
## 2 Not-in-family   8305
## 3 Other-relative   981
## 4 Own-child       5068
## 5 Unmarried       3446
## 6 Wife            1568
```



We see that Husband is the most common relationship identifier, while Not-in-family is second. Interestinly, just 1,569 identify as Wife despite the number identifying as Husband.

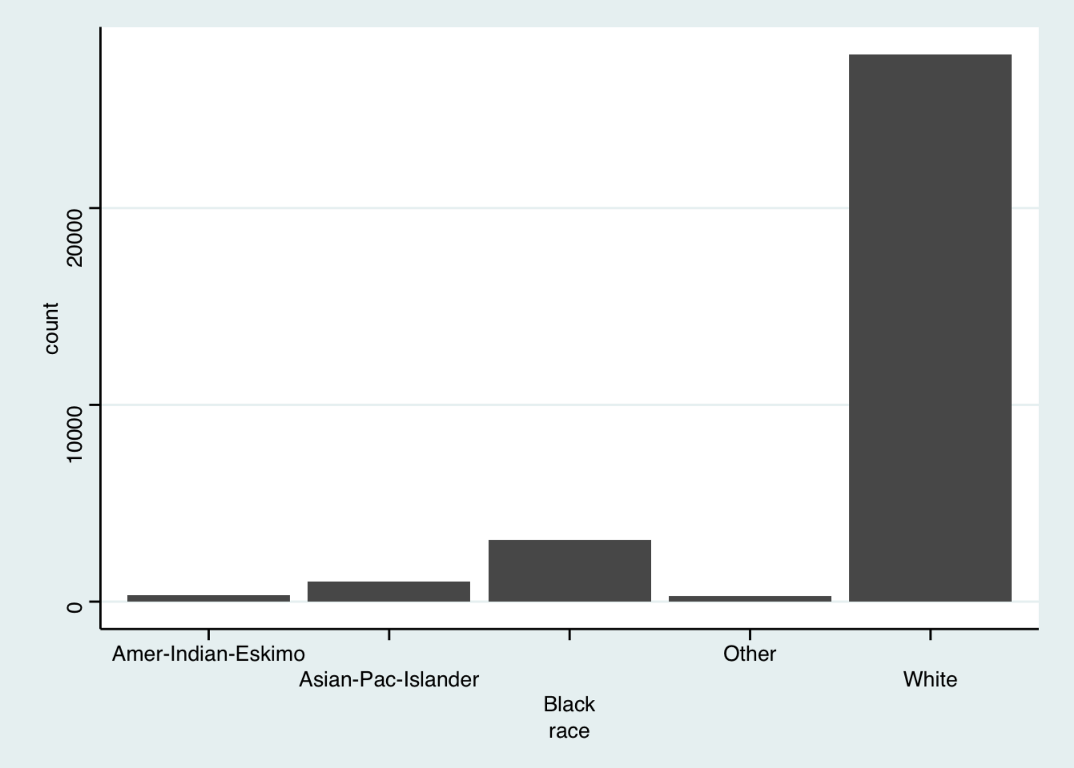
0.4.7 Exploring Race

We can explore the data by race.

```
income_data %>% group_by(race) %>%
  summarize(n=n()) %>% head()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 5 x 2
##   race                   n
##   <fct>              <int>
## 1 Amer-Indian-Eskimo   311
## 2 Asian-Pac-Islander  1039
## 3 Black               3124
## 4 Other                271
## 5 White              27816
```



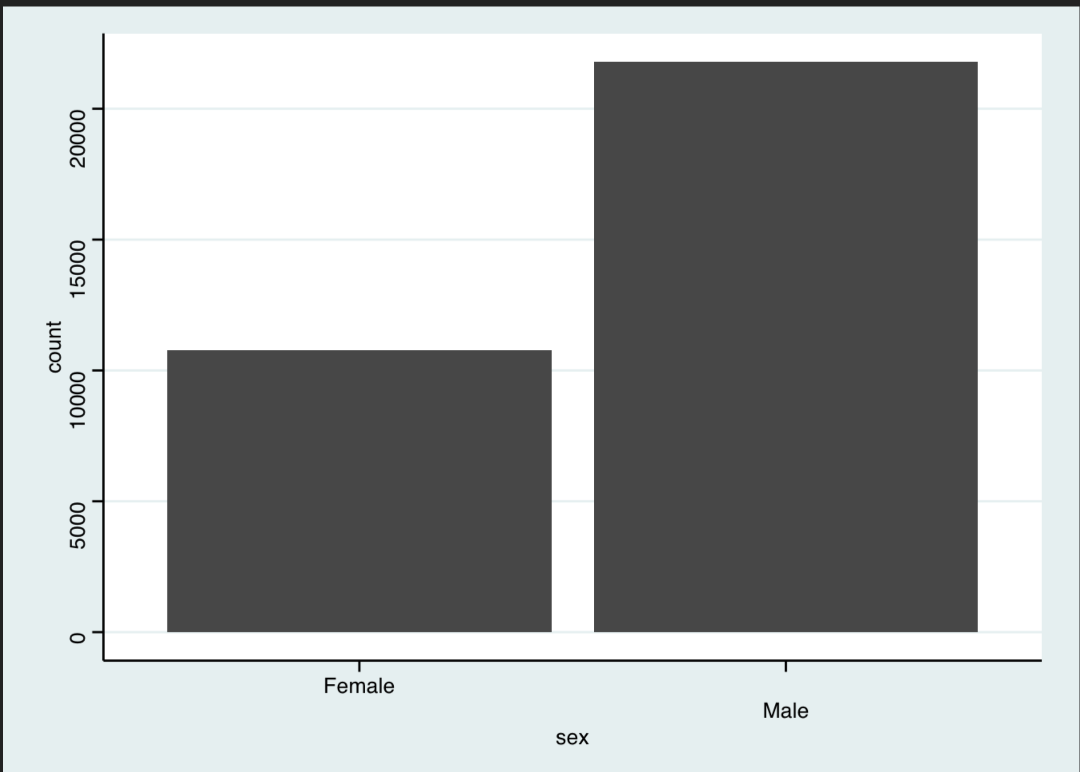
We see that by far the most common race is White, while the second most common race is Black.

0.4.8 Exploring Sex

```
income_data %>% group_by(sex) %>%
summarize(n=n()) %>% head()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 2
##   sex        n
##   <fct>  <int>
## 1 Female 10771
## 2 Male   21790
```



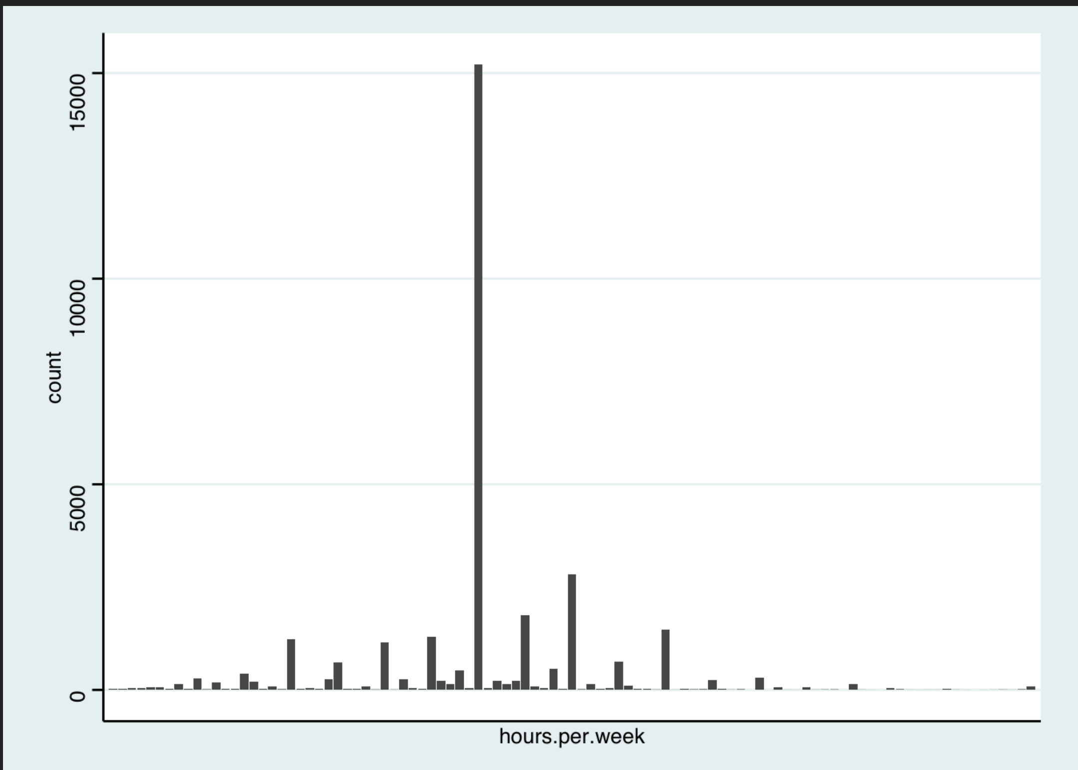
We see that have nearly double the number of Males than Females in this dataset.

0.4.9 Exploring Hours Per Week

```
income_data %>% group_by(hours.per.week) %>%
  summarize(n=n())
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 94 x 2
##    hours.per.week     n
##             <int> <int>
##  1              1    20
##  2              2    32
##  3              3    39
##  4              4    54
##  5              5    60
##  6              6    64
##  7              7    26
##  8              8   145
##  9              9    18
## 10             10   278
## # … with 84 more rows
```

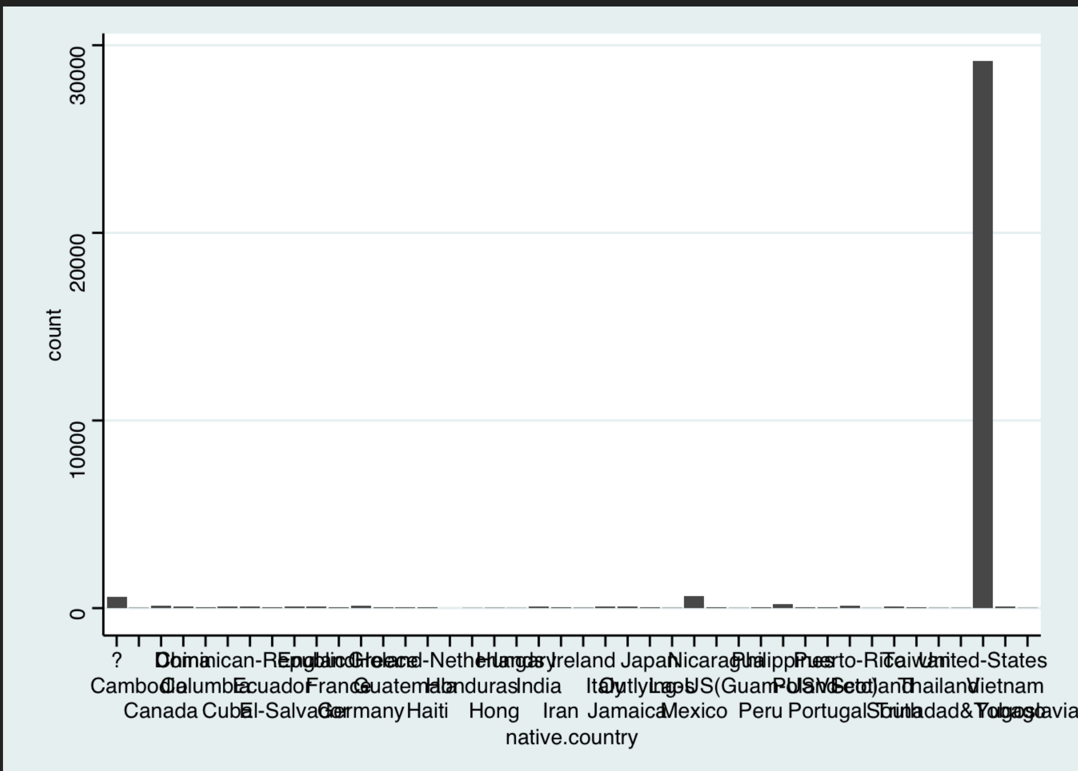


Exploring Native Country

```
income_data %>% group_by(native.country) %>%
  summarize(n=n())
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 42 x 2
##    native.country         n
##    <fct>              <int>
##  1 ?                    583
##  2 Cambodia              19
##  3 Canada               121
##  4 China                 75
##  5 Columbia              59
##  6 Cuba                  95
##  7 Dominican-Republic    70
##  8 Ecuador               28
##  9 El-Salvador          106
## 10 England               90
## # … with 32 more rows
```



0.4.10 Exploring Income

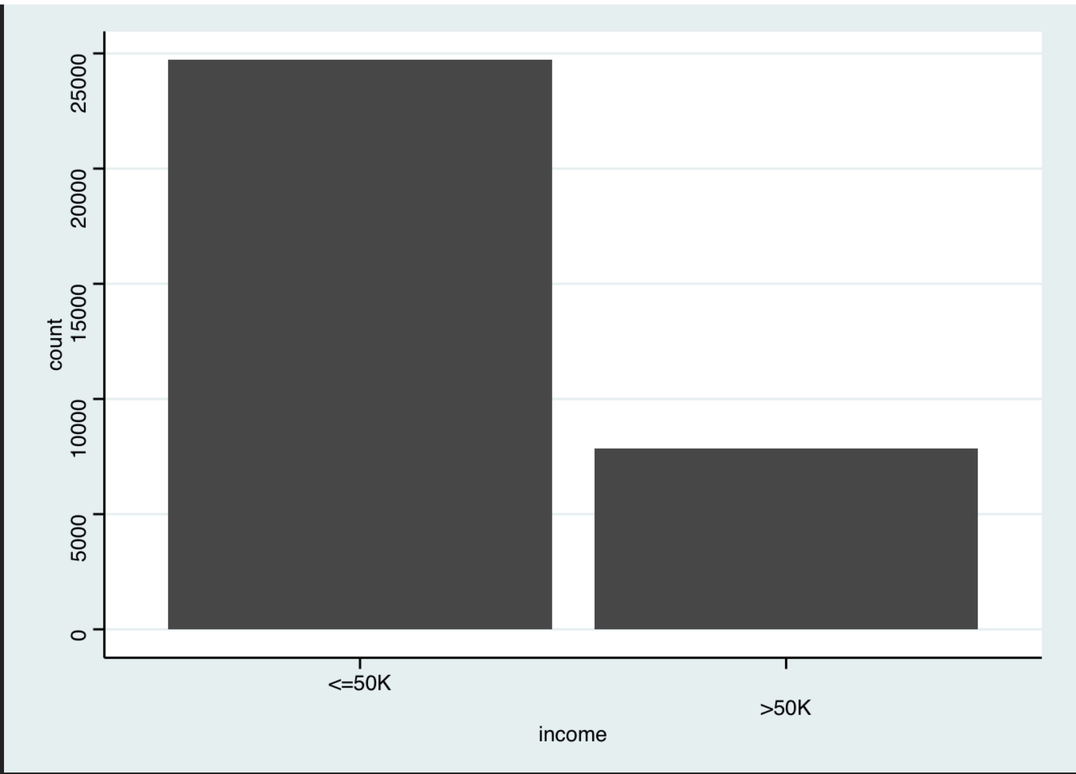
We can explore the number of people who earn less than 50K and more than 50K.

```
income_data %>% group_by(income) %>%
  summarize(n=n())
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 2
##   income     n
##   <fct>  <int>
## 1 <=50K  24720
## 2 >50K    7841
```

```
income_data %>% group_by(income) %>% ggplot(aes(income)) + geom_bar() + theme_stata() +
scale_x_discrete(guide = guide_axis(n.dodge=3))
```



We see that nearly three times more people earn less than 50K than they do more than 50K.

0.5 Partitioning the Data

We'll start by partitioning the data into a test set and a training set to train and test our models.

```
y <- income_data$income
set.seed(2, sample.kind = "Rounding") # if using R 3.5 or earlier, remove the sample.kind argument
```

```
## Warning in set.seed(2, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```
test_index <- createDataPartition(y, times = 1, p = 0.5, list = FALSE)
test_set <- income_data[test_index, ]
train_set <- income_data[-test_index, ]
```

0.6 Logistic Regression

We'll start by using the glm() function to see what variables have an influence on income.

```
summary(glm(income ~ ., family = binomial(), income_data))
```

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

```
##
## Call:
## glm(formula = income ~ ., family = binomial(), data = income_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.0885  -0.5044  -0.1822  -0.0251   3.7656
##
## Coefficients: (2 not defined because of singularities)
##                                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)                           -9.074e+00  4.405e-01 -20.601  < 2e-16
## age                                    2.552e-02  1.651e-03  15.460  < 2e-16
## workclassFederal-gov                   1.097e+00  1.538e-01   7.131 9.99e-13
```

```
## workclassLocal-gov                          4.118e-01  1.403e-01   2.934  0.00334
## workclassNever-worked                      -1.045e+01  2.722e+02  -0.038  0.96936
## workclassPrivate                            5.944e-01  1.252e-01   4.746 2.08e-06
## workclassSelf-emp-inc                       7.694e-01  1.497e-01   5.140 2.74e-07
## workclassSelf-emp-not-inc                   1.037e-01  1.371e-01   0.756  0.44954
## workclassState-gov                          2.835e-01  1.518e-01   1.868  0.06173
## workclassWithout-pay                       -1.221e+01  1.985e+02  -0.062  0.95095
## fnlwgt                                      7.072e-07  1.720e-07   4.111 3.93e-05
## education11th                               8.500e-02  2.107e-01   0.403  0.68670
## education12th                               4.891e-01  2.644e-01   1.850  0.06435
## education1st-4th                           -5.322e-01  4.895e-01  -1.087  0.27696
## education5th-6th                           -2.386e-01  3.248e-01  -0.735  0.46255
## education7th-8th                           -4.755e-01  2.320e-01  -2.050  0.04039
## education9th                               -1.939e-01  2.612e-01  -0.743  0.45771
## educationAssoc-acdm                         1.336e+00  1.763e-01   7.574 3.63e-14
## educationAssoc-voc                          1.352e+00  1.694e-01   7.981 1.45e-15
## educationBachelors                          1.936e+00  1.575e-01  12.296  < 2e-16
## educationDoctorate                          2.989e+00  2.142e-01  13.954  < 2e-16
## educationHS-grad                            8.134e-01  1.534e-01   5.302 1.15e-07
## educationMasters                            2.289e+00  1.679e-01  13.631  < 2e-16
## educationPreschool                         -2.109e+01  3.665e+02  -0.058  0.95410
## educationProf-school                        2.793e+00  2.002e-01  13.955  < 2e-16
## educationSome-college                       1.159e+00  1.556e-01   7.447 9.52e-14
## education.num                                      NA         NA      NA       NA
## marital.statusMarried-AF-spouse             2.686e+00  5.538e-01   4.849 1.24e-06
## marital.statusMarried-civ-spouse            2.206e+00  2.654e-01   8.312  < 2e-16
## marital.statusMarried-spouse-absent        -1.097e-02  2.298e-01  -0.048  0.96192
## marital.statusNever-married                -4.825e-01  8.751e-02  -5.513 3.52e-08
## marital.statusSeparated                    -1.334e-01  1.641e-01  -0.813  0.41647
## marital.statusWidowed                       1.284e-01  1.538e-01   0.835  0.40350
## occupationAdm-clerical                      1.095e-01  9.919e-02   1.104  0.26955
## occupationArmed-Forces                     -1.061e+00  1.543e+00  -0.688  0.49174
## occupationCraft-repair                      1.816e-01  8.487e-02   2.140  0.03239
## occupationExec-managerial                   8.965e-01  8.724e-02  10.276  < 2e-16
## occupationFarming-fishing                  -8.826e-01  1.420e-01  -6.214 5.16e-10
## occupationHandlers-cleaners                -5.698e-01  1.458e-01  -3.907 9.33e-05
## occupationMachine-op-inspct                -1.724e-01  1.062e-01  -1.624  0.10429
## occupationOther-service                    -7.152e-01  1.245e-01  -5.746 9.12e-09
## occupationPriv-house-serv                  -4.018e+00  1.664e+00  -2.415  0.01572
## occupationProf-specialty                    6.251e-01  9.365e-02   6.675 2.46e-11
## occupationProtective-serv                   6.864e-01  1.304e-01   5.265 1.40e-07
## occupationSales                             3.909e-01  9.015e-02   4.336 1.45e-05
## occupationTech-support                      7.657e-01  1.194e-01   6.415 1.41e-10
## occupationTransport-moving                         NA         NA      NA       NA
## relationshipNot-in-family                   5.695e-01  2.627e-01   2.168  0.03015
## relationshipOther-relative                 -3.729e-01  2.427e-01  -1.536  0.12442
## relationshipOwn-child                      -6.601e-01  2.600e-01  -2.539  0.01111
## relationshipUnmarried                       4.411e-01  2.786e-01   1.583  0.11338
## relationshipWife                            1.363e+00  1.026e-01  13.282  < 2e-16
## raceAsian-Pac-Islander                      6.650e-01  2.697e-01   2.465  0.01369
## raceBlack                                   3.940e-01  2.332e-01   1.690  0.09106
## raceOther                                   1.736e-01  3.537e-01   0.491  0.62365
## raceWhite                                   5.728e-01  2.217e-01   2.584  0.00978
## sexMale                                     8.618e-01  7.918e-02  10.883  < 2e-16
## capital.gain                                3.193e-04  1.031e-05  30.968  < 2e-16
## capital.loss                                6.474e-04  3.714e-05  17.431  < 2e-16
## hours.per.week                              2.970e-02  1.622e-03  18.316  < 2e-16
## native.countryCambodia                      1.482e+00  6.336e-01   2.338  0.01936
## native.countryCanada                        5.170e-01  2.952e-01   1.751  0.07989
## native.countryChina                        -5.080e-01  3.943e-01  -1.288  0.19766
## native.countryColumbia                     -1.930e+00  8.242e-01  -2.342  0.01919
## native.countryCuba                          5.339e-01  3.373e-01   1.583  0.11349
## native.countryDominican-Republic           -1.643e+00  1.049e+00  -1.566  0.11735
## native.countryEcuador                      -9.442e-02  7.292e-01  -0.129  0.89697
## native.countryEl-Salvador                  -4.230e-01  4.952e-01  -0.854  0.39301
## native.countryEngland                       4.954e-01  3.335e-01   1.486  0.13735
```

```
## native.countryFrance                      7.730e-01  5.289e-01   1.462  0.14385
## native.countryGermany                     6.197e-01  2.843e-01   2.179  0.02931
## native.countryGreece                     -7.982e-01  5.657e-01  -1.411  0.15824
## native.countryGuatemala                  -6.358e-02  7.625e-01  -0.083  0.93354
## native.countryHaiti                       1.359e-01  6.850e-01   0.198  0.84275
## native.countryHoland-Netherlands         -1.024e+01  8.827e+02  -0.012  0.99074
## native.countryHonduras                   -1.086e+00  2.356e+00  -0.461  0.64493
## native.countryHong                        8.706e-02  6.810e-01   0.128  0.89827
## native.countryHungary                     7.262e-02  7.759e-01   0.094  0.92543
## native.countryIndia                      -1.895e-01  3.284e-01  -0.577  0.56390
## native.countryIran                        2.341e-01  4.508e-01   0.519  0.60364
## native.countryIreland                     7.198e-01  6.448e-01   1.116  0.26424
## native.countryItaly                       9.944e-01  3.447e-01   2.885  0.00392
## native.countryJamaica                     2.285e-01  4.631e-01   0.493  0.62170
## native.countryJapan                       5.794e-01  4.214e-01   1.375  0.16914
## native.countryLaos                       -4.209e-01  8.630e-01  -0.488  0.62575
## native.countryMexico                     -3.643e-01  2.551e-01  -1.428  0.15325
## native.countryNicaragua                  -6.151e-01  8.040e-01  -0.765  0.44424
## native.countryOutlying-US(Guam-USVI-etc) -1.208e+01  2.098e+02  -0.058  0.95407
## native.countryPeru                       -6.498e-01  8.559e-01  -0.759  0.44772
## native.countryPhilippines                 6.104e-01  2.810e-01   2.173  0.02981
## native.countryPoland                      1.820e-01  4.216e-01   0.432  0.66608
## native.countryPortugal                    1.542e-01  6.332e-01   0.243  0.80763
## native.countryPuerto-Rico                -1.483e-01  4.041e-01  -0.367  0.71362
## native.countryScotland                    1.905e-01  7.892e-01   0.241  0.80929
## native.countrySouth                      -8.819e-01  4.414e-01  -1.998  0.04573
## native.countryTaiwan                      2.248e-01  4.724e-01   0.476  0.63409
## native.countryThailand                   -3.784e-01  8.356e-01  -0.453  0.65062
## native.countryTrinadad&Tobago            -1.977e-01  8.709e-01  -0.227  0.82041
## native.countryUnited-States               3.815e-01  1.380e-01   2.764  0.00570
## native.countryVietnam                    -9.593e-01  6.150e-01  -1.560  0.11884
## native.countryYugoslavia                  8.720e-01  6.824e-01   1.278  0.20131
##
## (Intercept)                              ***
## age                                      ***
## workclassFederal-gov                     ***
## workclassLocal-gov                       **
## workclassNever-worked
## workclassPrivate                         ***
## workclassSelf-emp-inc                    ***
## workclassSelf-emp-not-inc
## workclassState-gov                       .
## workclassWithout-pay
## fnlwgt                                   ***
## education11th
## education12th                            .
## education1st-4th
## education5th-6th
## education7th-8th                         *
## education9th
## educationAssoc-acdm                      ***
## educationAssoc-voc                       ***
## educationBachelors                       ***
## educationDoctorate                       ***
## educationHS-grad                         ***
## educationMasters                         ***
## educationPreschool
## educationProf-school                     ***
## educationSome-college                    ***
## education.num
## marital.statusMarried-AF-spouse          ***
## marital.statusMarried-civ-spouse         ***
## marital.statusMarried-spouse-absent
## marital.statusNever-married              ***
## marital.statusSeparated
```

```
## marital.statusWidowed
## occupationAdm-clerical
## occupationArmed-Forces
## occupationCraft-repair                   *
## occupationExec-managerial                ***
## occupationFarming-fishing                ***
## occupationHandlers-cleaners              ***
## occupationMachine-op-inspct
## occupationOther-service                  ***
## occupationPriv-house-serv                *
## occupationProf-specialty                 ***
## occupationProtective-serv                ***
## occupationSales                          ***
## occupationTech-support                   ***
## occupationTransport-moving
## relationshipNot-in-family                *
## relationshipOther-relative
## relationshipOwn-child                    *
## relationshipUnmarried
## relationshipWife                         ***
## raceAsian-Pac-Islander                   *
## raceBlack                                .
## raceOther
## raceWhite                                **
## sexMale                                  ***
## capital.gain                             ***
## capital.loss                             ***
## hours.per.week                           ***
## native.countryCambodia                   *
## native.countryCanada                     .
## native.countryChina
## native.countryColumbia                   *
## native.countryCuba
## native.countryDominican-Republic
## native.countryEcuador
## native.countryEl-Salvador
## native.countryEngland
## native.countryFrance
## native.countryGermany                    *
## native.countryGreece
## native.countryGuatemala
## native.countryHaiti
## native.countryHoland-Netherlands
## native.countryHonduras
## native.countryHong
## native.countryHungary
## native.countryIndia
## native.countryIran
## native.countryIreland
## native.countryItaly                      **
## native.countryJamaica
## native.countryJapan
## native.countryLaos
## native.countryMexico
## native.countryNicaragua
## native.countryOutlying-US(Guam-USVI-etc)
## native.countryPeru
## native.countryPhilippines                *
## native.countryPoland
## native.countryPortugal
## native.countryPuerto-Rico
## native.countryScotland
## native.countrySouth                      *
## native.countryTaiwan
## native.countryThailand
## native.countryTrinadad&Tobago
```

```
## native.countryTrinadad&Tobago
## native.countryUnited-States            **
## native.countryVietnam
## native.countryYugoslavia
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 35948  on 32560  degrees of freedom
## Residual deviance: 20565  on 32462  degrees of freedom
## AIC: 20763
##
## Number of Fisher Scoring iterations: 13
```

We see that some of the most influential variables on income in our dataset are age, class of work, education level, race, and sex.

0.7 Decision Tree

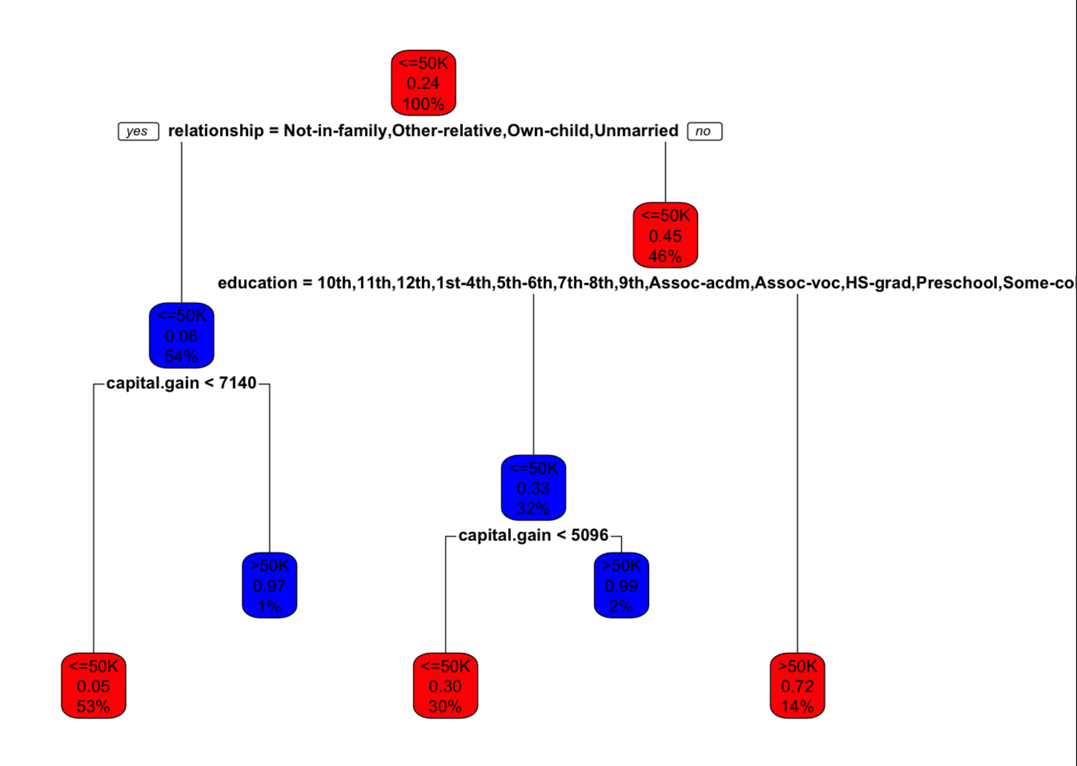
We'll use a decision tree to predict income.

```r
library(rpart)
library(rpart.plot)

fit_tree <- rpart(income~.,data=train_set,method = 'class')
print(fit_tree)
```

```
## n= 16280
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 16280 3920 <=50K (0.75921376 0.24078624)
##    2) relationship=Not-in-family,Other-relative,Own-child,Unmarried 8832  566 <=50K (0.93591486 0
.06408514)
##      4) capital.gain< 7139.5 8683  421 <=50K (0.95151445 0.04848555) *
##      5) capital.gain>=7139.5 149    4 >50K (0.02684564 0.97315436) *
##    3) relationship=Husband,Wife 7448 3354 <=50K (0.54967777 0.45032223)
##      6) education=10th,11th,12th,1st-4th,5th-6th,7th-8th,9th,Assoc-acdm,Assoc-voc,HS-
grad,Preschool,Some-college 5227 1744 <=50K (0.66634781 0.33365219)
##       12) capital.gain< 5095.5 4963 1483 <=50K (0.70118880 0.29881120) *
##       13) capital.gain>=5095.5 264    3 >50K (0.01136364 0.98863636) *
##      7) education=Bachelors,Doctorate,Masters,Prof-school 2221  611 >50K (0.27510131 0.72489869)
*
```

```r
rpart.plot(fit_tree, box.col=c("red", "blue"))
```




```
decision_prediction<- predict(fit_tree,newdata=test_set[-15],type = 'class')
TreeAcu<-confusionMatrix(decision_prediction,test_set$income)$overall[1]
TreeAcu
```

```
##  Accuracy
## 0.8439285
```

The decision tree has an accurace of ~84%.

0.8 Random Forest

In this section, I'll use Random Forest to predict income.

```
library("randomForest")
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
random_forest <- randomForest(income~., data = train_set, method = "class", ntree = 500, do.trace =
100)
```

```
## ntree      OOB      1      2
##   100:  13.97%  6.96% 36.07%
##   200:  13.99%  6.89% 36.38%
##   300:  14.00%  6.79% 36.73%
##   400:  13.95%  6.80% 36.51%
##   500:  13.93%  6.80% 36.40%
```

```
test_set$rf.predicted.income <- predict(random_forest, test_set, type = "class")

rfconfMat <- table(test_set$rf.predicted.income, test_set$income)
rfaccuracy <- sum(diag(rfconfMat))/sum(rfconfMat)
rfconfMat
```

```
##
##          <=50K  >50K
##   <=50K 11525  1402
##   >50K    835  2519
```

```
rfaccuracy
```

```
## [1] 0.8626006
```

We see that Random Forest gives us an accuracy of ~86%.

0.9 Results

We see from our analysis that we can use several socioeconomic factors to predict whether or not someone will earn more or less than $50,000 per year. Based on our logistic analysis, we know that some important factors are age, sex (males are more likely to earn more), race (white people are more likely to earn more), and education (higher education leads to people earning more).

Using the Random Forest method, we can use these factors to predict income level with ~86% accuracy.

0.10 Conclusion

This dataset appears to confirm much of what we know about socioeconomic status. If you're an older white male with higher education, you're more likely to earn more than $50,000 per year than people from other socioeconomic backgrounds. Based on this information, economists could then work to create initiatives that focus on increasing the living standard for people outside of this demographic.