Predicting Incomes From 1994

Justin Farnsworth

6/15/2020

Summary

In this project, a sample of the US population, originally from the 1994 Census, was taken and analyzed in an effort to generate an algorithm that could accurately predict whether an individual made over $50,000 or not. The variables that were used to predict income include but are not limited to age, race, sex, education, occupation, hours per week, and marital status.

Before generating numerous algorithms, an exploration of the dataset was conducted to identify patterns that could be useful when predicting income. We identified groups that were most likely to make over $50,000 based on the data.

A total of seven machine learning algorithms were used to predict income. Five of the models were supervised learning models, one was unsupervised, and the final model was an ensemble of the five supervised learning models. It was determined that the **random forest** model performed the best, with an **accuracy of 85.98%**. The ensemble also did comparatively well as it had an accuracy of 83.97%. Across all models, they were all capable of correctly predicting those who make $50,000 or less most of the time. However, they all struggled with correctly predicting those who made more than $50,000.

Each section has their methods and models explained, followed by their respective results.

The dataset can be accessed here: https://www.kaggle.com/uciml/adult-census-income/data

A copy of the dataset is also present in the project's GitHub repository: https://github.com/farnswj1/Predicting_Incomes_From_1994.git

Analysis

An exploration of the dataset was conducted to identify patterns/relationships in the dataset.

```r
# Required packages
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(gghighlight)) install.packages("gghighlight", repos = "http://cran.us.r-project.org")
if(!require(gam)) install.packages("gam", repos = "http://cran.us.r-project.org")
if(!require(tinytex)) install.packages("tinytex", repos = "http://cran.us.r-project.org")


# Create a temporary file and load the dataset into it
# NOTE: The CSV is already on this project's GitHub repo.
# Original Source: https://www.kaggle.com/uciml/adult-census-income/data
datafile = tempfile()
download.file(
  "https://raw.github.com/farnswj1/Predicting_Incomes_From_1994/master/adult.csv",
  datafile
)

# Read the data from the file
data <- read.csv(datafile)

# Delete the temporary file
rm(datafile)
```

Exploring the Dataset - Overview

After loading the dataset, we saw that there are 32561 rows (each row represented a person) and 15 columns. Here are the first 10 rows of the dataset:

```
# Show the first 10 rows of the dataset
head(data, 10)
```

```
##    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
## 7   38     Private 150601         10th             6      Separated
## 8   74   State-gov  88638    Doctorate            16  Never-married
## 9   68 Federal-gov 422013      HS-grad             9       Divorced
## 10  41     Private  70037 Some-college            10  Never-married
##           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
## 7       Adm-clerical      Unmarried White   Male            0         3770
## 8     Prof-specialty Other-relative White Female            0         3683
## 9     Prof-specialty  Not-in-family White Female            0         3683
## 10      Craft-repair      Unmarried White   Male            0         3004
##    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
## 7              40  United-States  <=50K
## 8              20  United-States   >50K
## 9              40  United-States  <=50K
## 10             60              ?   >50K
```

We see that there are missing values for some of the rows, which are represented as ?. However, let's check to see if there are any null values.

```
# Check if any values in the table are null
any(is.na(data))
```

```
## [1] FALSE
```

It seems that the dataset is fairly clean despite some unknown values. Now, let's check to see what the datatypes are for each column.

```
# Show column names and their datatypes
data.frame(
  column_names = colnames(data),
  data_type = map_chr(colnames(data), function(colname) {class(data[,colname])})
)
```

```
##      column_names data_type
## 1             age   integer
## 2       workclass    factor
## 3          fnlwgt   integer
## 4       education    factor
## 5   education.num   integer
## 6  marital.status    factor
## 7      occupation    factor
## 8    relationship    factor
## 9            race    factor
## 10            sex    factor
## 11   capital.gain   integer
## 12   capital.loss   integer
## 13 hours.per.week   integer
## 14 native.country    factor
## 15         income    factor
```

Exploring the Dataset - Age

Let's identify the range of ages that the dataset consists of.

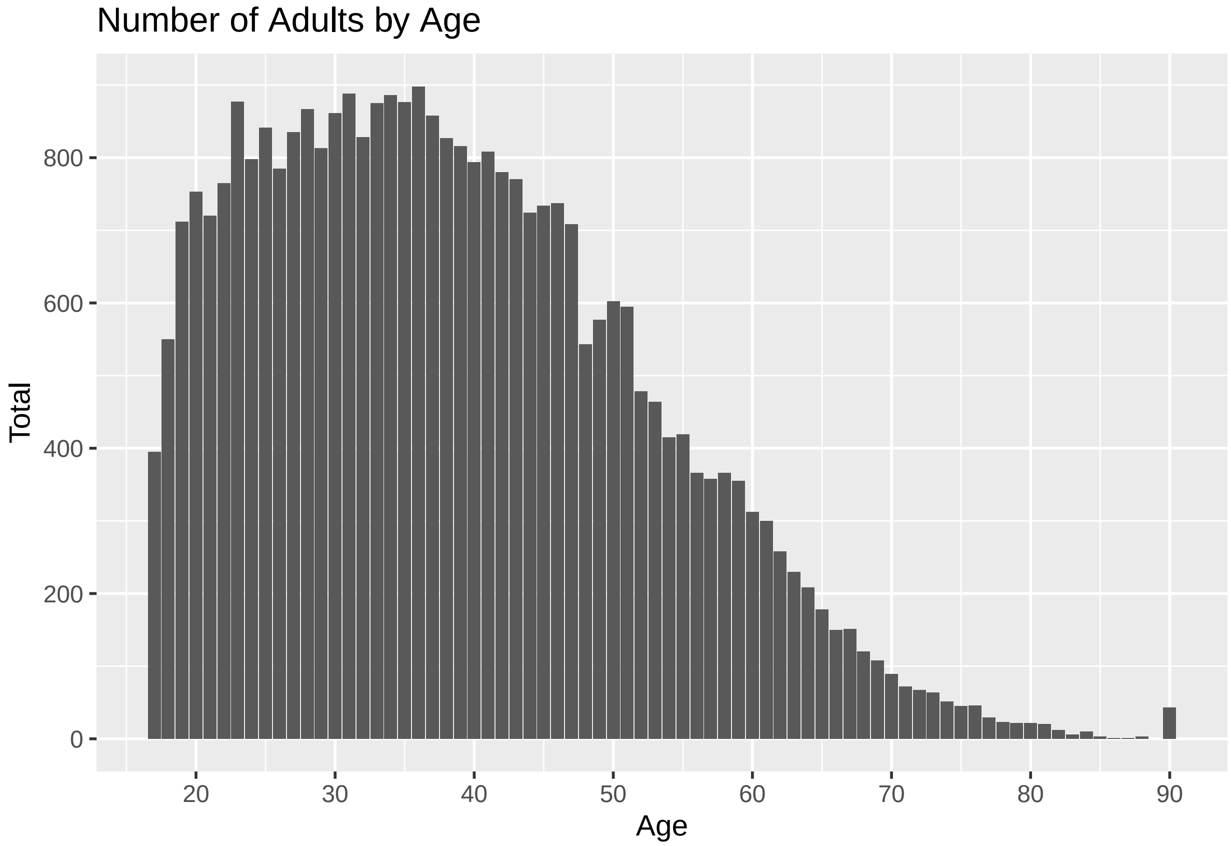
```
# Find the range of age values in the dataset
range(data$age)
```

```
## [1] 17 90
```

Given the wide range of ages, it might be more helpful to visualize the prevalance of each age group in the dataset. The following graph shows the total number of people for each age group.

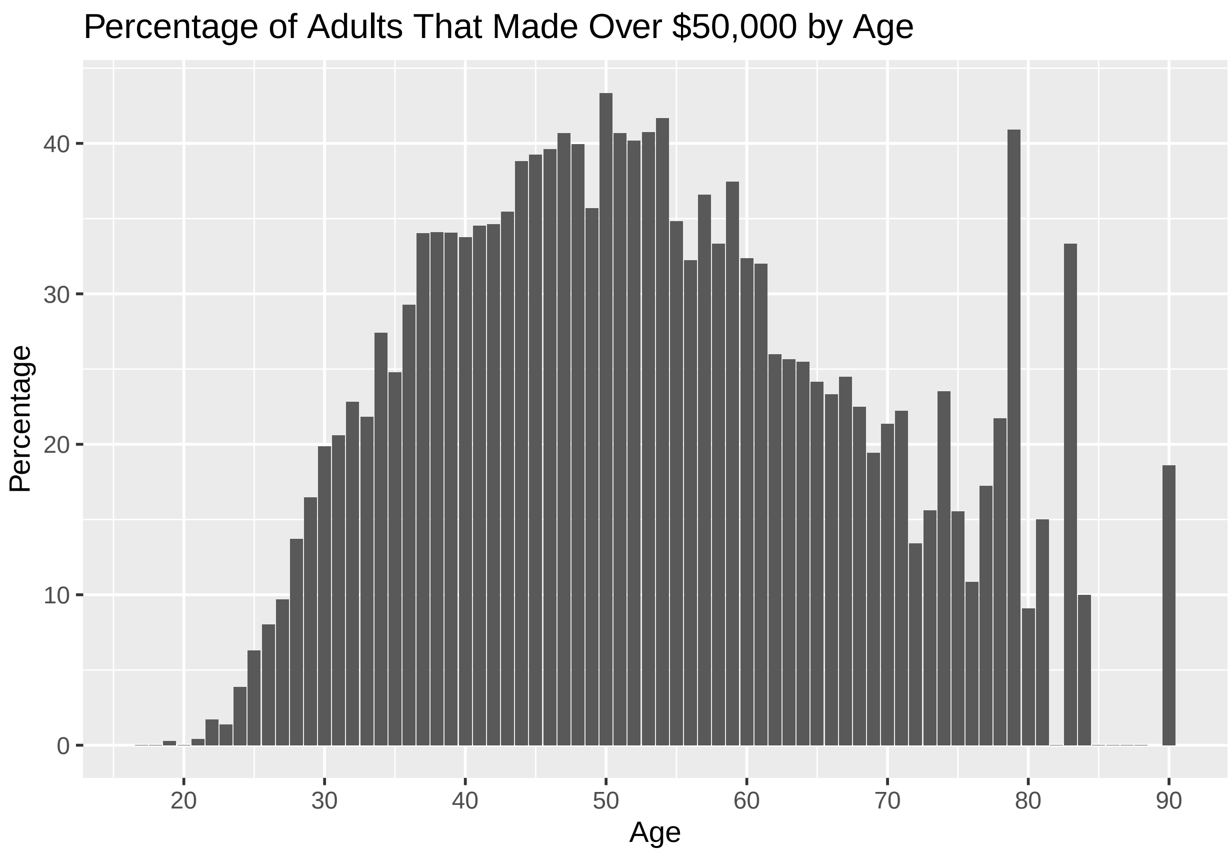
```
# Calculate the number of people and
# the percentage of people who made >$50k for each age
data_age_groups <- data %>%
  group_by(age) %>%
  summarize(total = n(), percentage = mean(income == ">50K") * 100)
```

```
# Plot the number of people in the dataset by age.
data_age_groups %>%
  ggplot(aes(age, total)) +
  geom_bar(stat = "identity") +
  ggtitle("Number of Adults by Age") +
  xlab("Age") +
  ylab("Total") +
  scale_x_continuous(labels = seq(20, 90, 10), breaks = seq(20, 90, 10)) +
  scale_y_continuous(labels = seq(0, 1000, 200), breaks = seq(0, 1000, 200))
```



As expected, the most prevalent age groups in the dataset are younger. It appears to peak in the mid-30s, then it declines afterwards. However, let's identify the percentage of people who made over $50,000 for each age group.

```
# Plot the percentage of people what made over $50k by age
data_age_groups %>%
  ggplot(aes(age, percentage)) +
  geom_bar(stat = "identity") +
  ggtitle("Percentage of Adults That Made Over $50,000 by Age") +
  xlab("Age") +
  ylab("Percentage") +
  scale_x_continuous(labels = seq(20, 90, 10), breaks = seq(20, 90, 10))
```



Interestingly, those that were about 50 years old were most likely to make over $50,000. We also see that there is a high percentage for specific age groups over 75. However, the prevalence of those over 75 years old isn't as high.

```
# Show the number of adults in the dataset that are over 75 by age
data_age_groups %>%
  group_by(age) %>%
  filter(age > 75) %>%
  select(total)
```

```
## # A tibble: 14 x 2
## # Groups:   age [14]
##      age total
##    <int> <int>
##  1    76    46
##  2    77    29
##  3    78    23
##  4    79    22
##  5    80    22
##  6    81    20
##  7    82    12
##  8    83     6
##  9    84    10
## 10    85     3
## 11    86     1
## 12    87     1
## 13    88     3
## 14    90    43
```

Exploring the Dataset - Work Class

Here are the different work classes in the dataset:

```
# Show the different types of work classes
unique(data$workclass)
```

```
## [1] ?                Private          State-gov        Federal-gov
## [5] Self-emp-not-inc Self-emp-inc     Local-gov        Without-pay
## [9] Never-worked
## 9 Levels: ? Federal-gov Local-gov Never-worked Private ... Without-pay
```

As mentioned previously, we see the `?` is listed as one of the values. However, let's look at the total number of people for each work class in the dataset as well as their percentages:

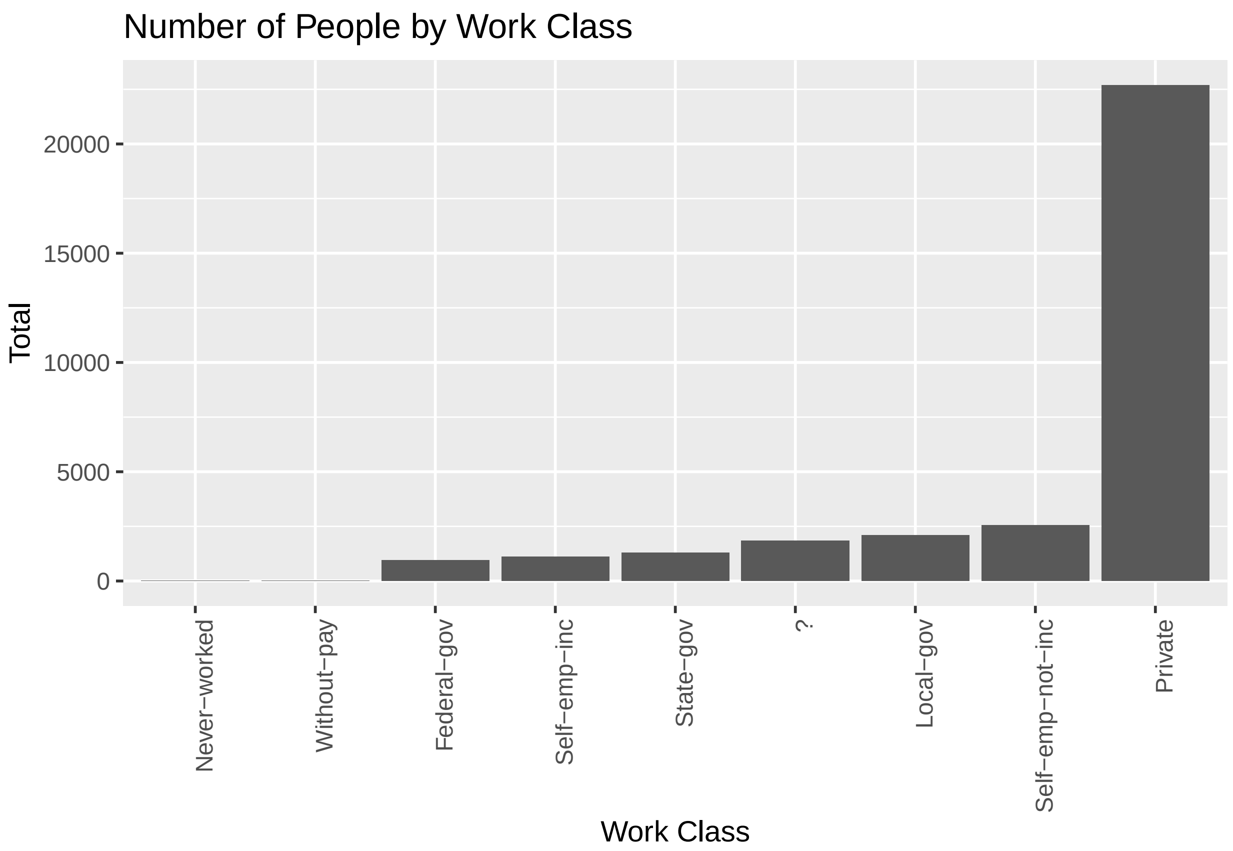
```
# Show the percentages and total number of people
data_work_classes <- data %>%
  group_by(workclass) %>%
  summarize(total = n(), percentage = mean(income == ">50K") * 100) %>%
  arrange(desc(percentage))

data_work_classes
```

```
## # A tibble: 9 x 3
##   workclass        total percentage
##   <fct>            <int>      <dbl>
## 1 Self-emp-inc      1116       55.7
## 2 Federal-gov        960       38.6
## 3 Local-gov         2093       29.5
## 4 Self-emp-not-inc  2541       28.5
## 5 State-gov         1298       27.2
## 6 Private          22696       21.9
## 7 ?                 1836       10.4
## 8 Never-worked         7        0
## 9 Without-pay         14        0
```

Unsurprisingly, those who never worked or aren't getting paid are not going to have high percentages. They were not receiving income and so they were almost certainly not going to earn over $50,000.

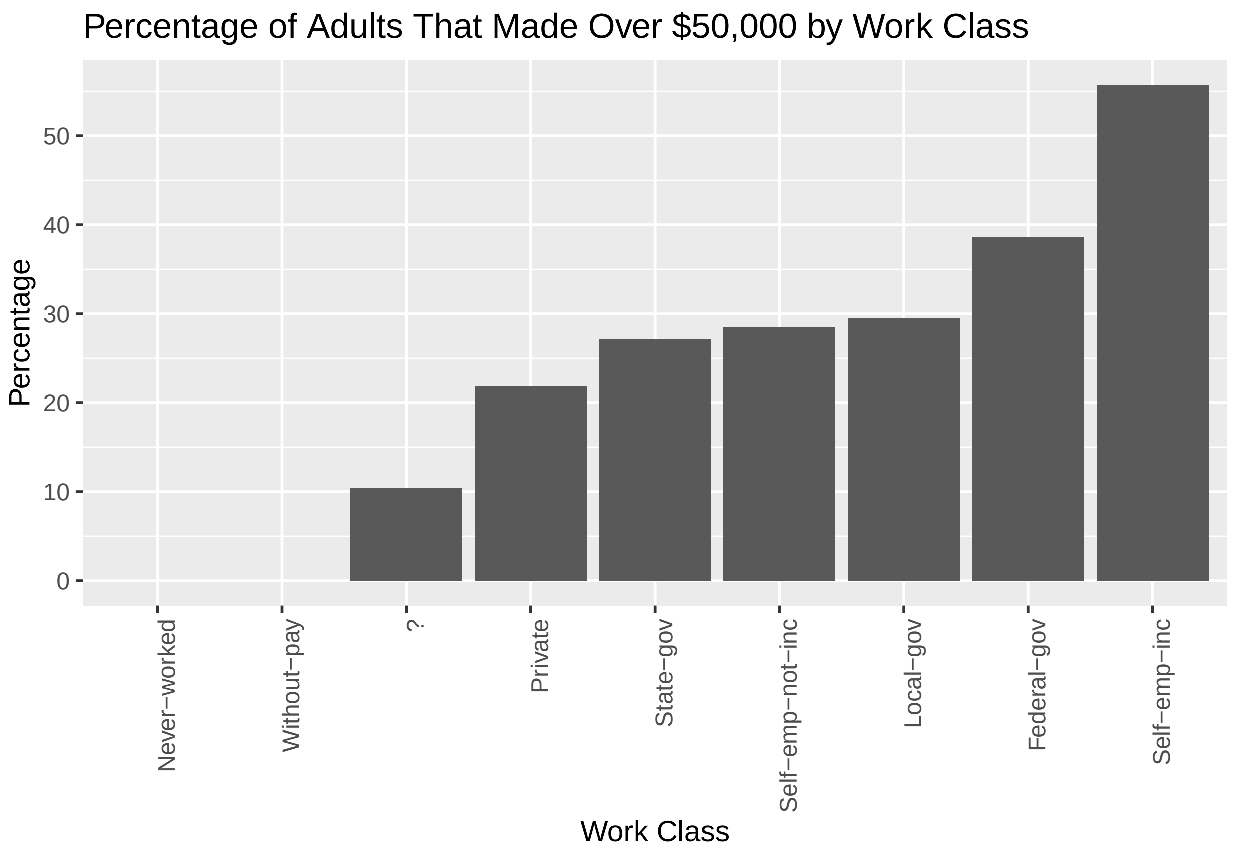
We also see that the private work class made up the majority of people in the dataset. To visualize the prevalence of the work class, the following graph shows the total number of people in each work class:

```
# Plot the total number of people from each work class
data_work_classes %>%
  mutate(workclass = reorder(workclass, total)) %>%
  ggplot(aes(workclass, total)) +
  geom_bar(stat = "identity") +
  ggtitle("Number of People by Work Class") +
  xlab("Work Class") +
  ylab("Total") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



Let's show the percentage of people who made over $50,000 for each work class:

```
# Plot the percentage of people what made over $50k by work class
data_work_classes %>%
  mutate(workclass = reorder(workclass, percentage)) %>%
  ggplot(aes(workclass, percentage)) +
  geom_bar(stat = "identity") +
  ggtitle("Percentage of Adults That Made Over $50,000 by Work Class") +
  xlab("Work Class") +
  ylab("Percentage") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  scale_y_continuous(labels = seq(0, 60, 10), breaks = seq(0, 60, 10))
```



We see that the private work class didn't have the highest percentage despite the high prevlance. Instead, it appears that those who were classified as part of the public sector or self-employed incorporated had the highest percentages. Particularly, the self-employed incorporated work class were twice as more likely than the private work class to make over than $50,000.

Exploring the Dataset - Education

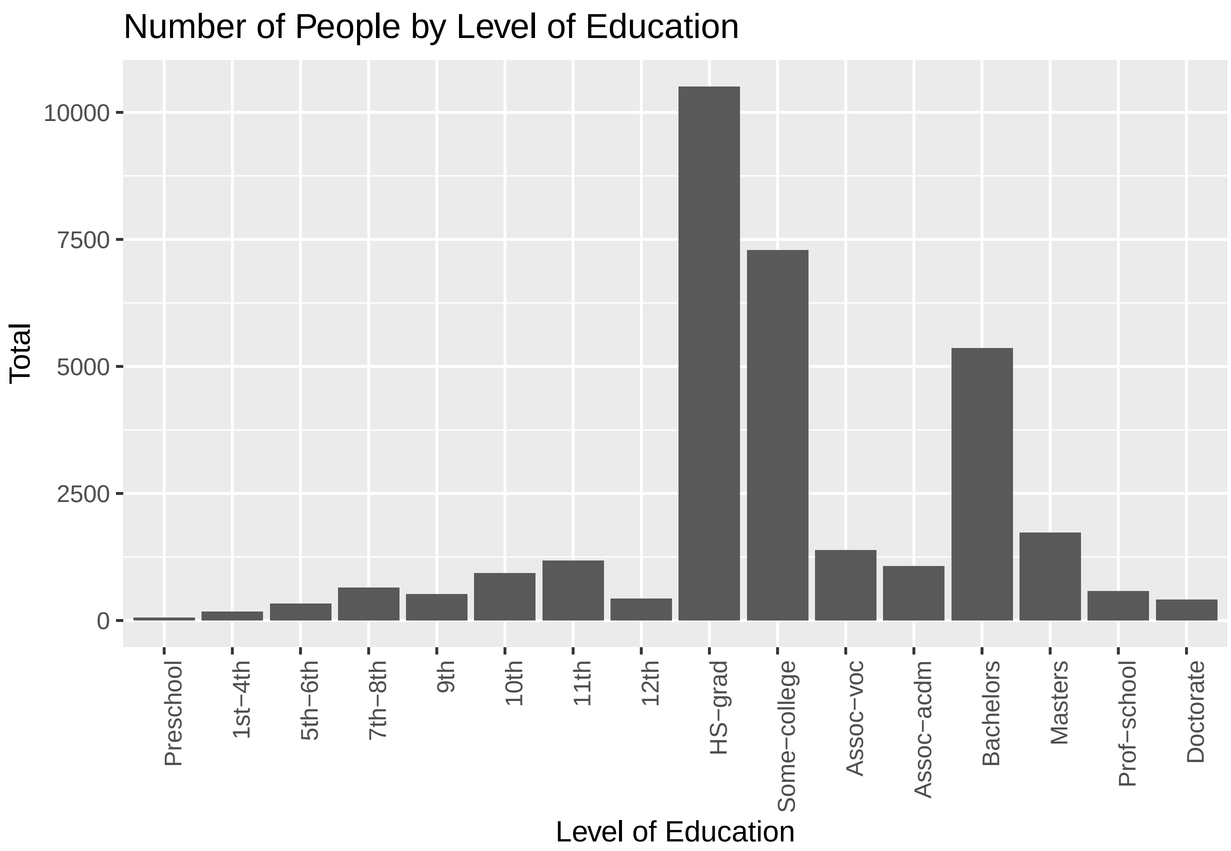
Let's have a look at the different levels of education in the dataset:

```r
# Show the different levels of education along with the totals and percentages
data_education <- data %>%
  select(education, education.num, income) %>%
  group_by(education.num, education) %>%
  summarize(total = n(), percentage = mean(income == ">50K") * 100) %>%
  arrange(desc(education.num)) %>%
  ungroup()

data_education
```

```
## # A tibble: 16 x 4
##    education.num education    total percentage
##            <int> <fct>        <int>      <dbl>
##  1            16 Doctorate      413       74.1
##  2            15 Prof-school    576       73.4
##  3            14 Masters       1723       55.7
##  4            13 Bachelors     5355      41.5
##  5            12 Assoc-acdm    1067      24.8
##  6            11 Assoc-voc     1382      26.1
##  7            10 Some-college  7291      19.0
##  8             9 HS-grad      10501      16.0
##  9             8 12th           433       7.62
## 10             7 11th          1175       5.11
## 11             6 10th           933       6.65
## 12             5 9th            514       5.25
## 13             4 7th-8th        646       6.19
## 14             3 5th-6th        333       4.80
## 15             2 1st-4th        168       3.57
## 16             1 Preschool       51       0
```

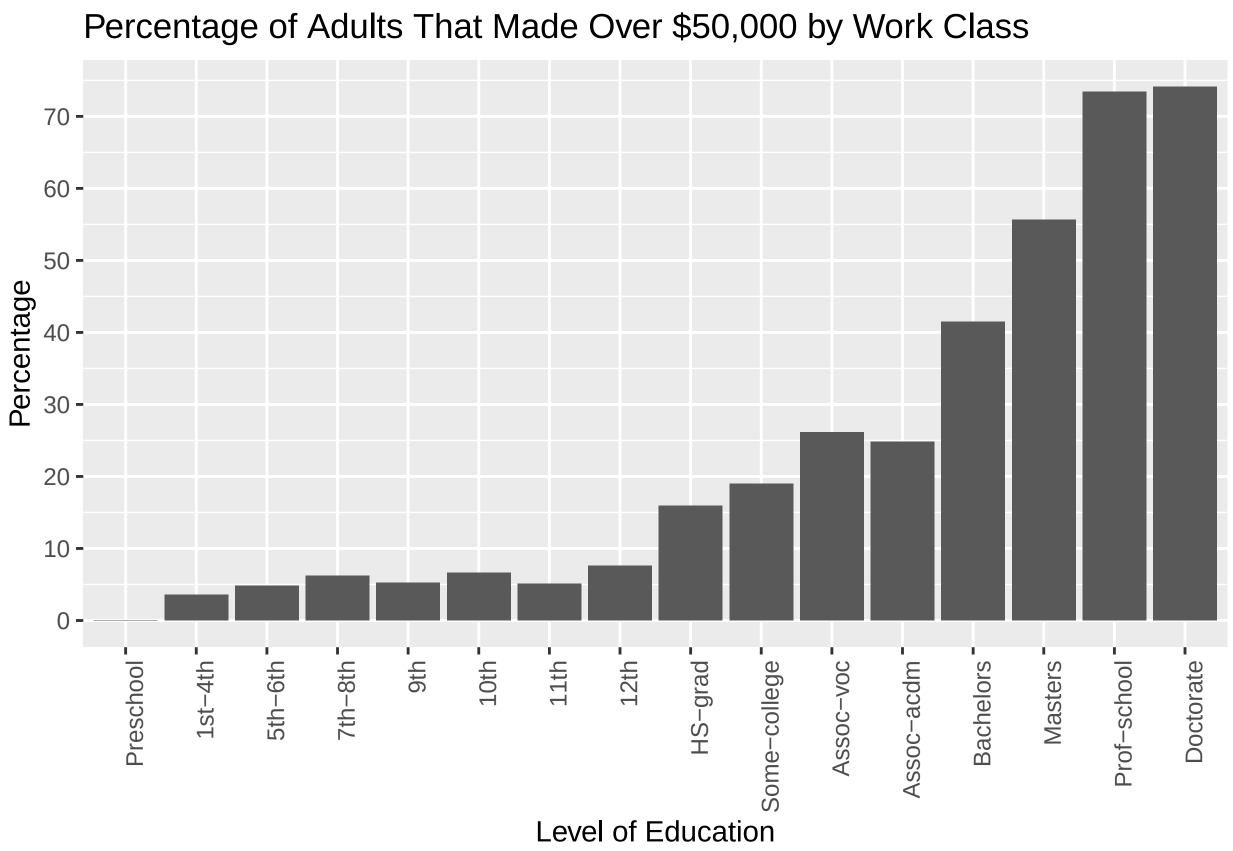
It is expected that those who have a higher level of education tend to have a better chance of making more money. Despite the wide range of levels of education, we see that the most common level of education is a high school graduate. A visualization of the total number of people for each level of education is shown as follows:

```
# Plot the number of people for each level of education
data_education %>%
  mutate(education = reorder(education, education.num)) %>%
  ggplot(aes(education, total)) +
  geom_bar(stat = "identity") +
  ggtitle("Number of People by Level of Education") +
  xlab("Level of Education") +
  ylab("Total") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



However, the percentages are visualized in the following:

```r
# Plot the percentage of people what made over $50k by level of education
data_education %>%
  mutate(education = reorder(education, education.num)) %>%
  ggplot(aes(education, percentage)) +
  geom_bar(stat = "identity") +
  ggtitle("Percentage of Adults That Made Over $50,000 by Work Class") +
  xlab("Level of Education") +
  ylab("Percentage") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  scale_y_continuous(labels = seq(0, 80, 10), breaks = seq(0, 80, 10))
```



Exploring the Dataset - Marital & Relatiionship Status

Let's show the total number of people in each category as well as the percentage of people who made over $50,000 for each group:

```r
# Show the total number of people and the percentage of
# people that made over $50k by marital status
data %>%
  group_by(marital.status) %>%
  summarize(total = n(), percentage = mean(income == ">50K") * 100) %>%
  arrange(desc(percentage))
```

```
## # A tibble: 7 x 3
##   marital.status        total percentage
##   <fct>                 <int>      <dbl>
## 1 Married-civ-spouse    14976      44.7
## 2 Married-AF-spouse        23      43.5
## 3 Divorced               4443      10.4
## 4 Widowed                 993       8.56
## 5 Married-spouse-absent   418       8.13
## 6 Separated              1025       6.44
## 7 Never-married         10683       4.60
```

The dataset suggests that those who were married had a significantly higher percentage than those that were not married. In fact, the percentage is 4 times higher than the next category, Divorced.

Let’s examine the relationship statuses next:

```
# Show the total number of people and the percentage of
# people that made over $50k by relationship status
data %>%
  group_by(relationship) %>%
  summarize(total = n(), percentage = mean(income == ">50K") * 100) %>%
  arrange(desc(percentage))
```

```
## # A tibble: 6 x 3
##   relationship   total percentage
##   <fct>          <int>      <dbl>
## 1 Wife            1568      47.5
## 2 Husband        13193      44.9
## 3 Not-in-family   8305      10.3
## 4 Unmarried       3446       6.33
## 5 Other-relative   981       3.77
## 6 Own-child       5068       1.32
```

This is consistent with the findings from the marital status column. Those that were married had a much higher probability of making over $50,000.

Exploring the Dataset - Occupation

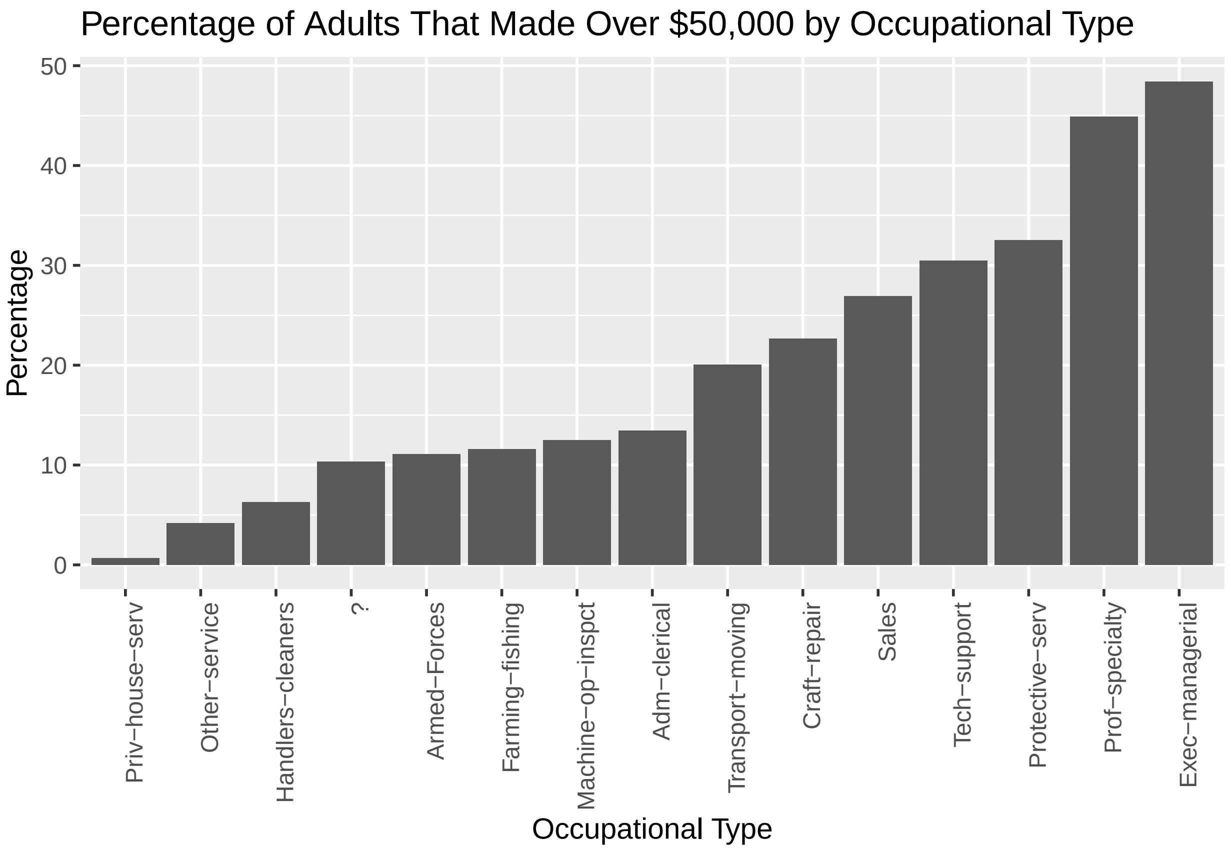
Let’s analyze the different occupational types.

```
# Show the number of people in each type of occupation along with the
# percentage of people who made over $50k for each occupation type.
data_occupations <- data %>%
  group_by(occupation) %>%
  summarize(total = n(), percentage = mean(income == ">50K") * 100) %>%
  arrange(desc(percentage))

data_occupations
```

```
## # A tibble: 15 x 3
##    occupation        total percentage
##    <fct>             <int>      <dbl>
##  1 Exec-managerial    4066     48.4
##  2 Prof-specialty     4140     44.9
##  3 Protective-serv     649     32.5
##  4 Tech-support        928     30.5
##  5 Sales              3650     26.9
##  6 Craft-repair       4099     22.7
##  7 Transport-moving   1597     20.0
##  8 Adm-clerical       3770     13.4
##  9 Machine-op-inspct  2002     12.5
## 10 Farming-fishing     994     11.6
## 11 Armed-Forces          9     11.1
## 12 ?                  1843     10.4
## 13 Handlers-cleaners  1370      6.28
## 14 Other-service      3295      4.16
## 15 Priv-house-serv     149      0.671
```

The occupational type by percentage was executive management, which is also one of the most prevalent types in the dataset. The only occupational type that remained under 1% was private house services. A visualization of the table is shown as follows:

```
# Plot the percentage of people what made over $50k by level of education
data_occupations %>%
  mutate(occupation = reorder(occupation, percentage)) %>%
  ggplot(aes(occupation, percentage)) +
  geom_bar(stat = "identity") +
  ggtitle("Percentage of Adults That Made Over $50,000 by Occupational Type") +
  xlab("Occupational Type") +
  ylab("Percentage") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  scale_y_continuous(labels = seq(0, 80, 10), breaks = seq(0, 80, 10))
```



Exploring the Dataset - Race & Sex

The dataset also included information about the individual's race and sex. Let's analyze race first.

Here are the total of number of people for each group as well as their percentages:

```
# Show the totals and percentages for each racial group
data_races <- data %>%
  group_by(race) %>%
  summarize(total = n(), percentage = mean(income == ">50K") * 100) %>%
  arrange(desc(percentage))

data_races
```

```
## # A tibble: 5 x 3
##   race               total percentage
##   <fct>              <int>      <dbl>
## 1 Asian-Pac-Islander  1039      26.6
## 2 White              27816      25.6
## 3 Black               3124      12.4
## 4 Amer-Indian-Eskimo   311      11.6
## 5 Other                271       9.23
```

While the majority of the people in the dataset were white, those of Asian/Pacific Islander descent had the highest percentage. The dataset suggests that those that were white or Asian/Pacific Islander were twice as likely to make over $50,000 than those that were black or American-Indian/Eskimo.

Now let's analyze the sexes:

```
# Show the totals and percentages for males and females
data_sexes <- data %>%
  group_by(sex) %>%
  summarize(total = n(), percentage = mean(income == ">50K") * 100) %>%
  arrange(desc(percentage))

data_sexes
```

```
## # A tibble: 2 x 3
##   sex    total percentage
##   <fct>  <int>      <dbl>
## 1 Male   21790       30.6
## 2 Female 10771       10.9
```

We see that men had almost triple the likelihood of making more than $50,000 when compared to women. However, the reason for this observation is not clearly explained by the dataset.

Let's analyze the two features together:

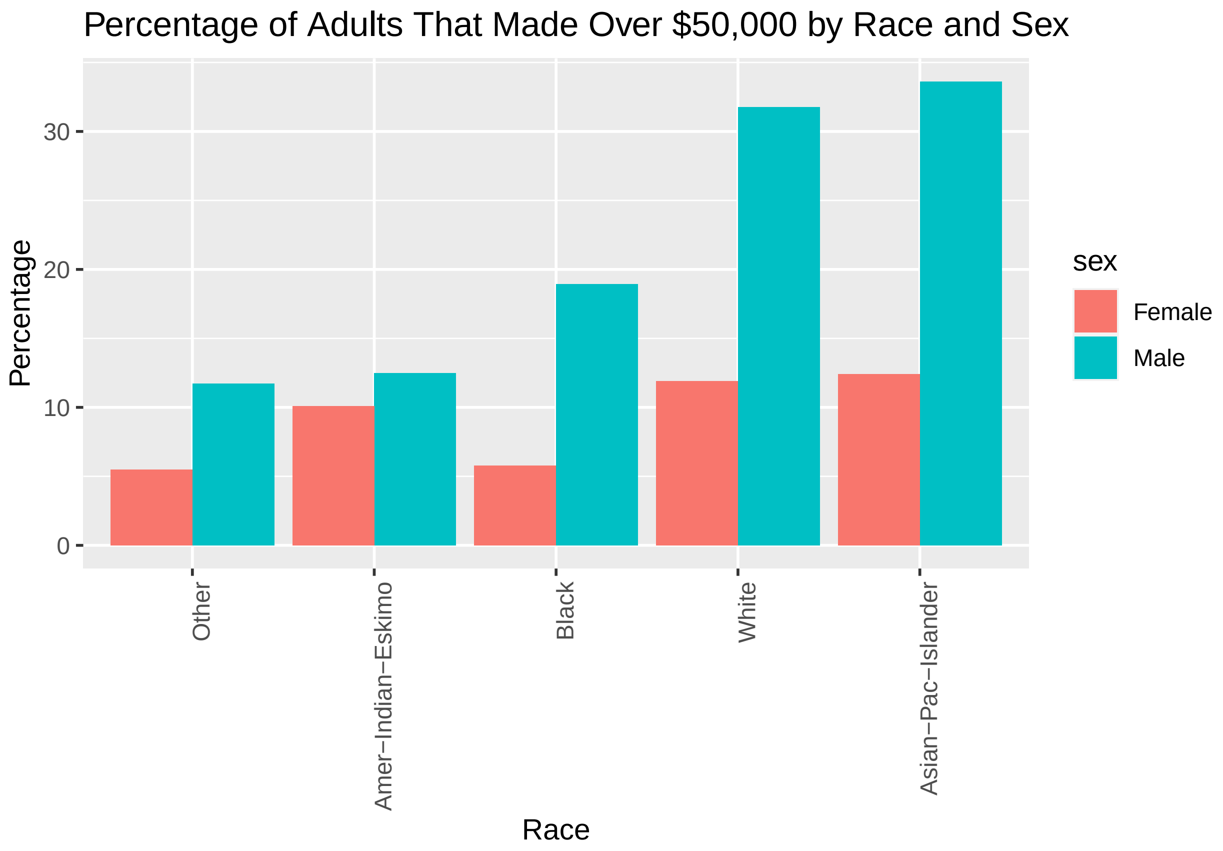
```
# Show the totals and percentages by race and sex together
data_races_and_sexes <- data %>%
  group_by(race, sex) %>%
  summarize(total = n(), percentage = mean(income == ">50K") * 100) %>%
  arrange(desc(percentage)) %>%
  ungroup()

data_races_and_sexes
```

```
## # A tibble: 10 x 4
##    race               sex    total percentage
##    <fct>              <fct>  <int>      <dbl>
##  1 Asian-Pac-Islander Male     693      33.6
##  2 White              Male   19174      31.8
##   3 Black              Male    1569      18.9
##   4 Amer-Indian-Eskimo Male     192      12.5
##   5 Asian-Pac-Islander Female   346      12.4
##   6 White              Female  8642      11.9
##   7 Other              Male     162      11.7
##   8 Amer-Indian-Eskimo Female   119      10.1
##   9 Black              Female  1555       5.79
##  10 Other              Female   109       5.50
```

We can see that across all races, men had a higher probability of earning more than $50,000 than women of the same race. It is also suggested by the data that some groups had a higher percentage than women of all racial groups. The only male group that didn't was those listed as Other.

A plot of the table above is shown below:

```
# Plot the percentages by race and sex
data_races_and_sexes %>%
  mutate(race = reorder(race, percentage)) %>%
  ggplot(aes(race, percentage, fill = sex)) +
  geom_bar(stat = "identity", position = "dodge") +
  ggtitle("Percentage of Adults That Made Over $50,000 by Race and Sex") +
  xlab("Race") +
  ylab("Percentage") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



NOTE: We do NOT encourage discrimination on the basis of race, sex, or any other immutable characteristic.

Exploring the Dataset - Capital

The dataset provides two columns: capital gains and capital losses. We can use this information to calculate net capital gains, which is defined as:

$$\textit{net capital gain} = \textit{capital gain} - \textit{capital loss}$$

We will also round the net capital gains for each row to the nearest thousand.

```r
# Show the totals and percentages by net capital gains (rounded to the nearest 1000)
data_net_capital_gains <- data %>%
  mutate(net_capital_gain = round((capital.gain - capital.loss) / 1000) * 1000) %>%
  group_by(net_capital_gain) %>%
  summarize(total = n(), percentage = mean(income == ">50K") * 100) %>%
  arrange(desc(net_capital_gain))

data_net_capital_gains %>% print(n = Inf)
```

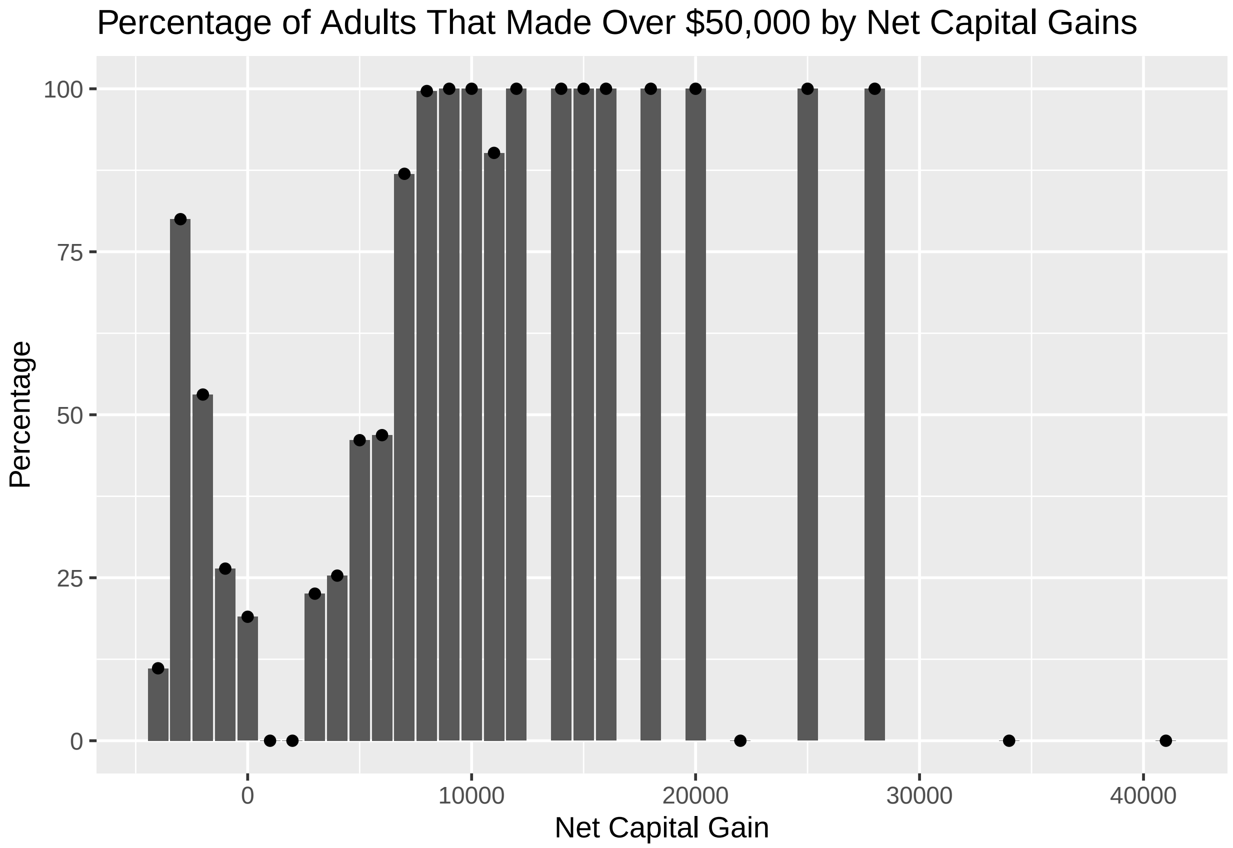
```
## # A tibble: 28 x 3
##    net_capital_gain total percentage
##               <dbl> <int>      <dbl>
##  1           100000   159      100
##  2            41000     2        0
##  3            34000     5        0
##  4            28000    34      100
##  5            25000    15      100
##  6            22000     1        0
##  7            20000    37      100
##  8            18000     2      100
##  9            16000     6      100
## 10            15000   352      100
## 11            14000    94      100
## 12            12000     2      100
## 13            11000    61       90.2
## 14            10000     4      100
## 15             9000    77      100
## 16             8000   288       99.7
## 17             7000   299       87.0
## 18             6000    32       46.9
## 19             5000   282       46.1
## 20             4000   229       25.3
## 21             3000   399       22.6
## 22             2000   218        0
## 23             1000   106        0
## 24                0 28349       19.0
## 25            -1000   125       26.4
## 26            -2000  1339       53.1
## 27            -3000    35       80
## 28            -4000     9       11.1
```

Most people have a net capital gain of 0. In other words, most people in the dataset either made or lost some money through their capital or they didn't have financial assets in 1994.

It is also no surprise that those who made over $50,000 in net capital gains have a 100% probability of having an income listed as more than $50,000. This is because they already earned more than $50,000 in net capital gains alone.

We also see a small number of people had a negative net capital gains. One user managed to make more than $50,000 for the year despite losing nearly $4,000! Also, most people who lost about $2,000 - $3,000 still made more than $50,000 that year. Here is the plot of the percentages by net capital gains:

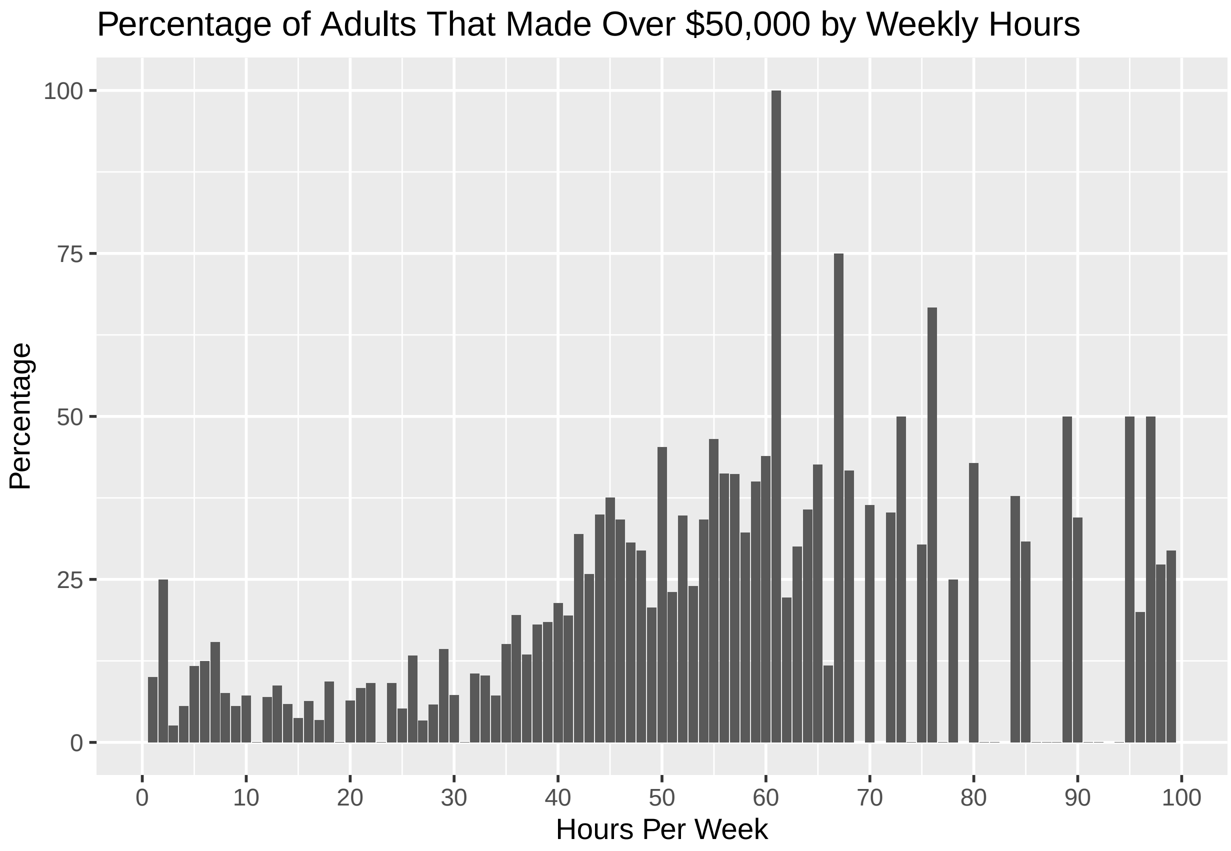
```r
# Plot the percentages by net capital gain
data_net_capital_gains %>%
  filter(net_capital_gain <= 50000) %>%
  ggplot(aes(net_capital_gain, percentage)) +
  geom_bar(stat = "identity") +
  geom_point() +
  ggtitle("Percentage of Adults That Made Over $50,000 by Net Capital Gains") +
  xlab("Net Capital Gain") +
  ylab("Percentage")
```



Exploring the Dataset - Hours Per Week

Intuitively, the more hours one works each week, the more money one makes. Below is the percentage of people who made over $50,000 by the number of hours per week:

```r
# Plot the percentage of people who made over $50k by weekly hours
data %>%
  group_by(hours.per.week) %>%
  summarize(percentage = mean(income == ">50K") * 100) %>%
  ggplot(aes(hours.per.week, percentage)) +
  geom_bar(stat = "identity") +
  ggtitle("Percentage of Adults That Made Over $50,000 by Weekly Hours") +
  xlab("Hours Per Week") +
  ylab("Percentage") +
  scale_x_continuous(labels = seq(0, 100, 10), breaks = seq(0, 100, 10))
```



As expected, we see that those who work more hours were more likely to have made more than $50,000 and vice versa.

Exploring the Dataset - Native Country

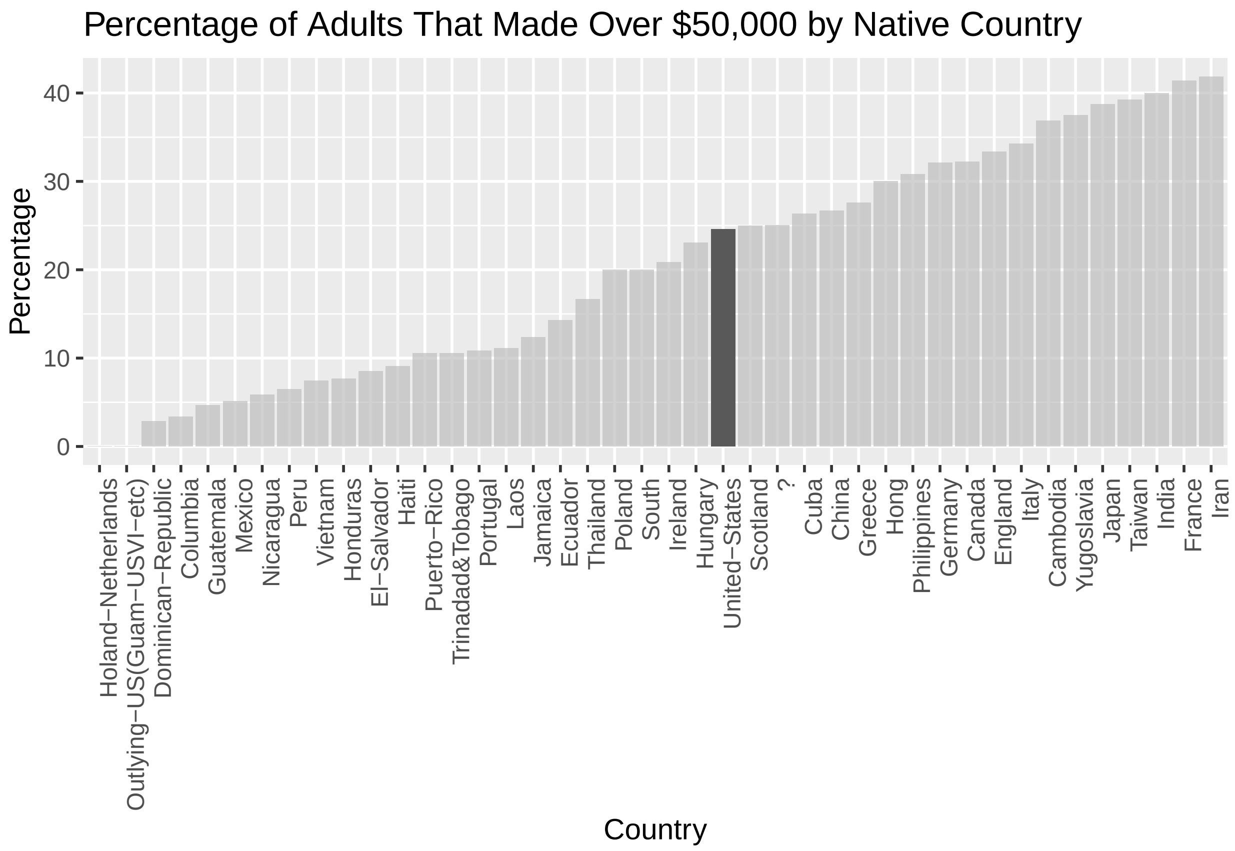
Here are the totals and percentages by country of origin:

```
# Show total and percentages by native country
data_native_countries <- data %>%
  group_by(native.country) %>%
  summarize(total = n(), percentage = mean(income == ">50K") * 100) %>%
  arrange(desc(total))

data_native_countries %>% print(n = Inf)
```

```
## # A tibble: 42 x 3
##    native.country             total percentage
##    <fct>                      <int>      <dbl>
##  1 United-States              29170      24.6
##  2 Mexico                       643       5.13
##  3 ?                            583      25.0
##  4 Philippines                  198      30.8
##  5 Germany                      137      32.1
##  6 Canada                       121      32.2
##  7 Puerto-Rico                  114      10.5
##  8 El-Salvador                  106       8.49
##  9 India                        100      40
## 10 Cuba                          95      26.3
## 11 England                       90      33.3
## 12 Jamaica                       81      12.3
## 13 South                         80      20
## 14 China                         75      26.7
## 15 Italy                         73      34.2
## 16 Dominican-Republic            70       2.86
## 17 Vietnam                       67       7.46
## 18 Guatemala                     64       4.69
## 19 Japan                         62      38.7
## 20 Poland                        60      20
## 21 Columbia                      59       3.39
## 22 Taiwan                        51      39.2
## 23 Haiti                         44       9.09
## 24 Iran                          43      41.9
## 25 Portugal                      37      10.8
## 26 Nicaragua                     34       5.88
## 27 Peru                          31       6.45
## 28 France                        29      41.4
## 29 Greece                        29      27.6
## 30 Ecuador                       28      14.3
## 31 Ireland                       24      20.8
## 32 Hong                          20      30
## 33 Cambodia                      19      36.8
## 34 Trinadad&Tobago               19      10.5
## 35 Laos                          18      11.1
## 36 Thailand                      18      16.7
## 37 Yugoslavia                    16      37.5
## 38 Outlying-US(Guam-USVI-etc)    14       0
## 39 Honduras                      13       7.69
## 40 Hungary                       13      23.1
## 41 Scotland                      12      25
## 42 Holand-Netherlands             1       0
```

As expected, most people in the dataset were born in the US. However, we can see that people from particular countries were more likely to make over $50,000. For example, Germany, Canada, and Cuba. A plot of the percentages for each country is shown below:

```
# Plot the percentages by country
data_native_countries %>%
  mutate(native.country = reorder(native.country, percentage)) %>%
  ggplot(aes(native.country, percentage)) +
  geom_bar(stat = "identity") +
  ggtitle("Percentage of Adults That Made Over $50,000 by Native Country") +
  xlab("Country") +
  ylab("Percentage") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  gghighlight(native.country == "United-States")
```



We can see that those born in the US were not the most likely to make more than $50,000. Out of the countries listed in the dataset, the US sits somewhere in the middle. The countries with the highest percentages were Iran, France, and India.

We also observe that some countries are listed as having a 0% probabaility of making over $50,000. This is not representative of immigrants of those countries collectively, as the data doesn't have a large prevalence of people from those countries.

```
# Show the native countries with the least amount of adults in the dataset
data_native_countries %>%
  arrange(total) %>%
  head(10)
```

```
## # A tibble: 10 x 3
##    native.country             total percentage
##    <fct>                      <int>      <dbl>
##  1 Holand-Netherlands             1       0
##  2 Scotland                      12      25
##  3 Honduras                      13       7.69
##  4 Hungary                       13      23.1
##  5 Outlying-US(Guam-USVI-etc)    14       0  
##  6 Yugoslavia                    16      37.5
##  7 Laos                          18      11.1
##  8 Thailand                      18      16.7
##  9 Cambodia                      19      36.8
## 10 Trinadad&Tobago               19      10.5
```

Only 1 person from the Netherlands was in the dataset and that person didn't make over $50,000. We also see that people from countries such as Cambodia and Yugoslavia had a small prevlance as well, but in particular, they had a higher percentage.

Let's try analyzing this column based on whether the person was born in the US or not. Here are the totals and percentages for both groups:

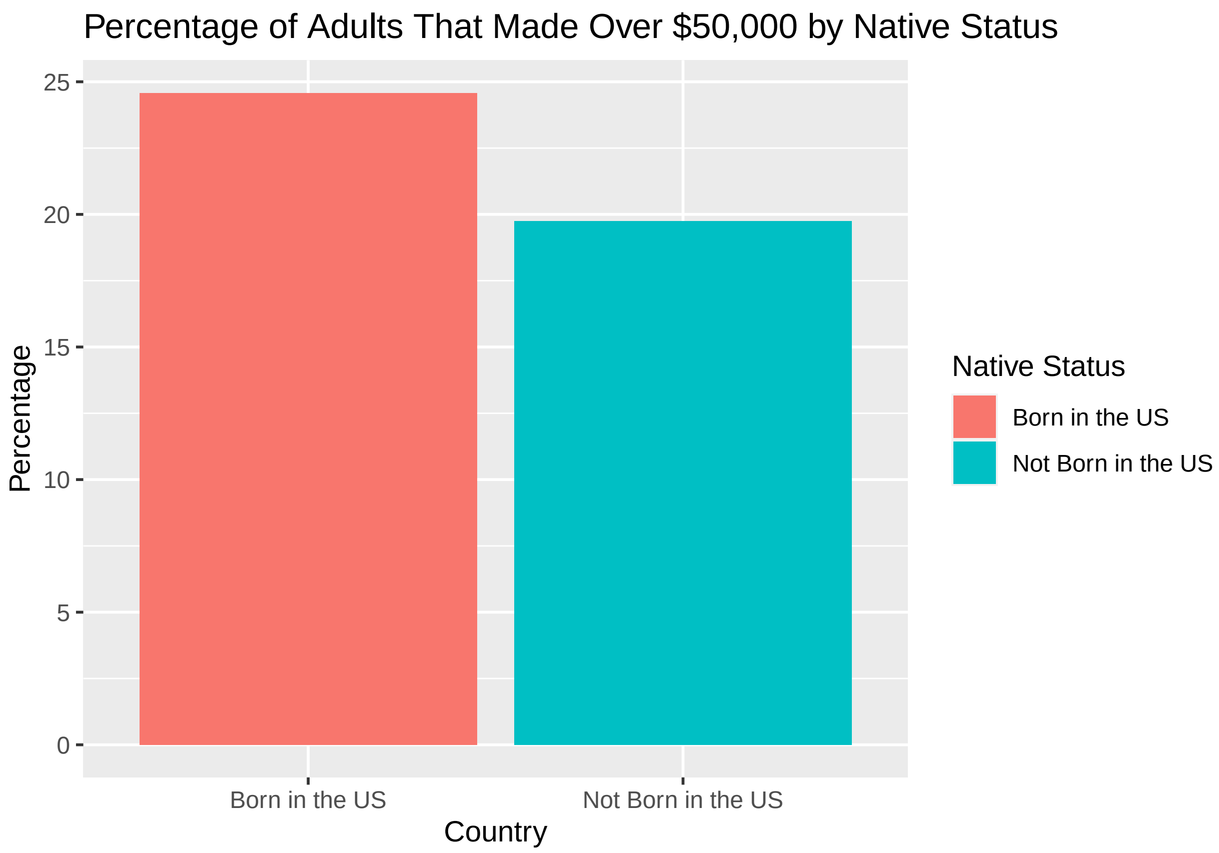
```
# Show the totals and percentages based on whether the adult is born in the US
data_us_born <- data %>%
  mutate(
    is_US_born = factor(
      ifelse(native.country == "United-States", "Born in the US", "Not Born in the US")
    )
  ) %>%
  group_by(is_US_born) %>%
  summarize(total = n(), percentage = mean(income == ">50K") * 100)
data_us_born
```

```
## # A tibble: 2 x 3
##   is_US_born         total percentage
##   <fct>              <int>      <dbl>
## 1 Born in the US     29170       24.6
## 2 Not Born in the US  3391       19.8
```

The dataset suggests that nearly 10% of people in the US were born in another country. We also see that US citizens had a higher probability of making over $50,000, but by nearly 5% more.

A visualization of percentages from the table above is shown below:

```
# Plot the percentages based on whether the adult is born in the US
data_us_born %>%
  ggplot(aes(is_US_born, percentage, fill = is_US_born)) +
  geom_bar(stat = "identity") +
  ggtitle("Percentage of Adults That Made Over $50,000 by Native Status") +
  xlab("Country") +
  ylab("Percentage") +
  labs(fill = "Native Status")
```



Exploring the Dataset - Final Weight

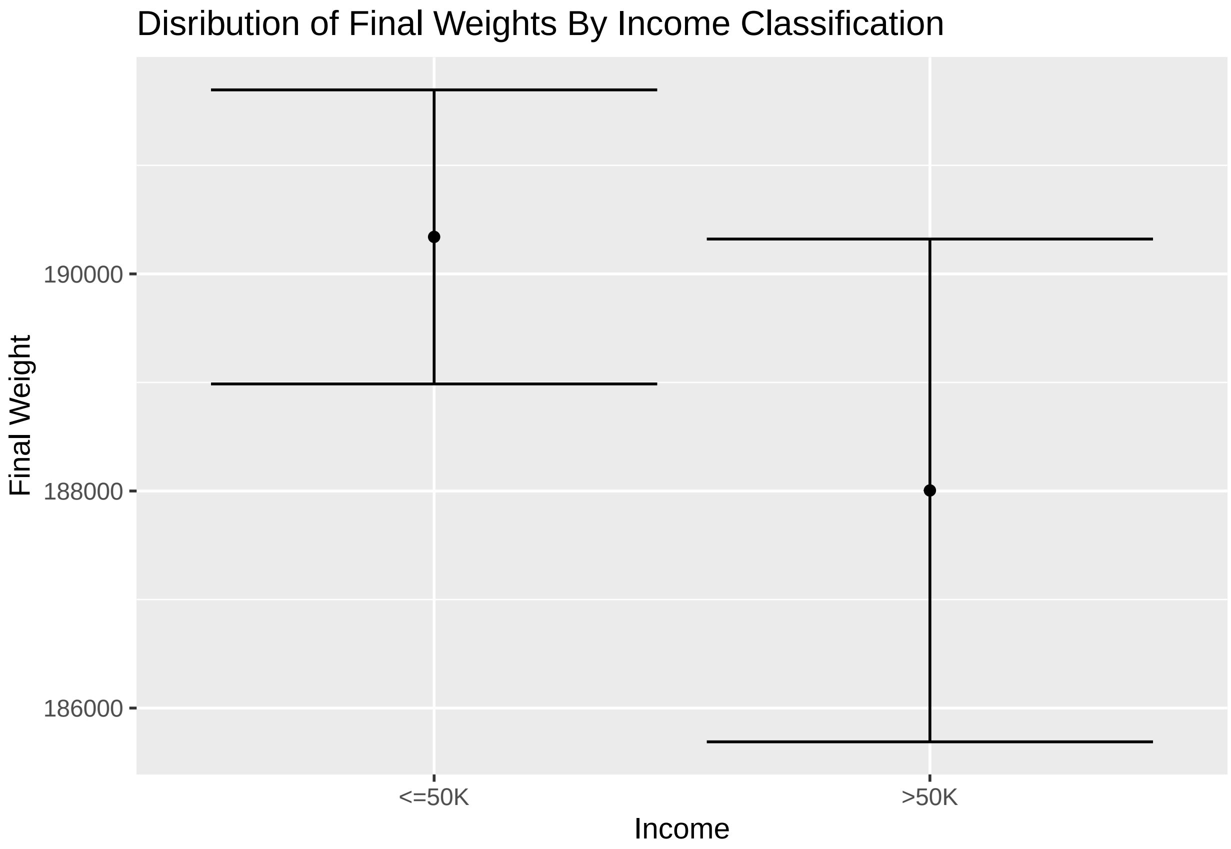
The dataset also provided a column called `fnlwgt`, or final weight. According to Ronny Kohavi and Barry Becker (see https://www.kaggle.com/uciml/adult-census-income/data), people from similar demographics should have similar final weight values. Due to the complexity of this calculation, we will just compare the distrubtion of final weights of those who made over $50,000 to the distribtuion of final weights of those that didn't.

```
# Calculate the mean, standard error, and total number of the final weights by income
data_final_weights <- data %>%
  group_by(income) %>%
  summarize(total = n(),
            proportion = n()/nrow(data),
            avg = mean(fnlwgt),
            se = sd(fnlwgt)/sqrt(n()),
            conf_low = avg - 2 * se,
            conf_high = avg + 2 * se)

data_final_weights
```

```
## # A tibble: 2 x 7
##   income total proportion    avg    se conf_low conf_high
##   <fct>  <int>      <dbl>  <dbl> <dbl>    <dbl>     <dbl>
## 1 <=50K  24720      0.759 190341.  677.  188986.   191695.
## 2 >50K    7841      0.241 188005  1158.  185689.   190321.
```

A visualization of the distributions above is shown below:

```
# Plot the mean and confidence intervals of the final weights by income
data_final_weights %>%
  ggplot(aes(income, avg, ymin = avg - 2 * se, ymax = avg + 2 * se)) +
  geom_point() +
  geom_errorbar() +
  ggtitle("Disribution of Final Weights By Income Classification") +
  xlab("Income") +
  ylab("Final Weight")
```



While there is some overlap, we can see that the averages are outside each other's confidence intervals.

Models

In this section, we try to use the features to generate models that can accurately predict the user's income classification. We will use the logistic regression, QDA, local regression, classification tree, random forest, k-means clustering, and ensemble models in an effort to predict the incomes.

Models - Preparing the Dataset

Before continuing, let's convert the columns into numerical values. This will be necessary for one of the models. Since the columns that are not numbers are already factors, the conversion should be simple. We will also add a net capital gains columns and remove columns that are redundant, such as education number, capital gains, and capital losses.

```
# Convert the columns to numerical values instead of factors.
# Then remove the columns that won't be used for the models
data <- data %>%
  mutate(workclass = as.numeric(workclass),
         fnlwgt = as.numeric(fnlwgt),
         education = as.numeric(education),
         marital.status = as.numeric(marital.status),
         occupation = as.numeric(occupation),
         relationship = as.numeric(relationship),
         race = as.numeric(race),
         sex = as.numeric(sex),
         net_capital_gain = as.numeric(capital.gain - capital.loss),
         hours.per.week = as.numeric(hours.per.week),
         native.country = as.numeric((native.country))
  ) %>%
  select(-c(education.num, capital.gain, capital.loss))
```

Models - Training & Test Sets

For this project, we will split the dataset into a training set, which will consist of 80% of the rows, and a test set, which consists of the remaining 20%. This should provide enough test cases to determine accuracy while providing enough training data for the models.

```
# Split the data into a training set (80%) and a test set (20%)
set.seed(1, sample.kind = "Rounding")
test_index <- createDataPartition(data$income, times = 1, p = 0.2, list = FALSE)
train_set <- data[-test_index,]
test_set <- data[test_index,]
rm(test_index)
```

The proportion of incomes less than or equal to $50,000 in the training set and test set is 0.7592138 and 0.7590972 respectively. Both sets have about the same proportion of income types.

For our baseline model, we will assume that everyone made under $50,000. While we would achieve an accuracy of 75.909719%, we would have specificity of 0%. In other words, everyone who made over $50,000 would be incorrectly predicted to have made $50,000 or less.

Models - Logistic Regression

The first model used was the logistic regression model, which is an improvement over the baseline model. However, it can be improved much more. The following code generates the model, makes the predictions, and displays the results.

```
# Train the model
set.seed(1, sample.kind = "Rounding")
train_glm <- train(income ~ .,
                   method = "glm",
                   data = train_set)
```

```
# Make the predictions
y_hat_glm <- predict(train_glm, test_set)

# Determine accuracy of the model
results_glm <- confusionMatrix(data = y_hat_glm, reference = test_set$income)
results_glm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction <=50K >50K
##      <=50K  4721 1175
##      >50K    223  394
##
##                Accuracy : 0.7854
##                  95% CI : (0.7752, 0.7953)
##     No Information Rate : 0.7591
##     P-Value [Acc > NIR] : 2.829e-07
##
##                   Kappa : 0.2598
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9549
##             Specificity : 0.2511
##          Pos Pred Value : 0.8007
##          Neg Pred Value : 0.6386
##              Prevalence : 0.7591
##          Detection Rate : 0.7249
##    Detection Prevalence : 0.9053
##       Balanced Accuracy : 0.6030
##
##        'Positive' Class : <=50K
##
```

Models - Quadratic Discriminant Analysis (QDA)

After the logistic model, we then tried using a quadratic discriminant analysis, or QDA, to predict the incomes. We also make small improvements here as well. The following code generates the model, makes the predictions, and displays the results.

```
# Train the model
set.seed(1, sample.kind = "Rounding")
train_qda <- train(income ~ .,
                   method = "qda",
                   data = train_set)

# Make the predictions
y_hat_qda <- predict(train_qda, test_set)

# Determine accuracy of the model
results_qda <- confusionMatrix(data = y_hat_qda, reference = test_set$income)
results_qda
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction <=50K >50K
##      <=50K  4774 1172
##      >50K    170  397
##
##                Accuracy : 0.794
##                  95% CI : (0.7839, 0.8037)
##     No Information Rate : 0.7591
##     P-Value [Acc > NIR] : 1.22e-11
##
##                   Kappa : 0.2796
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9656
##             Specificity : 0.2530
##          Pos Pred Value : 0.8029
##          Neg Pred Value : 0.7002
##              Prevalence : 0.7591
##          Detection Rate : 0.7330
##    Detection Prevalence : 0.9129
##       Balanced Accuracy : 0.6093
##
##        'Positive' Class : <=50K
##
```

Models - Local Regression (Loess)

Using local regression, we managed to achieve 80% accuracy. Once again, this model improves the accuracy, but only slightly. The following code generates the model, makes the predictions, and displays the results.

```
# Train the model
set.seed(1, sample.kind = "Rounding")
train_loess <- train(income ~ .,
                     method = "gamLoess",
                     data = train_set)

# Make the predictions
y_hat_loess <- predict(train_loess, test_set)

# Determine accuracy of the model
results_loess <- confusionMatrix(data = y_hat_loess, reference = test_set$income)
results_loess
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction <=50K >50K
##      <=50K  4607  920
##       >50K    337  649
##
##                Accuracy : 0.807
##                  95% CI : (0.7972, 0.8165)
##     No Information Rate : 0.7591
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3957
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9318
##             Specificity : 0.4136
##          Pos Pred Value : 0.8335
##          Neg Pred Value : 0.6582
##              Prevalence : 0.7591
##          Detection Rate : 0.7074
##    Detection Prevalence : 0.8486
##       Balanced Accuracy : 0.6727
##
##        'Positive' Class : <=50K
##
```

Models - Classification Tree

The classification tree significantly improved the accuracy by nearly 5%. The following code generates the model, makes the predictions, and displays the results.

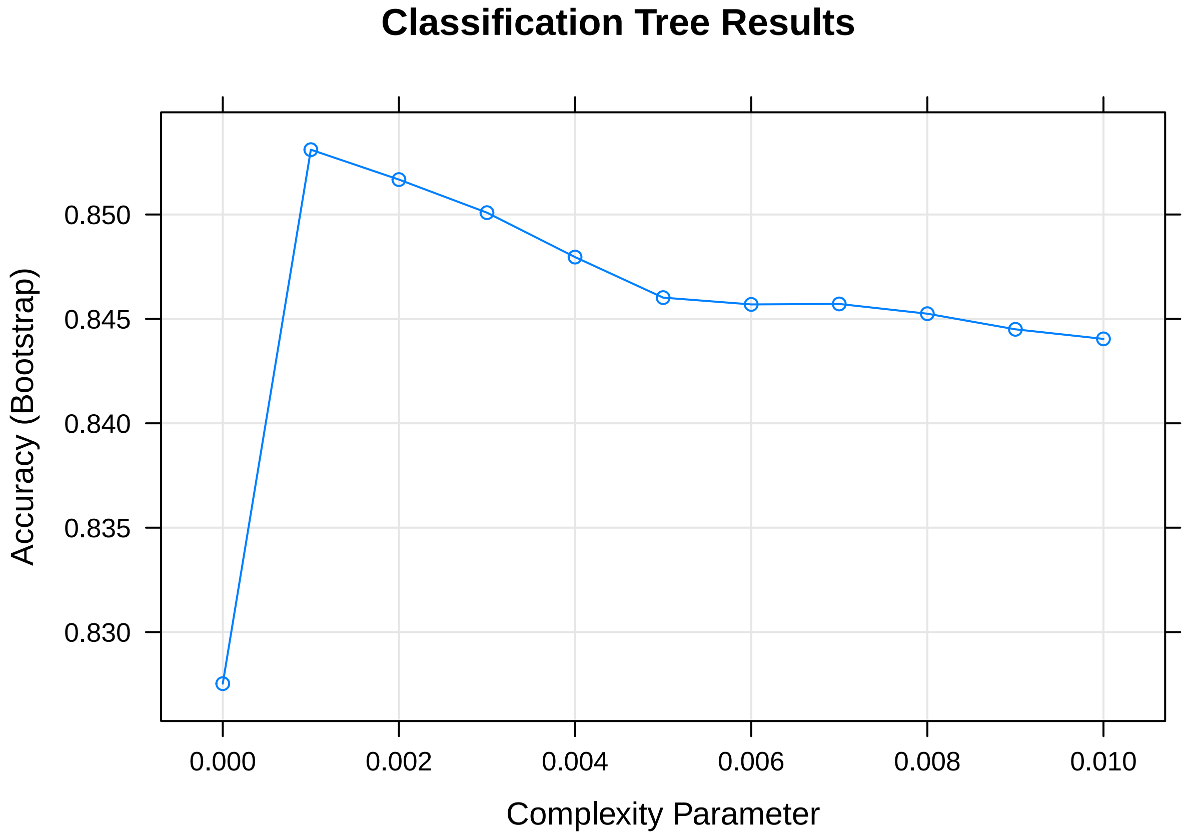
```
# Train the model
set.seed(1, sample.kind = "Rounding")
train_ct <- train(income ~ .,
                  method = "rpart",
                  data = train_set,
                  tuneGrid = data.frame(cp = seq(0, 0.01, 0.001)))

# Make the predictions
y_hat_ct <- predict(train_ct, test_set)

# Determine accuracy of the model
results_ct <- confusionMatrix(data = y_hat_ct, reference = test_set$income)
results_ct
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction <=50K >50K
##      <=50K  4658  682
##       >50K    286  887
##
##                Accuracy : 0.8514
##                  95% CI : (0.8425, 0.8599)
##     No Information Rate : 0.7591
##     P-Value [Acc > NIR] : < 2.2e-16
```

```
##
##                   Kappa : 0.5553
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9422
##             Specificity : 0.5653
##          Pos Pred Value : 0.8723
##          Neg Pred Value : 0.7562
##              Prevalence : 0.7591
##          Detection Rate : 0.7152
##    Detection Prevalence : 0.8199
##       Balanced Accuracy : 0.7537
##
##        'Positive' Class : <=50K
##
```

We can identify the optimal parameter value used and compare the accuracy obtained from that value to accuracies from other parameter values.

```
# Plot the model's accuracy for each complexity parameter
plot(train_ct, main = "Classification Tree Results")
```



```
# Show the most optimal paramater value
train_ct$bestTune
```

```
##      cp
## 2 0.001
```

We see that the best complexity paramater value is 0.001.

Models - Random Forest

After seeing the results of the classification tree, it was worth trying the random forest model to see if the accuracy improves even more. Using 200 trees, we see a slight improvement, however the model is close to an accuracy of 86%. The following code generates the model, makes the predictions, and displays the results.

```
# Train the model
# NOTE: This will take roughly 40 minutes to complete
set.seed(1, sample.kind = "Rounding")
train_rf <- train(income ~ .,
                  method = "rf",
                  data = train_set,
                  ntree = 200,
                  tuneGrid = data.frame(mtry = 1:5),
                  importance = TRUE)

# Make the predictions
y_hat_rf <- predict(train_rf, test_set)

# Determine accuracy of the model
results_rf <- confusionMatrix(data = y_hat_rf, reference = test_set$income)
results_rf
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction <=50K >50K
##      <=50K  4694  663
##      >50K    250  906
##
##                Accuracy : 0.8598
##                  95% CI : (0.8511, 0.8682)
##     No Information Rate : 0.7591
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5789
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9494
##             Specificity : 0.5774
##          Pos Pred Value : 0.8762
##          Neg Pred Value : 0.7837
##              Prevalence : 0.7591
##          Detection Rate : 0.7207
##    Detection Prevalence : 0.8225
##       Balanced Accuracy : 0.7634
##
```

```
##        'Positive' Class : <=50K
##
```

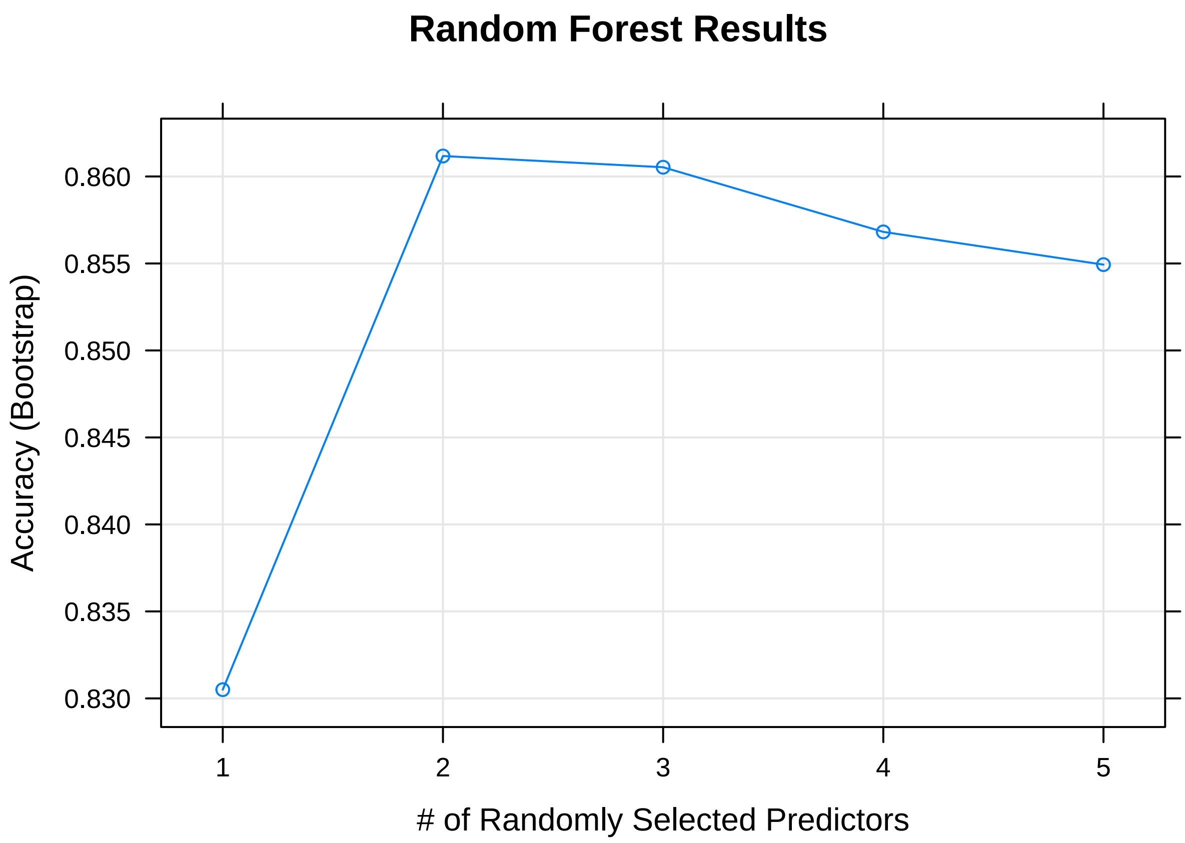
Here are the most important variables for this model. We can see that `net_capital_gain` is listed as the most important variable in the model, followed by education, occupation, age, hours per week, and marital status.

```r
# Show the most important variables in the model
varImp(train_rf)
```

```
## rf variable importance
##
##                  Importance
## net_capital_gain    100.000
## education            40.594
## occupation           35.726
## age                  29.530
## hours.per.week       26.306
## marital.status       24.929
## relationship         23.127
## workclass            19.588
## race                  5.690
## sex                   4.065
## fnlwgt                1.431
## native.country        0.000
```

We can see how the model performs across different number of predictors. In this model, we chose to use 1 through 5.

```r
# Plot the model and the accuracies for each predictor
plot(train_rf,
     main = "Random Forest Results",
     xlab = "# of Randomly Selected Predictors"
)
```



```
# Show the most optimal paramater value
train_rf$bestTune
```

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

We see that the most optimal number of predictors for this model is 2.

Models - K-Means Clustering

The k-means clustering model is the only unsupervised model used in this project. Unlike the previous models, this model fails to perform at least as well as the baseline model. The following code generates the model, makes the predictions, and displays the results.

```
# Train the model
set.seed(1, sample.kind = "Rounding")
train_kmeans <- kmeans(select(train_set, -income), centers = 3)

# Prediction function for the k-means clustering model
# Assigns each row to a cluster from k_means
predict_kmeans <- function(predictors, k_means) {
  # Get cluster centers
  centers <- k_means$centers
  # Calculate the distance from the cluster centers
  distances <- sapply(1:nrow(predictors), function(i) {
    apply(centers, 1, function(y) dist(rbind(predictors[i,], y)))
  })

  # Select the cluster that is closest to the center
  max.col(-t(distances))
}

# Make the predictions
y_hat_kmeans <- factor(ifelse(predict_kmeans(select(test_set, -income), train_kmeans) == 2, ">50K", "<=5

# Determine accuracy of the model
results_kmeans <- confusionMatrix(data = y_hat_kmeans, reference = test_set$income)
results_kmeans
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction <=50K >50K
##      <=50K  3157  970
##      >50K   1787  599
##
##                Accuracy : 0.5767
##                  95% CI : (0.5646, 0.5887)
##     No Information Rate : 0.7591
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0173
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.6386
##             Specificity : 0.3818
##          Pos Pred Value : 0.7650
##          Neg Pred Value : 0.2510
##              Prevalence : 0.7591
##          Detection Rate : 0.4847
##    Detection Prevalence : 0.6337
##       Balanced Accuracy : 0.5102
##
##        'Positive' Class : <=50K
##
```

Models - Ensemble

Using the previous models (except for the k-means clustering model), we use the predictions generated from each of the models to predict the incomes. It managed to achieve an accuracy of almost 84%. The following code generates the model, makes the predictions, and displays the results.

```
# Create the ensemble
ensemble <- data.frame(glm = y_hat_glm,
                       qda = y_hat_qda,
                       loess = y_hat_loess,
                       ct = y_hat_ct,
                       kmeans = y_hat_rf)

# Make the predictions
y_hat_ensemble <- factor(ifelse(rowMeans(ensemble == ">50K") > 0.5, ">50K", "<=50K"))

# Determine accuracy of the model
results_ensemble <- confusionMatrix(data = y_hat_ensemble, reference = test_set$income)
results_ensemble
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction <=50K >50K
##      <=50K  4793  893
##      >50K    151  676
##
##                Accuracy : 0.8397
##                  95% CI : (0.8306, 0.8485)
##     No Information Rate : 0.7591
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4774
##
##  Mcnemar’s Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9695
##             Specificity : 0.4308
##          Pos Pred Value : 0.8429
##          Neg Pred Value : 0.8174
##              Prevalence : 0.7591
##          Detection Rate : 0.7359
##    Detection Prevalence : 0.8730
##       Balanced Accuracy : 0.7002
##
##        ’Positive’ Class : <=50K
##
```

Results

We can condense the results of all the models into a table, where we can compare the models.

```
# Save the model names
models = c(
  "Logistic Regression",
  "QDA",
  "Loess",
  "Classification Tree",
  "Random Forest",
  "K-Means Clustering",
```

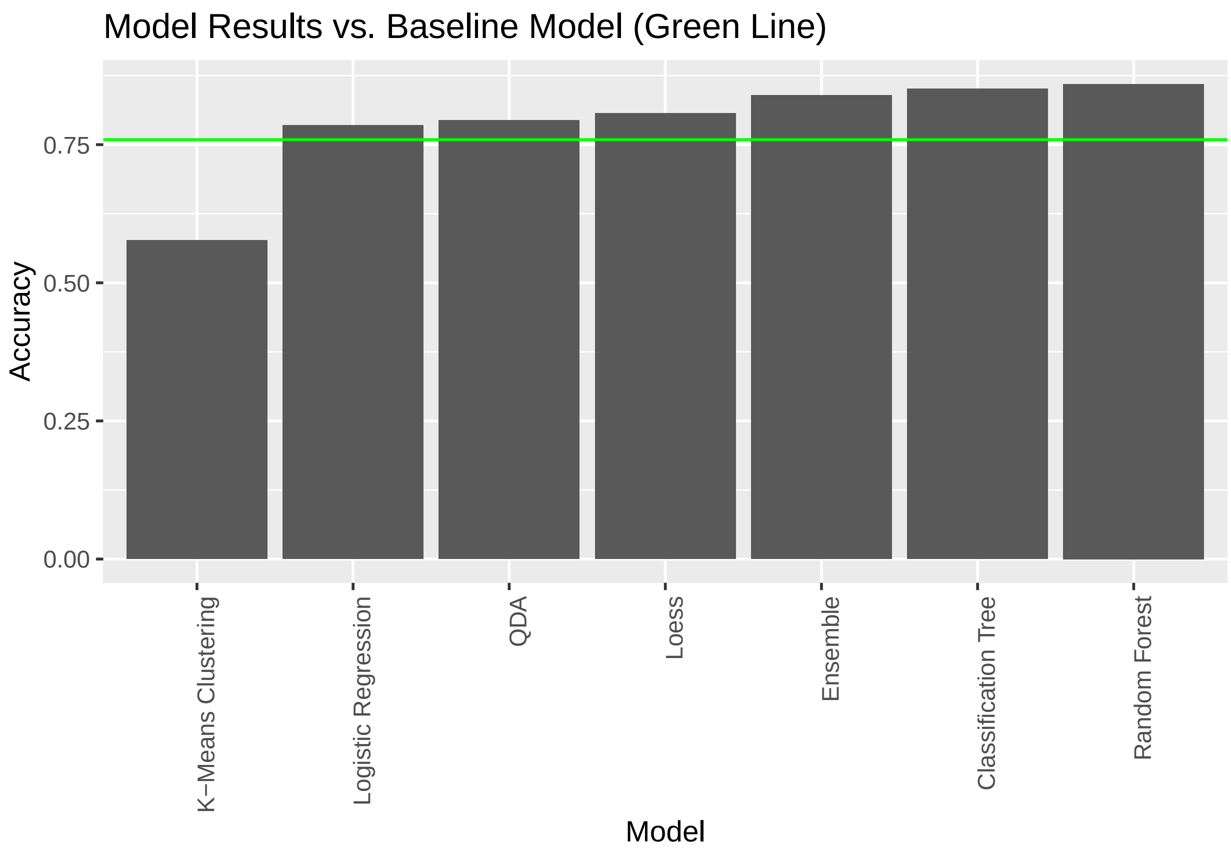
```r
  "Ensemble"
)

# Save the model accuracies
accuracies = c(
  mean(test_set$income == y_hat_glm),
  mean(test_set$income == y_hat_qda),
  mean(test_set$income == y_hat_loess),
  mean(test_set$income == y_hat_ct),
  mean(test_set$income == y_hat_rf),
  mean(test_set$income == y_hat_kmeans),
  mean(test_set$income == y_hat_ensemble)
)

# Save the model sensitivities
sensitivities = c(
  sensitivity(data = y_hat_glm, reference = test_set$income),
  sensitivity(data = y_hat_qda, reference = test_set$income),
  sensitivity(data = y_hat_loess, reference = test_set$income),
  sensitivity(data = y_hat_ct, reference = test_set$income),
  sensitivity(data = y_hat_rf, reference = test_set$income),
  sensitivity(data = y_hat_kmeans, reference = test_set$income),
  sensitivity(data = y_hat_ensemble, reference = test_set$income)
)

# Save the model specificities
specificities = c(
  specificity(data = y_hat_glm, reference = test_set$income),
  specificity(data = y_hat_qda, reference = test_set$income),
  specificity(data = y_hat_loess, reference = test_set$income),
  specificity(data = y_hat_ct, reference = test_set$income),
  specificity(data = y_hat_rf, reference = test_set$income),
  specificity(data = y_hat_kmeans, reference = test_set$income),
  specificity(data = y_hat_ensemble, reference = test_set$income)
)

# Save the model precision
precisions = c(
  precision(data = y_hat_glm, reference = test_set$income),
  precision(data = y_hat_qda, reference = test_set$income),
  precision(data = y_hat_loess, reference = test_set$income),
  precision(data = y_hat_ct, reference = test_set$income),
  precision(data = y_hat_rf, reference = test_set$income),
  precision(data = y_hat_kmeans, reference = test_set$income),
  precision(data = y_hat_ensemble, reference = test_set$income)
)

# Save the model F1 scores
F1s = c(
  F_meas(data = y_hat_glm, reference = test_set$income),
  F_meas(data = y_hat_qda, reference = test_set$income),
  F_meas(data = y_hat_loess, reference = test_set$income),
  F_meas(data = y_hat_ct, reference = test_set$income),
  F_meas(data = y_hat_rf, reference = test_set$income),
  F_meas(data = y_hat_kmeans, reference = test_set$income),
  F_meas(data = y_hat_ensemble, reference = test_set$income)
)

# Combine the results into a data frame, then display them
results <- data.frame(
  Model = models,
  Accuracy = accuracies,
  Sensitivity = sensitivities,
  Specificity = specificities,
  Precision = precisions,
  F1 = F1s
  )

results
```

```
##                  Model  Accuracy Sensitivity Specificity Precision        F1
## 1 Logistic Regression 0.7853524   0.9548948   0.2511154 0.8007123 0.8710332
## 2                 QDA 0.7939506   0.9656149   0.2530274 0.8028927 0.8767677
## 3               Loess 0.8070014   0.9318366   0.4136393 0.8335444 0.8799542
## 4 Classification Tree 0.8513742   0.9421521   0.5653282 0.8722846 0.9058732
## 5       Random Forest 0.8598188   0.9494337   0.5774379 0.8762367 0.9113678
## 6  K-Means Clustering 0.5766928   0.6385518   0.3817718 0.7649624 0.6960644
## 7            Ensemble 0.8397052   0.9694579   0.4308477 0.8429476 0.9017874
```

We can see that the **random forest** model had the highest accuracy, specificity, precision, and F1 score. The accuracy of the model was **85.98%**. It didn't have the highest sensitivity, although the differences are comparatively small across most models. The QDA model had the highest sensitivity, but it had one of lowest specificities. We can also see that the k-means clustering model performed poorly overall.

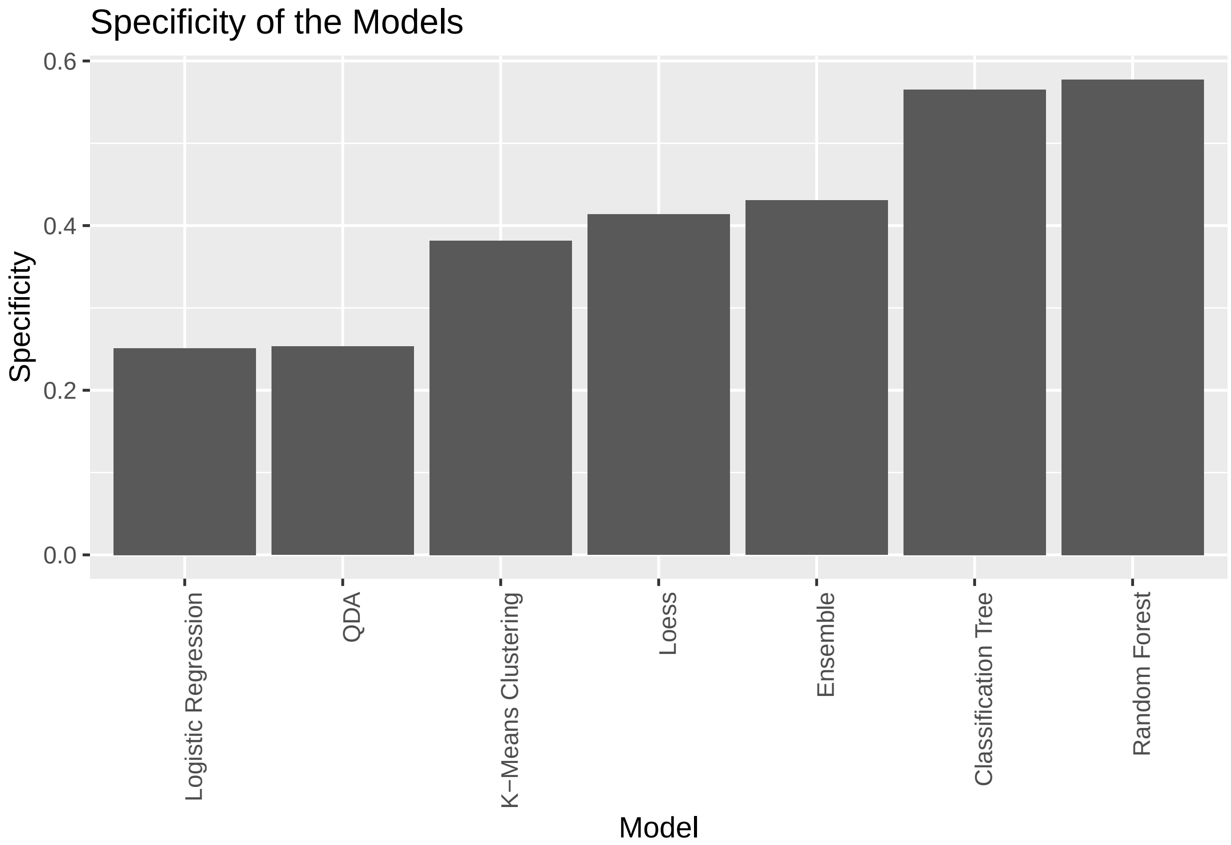
The following graph shows the accuracies of all the models and how they compare to the baseline model (0.7590972).

```
# Plot the accuracies of each model
results %>%
  mutate(Model = reorder(Model, Accuracy)) %>%
  ggplot(aes(Model, Accuracy)) +
  geom_bar(stat = "identity") +
  ggtitle("Model Results vs. Baseline Model (Green Line)") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  geom_hline(yintercept = mean(test_set$income == "<=50K"), color = "green")
```



The most variability observed from the results is from specificity. The range of values extend from about 25% to about 58%, a 33% difference! The graph below visualizes the specificities for all the models.

```
results %>%
  mutate(Model = reorder(Model, Specificity)) %>%
  ggplot(aes(Model, Specificity)) +
  geom_bar(stat = "identity") +
  ggtitle("Specificity of the Models") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



Conclusion

It was discovered that people who were about 50 years old had the highest probability of making over $50,000 than the other age groups. It also seems that people who had government jobs or were self-employed incorporated had a better chance of making more money than those in the private sector. Surprisingly, the dataset suggests that men had a higher probability of making over $50,000 than women, although the reason behind this is unclear. It is also noted that those of Asian/Pacific Island descent had the highest probability despite having a small prevalance. Another surprising observation was that US citizens didn't have the highest probability. The top 3 probabilities by ethnic groups were Iranian, French, and Indian.

When predicting the incomes, the random forest model performed the best overall. It determined that net capital gain was the most important variable when predicting incomes. Education, occupation, age, hours per week, and marital status were also among one of the most important variables as well. Immutable characteristics such as race and sex were not considered to be as important, according to the model.

The findings indicate that personal choices are one of the biggest determinants of income. Those that pursued a higher education, were married, worked more hours, worked in higher-paying occupations, and invested in capital were more likely to earn over $50,000 in 1994.

An important note to consider is that the data is over 25 years old. However, it is likely that these observations can still be utilized and applied today. For instance, investing, pursuing a higher education and working more hours all can improve one's chances of making more than $50,000.