HX Capstone - Adult Census Data

1. Dataset description and project goals

These data were extracted from the 1994 Census bureau database by Ronny Kohavi and Barry Becker (Data Mining and Visualization, Silicon Graphics). A set of reasonably clean records was extracted using the following conditions: ((AAGE>16) && (AGI>100) && (AFNLWGT>1) && (HRSWK>0)). The prediction task is to determine whether a person makes over $50K a year.

Description of fnlwgt (final weight):

The weights on the Current Population Survey (CPS) files are controlled to independent estimates of the civilian population of the US. These are prepared monthly for us by Population Division here at the Census Bureau. We use 3 sets of controls. These are:

```
A single cell estimate of the population 16+ for each state.
Controls for Hispanic Origin by age and sex.
Controls by Race, age and sex.
```

We use all three sets of controls in our weighting program and "rake" through them 6 times so that by the end we come back to all the controls we used. The term estimate refers to population totals derived from CPS by creating "weighted tallies" of any specified socio-economic characteristics of the population. People with similar demographic characteristics should have similar weights. There is one important caveat to remember about this statement. That is that since the CPS sample is actually a collection of 51 state samples, each with its own probability of selection, the statement only applies within state.

Relevant papers: Ron Kohavi, "Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid", Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, 1996. (PDF)

Once again, we will be building a model to predict whether a person's income exceeds $50K/yr based on census data.

The datasets (adult.data and adult.test) can be downloaded directly from http://archive.ics.uci.edu/ml/machine-learning-databases/adult/ or by using the code below. Alternatively, the datasets and the code are also available on my github site https://github.com/oster4/HX-DS-Capstone.

2. Ingesting and exploring the data

First, let's download the required libraries.

```r
suppressMessages(if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project
```

```
## Warning: package 'tibble' was built under R version 3.5.2
```

```
## Warning: package 'purrr' was built under R version 3.5.2
```

```r
suppressMessages(if(!require(corrplot)) install.packages("corrplot", repos = "http://cran.us.r-project.o
suppressMessages(if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org"))
suppressMessages(if(!require(e1071)) install.packages("e1071", repos = "http://cran.us.r-project.org"))
```

```
## Warning: package 'e1071' was built under R version 3.5.2
```

```r
suppressMessages(if(!require(randomForest)) install.packages("randomForest", repos = "http://cran.us.r-p
suppressMessages(if(!require(magrittr)) install.packages("magrittr", repos = "http://cran.us.r-project.o
```

Let's download the training and testing datasets:

```r
tmp_train <- tempfile()
download.file("http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data", tmp_train)
adult_train <- read.csv(tmp_train, header = FALSE, sep = ",")
```

```r
tmp_test <- tempfile()
download.file("http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.test", tmp_test)
adult_test <- read.csv(tmp_test, skip = 1, header = FALSE, sep = ",")
```

Let's attach comlumn names:

```r
headers = c("age", "workclass", "fnlweight", "education", "eduyears", "marital", "occupation", "relatio
            "caploss", "hours", "country", "income")
colnames(adult_train) <- headers
colnames(adult_test) <- headers
```

Let's create respective csv files on the hard drive for future use (optional, uncomment if you'd like to run):

```r
# write.csv(adult_train, file = "adult_train.csv", row.names = FALSE)
# write.csv(adult_test, file = "adult_test.csv", row.names = FALSE)
```

Let's combine the training and testing datasets for holistic overview, and check for NA values:

```r
adult_all <- rbind(adult_train, adult_test)
sum(is.na(adult_all))
```

```
## [1] 0
```

The dataset appears to be well populated, but let's summarize it to check for any other issues:

```r
summary(adult_all)
```

```
##       age                    workclass       fnlweight
##  Min.   :17.00    Private         :33906   Min.   :  12285
##  1st Qu.:28.00    Self-emp-not-inc: 3862   1st Qu.: 117550
##  Median :37.00    Local-gov       : 3136   Median : 178144
##  Mean   :38.64    ?               : 2799   Mean   : 189664
##  3rd Qu.:48.00    State-gov       : 1981   3rd Qu.: 237642
##  Max.   :90.00    Self-emp-inc    : 1695   Max.   :1490400
##                  (Other)          : 1463
##          education        eduyears                      marital
##   HS-grad     :15784   Min.   : 1.00    Divorced              : 6633
##   Some-college:10878   1st Qu.: 9.00    Married-AF-spouse     :   37
##   Bachelors   : 8025   Median :10.00    Married-civ-spouse    :22379
##   Masters     : 2657   Mean   :10.08    Married-spouse-absent:  628
##   Assoc-voc   : 2061   3rd Qu.:12.00    Never-married        :16117
##   11th        : 1812   Max.   :16.00    Separated             : 1530
##  (Other)      : 7625                    Widowed               : 1518
##             occupation           relationship
##   Prof-specialty : 6172    Husband        :19716
##   Craft-repair   : 6112    Not-in-family :12583
##   Exec-managerial: 6086    Other-relative: 1506
##   Adm-clerical   : 5611    Own-child      : 7581
##   Sales          : 5504    Unmarried      : 5125
##   Other-service  : 4923    Wife           : 2331
##  (Other)         :14434
##                   race             sex           capgain
##   Amer-Indian-Eskimo:  470    Female:16192   Min.   :    0
```

```
##   Asian-Pac-Islander: 1519    Male  :32650   1st Qu.:    0
##   Black             : 4685                  Median :    0
##   Other             :  406                  Mean   : 1079
##   White             :41762                  3rd Qu.:    0
##                                             Max.   :99999
##
##     caploss           hours                 country         income
##  Min.   :   0.0   Min.   : 1.00    United-States:43832    <=50K :24720
##  1st Qu.:   0.0   1st Qu.:40.00    Mexico       :  951    >50K  : 7841
##  Median :   0.0   Median :40.00    ?            :  857    <=50K.:12435
##  Mean   :  87.5   Mean   :40.42    Philippines  :  295    >50K. : 3846
##  3rd Qu.:   0.0   3rd Qu.:45.00    Germany      :  206
##  Max.   :4356.0   Max.   :99.00    Puerto-Rico  :  184
##                                   (Other)       : 2517
```

Workclass column has "?" and several columns have "Other", so we need to break those down to understand if any adjustments are necessary:

```
workclass_values <- unique(adult_all$workclass); workclass_values
```

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

Review of the data structure:

```
str(adult_all)
```

```
## 'data.frame':    48842 obs. of  15 variables:
##  $ age         : int  39 50 38 53 28 37 49 52 31 42 ...
##  $ workclass   : Factor w/ 9 levels " ?"," Federal-gov",..: 8 7 5 5 5 5 5 7 5 5 ...
##  $ fnlweight   : int  77516 83311 215646 234721 338409 284582 160187 209642 45781 159449 ...
##  $ education   : Factor w/ 16 levels " 10th"," 11th",..: 10 10 12 2 10 13 7 12 13 10 ...
##  $ eduyears    : int  13 13 9 7 13 14 5 9 14 13 ...
##  $ marital     : Factor w/ 7 levels " Divorced"," Married-AF-spouse",..: 5 3 1 3 3 3 4 3 5 3 ...
##  $ occupation  : Factor w/ 15 levels " ?"," Adm-clerical",..: 2 5 7 7 11 5 9 5 11 5 ...
##  $ relationship: Factor w/ 6 levels " Husband"," Not-in-family",..: 2 1 2 1 6 6 2 1 2 1 ...
##  $ race        : Factor w/ 5 levels " Amer-Indian-Eskimo",..: 5 5 5 3 3 5 3 5 5 5 ...
##  $ sex         : Factor w/ 2 levels " Female"," Male": 2 2 2 2 1 1 1 2 1 2 ...
##  $ capgain     : int  2174 0 0 0 0 0 0 0 14084 5178 ...
##  $ caploss     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hours       : int  40 13 40 40 40 40 16 45 50 40 ...
##  $ country     : Factor w/ 42 levels " ?"," Cambodia",..: 40 40 40 40 6 40 24 40 40 40 ...
##  $ income      : Factor w/ 4 levels " <=50K"," >50K",..: 1 1 1 1 1 1 1 2 2 2 ...
```

```
education_values <- unique(adult_all$education); education_values
```

```
##  [1]  Bachelors     HS-grad       11th          Masters       9th
##  [6]  Some-college  Assoc-acdm    Assoc-voc     7th-8th       Doctorate
## [11]  Prof-school   5th-6th       10th          1st-4th       Preschool
## [16]  12th
## 16 Levels:  10th  11th  12th  1st-4th  5th-6th  7th-8th ...  Some-college
```

```
country_values <- unique(adult_all$country); country_values
```

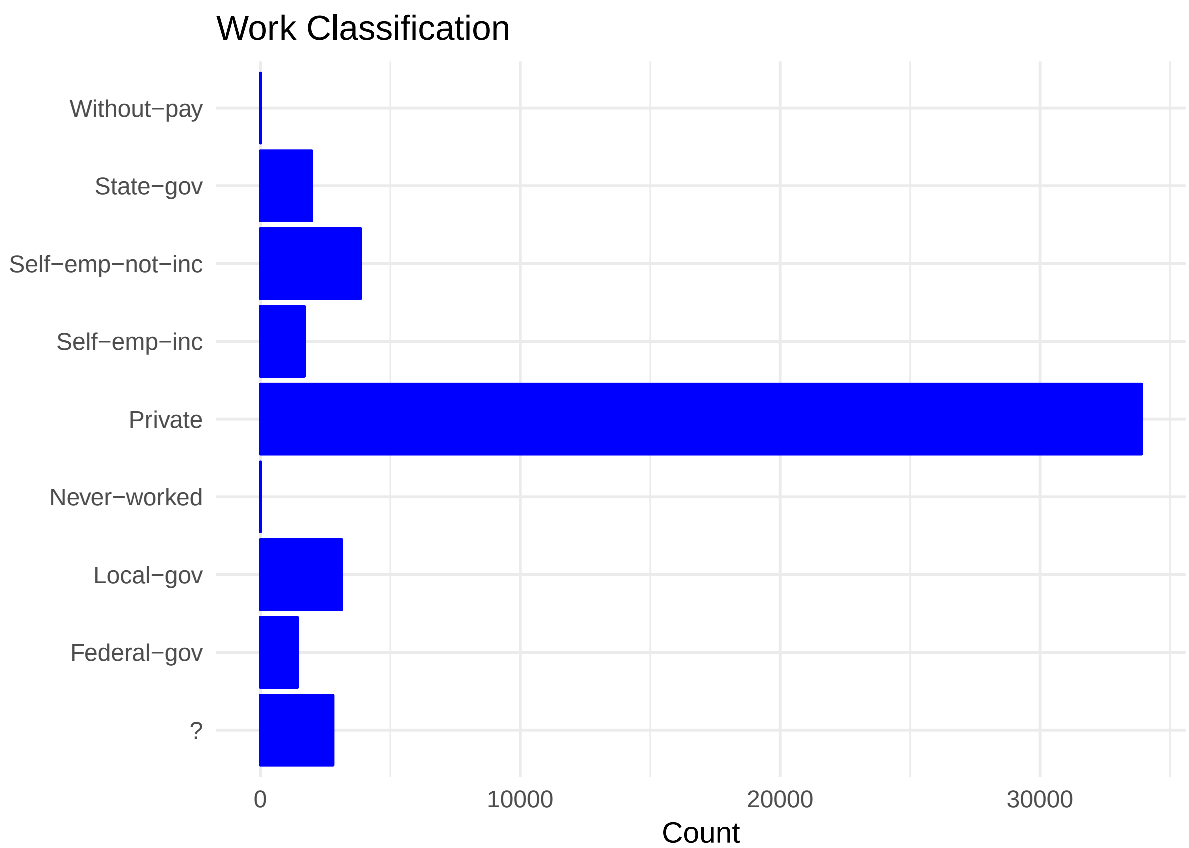
```
##  [1]  United-States               Cuba
##  [3]  Jamaica                     India
```

```
##  [5]  ?                          Mexico
##  [7]  South                      Puerto-Rico
##  [9]  Honduras                   England
## [11]  Canada                     Germany
## [13]  Iran                       Philippines
## [15]  Italy                      Poland
## [17]  Columbia                   Cambodia
## [19]  Thailand                   Ecuador
## [21]  Laos                       Taiwan
## [23]  Haiti                      Portugal
## [25]  Dominican-Republic         El-Salvador
## [27]  France                     Guatemala
## [29]  China                      Japan
## [31]  Yugoslavia                 Peru
## [33]  Outlying-US(Guam-USVI-etc)  Scotland
## [35]  Trinadad&Tobago            Greece
## [37]  Nicaragua                  Vietnam
## [39]  Hong                       Ireland
## [41]  Hungary                    Holand-Netherlands
## 42 Levels:  ?  Cambodia  Canada  China  Columbia ...  Yugoslavia
```

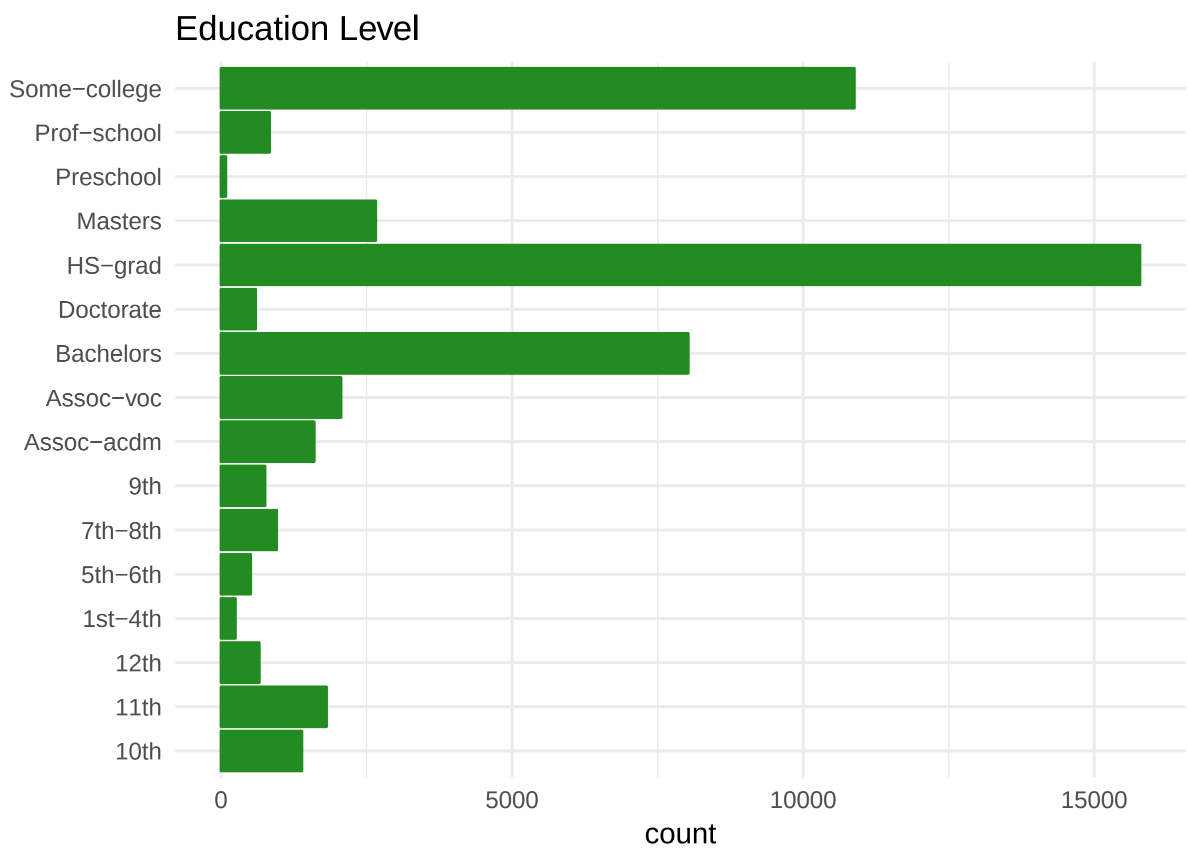
Later we'll need to replace "?" with "Unknown", and consider whether using the highest attained degree ("education") is meaningful while years of education ("eduyears") is also available. Also, the testing dataset has an extra dot available in the income column, which we'll need to remove.

Let's look at some charts to get a better sense of the distributions:

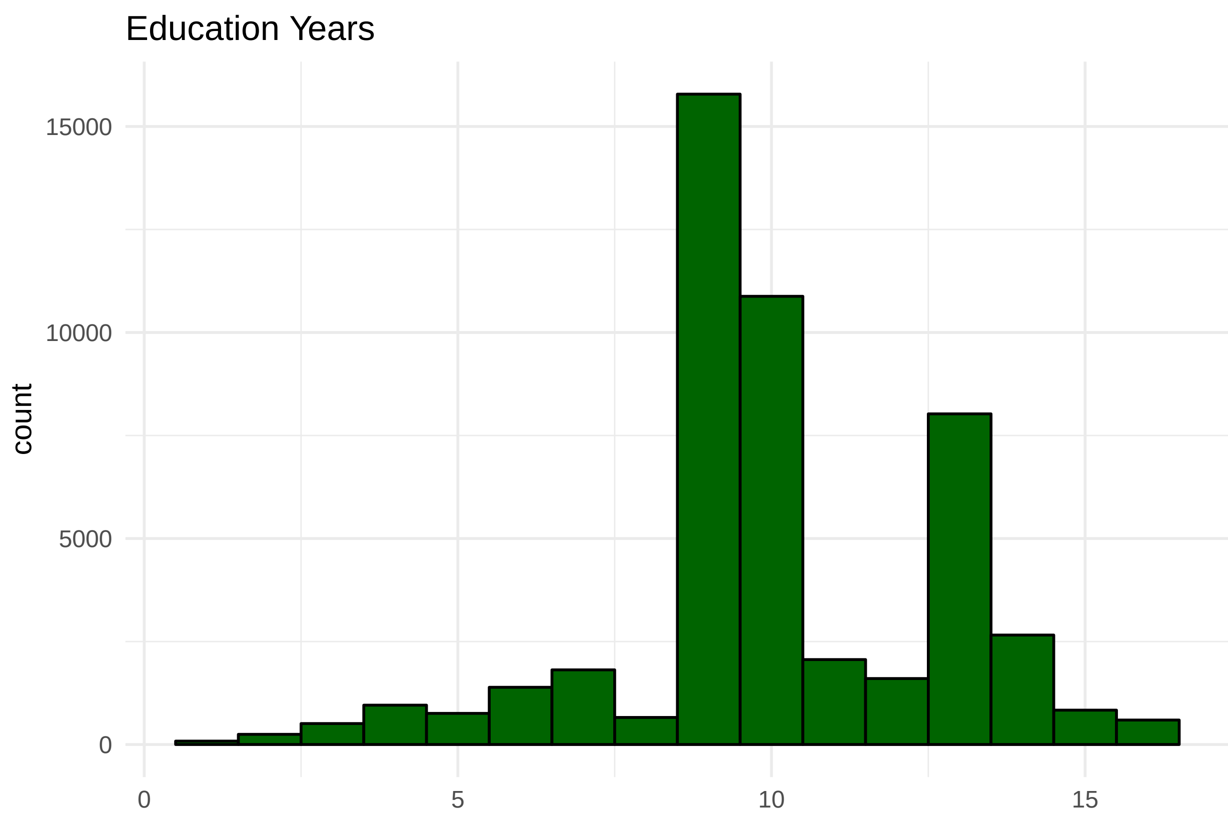
```
ggplot(adult_all, aes(workclass)) + geom_bar(colour="blue", fill="blue") + ggtitle("Work Classification
  theme_minimal() + coord_flip() + ylab("Count") + xlab("")
```



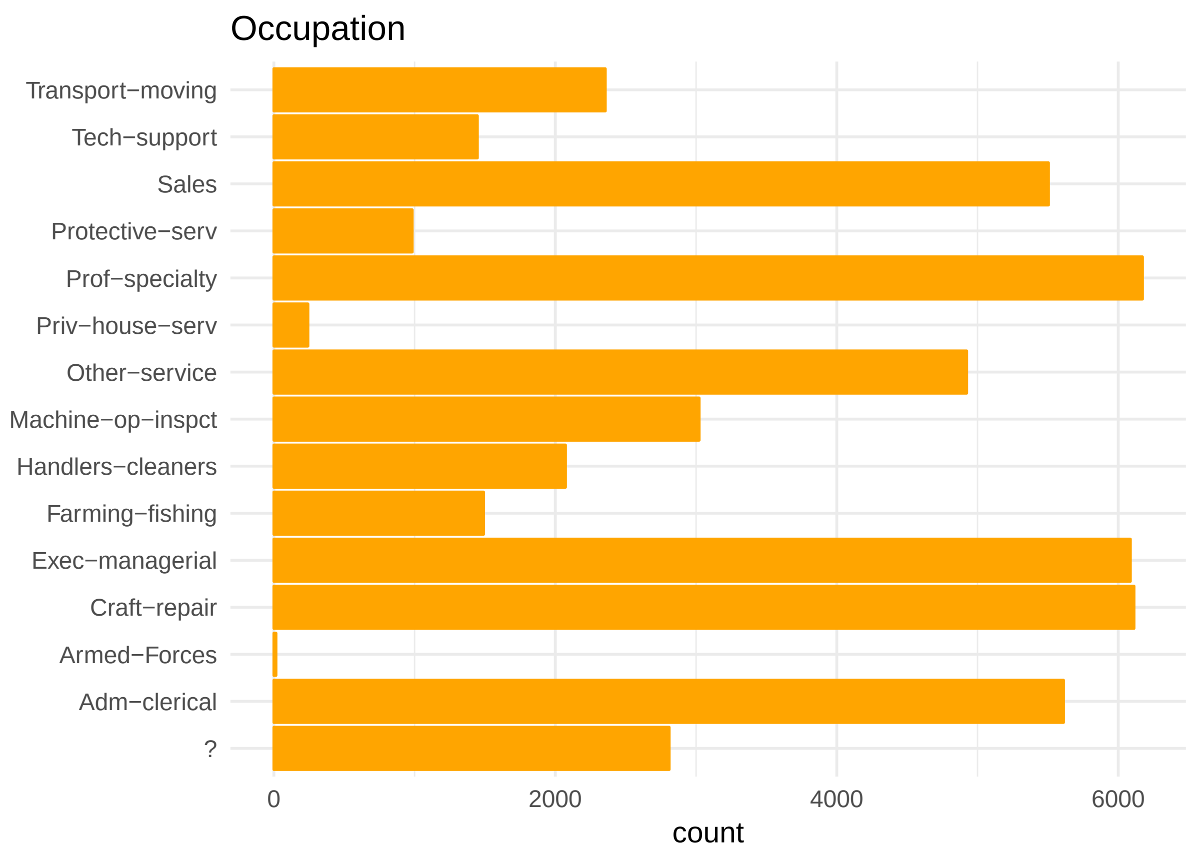

```
ggplot(adult_all, aes(education)) + geom_bar(colour="forestgreen", fill="forestgreen") + ggtitle("Educat
  theme_minimal() + coord_flip() + xlab("")
```



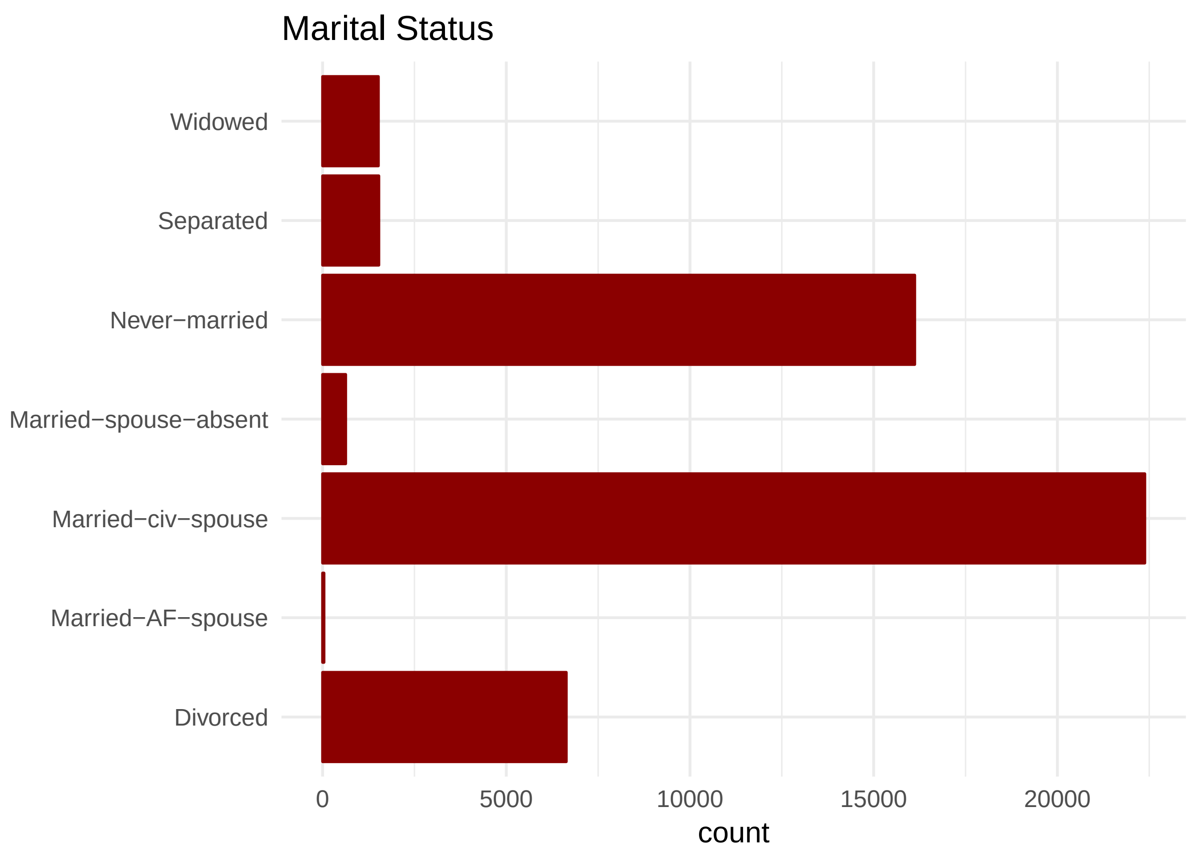

```
ggplot(adult_all, aes(eduyears)) + geom_histogram(colour="black", fill="darkgreen", binwidth = 1) + ggt
  theme_minimal() + xlab("")
```



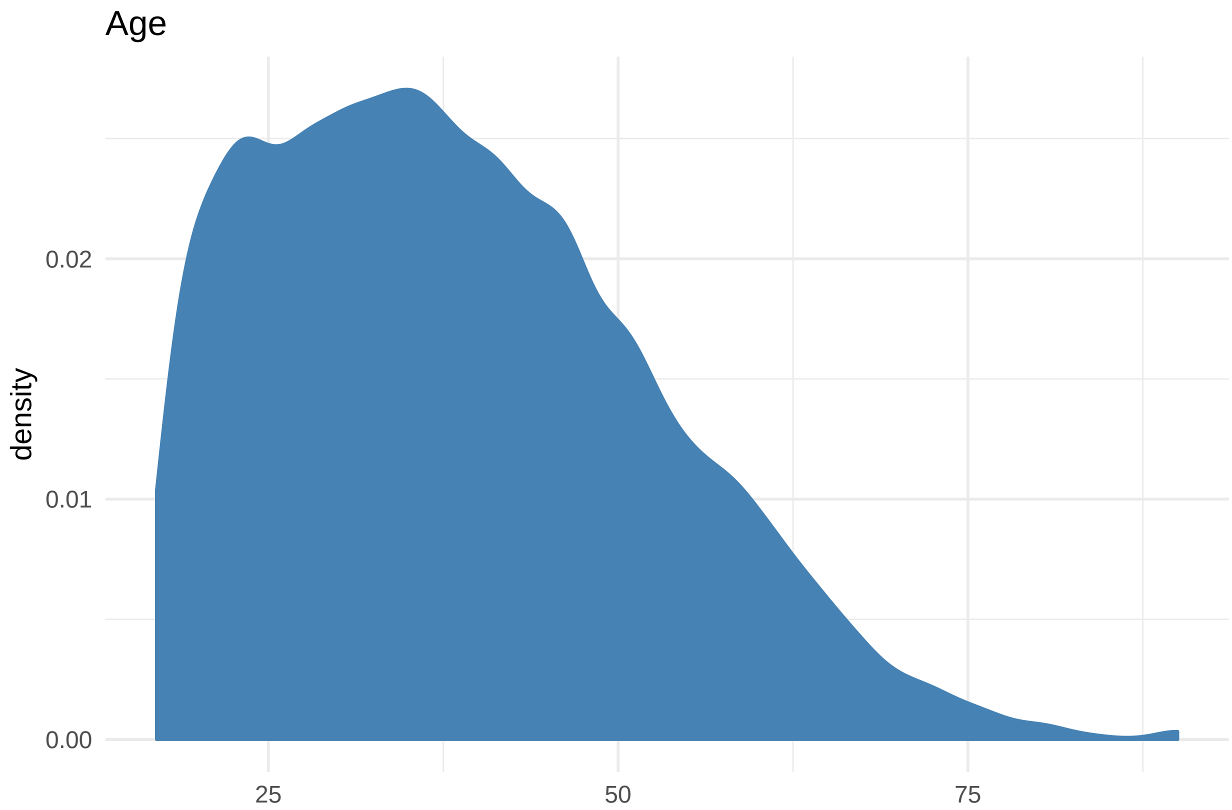

```r
ggplot(adult_all, aes(occupation)) + geom_bar(colour="orange", fill="orange") + ggtitle("Occupation") +
  theme_minimal() + coord_flip() + xlab("")
```



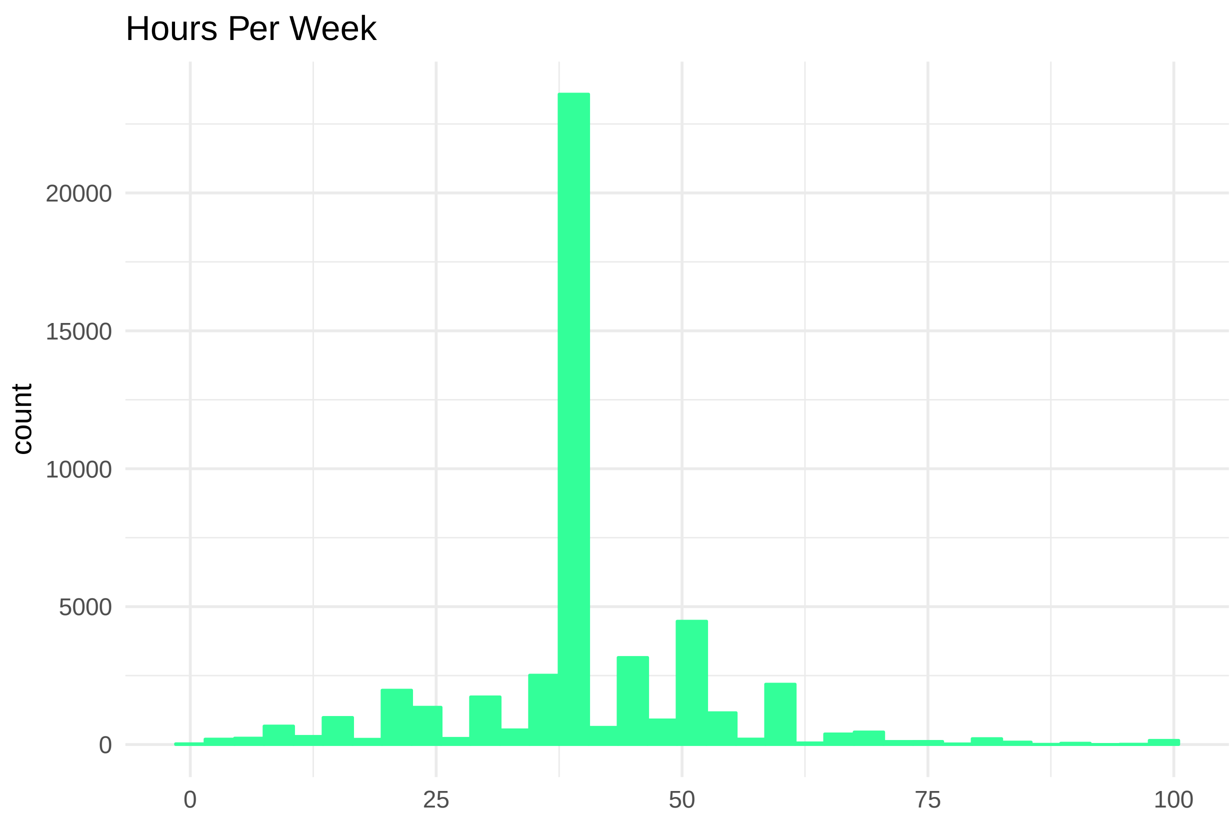

```
ggplot(adult_all, aes(marital)) + geom_bar(colour="darkred", fill="darkred") + ggtitle("Marital Status")
  theme_minimal() + coord_flip() + xlab("")
```



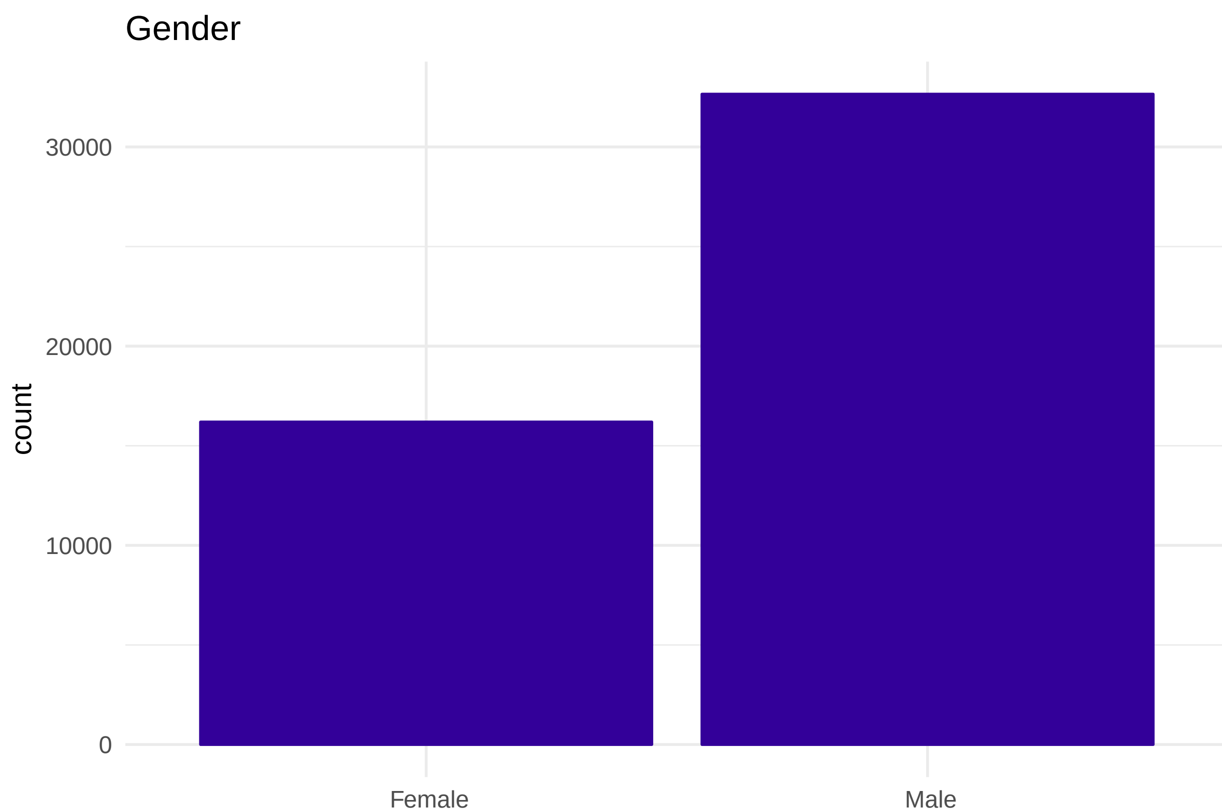

```
ggplot(adult_all, aes(age)) + geom_density(colour="steelblue", fill="steelblue") + ggtitle("Age") + ther
```




```
ggplot(adult_all, aes(hours)) + geom_histogram(colour="#33FF99", fill="#33FF99", binwidth = 3) + ggtitl
  theme_minimal() + xlab("")
```




```
ggplot(adult_all, aes(sex)) + geom_bar(colour="#330099", fill="#330099") + ggtitle("Gender") + theme_mi
```



We can see that the majority of subjects work in private businesses, ara high school graduates, obtained bachelors degree or some college (with the corresponding peaks in education years). Occupation-wise, the distribution is rather broad. Majority of the subjects are between 20 and 40 years old, and two thirds are male.

Time to do some clean-up on the training and testing datasets. First, let's confirm that the "50k" column is factorized:

```
str(adult_train$income)
```

```
##  Factor w/ 2 levels " <=50K"," >50K": 1 1 1 1 1 1 1 2 2 2 ...
```

Income is already factorized, so can keep the existing values, except we need to remove "." from the test set predicted values to ensure that the predicted values are identical (there is no "." at the end of the predicted value in the training set).

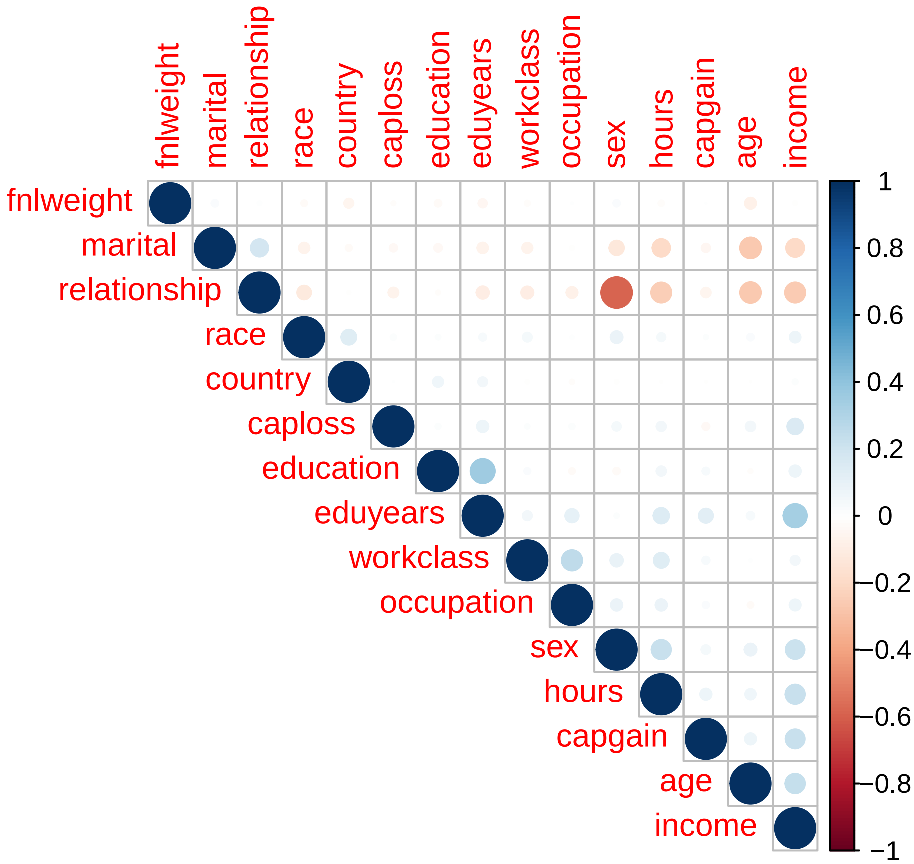
```
adult_test <- adult_test %>% mutate(income = recode(income, " <=50K." = " <=50K", " >50K." = " >50K"))
```

Let's also convert "?" in work classification into "Unknown". I do not plan on categorizing this entry as an NA, in part due to its abundance and in part due to the possibility that there is something about people who do not disclose this information that could yield predictive value.

```
adult_train <- adult_train %>% mutate(workclass = recode(workclass, " ?" = "Unknown"))
adult_test <- adult_test %>% mutate(workclass = recode(workclass, " ?" = "Unknown"))
```

Let's look at the correlations, but first convert the dataframe into a matrix:

```
adult_train_num <- as.matrix(sapply(adult_train, as.numeric))
correlation <- cor(adult_train_num, method = c("pearson"))
corrplot(correlation, method = "circle", type = 'upper', order = 'hclust')
```



The only somewhat interesing positive correlation for our target prediction (income above or below 50k) is with years of education. Let's move into the analysis stage.

3. Analysis and Results

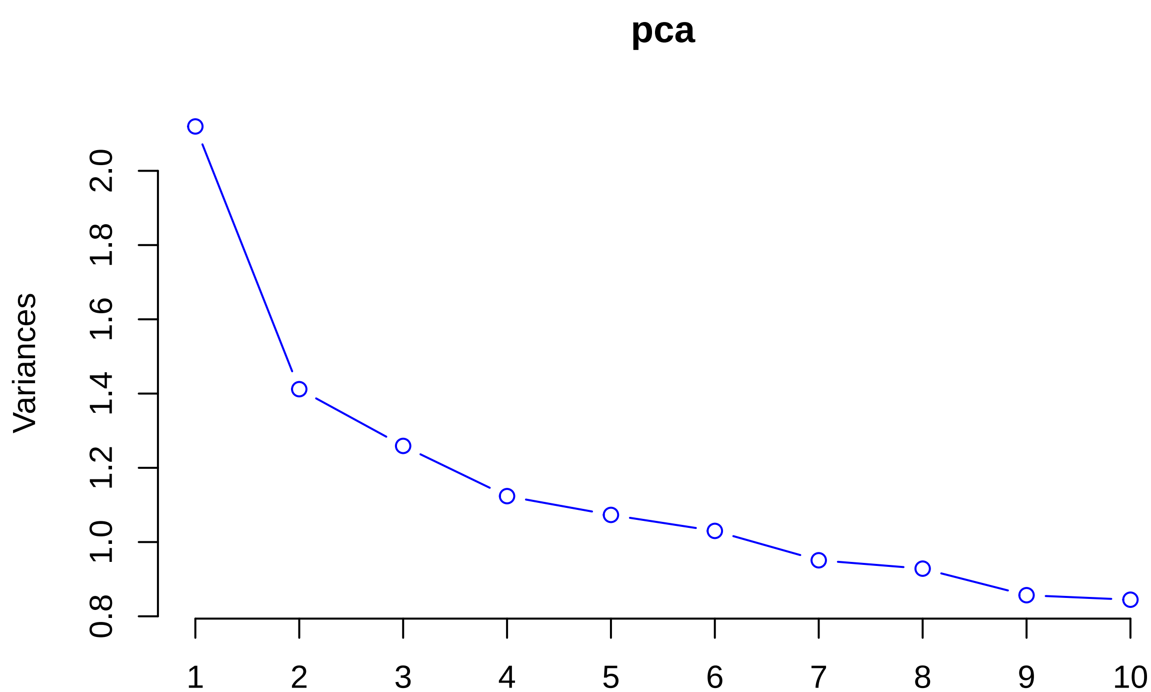
Let's start with the principal component analysis to see how much variance all of the features explain individually, and if some of them can be immediately dropped.

```
pca <- prcomp(adult_train_num[,1:14], scale. = TRUE)
summary(pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6
## Standard deviation     1.4559 1.1882 1.12204 1.06005 1.03591 1.01489
## Proportion of Variance 0.1514 0.1008 0.08993 0.08026 0.07665 0.07357
## Cumulative Proportion  0.1514 0.2522 0.34217 0.42243 0.49908 0.57265
##                            PC7     PC8    PC9    PC10    PC11    PC12
## Standard deviation     0.97515 0.96352 0.9256 0.91910 0.86506 0.82529
## Proportion of Variance 0.06792 0.06631 0.0612 0.06034 0.05345 0.04865
## Cumulative Proportion  0.64058 0.70689 0.7681 0.82843 0.88188 0.93053
##                           PC13    PC14
## Standard deviation     0.76716 0.61970
## Proportion of Variance 0.04204 0.02743
## Cumulative Proportion  0.97257 1.00000
```

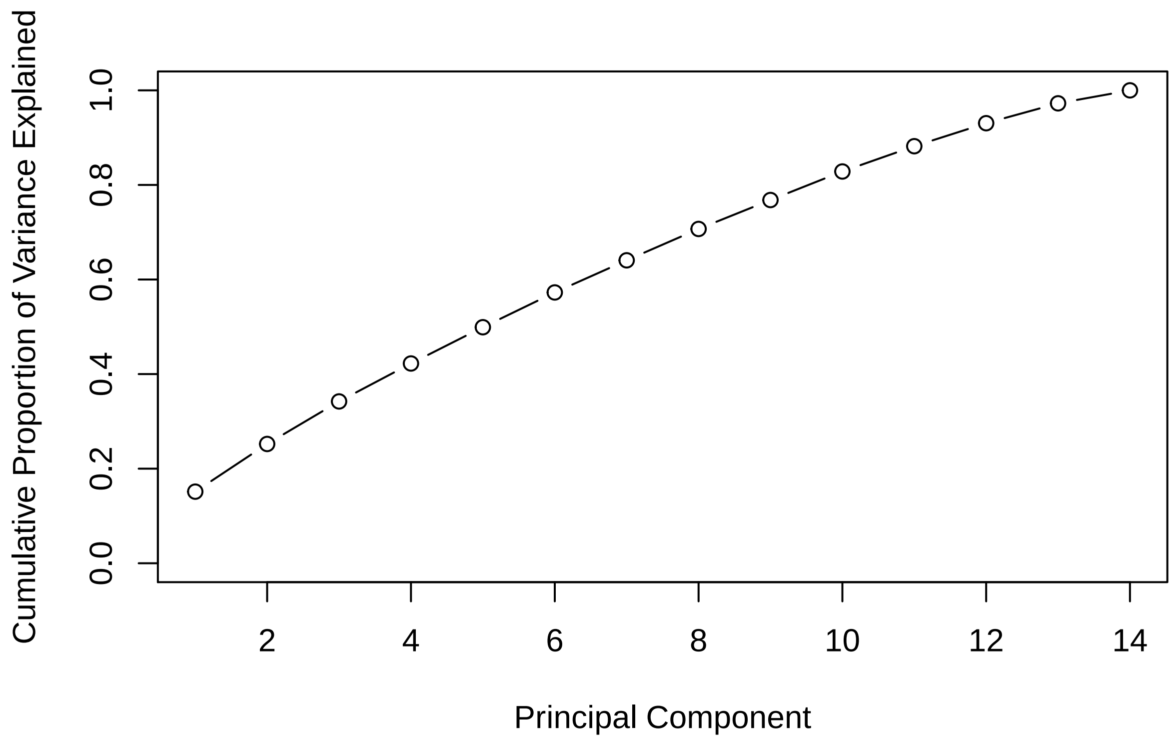
It appears all 14 variables have a meaningful role in explaining variances, and especially top 8:

```r
screeplot(pca, type="lines",col="blue")
```



Here, we can chart cumulative contribution of all 14 principal components:

```r
var <- pca$sdev^2
propvar <- var/sum(var)
plot(cumsum(propvar), xlab = "Principal Component", ylab = "Cumulative Proportion of Variance Explained
```



All of the components have significant enough contribution in explaining variance. Next, let's start our predictive analytics with Naive Bayes:

```
model_naiveBayes <- naiveBayes(income ~ ., data = adult_train)
pred_naiveBayes <- predict(model_naiveBayes, newdata=adult_test)
(table_naiveBayes <- table(adult_test$income, pred_naiveBayes))
```

```
##         pred_naiveBayes
##           <=50K  >50K
##    <=50K  11560   875
##    >50K    1951  1895
```

```
confusionMatrix(pred_naiveBayes, adult_test$income)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  <=50K  >50K
##      <=50K  11560  1951
##      >50K     875  1895
##
##                Accuracy : 0.8264
##                  95% CI : (0.8205, 0.8322)
##     No Information Rate : 0.7638
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4675
##  Mcnemar's Test P-Value : < 2.2e-16
```

```
##
##             Sensitivity : 0.9296
##             Specificity : 0.4927
##          Pos Pred Value : 0.8556
##          Neg Pred Value : 0.6841
##              Prevalence : 0.7638
##          Detection Rate : 0.7100
##    Detection Prevalence : 0.8299
##       Balanced Accuracy : 0.7112
##
##        'Positive' Class :  <=50K
##
```

```
error_naiveBayes <- 1 - sum(table_naiveBayes[row(table_naiveBayes)==col(table_naiveBayes)])/sum(table_na
(error_rate <- data_frame(Method = "Naive Bayes", Error_Rate = error_naiveBayes))
```

```
## Warning: `data_frame()` is deprecated, use `tibble()`.
## This warning is displayed once per session.
```

```
## # A tibble: 1 x 2
##   Method      Error_Rate
##   <chr>            <dbl>
## 1 Naive Bayes      0.174
```

The method resulted in just over 17% error rate, let's see if we can do better with logistic regression:

```
model_logReg <- glm(income ~ . , data = adult_train, family = "binomial")
```

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

```
pred_logReg <- predict(model_logReg, newdata=adult_test, type = "response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
predbinary_logReg <- as.factor(ifelse(pred_logReg > 0.5, " >50K", " <=50K"))
(table_logReg <- table(adult_test$income, predbinary_logReg))
```

```
##         predbinary_logReg
##           <=50K  >50K
##    <=50K  11578   857
##    >50K    1543  2303
```

```
confusionMatrix(predbinary_logReg, adult_test$income)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  <=50K  >50K
##      <=50K  11578  1543
##      >50K     857  2303
##
##                Accuracy : 0.8526
##                  95% CI : (0.847, 0.858)
##     No Information Rate : 0.7638
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5647
##  Mcnemar's Test P-Value : < 2.2e-16
```

```
##
##             Sensitivity : 0.9311
##             Specificity : 0.5988
##          Pos Pred Value : 0.8824
##          Neg Pred Value : 0.7288
##              Prevalence : 0.7638
##          Detection Rate : 0.7111
##    Detection Prevalence : 0.8059
##       Balanced Accuracy : 0.7649
##
##        'Positive' Class :  <=50K
##
```

```
error_logReg <- 1 - sum(table_logReg[row(table_logReg)==col(table_logReg)])/sum(table_logReg)
error_rate <- bind_rows(error_rate, data_frame(Method="Logistic Regression", Error_Rate = error_logReg))
error_rate %>% knitr::kable()
```

Method	Error_Rate
Naive Bayes	0.1735766
Logistic Regression	0.1474111

Finally, let's try Random Forests, but first need convert factor variables into numeric:

```
numCols <- c('workclass', 'education', 'marital', 'occupation', 'relationship', 'race', 'sex', 'country
adult_train_num <- adult_train
adult_train_num[,numCols] %<>% lapply(function(x) as.numeric(x))
adult_test_num <- adult_test
adult_test_num[,numCols] %<>% lapply(function(x) as.numeric(x))
```

```
model_RF <- randomForest(income~., data=adult_train_num, ntree=400)
pred_RF <- predict(model_RF, adult_test_num, type = "response")
(table_RF <- table(adult_test_num$income, pred_RF))
```

```
##         pred_RF
##           <=50K  >50K
##    <=50K  11710   725
##    >50K    1504  2342
```

```
error_RF = 1 - sum(table_RF[row(table_RF)==col(table_RF)])/sum(table_RF)
confusionMatrix(pred_RF, adult_test_num$income)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  <=50K  >50K
##      <=50K  11710  1504
##      >50K     725  2342
##
##                Accuracy : 0.8631
##                  95% CI : (0.8577, 0.8683)
##     No Information Rate : 0.7638
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5921
##  Mcnemar's Test P-Value : < 2.2e-16
```

```
##
##             Sensitivity : 0.9417
##             Specificity : 0.6089
##          Pos Pred Value : 0.8862
##          Neg Pred Value : 0.7636
##              Prevalence : 0.7638
##          Detection Rate : 0.7192
##    Detection Prevalence : 0.8116
##       Balanced Accuracy : 0.7753
##
##        'Positive' Class :  <=50K
##
```

```r
error_rate <- bind_rows(error_rate, data_frame(Method="Random Forests", Error_Rate = error_RF))
error_rate %>% knitr::kable()
```

Method	Error_Rate
Naive Bayes	0.1735766
Logistic Regression	0.1474111
Random Forests	0.1369081

Conclusion

Having applied three algorithms - Naive Bayes, Logistic Regression, and Random Forests - the latter turned out to be the most accurate in predicting whether an individual is earning above 50k or not. Random Forests accuracy reached 86.3% (error rate of 13.7%) and, even more importantly, the Kappa value (a metric that compares an Observed Accuracy with random chance) was a significant 0.59.