ACIProject

Mandy Simpson

18/10/2019

Predicting income from census data

Introduction

The project is part of the HarvardX Profeessional Certificate in Data Science Course, PH:125.9x - Capstone.

For the self-directed project I chose to explore the Adult Census Income dataset ("ACI") from University of California at Irvine, which can be found at https://www.kaggle.com/uciml/adult-census-income. This dataset was originally derived from the US Census Bureau 1994 database. It includes an indicator column showing whether the individual described by that entry earned under or over $50k. It is this indicator that I am seeking to predict using the other information in the dataset.

I started the project by exploring the full dataset to ensure I understood the information contained. While the data was already broadly clean, and in tidy format, I removed some rows with missing data, and some unproductive attributes.

Once this was complete, I split the dataset into training and validation sets (80%/20%), and then split the training set further into training and test sets (also 80%/20%), to allow a number of algorithms to be trialled on this data before selecting one for use on the validation set.

I tried the k-nearest neighbours, classification tree and random forest algorithms, before settling on the random forest algorithm for use on the validation set.

Downloading and preprocessing the Adult Census Income dataset

The file can be downloaded from github at https://github.com/mandysimpson/adult-census-income/raw/master/adult.csv

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

#download file
adult_census_income <- read.csv("https://github.com/mandysimpson/adult-census-income/raw/master/adult.c
```

The ACI dataset has 32561 observations each representing an individual person in the census.

There are 15 attributes as follows:

Name	Type	Description
age	Integer	Age of respondent
workclass	Factor (9 levels)	Employer type
fnlwgt	Integer	Population weighting factor

Name	Type	Description
education	Factor (16 levels)	Highest education level achieved
education.num	Integer	Numerical representation of education
marital.status	Factor (7 levels)	Marital Status of respondent
occupation	Factor (15 levels)	Type of employment
relationship	Factor (6 levels)	Position in household
race	Factor (5 levels)	Race of respondent
sex	Factor (2 levels)	Sex of respondent
capital.gain	Integer	Investment gains
capital.loss	Integer	Investment losses
hours.per.week	Integer	Working hours
native.country	Factor (42 levels)	Country of birth
income	Factor (2 levels)	Income under or over $50k

A number of observations include missing data in the `workclass`, `occupation` and `native.country` attributes - represented by a "?". I removed these from the data set.

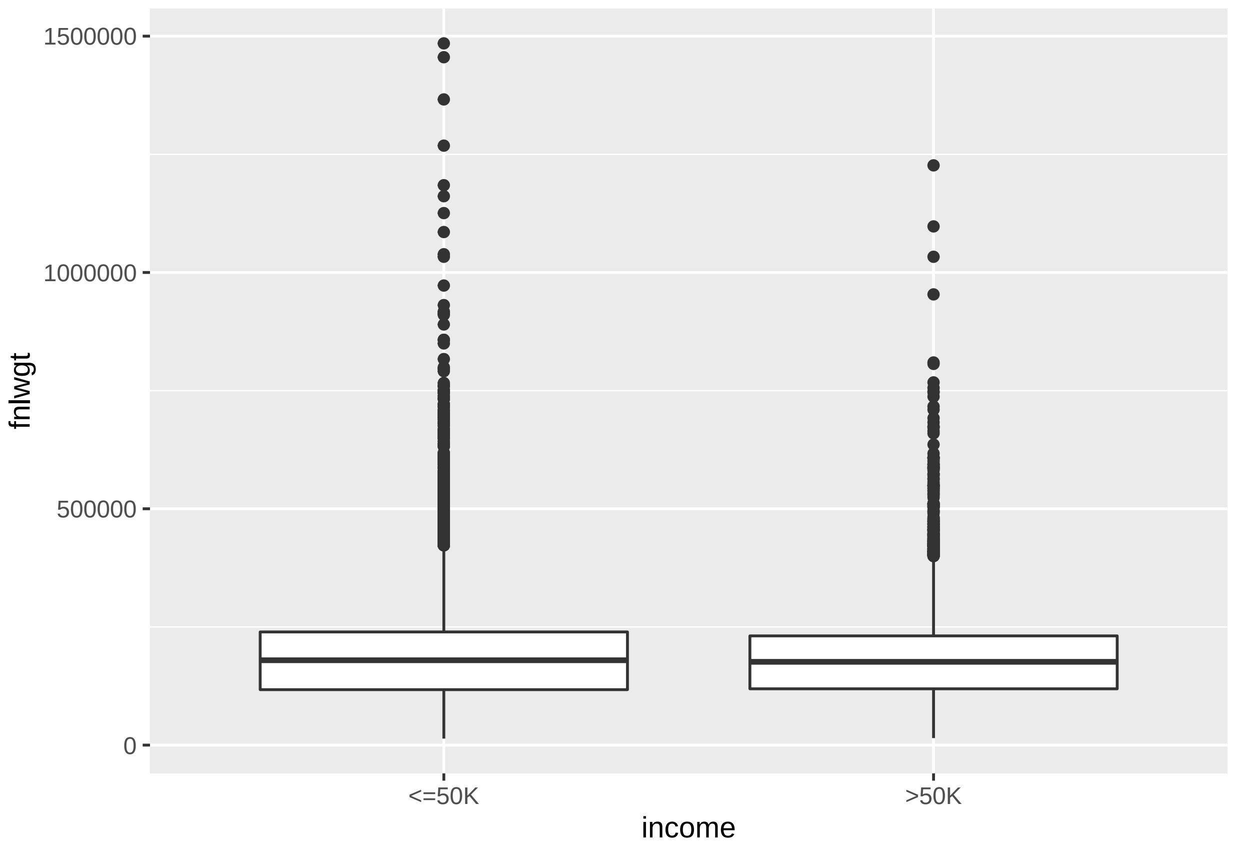
```r
aci <- filter(adult_census_income, !workclass == "?", !occupation == "?", !native.country == "?")
aci <- droplevels(aci)
```

This removes 2399 rows, leaving 30162.

For observations remaining in the dataset, 24.89% indicate an income of over 50k.

Deleting unnecessary attributes

The attribute `fnlwgt` is a measure of the units of population represented by this observation. As such it doesn't seem likely to be relevant for determining income. To check this I plotted fnlwgt against income.



There appears to be very little difference, and so I deleted this attribute. On inspection, the attributes `education` and `education.num` appear to be the same, with `education.num` being a numerical representation of `education`. I therefore also deleted `education`.

```
aci <- aci %>% select(-c(fnlwgt,education))
```

I next ran the `nearZeroVar` function of the `caret` package to understand which attributes could be removed due to minimal variance.

```
nzv <- nearZeroVar(aci)
nzv
```

```
## [1]  9 10 12
```

The function recomends the removal of the `capital.gain`, `capital.loss` and `native.country` attributes. On inspection we can see that 91.59% of the `capital.gain` values and 95.27% of the `capital.loss` values are actually 0, and that 91.19% of the `native.country` values are "United-States".

I therefore removed these attributes from the dataset.

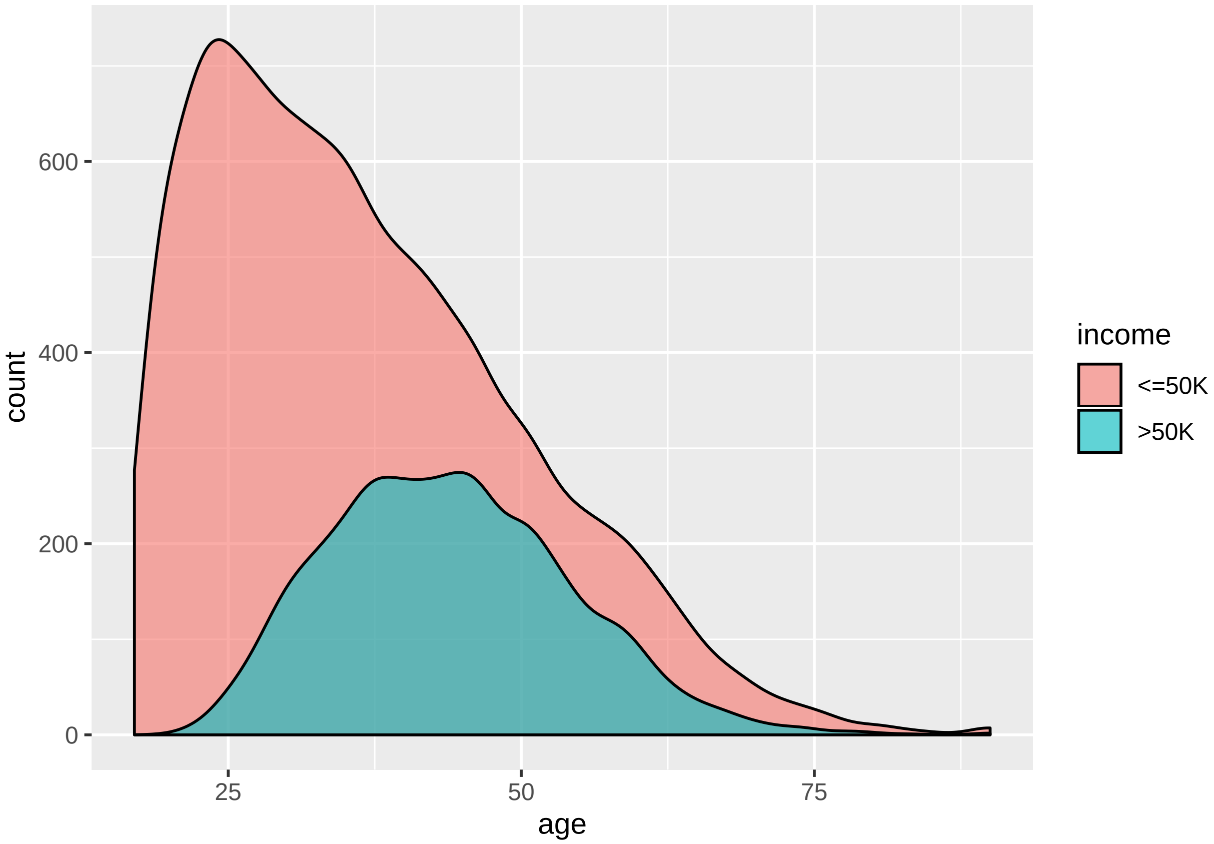
```
aci <- aci %>% select(-c(capital.gain,capital.loss,native.country))
```

We're left with 9 predictors for the income attribute.

Visualising the data

Age

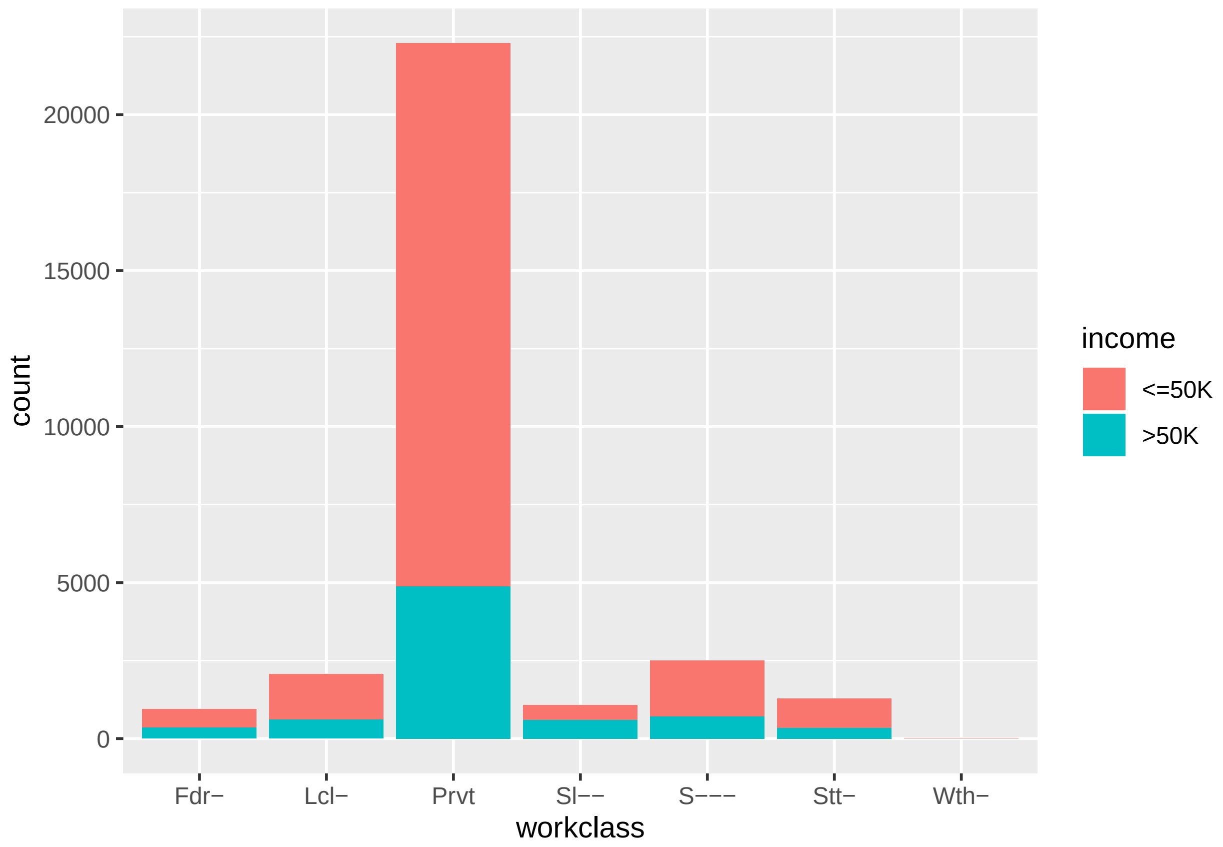
I created a density plot showing the count of age range split by income.



We can see significant differences in the data. While there is no age at which the proportion earning over 50k increases above 50%, the proportion does increase with age. For under 25's the proportion is just 1.19% while for those between 45-49 it is 39.49%.

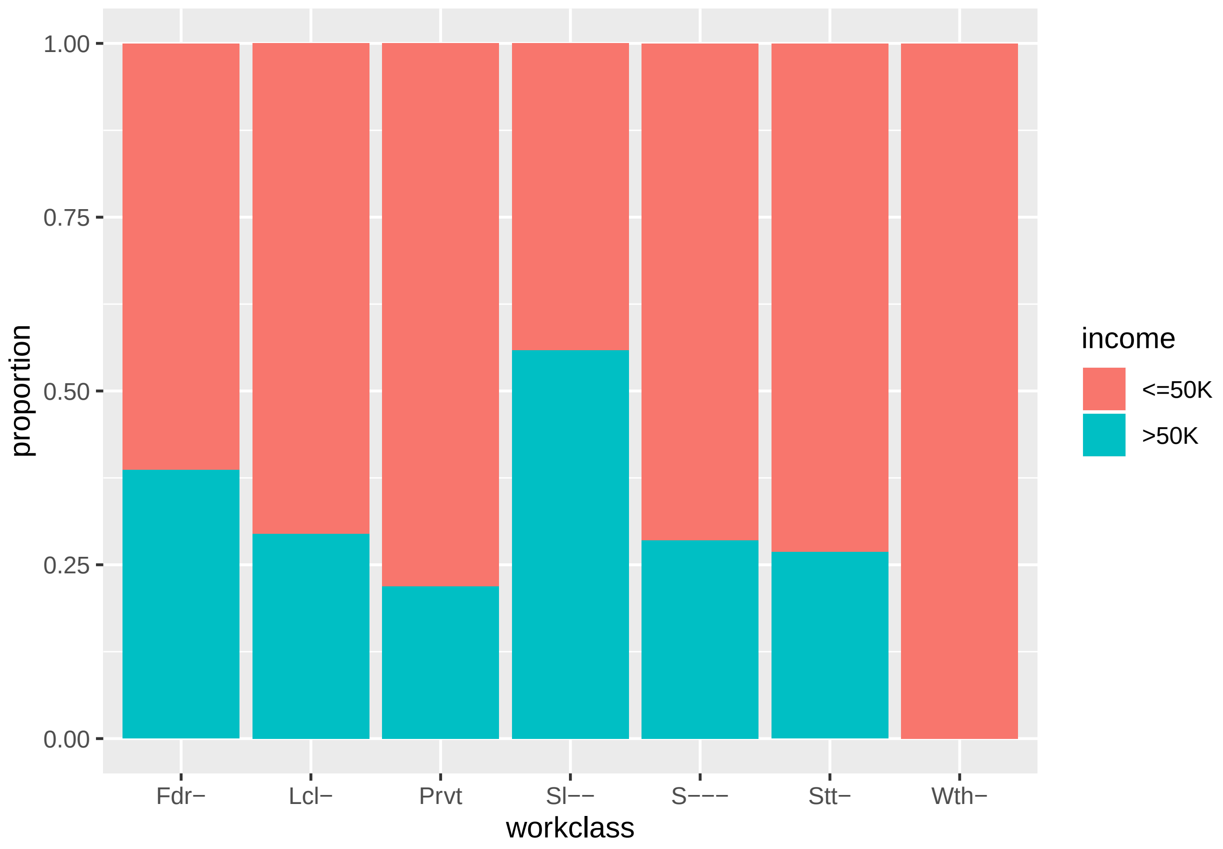
Workclass

For the workclass attribute I used a bar graph showing the split of each workclass factor by income.



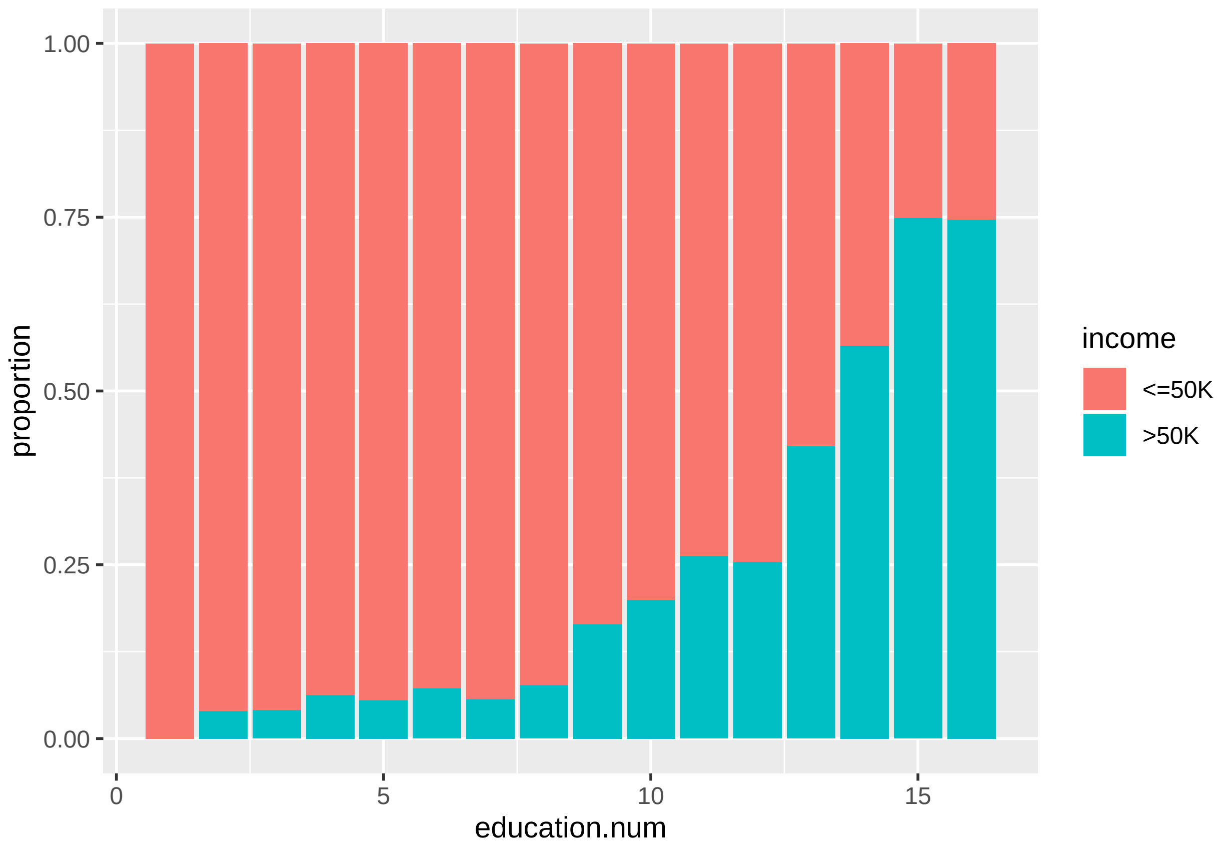
The workclass "Private" is by far the biggest, making up 73.89% of the total.

Drawing this bar graph again by proportion shows that greater than 50% of the `Self-emp-inc` group has income over 50k - in fact this is 55.87%



Education

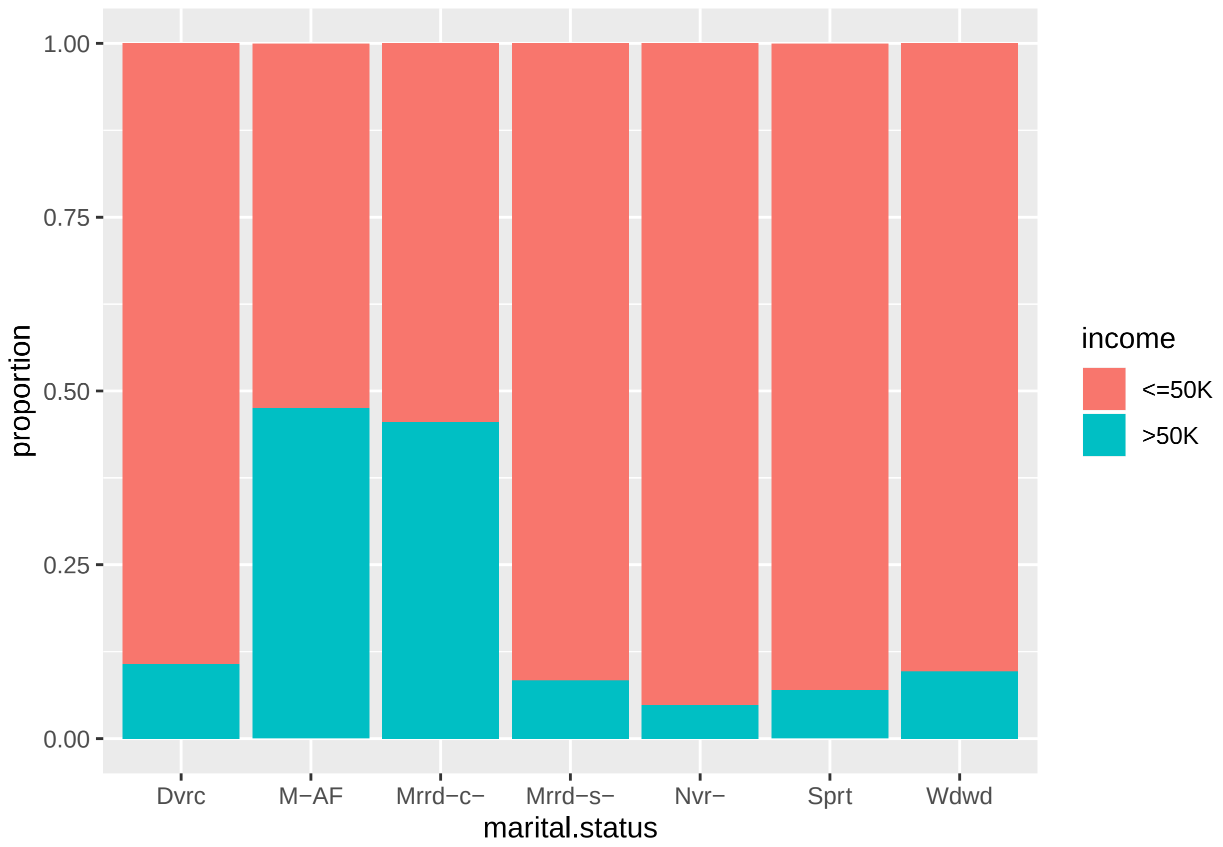
For the education attribute I used a bar graph showing the split of each education level by income, in this case using proportion view straight away.



Unsurprisingly as education levels increase so does the proportion of people earning over 50k.

Marital Status

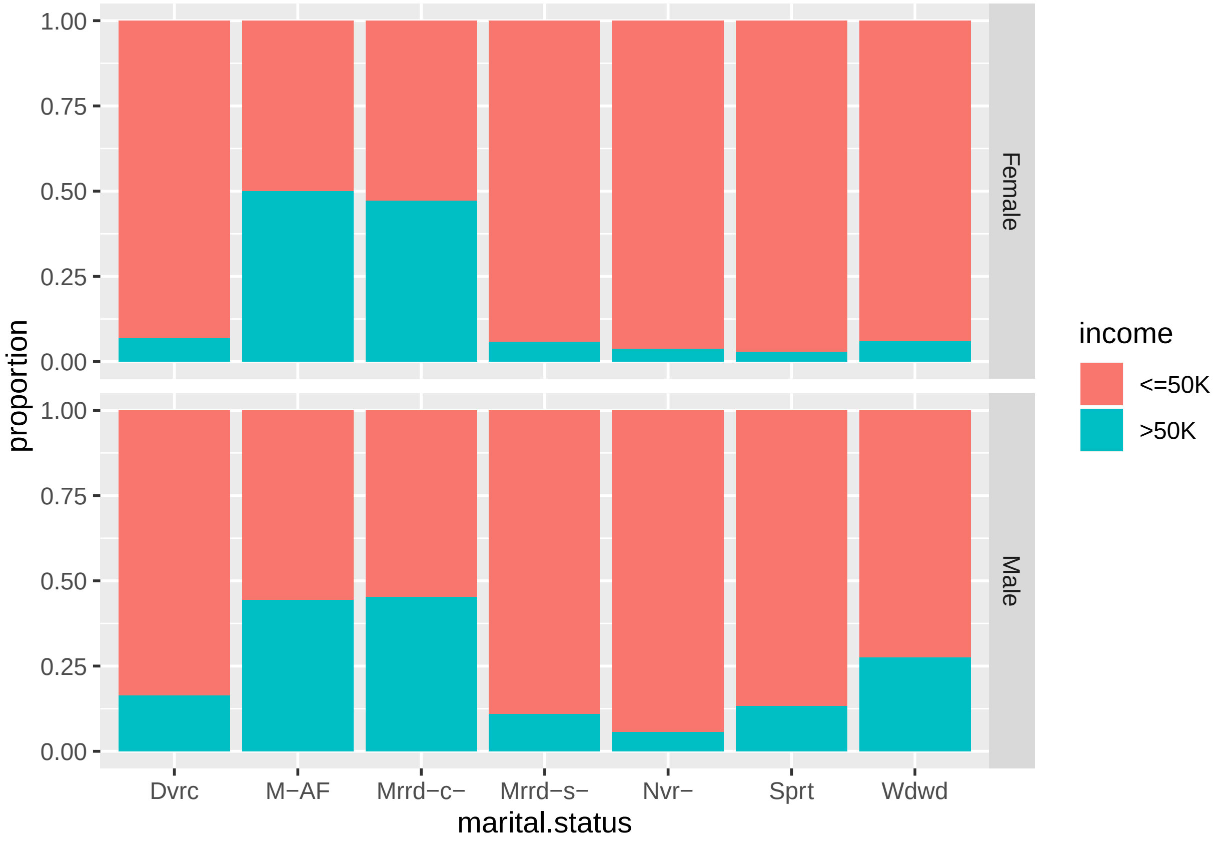
For this attribute I again used a bar graph showing the split by income.



What we seem to see here is a strong difference in the proportions of higher incomes for someone who is married (with their spouse present):

Marital Status	% over 50K
Married-civ-spouse	45.50
Married-AF-spouse	47.62
Divorced	14.06
Married-spouse-absent	8.38
Never-married	4.83
Separated	7.03
Widowed	9.67

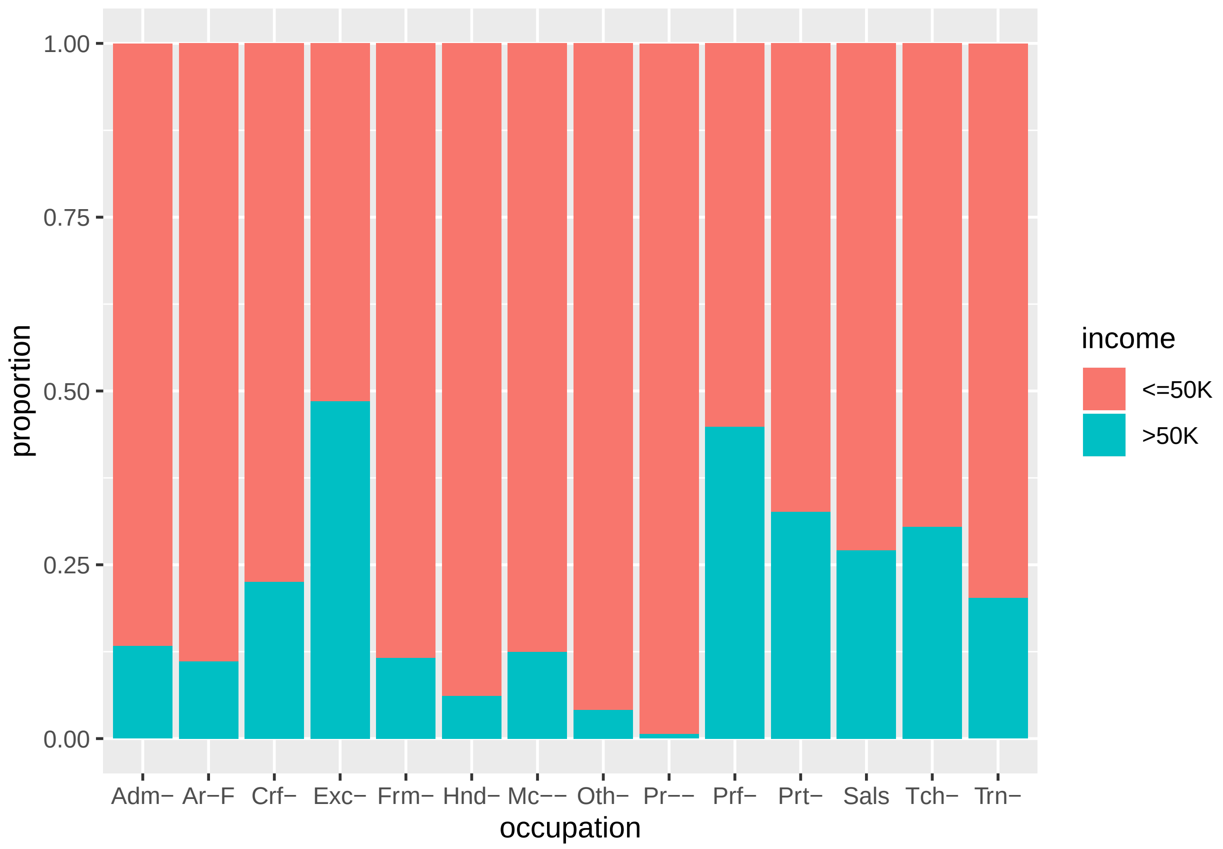
I wondered whether this might be different for each sex:



They appear broadly similar, although what we do see is that the proportion of female respondents earning more than 50k is even more clearly delineated across the categories with spouse present and those without.

Occupation

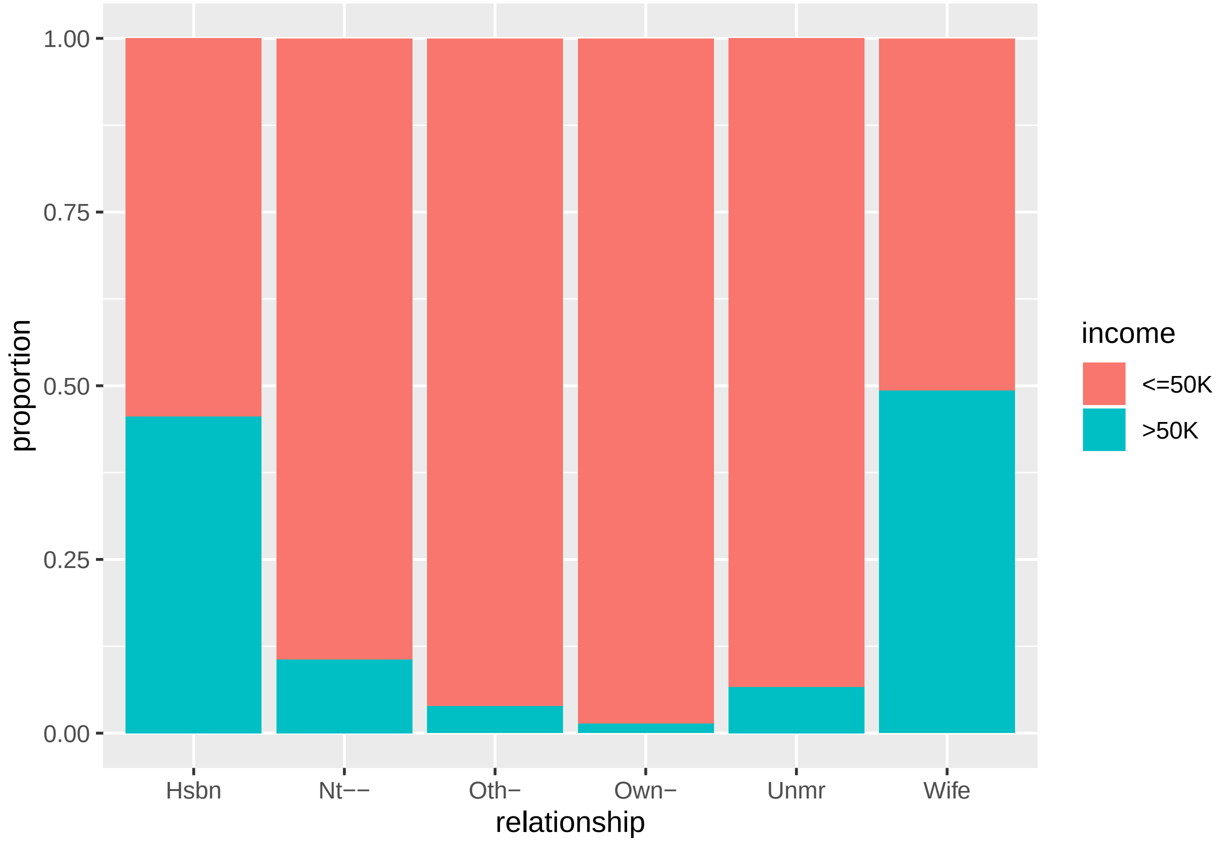
Again I plotted this on a proportion basis by income



There is a lot of variance across these, with the highest being “Exec-managerial” and “Prof-specialty”.

Relationship

I wasn’t sure from the documentation of this dataset, what connection (if any) there is between `relationship`, `marital.status` and `sex`.



Here it seems that the `relationship` attribute is repeating the information we gained from the `marital.status` attribute, at least for those describing themselves as “Husband” or “Wife” - these are 100% contained in the Married-AF-spouse and “Married-civ-spouse” categories.

```
aci %>% filter(relationship == "Wife" | relationship == "Husband") %>% group_by(marital.status) %>% sum
```

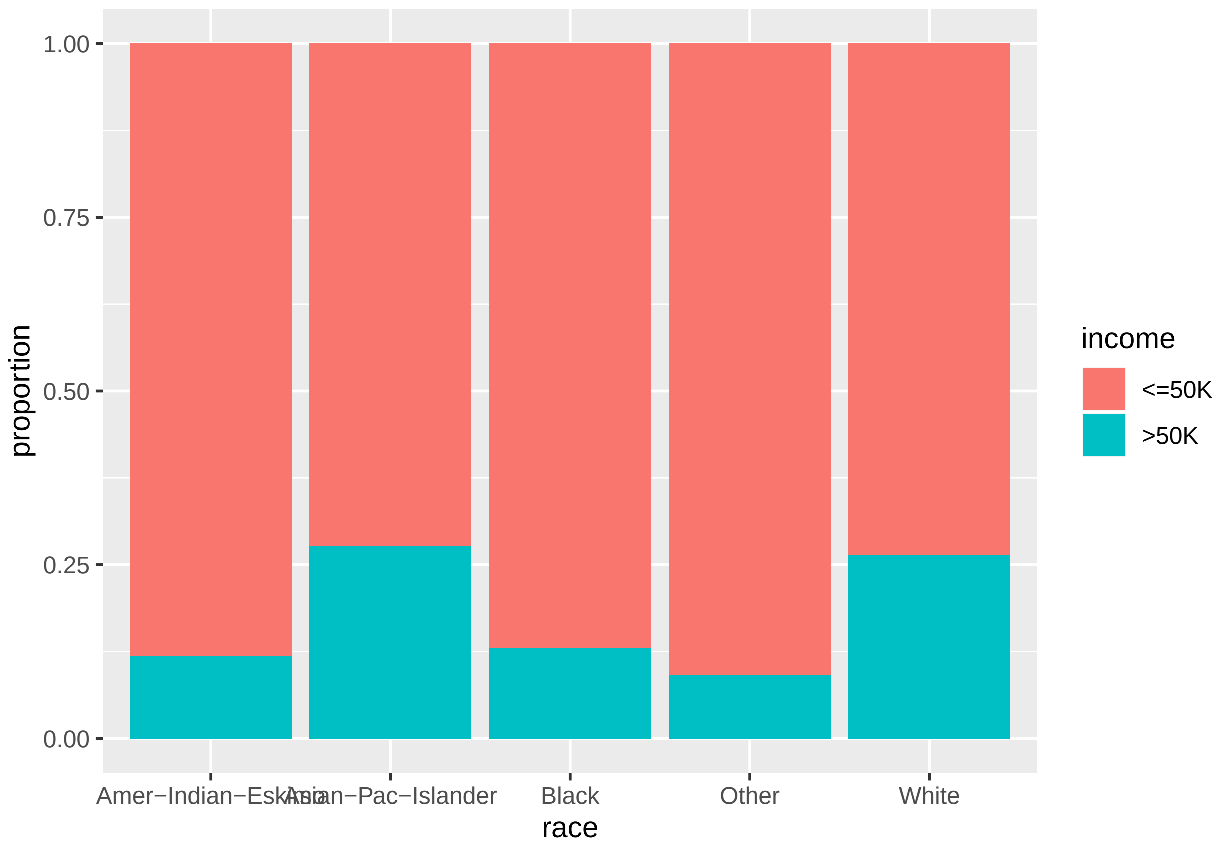
```
## # A tibble: 2 x 2
##   marital.status     count
##   <fct>              <int>
## 1 Married-AF-spouse     19
## 2 Married-civ-spouse 13850
```

Given the similarity of these I decided to remove this attribute as well.

```
aci <- aci %>% select(-relationship)
```

Race

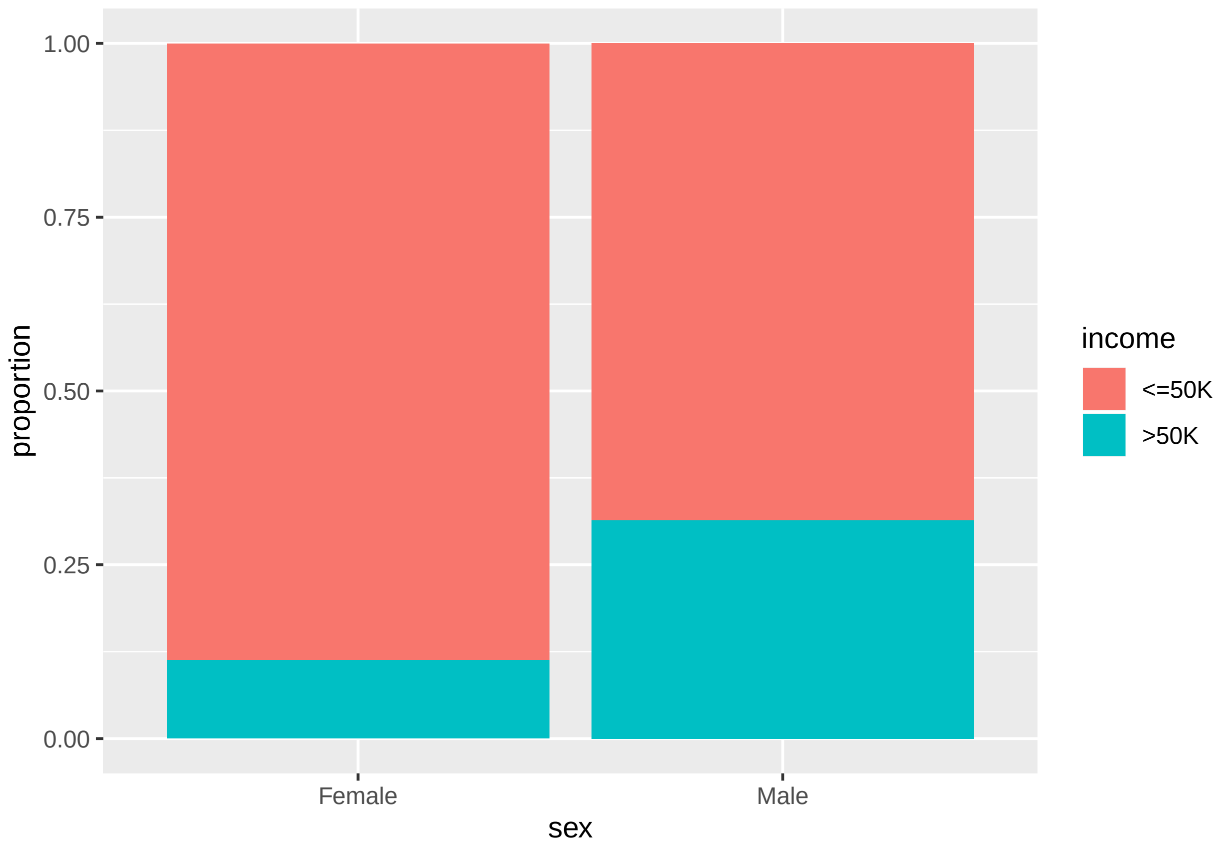
85.98% of the dataset is “White”. The bar chart shows the proportions by income.



There is a significant difference between the income proportion of “White” or “Asian-Pac_Islander” at roughly 25% over 50k and all other races at around half that level.

Sex

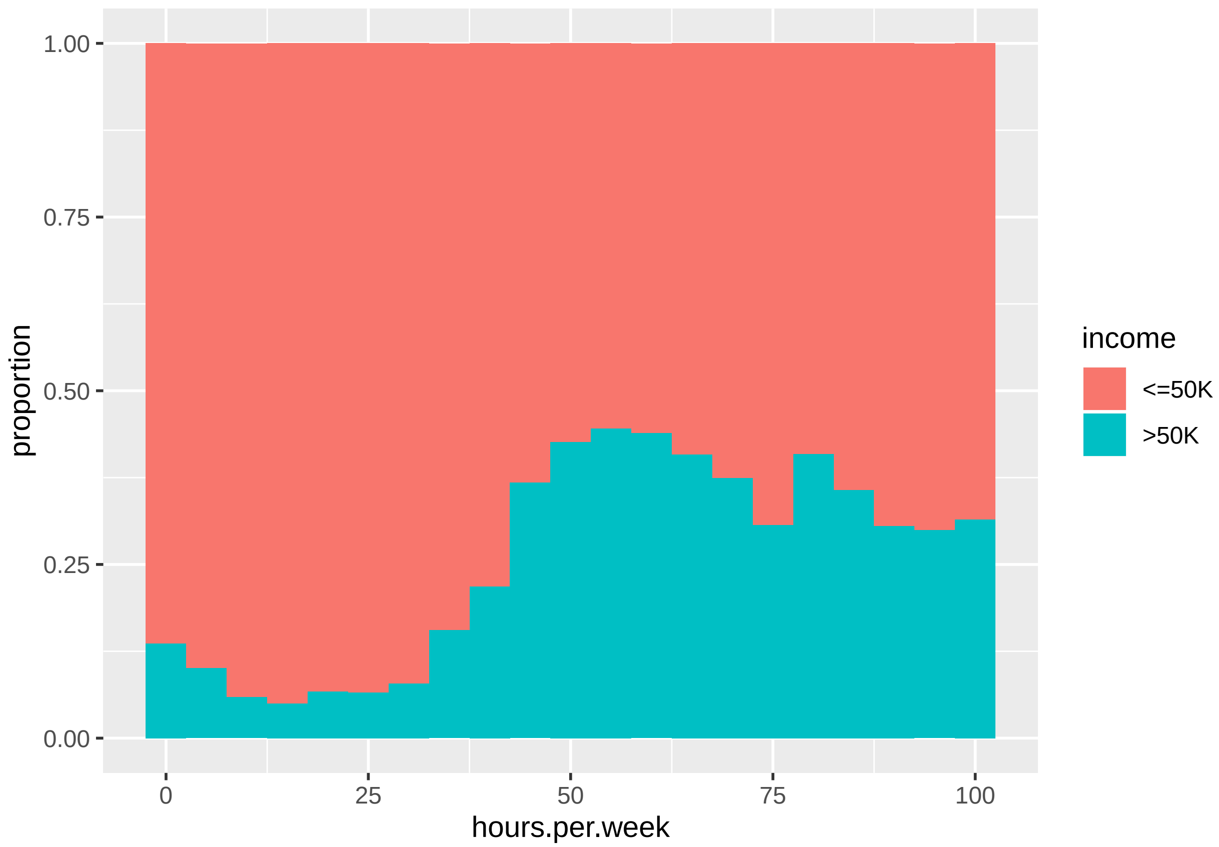
67.57% of the dataset is “Male”. The bar chart again shows the proportions by income.



Again there is a noticeable difference in income level between sexes with the proportion of "Male" respondents earning over 50k at 31.38%, with "Female" respondents only 11.37%.

Hours per week

I used a histogram to plot the hours per week data, to allow for the fact that most of the responses are rounded to the nearest 5 hours.



As we might expect the proportion of those earning over 50k rises significantly at around 40-50 hours per week, representing the cultural norm of fulltime work. Interestingly it doesn't rise further, as hours continue to go up, and in fact appears to fall slightly.

Analysis

Firstly I split the dataset (which is now 30162 observations) into training and validation sets.

```
set.seed(1,sample.kind = "Rounding")
#if using R3.5 or earlier set.seed(1)

test_index <- createDataPartition(aci$income, times = 1, p = 0.2, list = FALSE)
aci_validation <- aci[test_index, ]
aci_training <- aci[-test_index, ]
```

To enable me to try out a number of different approaches before settling on a final algorithm I also split the `aci_training` set into training and test.

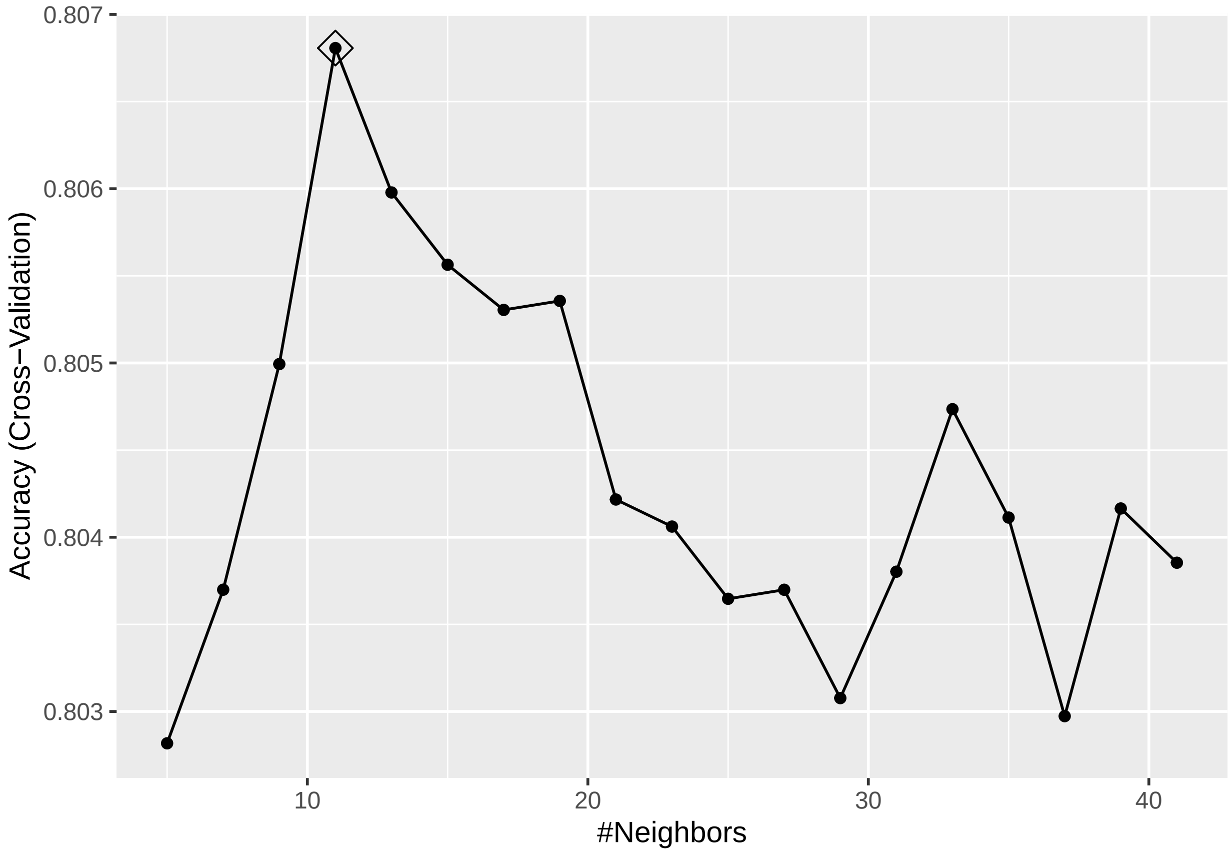
```
set.seed(10,sample.kind = "Rounding")  #if using R3.5 or earlier set.seed(10)
test_index2 <- createDataPartition(aci_training$income, times = 1, p = 0.2, list = FALSE)
testing <- aci_training[test_index2, ]
training <- aci_training[-test_index2, ]
```

The training set now has 19302 observations.

k Nearest Neighbours

I start with a k-nearest neighbours model. I'm not sure what value of k to use so I use the cross-validation built into the caret package on the range 4 to 41. I reduce the default to 10-fold cross validation to reduce the time taken to run.

```
set.seed(3,sample.kind = "Rounding") #if using R3.5 or earlier set.seed(3)
control <- trainControl(method = "cv", number = 10, p = .9)
train_knn <- train(income ~ ., method = "knn", data = training, tuneGrid = data.frame(k = seq(5,41,2)),
ggplot(train_knn,highlight = TRUE)
```



```
train_knn$bestTune
```

```
##    k
## 4 11
```

I then use this model to make predictions on the testing dataset.

```
knn_accuracy <- confusionMatrix(predict(train_knn, testing, type = "raw"), testing$income)$overall["Accu
knn_accuracy
```

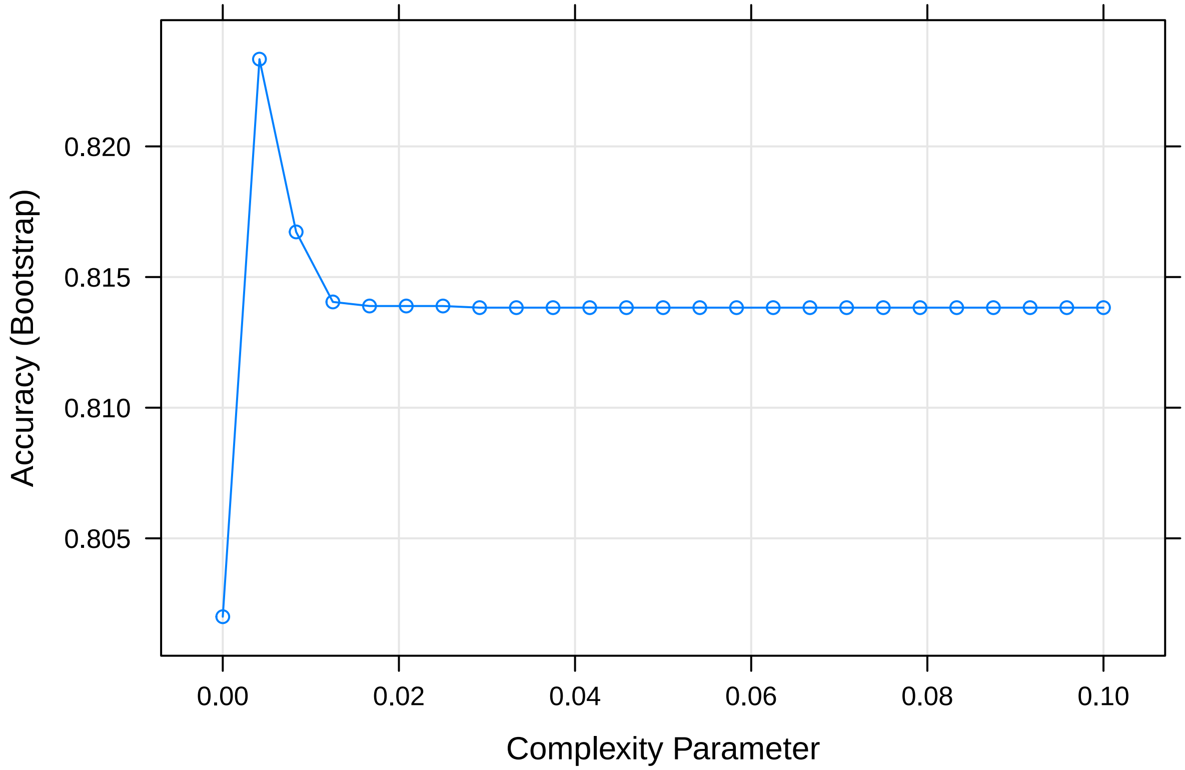
```
##  Accuracy
## 0.8058836
```

The accuracy is 80.59% on the testing set.

Classification Tree

I considered using Quadratic Discriminant Analysis or Linear Discriminant Analysis, but discarded these approaches due to the number of predictors and the fact that most do not appear normally distributed. Instead I settle on trying a classification tree as the second approach, using cross-validation to select the complexity parameter.

```
set.seed(3,sample.kind = "Rounding") #if using R3.5 or earlier set.seed(3)
train_rpart <- train(income ~ ., method = "rpart", tuneGrid = data.frame(cp = seq(0.0, 0.1, len=25)), da
plot(train_rpart)
```



```
train_rpart$bestTune
```

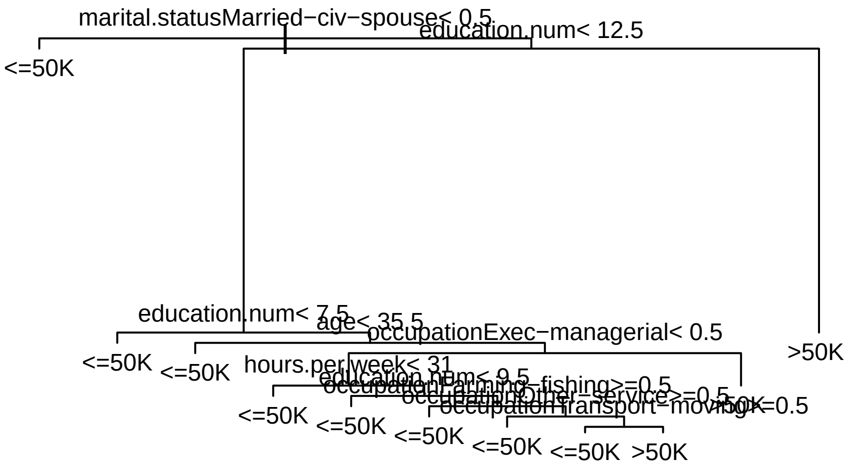
```
##            cp
## 2 0.004166667
```

Again I use this model to make predictions on the testing dataset.

```
rpart_accuracy <- confusionMatrix(predict(train_rpart, testing), testing$income)$overall["Accuracy"]
rpart_accuracy
```

```
##  Accuracy
## 0.8253574
```

The accuracy is higher than for the kNN algorithm - now 82.54% it is also interesting to be able to see the key decision points. As expected marital status plays a key role, with anyone who is not "Married-civ-spouse" predicted to have income of 50k or less.


marital.statusMarried−civ−spouse< 0.5
education.num< 12.5
<=50K
education.num< 7.5
age< 35.5
occupationExec−managerial< 0.5
>50K
<=50K
<=50K
hours.per.week< 31
education.num< 9.5
occupationFarming−fishing>=0.5
occupationOther−service>=0.5
occupationTransport−moving>=0.5
>50K
<=50K
<=50K
<=50K
<=50K
<=50K
>50K

Random Forest

My final approach is a random forest algorithm.

```r
set.seed(3, sample.kind = "Rounding")
#if using R3.5 or earlier set.seed(3)
train_rf <- randomForest(income ~ ., data = training)
rf_accuracy <- confusionMatrix(predict(train_rf, testing), testing$income)$overall["Accuracy"]
rf_accuracy
```

```
##  Accuracy
## 0.8394448
```

The accuracy is improved again, now to 83.94%.

While we cannot plot the tree in the same way as for the classification tree above, it is still possible to see the importance of each variable, measured by the Mean Decrease in Gini.

```r
importance(train_rf)
```

```
##                 MeanDecreaseGini
## age                     910.5969
## workclass               279.5286
## education.num           812.5369
## marital.status         1145.5805
## occupation              736.0640
## race                    120.9722
## sex                     152.3775
## hours.per.week          540.7127
```

As expected, `marital.status` remains the most important, followed by `age`, `education.num` and `occupation`.

Validation and Results

I have selected the random forest algorithm from the ones tested above to try on the validation set. I trained the algorithm again, now using the entire aci_training set avaiable to me, and then used this to predict the income results on the aci_validation set.

```
set.seed(3, sample.kind = "Rounding") #if using R3.5 or earlier set.seed(3)
final_train_rf <- randomForest(income ~ ., data = aci_training)
final_accuracy <- confusionMatrix(predict(final_train_rf, aci_validation), aci_validation$income)$overal
final_accuracy
```

```
##  Accuracy
## 0.8380574
```

```
importance(final_train_rf)
```

```
##                 MeanDecreaseGini
## age                    1109.3035
## workclass               326.0302
## education.num           986.3502
## marital.status         1451.9419
## occupation              915.9308
## race                    144.2865
## sex                     186.2557
## hours.per.week          670.0404
```

The accuracy remains similar to that achieved during the testing phase, with the final model achieving 83.81% accuracy on the validation dataset. The importance factors remain the same, with `marital.status`, `age`, `education.num` and `occupation` being the key factors in predicting income level.

Conclusion

I used the Adult Census Income dataset from UCI to predict income levels as being under or over 50k. Using a Random Forest algorithm I achieved an accuracy level on a randomly selected validation set of 83.81%.

Prior to selecting this algorithm I also tested a k-nearest neighbours algorithm, achieving an accuracy rate on a reserved (for testing) part of the training set of 80.59%, and using a Classification Tree, achieving an accuracy of 82.54%. It is possible therefore that further accuracy could be achieved by using an ensemble approach of the kNN, Classification Tree and Random Forest algorithms.

It is also worth considering the possible limitations of this as a predictive algorithm for use outside the Adult Census Income data. The chosen dataset differs to the US population in a number of respects. Notably it has more men, more white people, and fewer immigrants than the standard population, which is currently approximately 51% female, 40% non-white and approximately 28% immigrant (includes US born children). It is also now 25 years old. All of this would cause concern if using the algorithm to create predictions on current populations.