STAT 5703 Final Project

ANALYSIS OF CENSUS INCOME DATASET

Xue Cao 101020185

Xiyu Liang 101086285

Ke Xu(vicky) 101094337

Abstract

This technical report explores and analyzes the census income dataset from the UCI Machine Learning Repository. Since there are both quantitative and qualitative variables in our dataset, so we visualized data and used MCA (Multiple Correspondence Analysis) to do the dimension reduction analysis. We also performed the variable clustering analysis for our dataset. In unsupervised learning part, we used 'K-prototypes' for this mixture model to do the clustering and we also used association rule to explore further relationship between the variables in our data. Then we tried to find a model that best predict our data (supervised learning), we fit logistic regression, Naïve Bayes, SVM, Neural Network, K-nearest neighbors, regression tree and Random Forest models. Finally, we used ROC and the area under the curve (AUC) to conclude that the Random Forest has the best prediction in those seven models.

Table of Contents

Introduction

As we've already used the Heart dataset to do the classification trees and random forest, we decided to try a new dataset for our final project, which is called the census dataset. This multivariate dataset contains 14 attributes and 48842 observations, and it is divided into the training (32561 rows) and testing dataset (16281 rows). The response variable is 'income', which indicates whether a person makes over 50K annually. We are going to understand our dataset in five dimensions: visualization, dimension reduction, data reduction, unsupervised learning and supervised learning. We find the best model with the highest classification rate.

Data Import and Cleaning

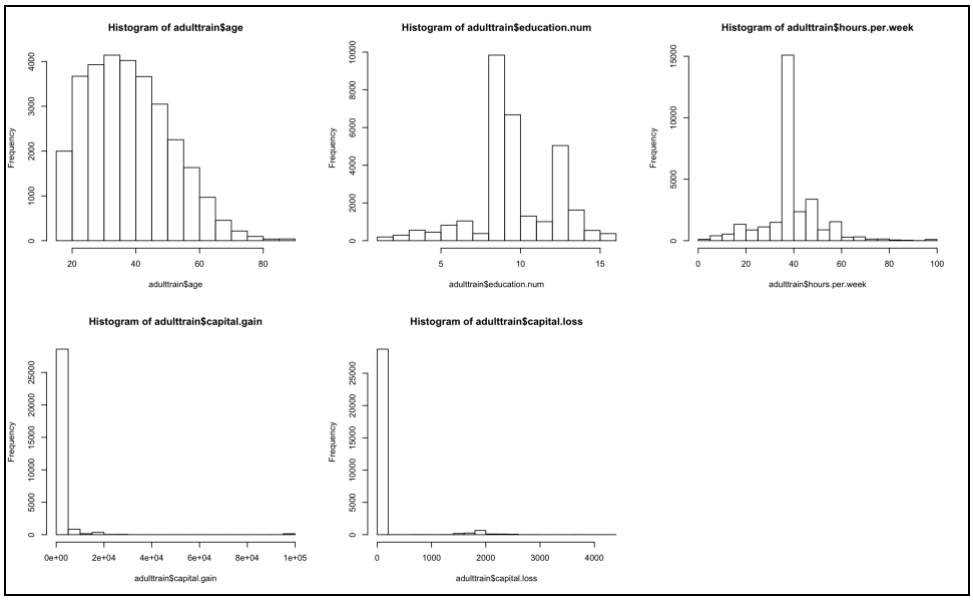
Before we get started, we first need to load the dataset into R by using the following code:

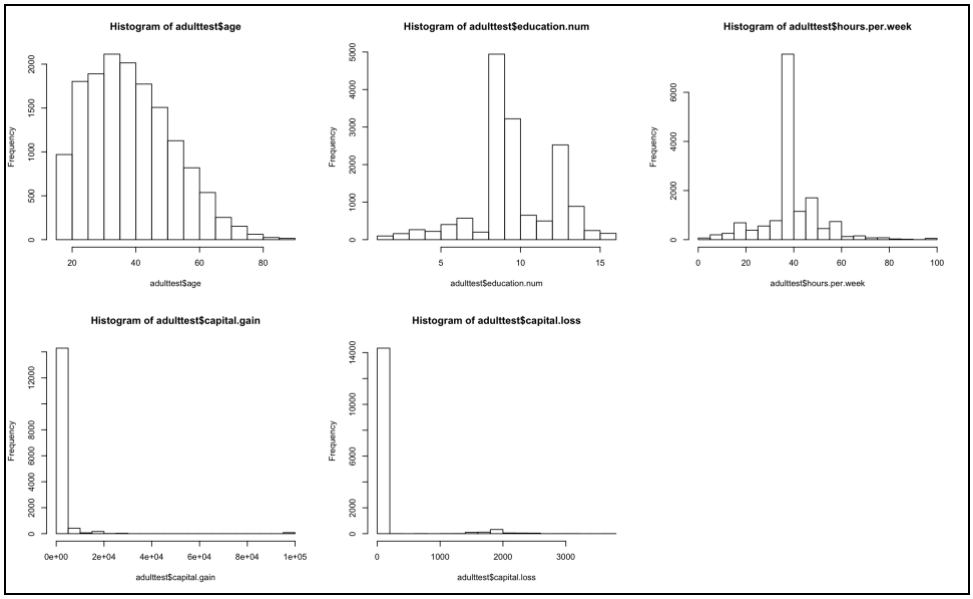
```
datatrainurl <- 'http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data'
datatesturl <-'http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.test'
datanameurl <- 'http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.names'
adulttrain <- read.table(datatrainurl,sep = ',',stringsAsFactors = FALSE)
adulttest <- readLines(datatesturl)[-1]
adulttest <- read.table(textConnection(adulttest),sep = ',',stringsAsFactors = FALSE)

adultnames <- readLines(datanameurl)[97:110]
adultnames <- as.character(lapply(strsplit(adultnames,':'), function(x) x[1]))
adultnames <- c(adultnames,'income')
colnames(adulttrain) <- adultnames
colnames(adulttest) <- adultnames
```

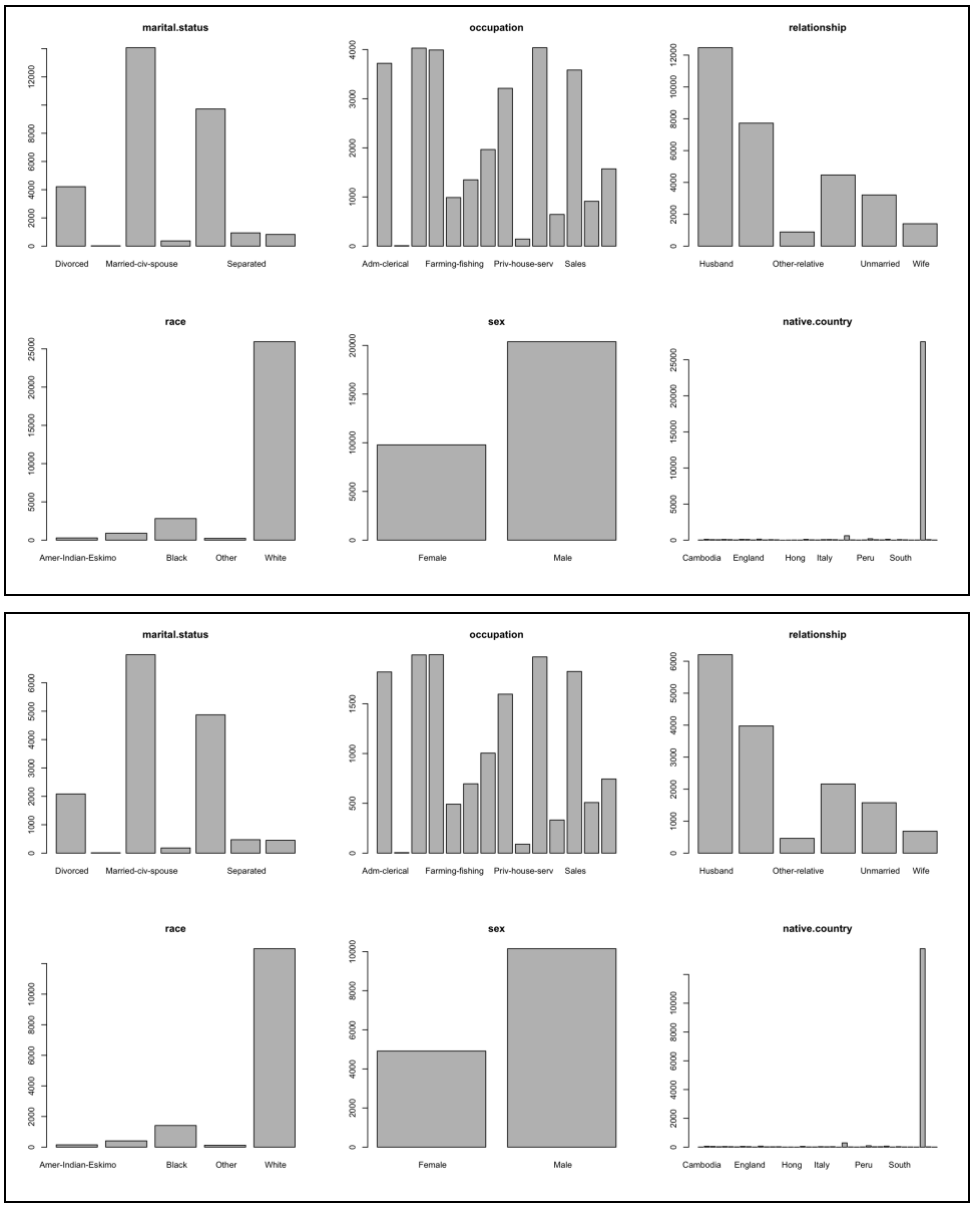
After loading, we first noticed that the question mark '?' denotes the missing value. We can remove them by detecting and subsetting:

```
# We first remove missing value (ones with ' ?')
no.question.mark <- apply(adulttrain, 1, function(r) !any(r %in% ' ?'))
adulttrain <- adulttrain[no.question.mark,]

no.question.mark <- apply(adulttest, 1, function(r) !any(r %in% ' ?'))
adulttest <- adulttest[no.question.mark,]

adulttrain <- as.data.frame(unclass(adulttrain),stringsAsFactors = T)
adulttest <- as.data.frame(unclass(adulttest),stringsAsFactors = T)
```

We then looked at the structure of the data, analyzed whether all variables should be kept. We first noticed that 'education' is the replication of 'education.num', thus we removed the column 'education'. The variable 'fnlwgt' is the weights on the CPS files are controlled to independent estimates of the civilian non-institutional population of the US, so we can also remove it as well. (For more information and analysis about those two variables, please see the **DataVisualization** part) As this dataset contains both categorical and quantitative variables, we need to take a look at the distributions of all numerical covariates by looking at the histograms, and all possible values of all categorical covariates (which are also the possible factor levels) by using the 'table' function:


Histogram of adulttrain$age
Frequency
adulttrain$age
Histogram of adulttrain$education.num
Frequency
adulttrain$education.num
Histogram of adulttrain$hours.per.week
Frequency
adulttrain$hours.per.week
Histogram of adulttrain$capital.gain
Frequency
adulttrain$capital.gain
Histogram of adulttrain$capital.loss
Frequency
adulttrain$capital.loss


Histogram of adulttest$age
Frequency
adulttest$age
Histogram of adulttest$education.num
Frequency
adulttest$education.num
Histogram of adulttest$hours.per.week
Frequency
adulttest$hours.per.week
Histogram of adulttest$capital.gain
Frequency
adulttest$capital.gain
Histogram of adulttest$capital.loss
Frequency
adulttest$capital.loss



For categorical variables, most of them have reasonable numbers of classes. The variable 'native.country' has 41 levels but the majority observations of it are 'united.states', the rest of them have only a few observations, we thus decided to use the variable 'race' instead. On the numerical side, both 'capital.gain' and 'capital.loss' are highly skewed, this is a typical phenomenon because most people have neither capital gain nor loss. So we combined the 'capital.gain' and 'capital.loss' columns together and create a new column called 'capital.change'. Since the proportions of people who have capital changes in the training and testing datasets are around 13%, we keep 'capital.change' for analysis. We also changed the classes of response variable from '<=50K' and '>50K' to factors '0' and '1'.

The final step of data preprocessing is outlier detection. As 'capital.change' and 'hours.per.week' are highly concentrated variables, and 'education.num' is an ordinal variable. Therefore, we only detect outliers for the variable 'age'. Traditionally, an observation would be diagnosed as an outlier by using the following formula:

$$obs > Q3 + 1.5 * IQR \mid obs < Q1 - 1.5 * IQR$$

In R, we used 'boxplot.stat()$out' to detect vector's outliers, and we wrote a wrapped function to automate the process:

```
no.outlier<-function(data,x)
{
  for (i in x)
  {
    a<-boxplot.stats(data[,i])$out
    data<-data[!data[,i]%in%a,]
    print(length(a))
  }
  return(data)
}
str(adulttrain)

adulttrain <- no.outlier(adulttrain,1)
adulttest <- no.outlier(adulttest,1)
dim(adulttrain)
dim(adulttest)
```

After outlier removal, we had 29993 observations in the training dataset and 15003 observations in the testing dataset. We commenced the analysis by data visualization.

Visualization

First, we detected skewed variables and got the results as below:

```
[1] "age: 0.532798220751413"
[1] "fnlwgt: 1.44746753851606"
[1] "education.num: -0.310610643179668"
[1] "capital.gain: 11.7886111390678"
[1] "capital.loss: 4.51615434716475"
[1] "hours.per.week: 0.34053384790218"
```

If a variable's absolute value is greater than 1, then it can be considered highly skewed. According to the above result, 'fnlwgt, 'capital.gain' and 'capital.loss' are all highly skewed and we treat them by taking the log transformation in the visualization procedure.

Next, we removed the outliers in the 'age' and 'fnlwgt' columns. After these step, we save the dataset named 'Adultdata'.

In order to gain an idea for each attribute of the dataset, we plot distributions of each column.

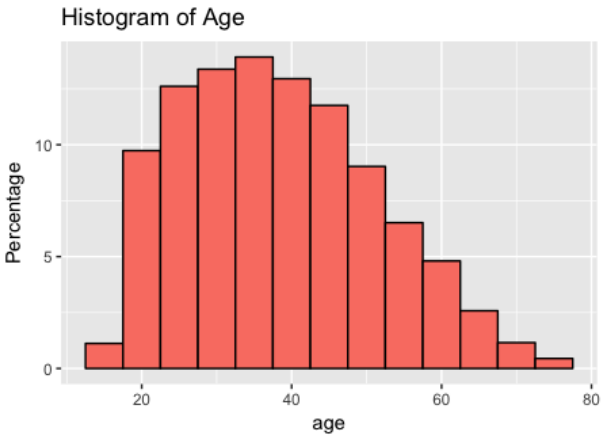
Explore Numerical Variable

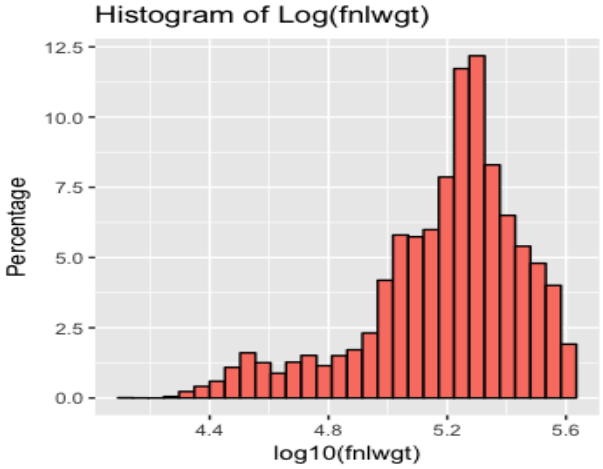
First, we calculate the correlations between numerical variables:

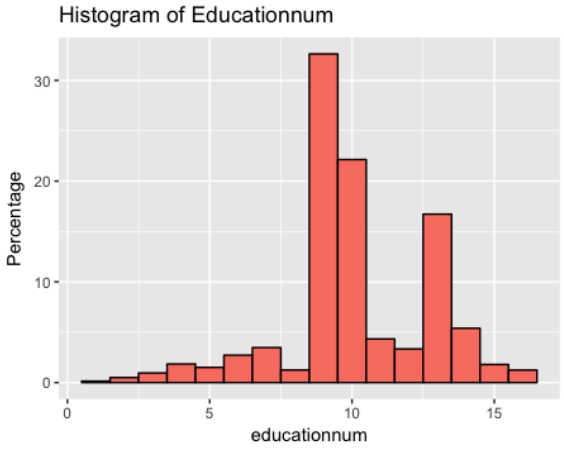
	age	fnlwgt	educationnu	capitalgain	capitalloss	hoursperweek
age	1	-0.075792	0.03762295	0.07968324	0.05935058	0.10199224
fnlwgt	-0.075792	1	-0.041993	-0.0041105	-0.0043488	-0.01867873
educationnun	0.03762295	-0.041993	1	0.1269068	0.08171132	0.14620624
capitalgain	0.07968324	-0.0041105	0.1269068	1	-0.0321023	0.08388042
capitalloss	0.05935058	-0.0043488	0.08171131	-0.0321023	1	0.05419487
hoursperwee	0.10199224	-0.0186787	0.14620624	0.08388042	0.05419487	1

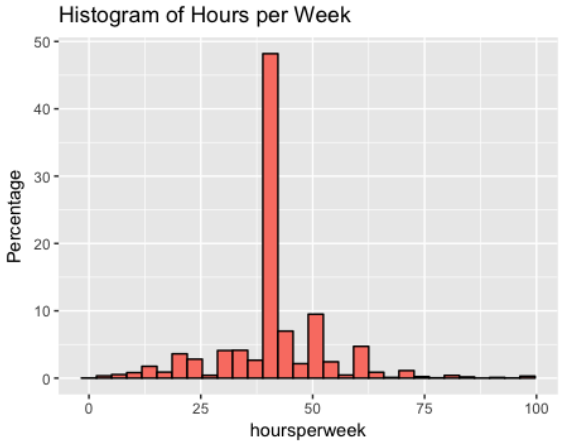
The result above shows that numerical variables are nearly uncorrelated.

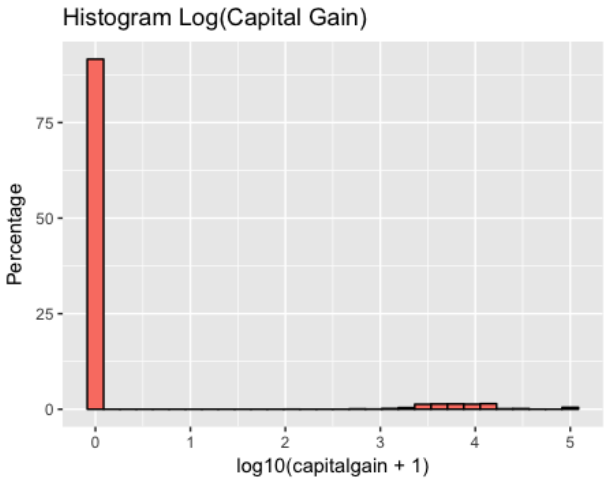
Next, we plot all the histograms of all the numerical variables as below:

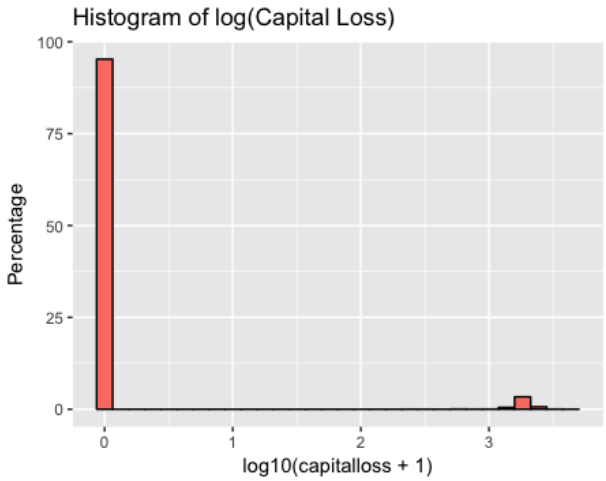








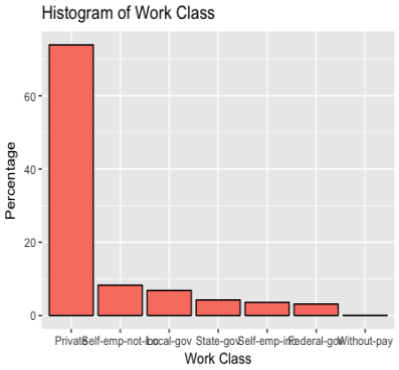


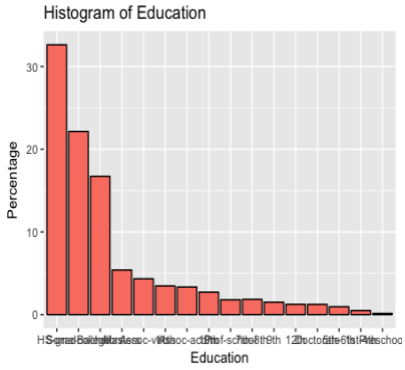


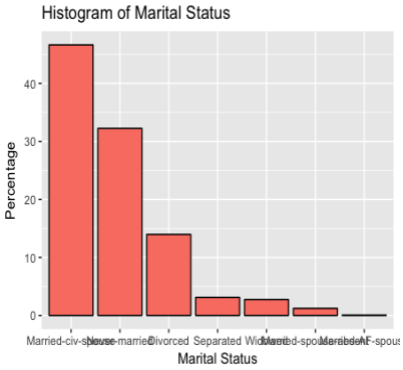
From the plots above we can find the variable 'age', 'log(fnlwgt)', 'education.num' and 'hours.per.week' have broad distributions, hence we will keep these variables in later regression analysis. However, 'capital.gain' and 'capital.loss' have very narrow distributions. They are quite skewed and mostly concentrated at zero value: 91.61912% of Capital Gain and 95.26779% of Capital Loss have zero value. Therefore, we will remove them from the model.

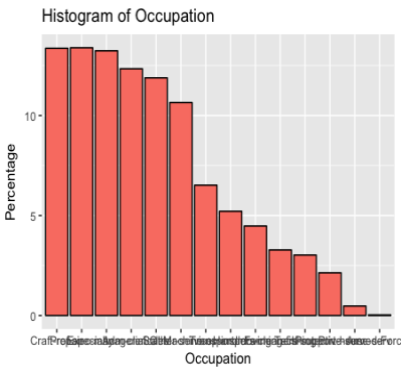
Explore Categorical Data

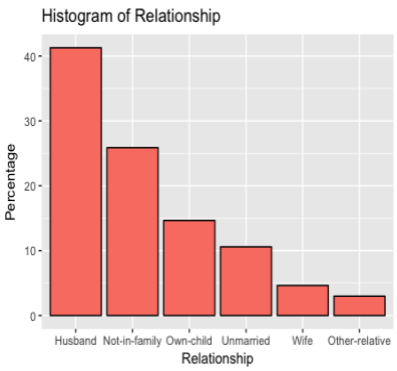
Next, we plot all the histograms of all the categorical variables as below:

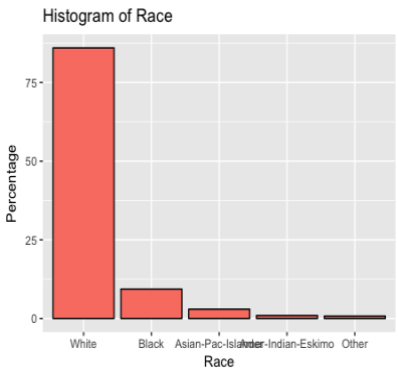


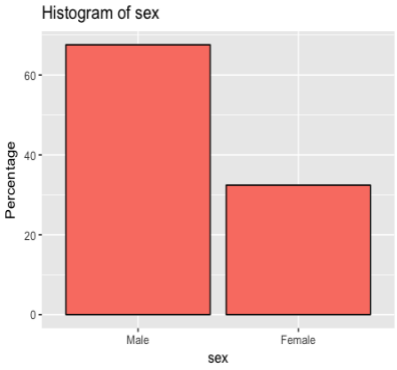


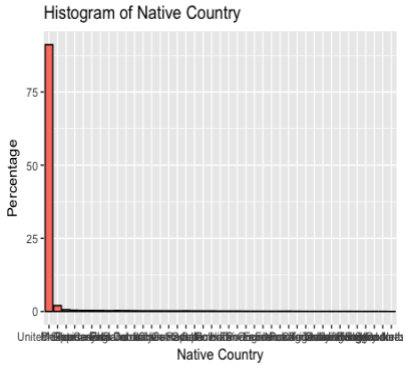










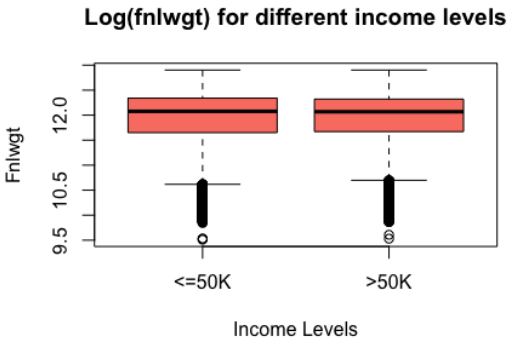


From the plots above we can find the variable 'workclass', education', 'marital.status', 'occupation', ' relationship', 'race' and 'sex' have broad distributions, hence we will use these variable in later regression analysis. However, Native Country has a very narrow distribution. As we can see, 90% of population coming from the United States, therefore we will remove 'Native Country' from the model.

In addition, we can observe that correlation between 'education' and 'education.num' is high. This is because education and 'education.num'exhibit same information but in different pattern: 'education.num' is the numeric representation of 'education' Hence, we will only choose one of them in the future model, which also verified our assumption at the beginning.

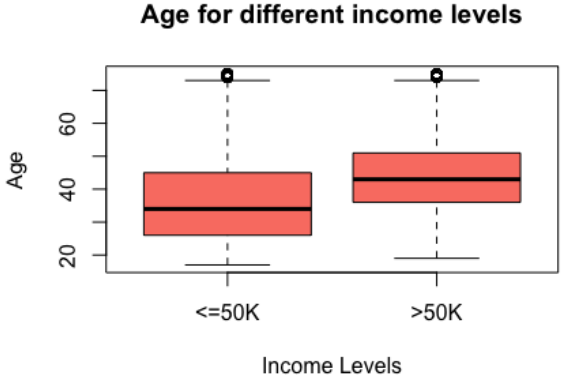
Correlation between numerical variables and income class

To explore the relationship between numerical variables and income, we got some boxplot as below:



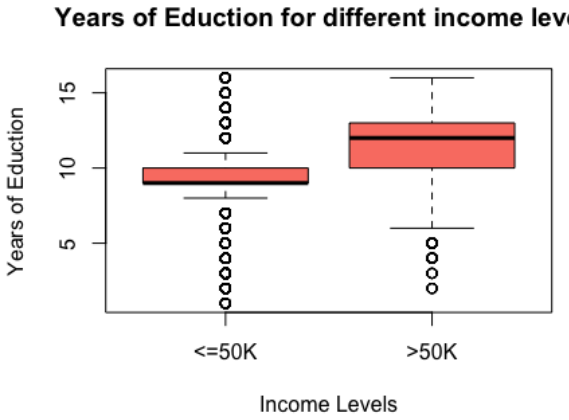
From the boxplot1 on the left, we noticed that 'fnlwgt' shows little variation with Income levels, therefore 'fnlwgt' will be excluded from the model. This plot verified our assumption at the beginning.

From the boxplot1 on the left, we that most of the adults in this dataset between 25 to 50 years of age. The who earn <=50k a year are mostly



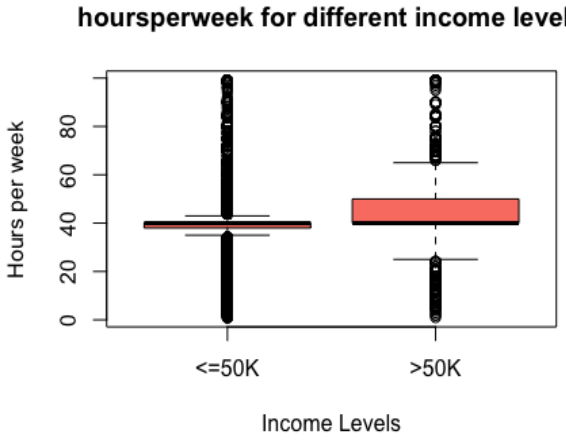
found are adults between

25 to 45(average 33), while who earn >50k a year are mostly between 35 to 50 (average 42). It seems like older people would have higher salary.



From the boxplot3 on the left, we found that most of the adults' education level are between 9 to 13. The adults who earn <=50k a year are mostly between level 9 to level 10, while who earn >50k a year are mostly between level 10 to level 13. We can conclude that the salary increases as the years of education increases.

From the boxplot4 on the left, most of adults work between 38 to 50 hours a The adults who earn <=50k a year are work between 38 to 40 hours per while who earn >50k a year are work between 40 to 50 hours per This indicate that people who invest time on work tend to be earning more



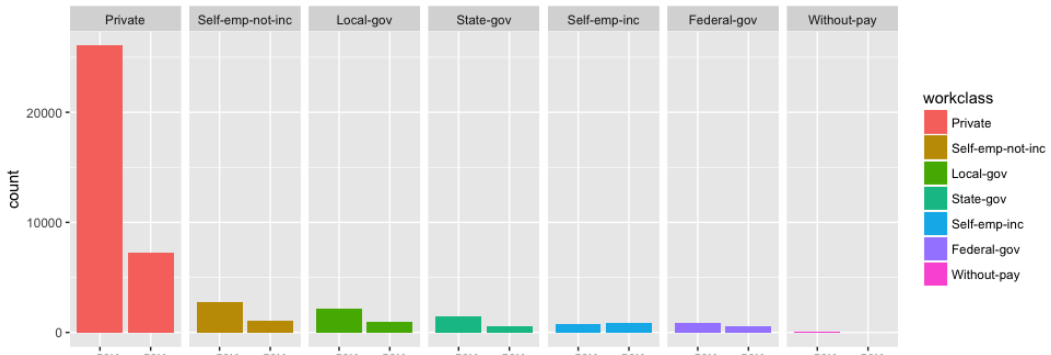
the week. mostly week, mostly week. more salary.

Variables 'age', 'education' and 'hours.per.week' have significant correlation with Income hence we will keep them in the model.

Correlation between categorical variables and income class

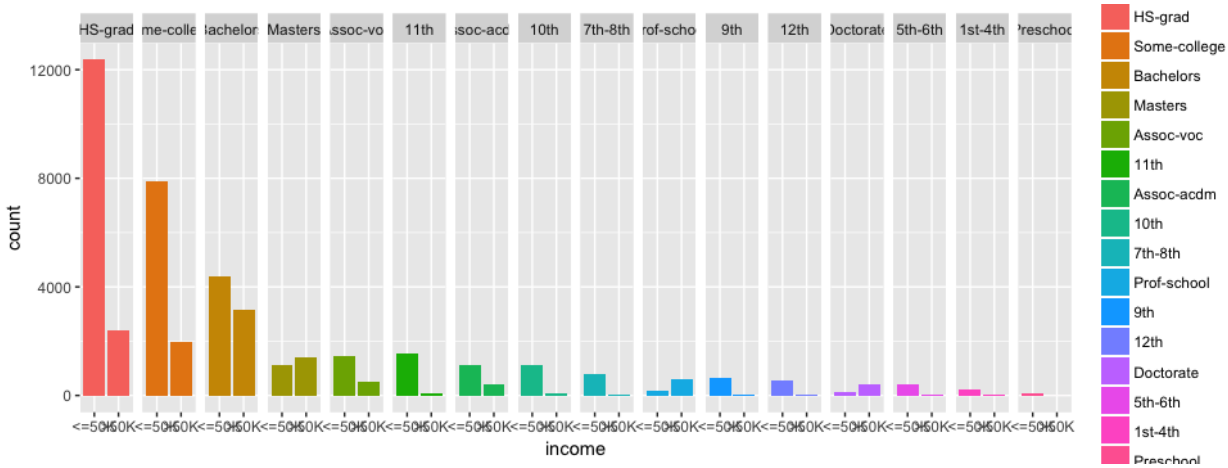
To explore the relationship between categorical variables and income, we got some bar plot as below:

- 'workclass' vs 'income'



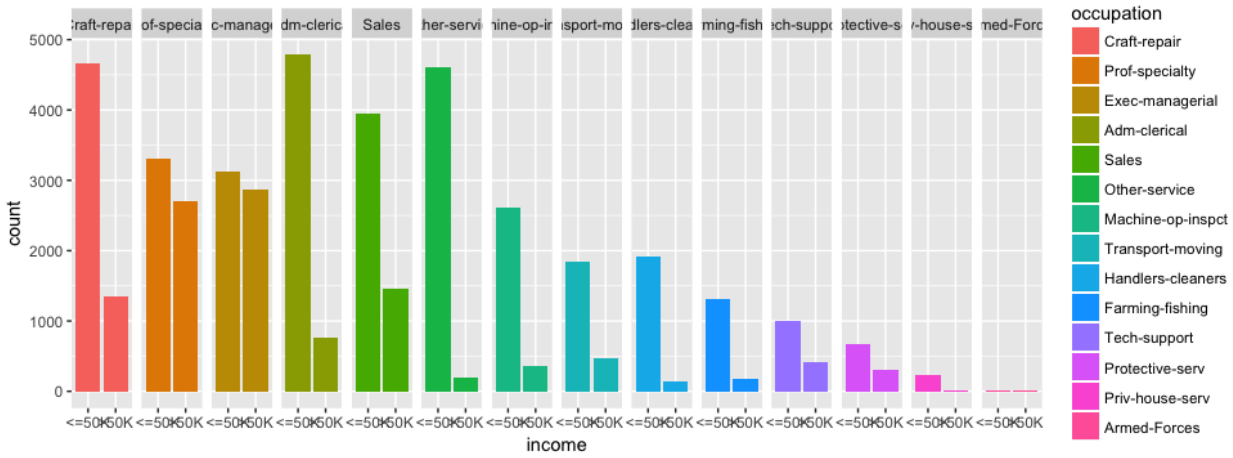
From the above plot, the ‘private’ section has the most people work in and has the largest number of population that earn more than 50K per year. Hence, those who are self-employed have the highest tendency of making greater than $50K a year.

- ‘education’ vs ‘income’



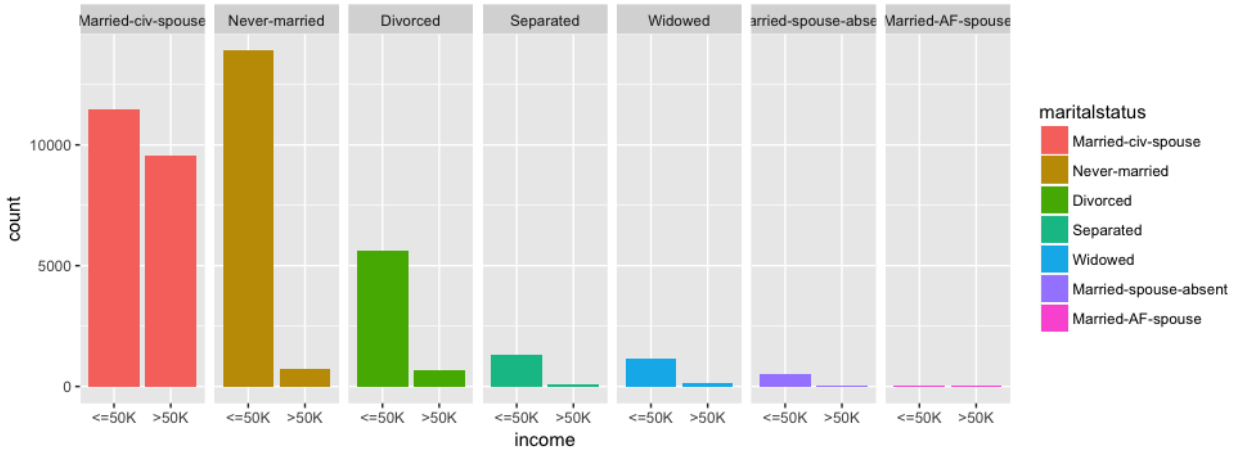
From the above plot, we can see that most of adults have the education level between high school to Bachelor degree. Higher education level may result in a higher possibility to earn high salary. For those less than or equal to 7 years of education people, most have an annual income of less than $50K, only 10% have an annual income of greater than $50K. While for those people with doctorate degrees, nearly 75% makes greater than $50K a year, however, there are roughly 25% of people with doctorate degrees earning less $50K a year.

- ‘occupation’ vs ‘income’

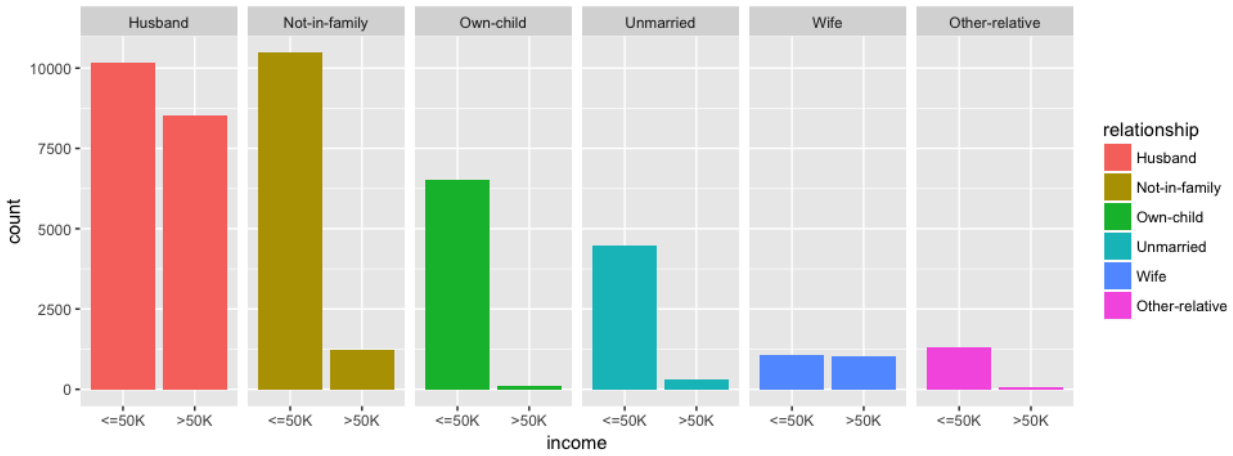


From the above plot, we can notice that income varies greatly across different occupations. Nearly half of 'Prof-specialty' and 'Exec-managerial'(professional/high skill occupation) makes greater than $50,000 a year and craft-repair, still have around 30% of them that have an annual income of greater than $50K. The percentage is less than 10% for 'other-service'(non-professional/low skill occupation).

- 'marital.status'/'relationship' vs 'income'

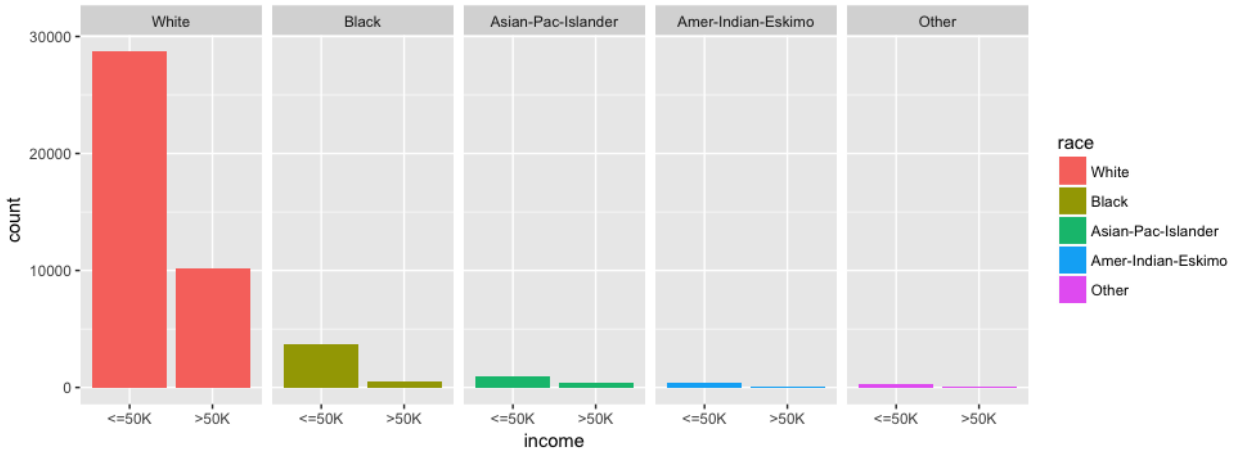


From the above plot, almost half of married people are making greater than $50K a year. For the people who are not married/do not have spouses, less than 10% of them making greater than $50K a year. It can be inferred that the marital relationship has a significate influence on salary.



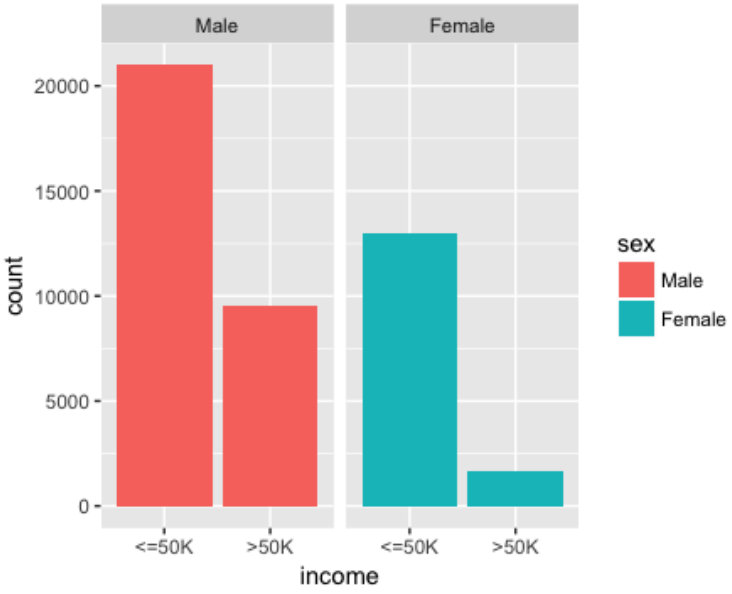
Relationship exhibits almost the same information compared to ‘marital.status’.

- ‘race’ vs ‘income’



From the above plot, nearly 30% of Whites and nearly 50% of Asian-Pac-Islander have an annual income of greater than $50K, less than 20% of blacks, Amer-Indian-Eskimo and other have an annual income of greater than $50K.

- Sex VS Income



From the left plot, almost half of males have an annual income of greater than $50K, but less than 20% of female have an annual income of greater than $50K. The data shows that male employees are more competitive in terms of salary.

Dimension reduction (MCA and Variable Clustering)

MCA is an extension of correspondence analysis (CA), which can be seen as a generalization of principal component analysis when the variables to be analyzed are categorical instead of quantitative. It is not just a multivariate data analysis tool for finding groups of variables that are as correlated as possible among themselves, one can also use this method to understand the relationship between the target variable and input variables. Our data is multivariate and there are both categorical and numerical independent variables in our dataset. Therefore, instead of using PCA, MCA is a better choice.

Data pre-processing

In the data visualization part, we realized that the variable ‘fnlwgt’ has no relationship with the target variable. And there is a strong relationship between ‘education’ and ‘education.num’, these two variables may provide almost the same information. However, since we only have 14 active variables and we do not want to lose any information, so in this part we believe it is better to assume all active variables will influence an individual’s income first.

We next convert some of the remaining numerical variables to categorical variables by using the ‘quantcut’ function. First, we discretize variables ‘fnlwgt’, ‘education.num’, ‘capital.gain’, ‘capital.loss’ and ‘hours.per.week’ into quartiles. Then we take the ‘Age’ variable and turn it into 5 groups (under 25, 26 to 35, 36 to 49, 50 to 64, 65 and up).

After performing these data transformations, we obtain a dataset as follows:

```
> summary(Adult)
               age                    workclass                             fnlwgt              education                     education.num
 age:25 and under:5424   Federal-gov     :  900   fnlwgt [1.38e+04,1.16e+05]:7275   HS-grad     :9500   education.num [1,9]   :13031
 age:26 to 35    :7801   Local-gov       : 2004   fnlwgt (1.16e+05,1.76e+05]:7273   Some-college:6462   education.num (9,10]  : 6462
 age:36 to 49    :9886   Private         :21496   fnlwgt (1.76e+05,2.29e+05]:7274   Bachelors   :4881   education.num (10,13]: 7141
 age:50 to 64    :5199   Self-emp-inc    : 1035   fnlwgt (2.29e+05,4.18e+05]:7274   Masters     :1575   education.num (13,16]: 2462
 age:65 and up   : 786   Self-emp-not-inc: 2413                                     Assoc-voc   :1278
                         State-gov       : 1234                                     11th        :1013
                         Without-pay     :   14                                    (Other)      :4387
                marital.status              occupation               relationship                   race              sex
  Divorced             : 4100   Prof-specialty :3919   Husband       :12038   Amer-Indian-Eskimo:  285   Female: 9485
  Married-AF-spouse    :   19   Craft-repair   :3894   Not-in-family : 7425   Asian-Pac-Islander:  888   Male  :19611
  Married-civ-spouse   :13590   Exec-managerial:3848   Other-relative:  845   Black             : 2620
  Married-spouse-absent:  357   Adm-clerical   :3596   Own-child     : 4326   Other             :  227
  Never-married        : 9361   Sales          :3456   Unmarried     : 3098   White             :25076
  Separated            :  905   Other-service  :3088   Wife          : 1364
  Widowed              :  764  (Other)         :7295
           capital.gain                   capital.loss                 hours.per.week              native.country      income
 capital.gain 0    :26654   capital.loss 0       :27706   hours.per.week [1,40) : 6418   United-States:26599   <=50K:21819
 capital.gain (0,1]: 2442   capital.loss (0,3.9]: 1390   hours.per.week 40     :13742   Mexico       :  521   >50K : 7277
                                                          hours.per.week (40,45]: 2285   Philippines  :  186
                                                          hours.per.week (45,99]: 6651   Germany      :  126
                                                                                         Puerto-Rico  :  105
                                                                                         Canada       :  104
                                                                                        (Other)       : 1455
```

From the summary, we can see that the most occurring individuals are between 36 and 49, work for private employers, who are male (husbands), white, have at least a high school education, work 40 hours a week, come from the US and make less than 50 thousand dollars a year.

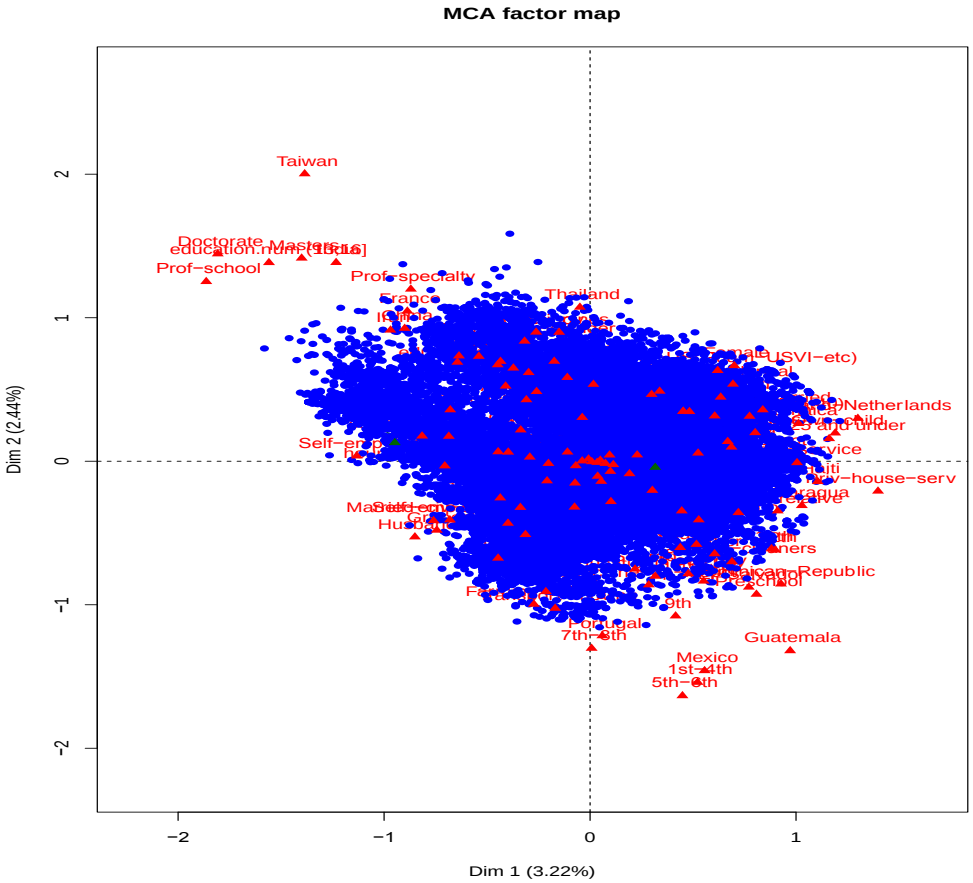
MCA

Now all of our variables are factors, we can perform MCA on our dataset. From the MCA output, we can extract all the results (coordinates, squared cosine and contributions) for the active individuals/variable categories. The package we used are 'PCAmix' and 'FactoMineR'.

In this dataset, we have 14 active variables and 1 target variable which is the total income. In R, we specify the target variable as the 'Supplementary qualitative variable'.

```
> summary(res,ncp=3,nbelements=Inf)

Call:
MCA(X = Adult, quali.sup = c(15))


Eigenvalues
                       Dim.1   Dim.2
Variance               0.242   0.183
% of var.              3.225   2.440
Cumulative % of var.   3.225   5.665
```

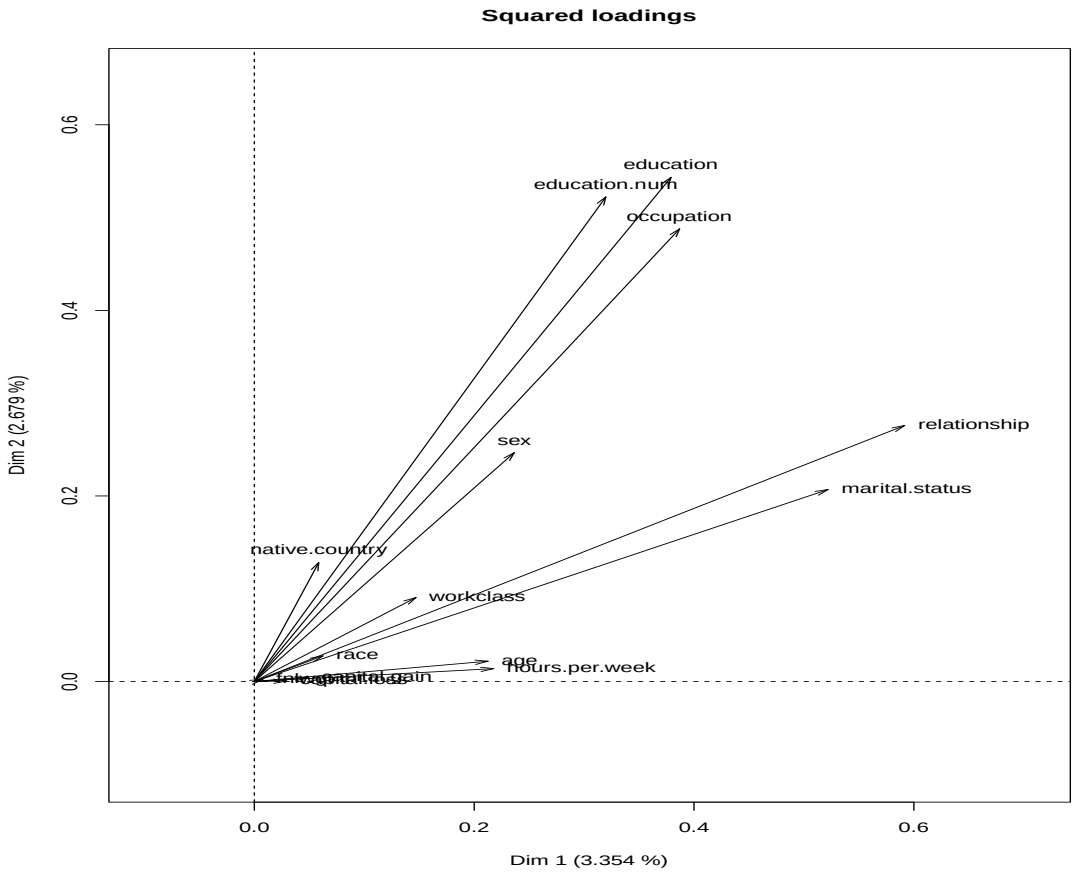
We can see that the first and the second dimension only explained 5.665% variance in total. It seems like the first two dimensions cannot explain most information of our dataset. While since the MCA method converts all factors to dummy variables (one categorical variable is coded with several columns). As a consequence, the variance of the solution space is artificially inflated and therefore the percentage of variance explained by the first dimension is severely underestimated. Taking into account of this specific coding schema, we still can use the first two dimensions to visualize our data.



The figure above displays all variable categories and all individuals, since the size of our dataset is big, the plot is quite loaded and hard to read. Also, it is difficult to extract useful information since most central ones are overlapping. Usually, this factor map can only be used for understanding the global distribution of our data as we as identifying outliers.

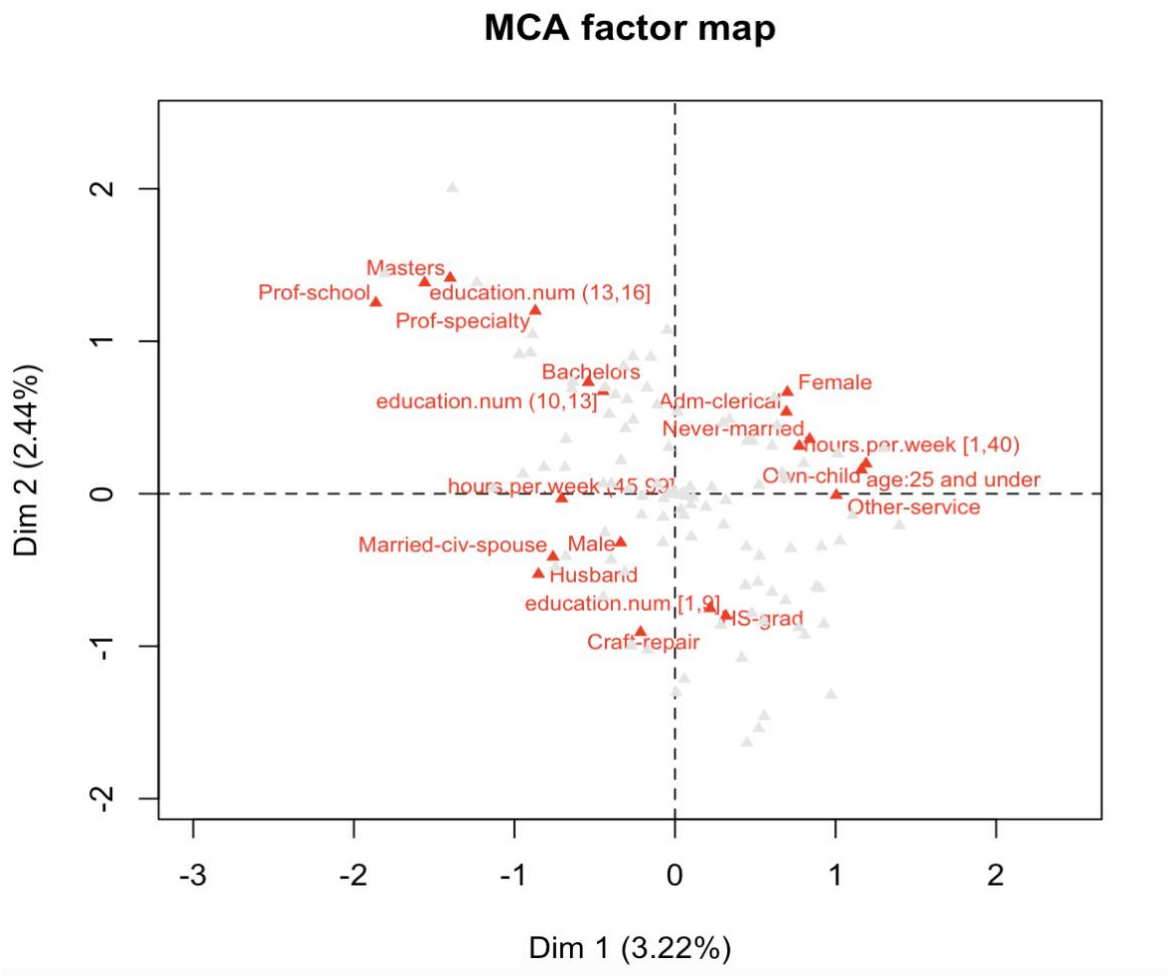
By using the 'FactoMineR' package, we can plot individuals and variables separately but since these two plots are still hard to read so we do not show them in this paper.

Then we used the 'PCAmix' function to obtain the 'Squared loading' plot.

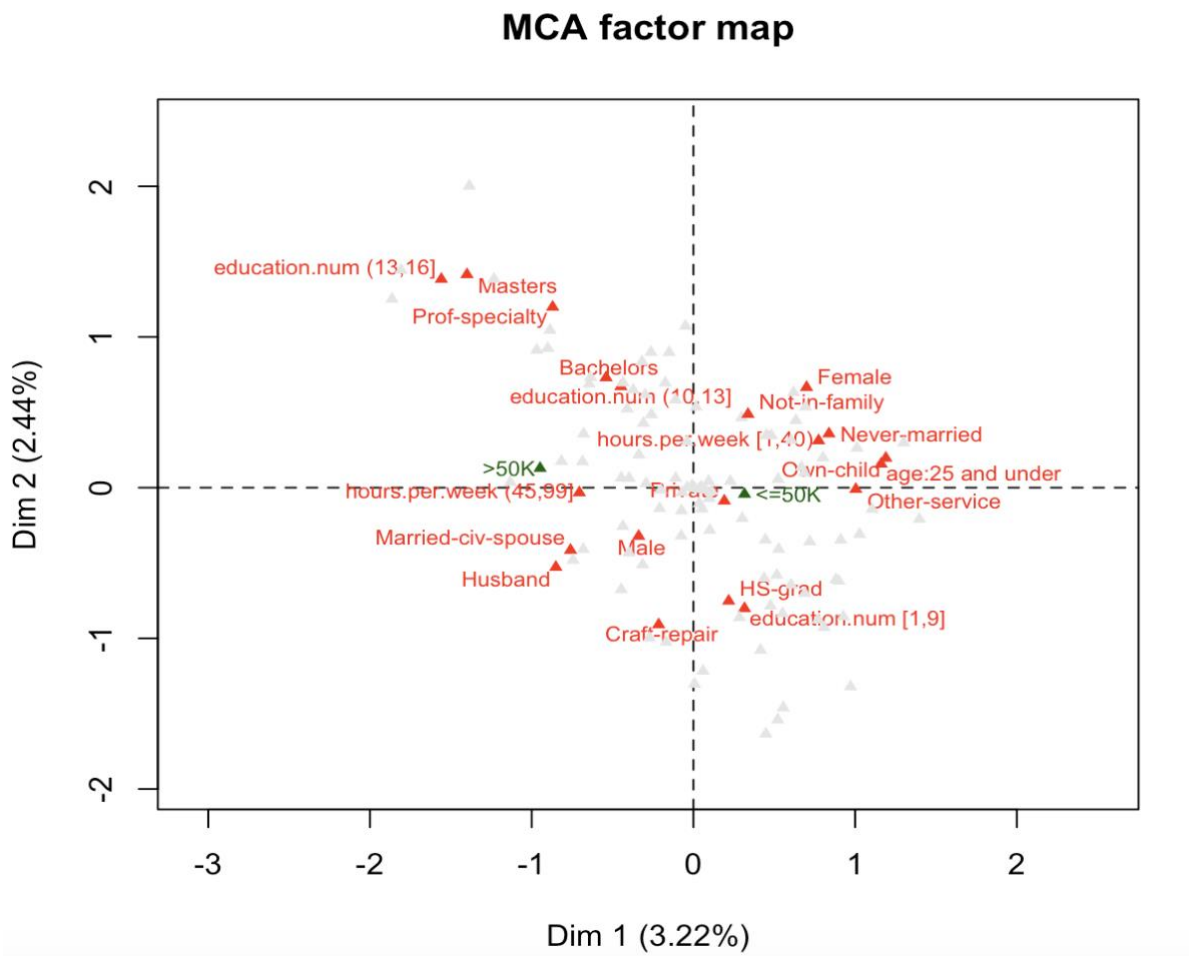


'PCAmix' includes the ordinary principal component analysis (PCA) and multiple correspondence analysis (MCA) as special cases. Squared loadings are correlation ratios for qualitative variables and squared correlation for quantitative variables.

This plot is easier to read and we can tell the correlations between variables and dimensions. We can see that variables 'education', 'education.num' and 'occupation' are clustered and they are further from the center, which means that they are the most relevant variables. Also, these three variables have the strongest correlation with the second dimension. On the other hand, variable 'relationship' and 'marital.status' are correlated as well as be highly related with the first dimension. Compared to the distribution of individuals, we are more interested in that of variables. So, now we focus on variables which contributed/correlated most.



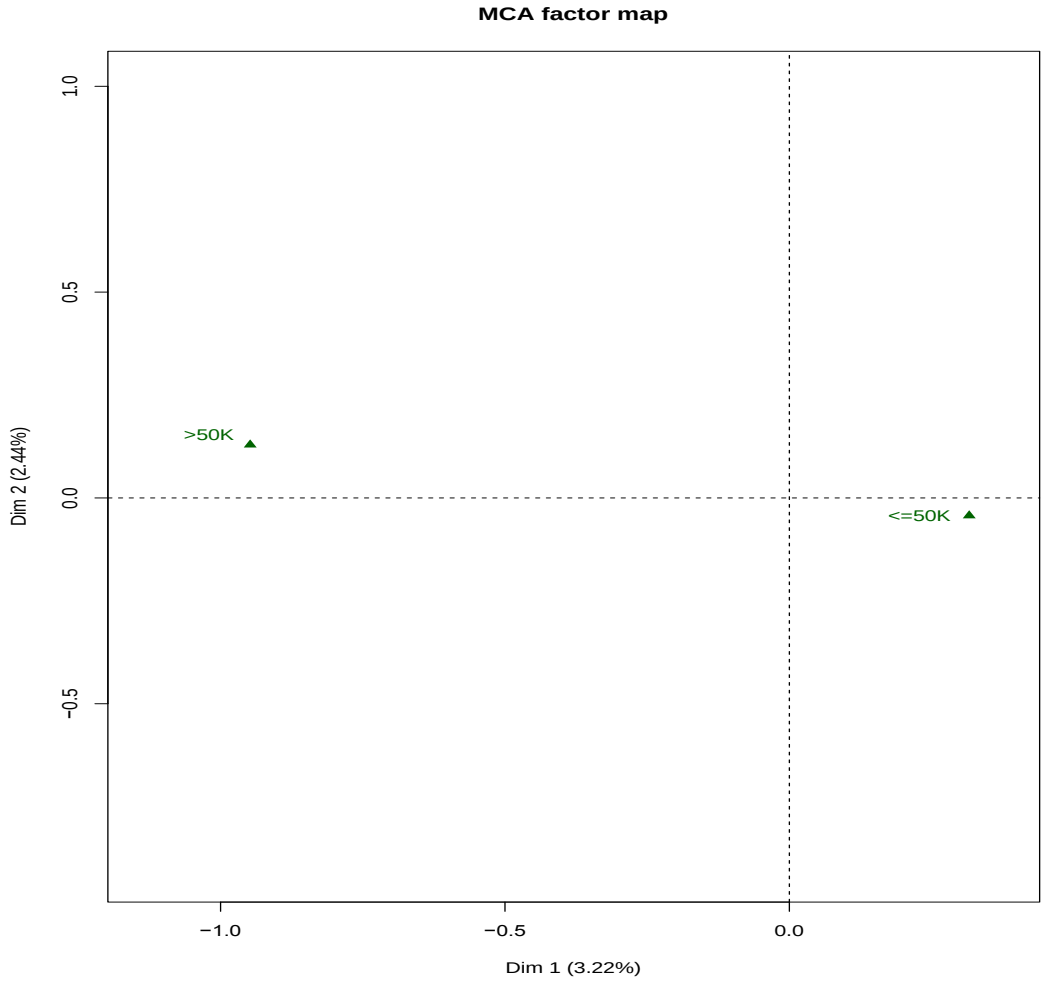
This plot shows the 20 variable which contributed most to the dimensions. It is not hard to notice some pattern in the distribution of them. For example, ‘female’ and ‘male’ are be opposites. Highly educated people (Bachelors, Masters, etc.) clustered at the top left, whereas the low educated people are in the opposite position. Very young people would be in the middle right of the plot, also the worked hours per week seems to follow a straight line along the first dimension.



We can also plot the 20 variables that are most correlated to the dimensions. The two levels of our target variable (income) appear on this plot, with the distribution follows the first-dimension axis and we can see that the 'right' part of the plot would be the one containing people earning less than 50K a year while the 'left' part would be people earning more than 50K a year. Therefore, we can conclude that the first dimension could also be the dimension of 'wealth'.

Combine the distribution of the target variable and the other active variables, we can keep reading information from this plot. For instance, people under 25 years old are less likely earning more than 50K dollars a year. On the second dimension, the 'female' variable is quite high while the 'male' is relatively low. Besides, the 'female' is in the right part of the plot, the 'male' is in the contrary left part of the plot. Considering the education variables and the target income variable, we can conclude that in general, women are highly educated but with lower income, men are less educated but can earn more relatively. (Note that this is just the conclusion we can get according to the information given in the plot, there is a need for much more careful study to verify it.)

We can also notice that compared with those people who work less than 40 hours a week, people who spend more time on working per week can probably make more money.



It is possible to hide active variables and individuals then look at the target variable only. The plot above shows two modalities of the target variable (income). We can see that the level '<=50K' may be more common since compared to the level '>=50K', it is more central. Again, based on this, we can consider the first dimension as the dimension of 'wealth'.

We can also give a more formal description of the dimensions, which can be quite useful when we have a lot of variables, using the 'dimdesc' function. In the figures below, we only kept those which the p-values equal to zero.

The tables '$`Dim 1`$quali' and '$`Dim 2`$quali' show the relevant variables (active and Supplementary, whereas the second ones show the '$category' that is the most relevant modalities.

```
$`Dim 1`
$`Dim 1`$quali
                      R2       p.value
age             0.340005113  0.000000e+00
workclass       0.125735661  0.000000e+00
education       0.376145964  0.000000e+00
education.num   0.345137180  0.000000e+00
marital.status  0.535235372  0.000000e+00
occupation      0.377670273  0.000000e+00
relationship    0.606405337  0.000000e+00
race            0.053398971  0.000000e+00
sex             0.236762377  0.000000e+00
capital.gain    0.060944521  0.000000e+00
hours.per.week  0.262917721  0.000000e+00
income          0.299775664  0.000000e+00

$`Dim 1`$category
                                 Estimate       p.value
 <=50K                          0.310877087  0.000000e+00
hours.per.week [1,40)           0.420458616  0.000000e+00
capital.gain 0                  0.218926050  0.000000e+00
 Female                         0.255253639  0.000000e+00
 Own-child                      0.412120236  0.000000e+00
 Other-service                  0.426835073  0.000000e+00
 Adm-clerical                   0.274005907  0.000000e+00
 Never-married                  0.256923275  0.000000e+00
education.num (9,10]            0.374631237  0.000000e+00
education.num [1,9]             0.307854232  0.000000e+00
 Some-college                   0.249705831  0.000000e+00
 Private                        0.237363498  0.000000e+00
age:25 and under                0.558373800  0.000000e+00

 >50K                          -0.310877087  0.000000e+00
hours.per.week (45,99]         -0.307100687  0.000000e+00
capital.gain (0,1]             -0.218926050  0.000000e+00
 Male                          -0.255253639  0.000000e+00
 Husband                       -0.591341513  0.000000e+00
 Prof-specialty                -0.495082686  0.000000e+00
 Exec-managerial               -0.400948394  0.000000e+00
 Married-civ-spouse            -0.529054559  0.000000e+00
education.num (13,16]          -0.614342127  0.000000e+00
education.num (10,13]          -0.068143342  0.000000e+00
 Prof-school                   -0.888950296  0.000000e+00
 Masters                       -0.661163288  0.000000e+00
 Bachelors                     -0.238134438  0.000000e+00
age:36 to 49                   -0.207904530  0.000000e+00
```

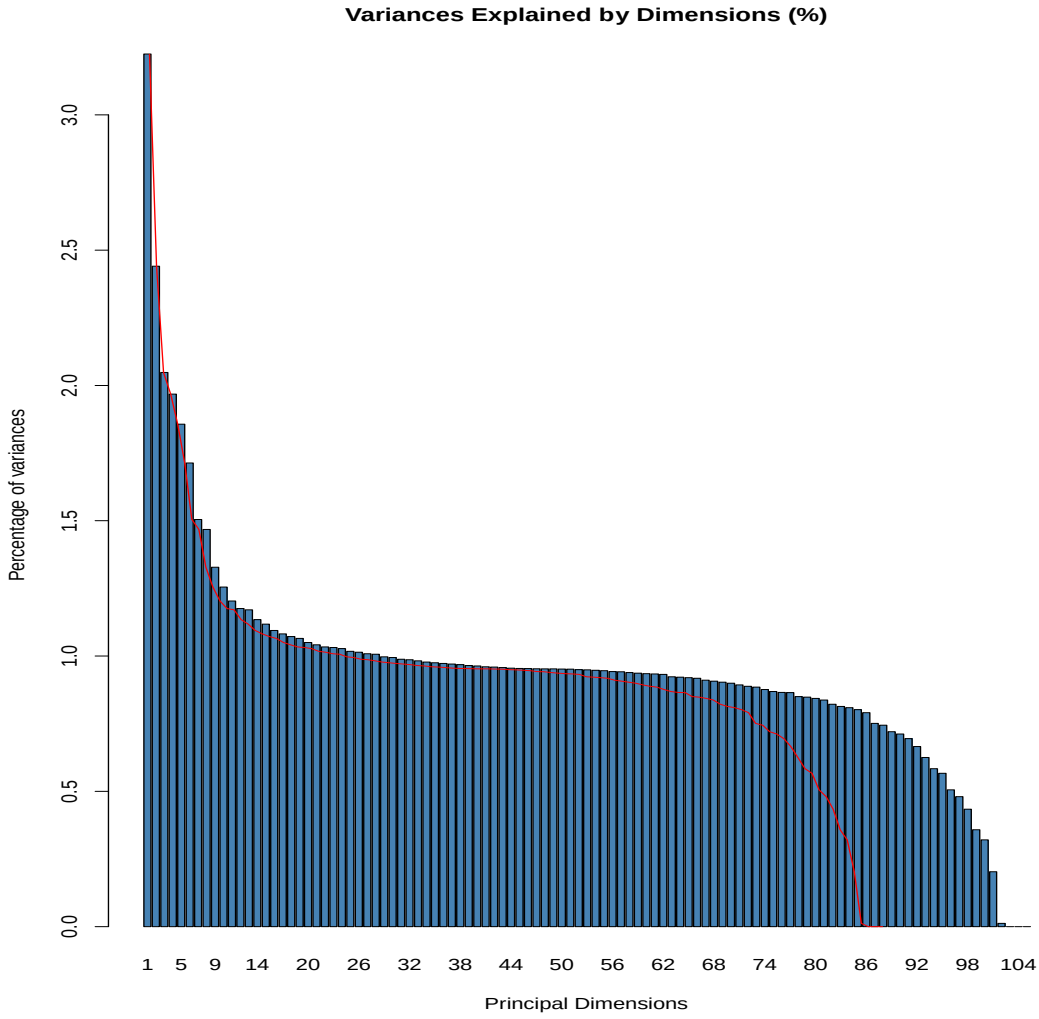
Although the first dimension only explained 3.22% variation, it is still very discriminative. We already known that this dimension separates the two levels of the target variable. And we can point out that the first dimension separates the working hour modalities: on the right (<=50K) there will be people working 1 to 40 hours per week, whereas on the left (>50K) there will be people working more than 45 hours per week. There is a negative relationship between a positive capital gain and the first dimension, that make senses because usually the rich will make money from capital gains. In addition, 'education' and 'education number' is also distributed on the first dimension, with lower modalities 'education.num (9,10)', 'education.num [1,9]' and 'Some-college' on the right, higher education modalities like 'Bachelors' and 'Masters' in the left part of the plot.

The first dimension also includes some 'social' information. For example, it differentiates the age categories: the younger people in the right, while middle age people (36 to 49) in the left. This may be correlated with the social phenomenon that young people (most of them are students) do not work, and usually middle age ones are more professional, so we can expect that the salary of someone will reach a maximum when he or she is middle-aged. And some higher professional occupations (such as exec-managerial) are on the left (negative correlation).

```
$`Dim 2`
$`Dim 2`$quali
                          R2       p.value
workclass       0.090401993  0.000000e+00
education       0.607832691  0.000000e+00
education.num   0.587243075  0.000000e+00
marital.status  0.155223734  0.000000e+00
occupation      0.523088375  0.000000e+00
relationship    0.222198656  0.000000e+00
sex             0.214379590  0.000000e+00
native.country  0.079061881  0.000000e+00

$`Dim 2`$category
                                Estimate       p.value
 Female                       0.211294211  0.000000e+00
 Not-in-family                0.148760433  0.000000e+00
 Prof-specialty               0.573004767  0.000000e+00
 Never-married                0.103567786  0.000000e+00
education.num (13,16]         0.420745811  0.000000e+00
education.num (10,13]         0.116011194  0.000000e+00
 Masters                      0.685173062  0.000000e+00
 Bachelors                    0.392401815  0.000000e+00

 Male                        -0.211294211  0.000000e+00
 Husband                     -0.285555706  0.000000e+00
 Transport-moving            -0.378210074  0.000000e+00
 Machine-op-inspct           -0.308550850  0.000000e+00
 Craft-repair                -0.328704816  0.000000e+00
 Married-civ-spouse          -0.226987709  0.000000e+00
education.num [1,9]          -0.514174247  0.000000e+00
 HS-grad                     -0.241791645  0.000000e+00
```

Clearly, the second dimension is the dimension of education, with highest education on the top, and lower ones at the bottom. It also separates female and male well, while I believe it does not mean that in general, women are more educated than men. Besides, we notice that some occupations appear as relevant, that is because usually those occupations require corresponding high or low education levels. For instance, in most cases, college professors must have a doctorate degree. In contrast, for some physical works, people's education level is not that important.

Like PCA, MCA is also a dimension reduction tool. From the MCA analysis, we can decide how many dimensions we want to keep. The scree plot below shows the variances explained by dimensions.

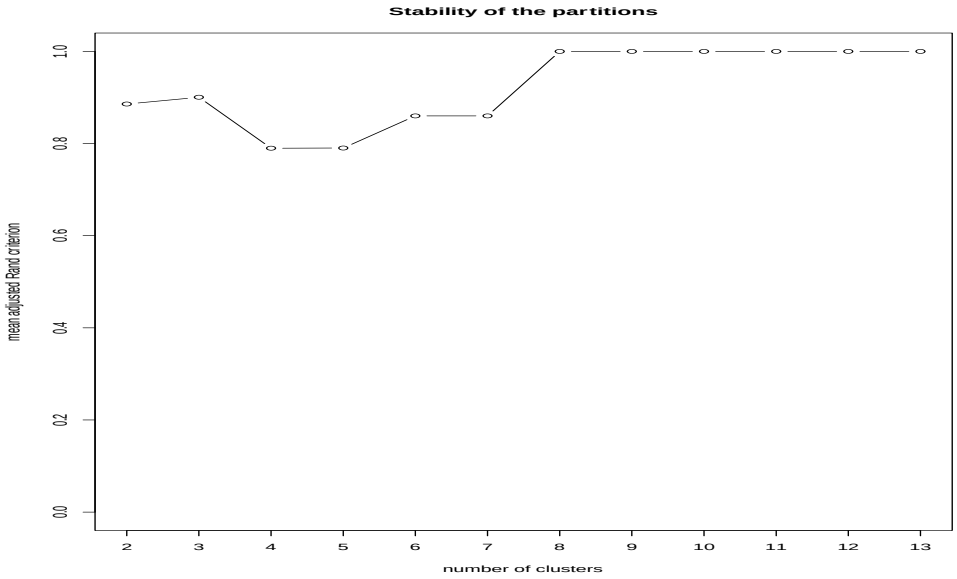


We have 105 dimensions in total, and there are two ways to determine how many dimensions we need to keep. The first rule is to keep the dimensions for which the eigenvalue is higher than the average of all eigenvalues. The second rule is to keep the dimensions for which the eigenvalue is higher than $\frac{1}{14}$ (i.e, one over the number of active variables). In our case, we get the same result by using these two rules. Therefore, we can keep 49 dimensions.

Variable Clustering

We also performed the variable clustering of categorical variables in order to arrange variables into homogenous clusters, i.e., groups of variables which are strongly related to each other and thus bring the same information. By doing this, we can identify redundant variables. The package we used is 'ClustOfVar', which can be used for the clustering of mixtures of quantitative and qualitative variables. The 'hclustvar' function produces a tree of variable groups.

The dendrogram above suggests that the 14 input variables can be combined into approximately 4– 8 groups of variables. The figure above also displays the link between the variables. For instance, the two variables ‘marital.status’ and ‘relationship’ are linked as well as the two variables ‘education’ and ‘education.num’, but we need to keep in mind that this dendrogram does not indicate the sign of these relationships. It is better to perform more careful study (conclusions we got by using the association rule are very helpful). We can also use the ‘stability’ function to have an idea of the stability of the partitions of the dendrogram represented



in the figure below.

Since our dataset is relatively large, so the stability function was used to run just 25 bootstrap samples of the trees. We can see that the plot of stability of variable cluster partitions suggests approximately a 6 to 8 cluster solution.

In conclusion, from the MCA analysis, we can reduce the number of dimensions of our data from 105 to 49. And from the variable clustering analysis, we can combine the 14 variables into 4-8 groups of variables.

Unsupervised learning & clustering

Clustering (k-prototypes)

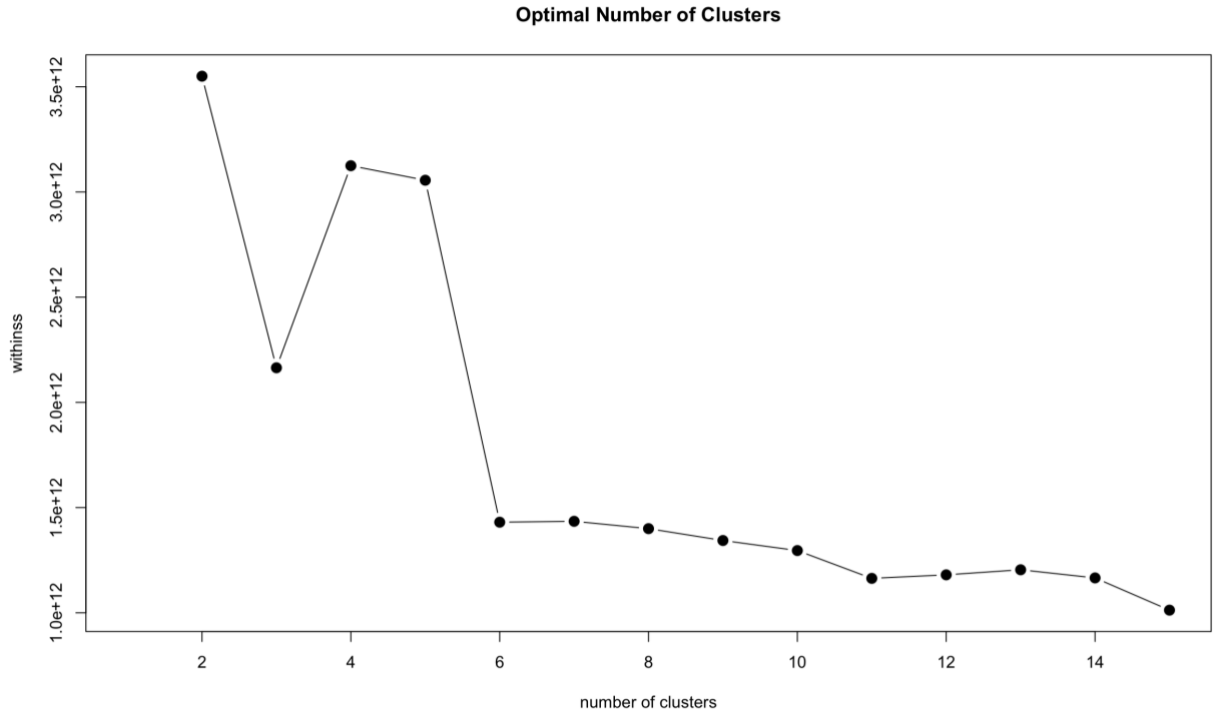
In unsupervised learning, we tend to use clustering to detect the characteristics of a dataset, such as K-Means, PCA or ICA. However, since this dataset is a mixture of categorical and quantitative variables, many clustering methods that are distance-calculation based would fail. Luckily, there is a remedy called the k-modes algorithm and it came out in a paper of 1998 by Zhexue Huang. It uses dissimilarity between classes to calculate the 'distance', and instead of using the mean, it uses mode to select the elements that would minimize the dissimilarities. But to cluster our dataset, we need to use the third kind of clustering, which is the combination of k-means and k-modes, called k-prototypes. This algorithm is similar to k-means, which iteratively recomputes cluster prototypes and reassigns clusters.

https://cran.r-project.org/web/packages/clustMixType/clustMixType.pdf

Clusters are assigned using:

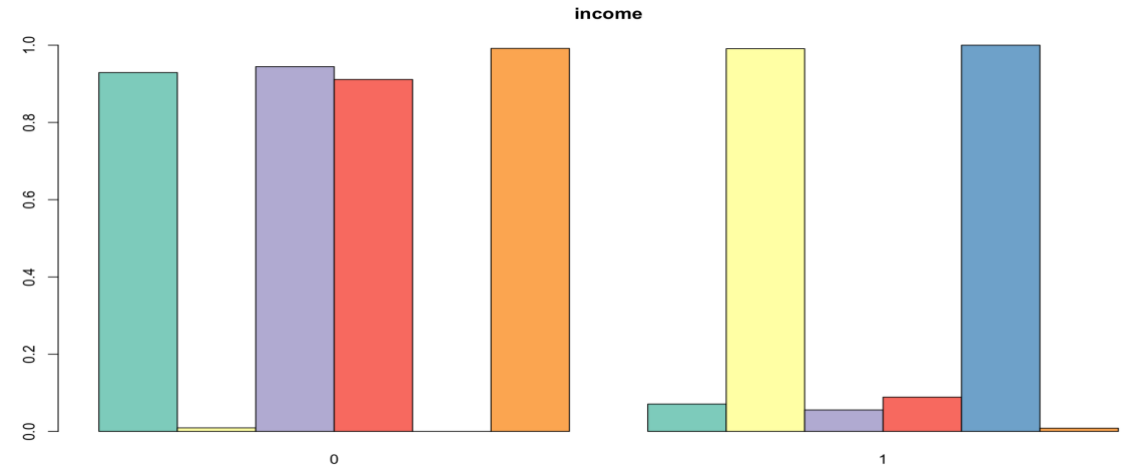
$$d(x,y) = d_{euclid}(x,y) + \lambda d_{simple\ matching}(x,y)$$

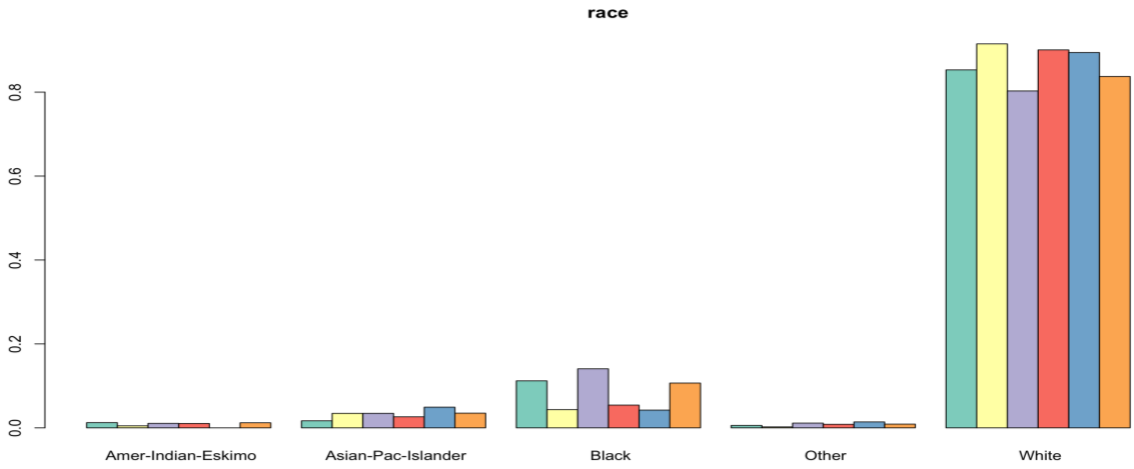
To demonstrate how it works, we used the 'kproto' function in the 'clustMixType' package. We first determined the optimal number of clusters by selecting the 'elbow' point on the following graph:

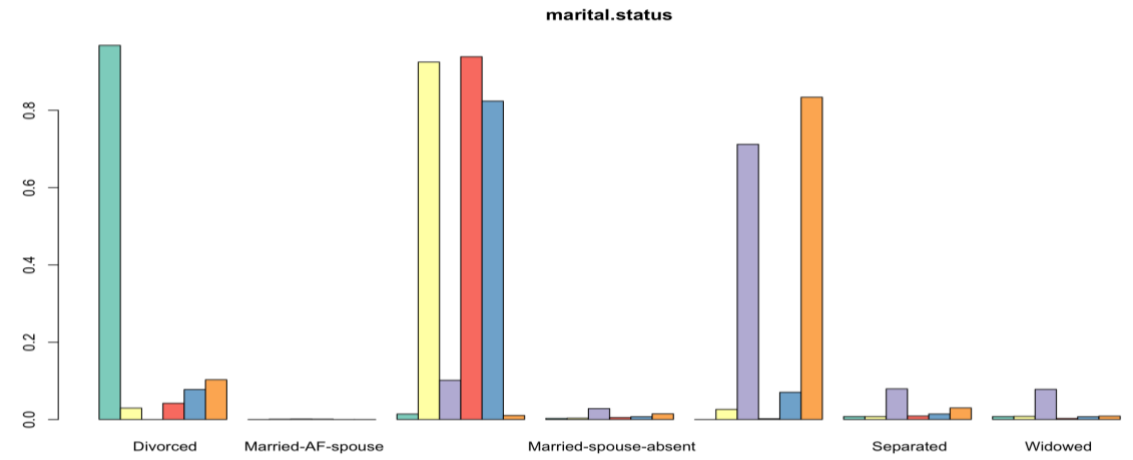


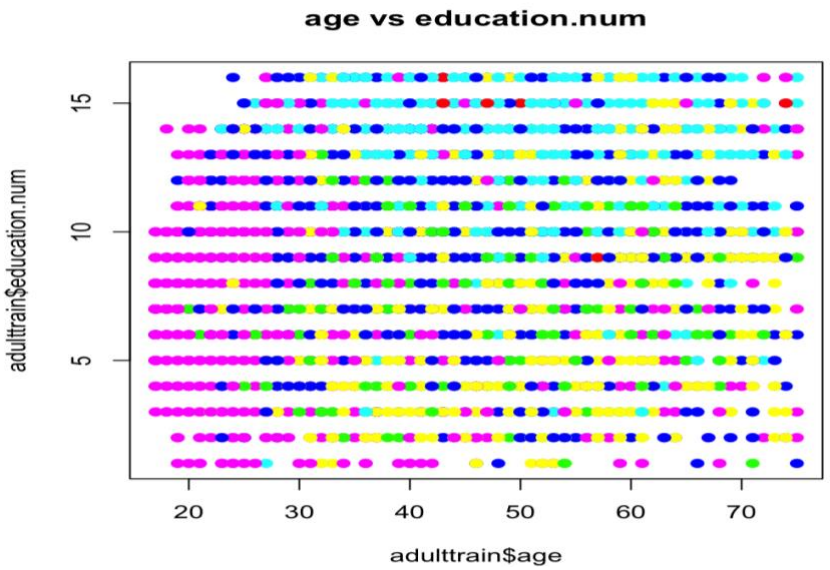
The logic is similar to scree plot, where we choose the point that reduces the most amount of variation. In this graph, we decided to choose the number of clusters as 6. To visualize our

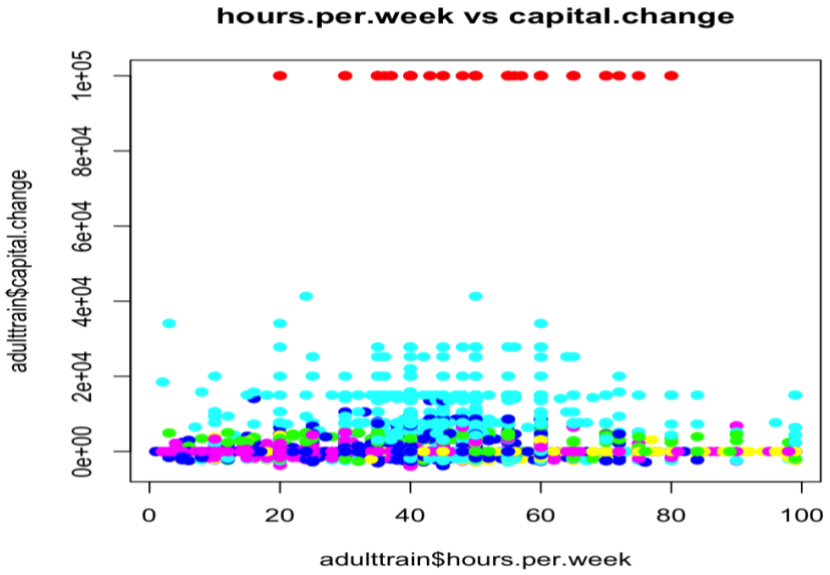
clusters, we can plot every covariate verses the response variable ‘income’. Continuous variables are visualized in box plots, and categorical variables are visualized in proportions.

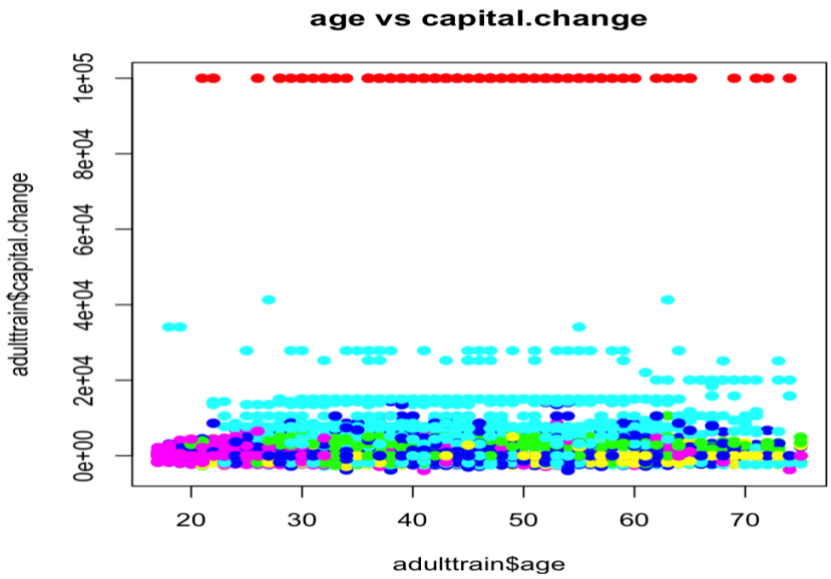







age vs education.num
adulttrain$education.num
5
10
15
20
30
40
50
60
70
adulttrain$age


hours.per.week vs capital.change
adulttrain$capital.change
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
0
20
40
60
80
100
adulttrain$hours.per.week


age vs capital.change
adulttrain$capital.change
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
20
30
40
50
60
70
adulttrain$age

We used the ‘rainbow’ function to specify the colors of clusters, the color for each cluster is as shown as the following:

Cluster	Color
1	Red
2	Yellow
3	Green
4	Cyan
5	Blue
6	Magenta

Some of these graphs give good clustering results. For bar charts, if we take a closer look at the bar plot for ‘income’, we could observe that each cluster did a good job distinguishing this binary variable, that is being said, the majority of each cluster belongs to either class ‘0’ or ‘1’. Similar clustering result can be also found in the bar plot for ‘marital.status’. For the class ‘divorced’, the green cluster (which is the cluster 3) not only occupies the majority of observations within the cluster but also among all clusters. This is the most desirable result that we are seeking for because that means our clusters could separate our dataset in a perfect manner. On the contrast, the clustering result produced for the variable ‘race’ was unsatisfying. Even though all clusters have the highest proportions of the class ‘white’, they cannot identify other races. The reason is understandable: the majority observations for the race are ‘white’, and the rest has comparatively small amounts of rows.

Numerical variables were visualized in pairs, and their results can be interpreted more easily. In the plots ‘age vs capital.change’ and ‘hours.per.week vs capital.change’, data was more clearly divided in the ‘capital.change’ dimension than either ‘age’ or ‘hours.per.week’ dimension. In the plot ‘age vs education.num’, there was a slight tendency of partitioning between clusters 2,4,6.

However, since we only have six clusters, we cannot see this most favorable situation quite often. But on the other hand, we can approximately separate our dataset with a simpler clustering setting. Please refer to Appendix for all graphical results.

Association Rule

After the data visualization step, our dataset reduce to 45222 obervations with 10 variables (though we have 11 variables, education and ‘education.num’ exhibits same information so we will only choose either of them).

```
dim(Adultdata)
```

```
[1] 43630    11
```

First, we remove the attribute ‘education.num’(numerical) and keep the attribute education (categorical). Then we transfered the two numerical attributes, ‘age’ and ‘hours.per.week’ to categorical attributes. We divide the ‘age’ into suitable categories using ‘Young’, ‘Middle-aged’, ‘Senior’, ‘Old’ four groups; divide ‘hours-per-week’ into suitable categories using ‘Part-time’,

'Full-time', 'Over-time', 'Workaholic' four groups. Then we read the dataset 'Adultdata' as transactions and save it as 'Adult'.

Nest step, we do association rule mining using the Apriori algorithm in arules.

```
> rules <- apriori(Adult, parameter = list(support = 0.01, confidence = 0.7))
```

By using summary(rules), we can see the result is a set of 27077 rules

Since we are interested in whether an adult has an income higher than $50K per year or not, here we added the attribute income with levels small and large, representing an income of <=$50K and >$50K respectively.

Sort by the lift measure exceeds 1.2, we obtain the top ten rules by the R output as below:

- **Top ten rules that earing <=$50K a year**

	lhs	rhs	support	confidence	lift	count
[1]	{maritalstatus=Never-married, occupation=Handlers-cleaners, relationship=Own-child}	=> {income=<=50K}	0.01242264	1	1.331766	542
[2]	{occupation=Other-service, relationship=Own-child, hours.per.week=Part-time}	=> {income=<=50K}	0.01400413	1	1.331766	611
[3]	{age=Young, occupation=Other-service, hours.per.week=Part-time}	=> {income=<=50K}	0.01620445	1	1.331766	707
[4]	{occupation=Sales, relationship=Own-child, hours.per.week=Part-time}	=> {income=<=50K}	0.01038276	1	1.331766	453
[5]	{relationship=Own-child, sex=Male, hours.per.week=Part-time}	=> {income=<=50K}	0.02053633	1	1.331766	896
[6]	{maritalstatus=Never-married, occupation=Handlers-cleaners, relationship=Own-child, sex=Male}	=> {income=<=50K}	0.01100160	1	1.331766	480
[7]	{workclass=Private, maritalstatus=Never-married, occupation=Handlers-cleaners, relationship=Own-child}	=> {income=<=50K}	0.01184964	1	1.331766	517

[8] {maritalstatus=Never-married,
occupation=Handlers-cleaners,
relationship=Own-child,
race=White} => {income=<=50K} 0.01061196 1 1.331766 463
[9] {age=Young,
workclass=Private,
maritalstatus=Never-married,
occupation=Handlers-cleaners} => {income=<=50K} 0.01375201 1 1.331766 600
[10] {age=Young,
occupation=Other-service,
relationship=Own-child,
hours.per.week=Part-time} => {income=<=50K} 0.01276645 1 1.331766 557

- **Top ten rules that earing >$50K a year**

lhs rhs support confidence lift count
[1] {education=Prof-school,
relationship=Husband} => {income=>50K} 0.01006188 0.8624754 3.462122 439
[2] {education=Prof-school,
maritalstatus=Married-civ-spouse,
relationship=Husband} => {income=>50K} 0.01006188 0.8624754 3.462122 439
[3] {education=Prof-school,
relationship=Husband,
sex=Male} => {income=>50K} 0.01006188 0.8624754 3.462122 439
[4] {education=Prof-school,
maritalstatus=Married-civ-spouse,
sex=Male} => {income=>50K} 0.01006188 0.8624754 3.462122 439
[5] {education=Prof-school,
maritalstatus=Married-civ-spouse,
relationship=Husband,
sex=Male} => {income=>50K} 0.01006188 0.8624754 3.462122 439
[6] {education=Prof-school,
maritalstatus=Married-civ-spouse} => {income=>50K} 0.01074948 0.8621324 3.460745 469
[7] {education=Bachelors,
occupation=Exec-managerial,
relationship=Husband,
race=White,
hours.per.week=Over-time} => {income=>50K} 0.01086408 0.8419183 3.379602 474
[8] {education=Bachelors,

maritalstatus=Married-civ-spouse,
occupation=Exec-managerial,
relationship=Husband,
race=White,
hours.per.week=Over-time} => {income=>50K} 0.01086408 0.8419183 3.379602 474

[9] {education=Bachelors,
occupation=Exec-managerial,
relationship=Husband,
race=White,
sex=Male,
hours.per.week=Over-time} => {income=>50K} 0.01086408 0.8419183 3.379602 474

[10] {education=Bachelors,
maritalstatus=Married-civ-spouse,
occupation=Exec-managerial,
relationship=Husband,
race=White,
sex=Male,
hours.per.week=Over-time} => {income=>50K} 0.01086408 0.8419183 3.379602 474

Interpretation of the top ten rules that earing <=$50K a year:

Rule1:

Support—the probability that an never-married adult, who work as handlers-cleaner and have child among all the dataset is 1.242264%.

Confidence—for an never-married adult, who work as handlers-cleaner and have child, the probability that this adult earning less than $50K a year is 100%.

Lift— the increase in probability that an adult earning less than $50K a year, given by he/she is an never-married adult, who work as handlers-cleaner and have child is 133.18%.

Rule2:

Support—the probability that an adult who has a part-time other-service job and have child among all the dataset is 1.400413%.

Confidence—for an adult who has a part-time other-service job and have child, the probability that this adult earning less than $50K a year is 100%.

Lift— the increase in probability that an adult earning less than $50K a year, given by he/she is an adult who has a part-time other-service job and have child is 133.18%.

Rule3:

Support—the probability that a young adult who has a part-time other-service job among all the dataset is 1.620445%.

Confidence—for a young adult who has a part-time other-service job, the probability that this adult earning less than $50K a year is 100%.

Lift— the increase in probability that an adult earning less than $50K a year, given by he/she is a young adult who has a part-time other-service job is 133.18%.

Rule4:

Support—the probability that an adult who has a part-time sales job and have child among all the dataset is 1.038276%.

Confidence—for an adult who has a part-time sales job and have child, the probability that this adult earning less than $50K a year is 100%.

Lift— the increase in probability that an adult earning less than $50K a year, given by he/she is an adult who has a part-time sales job and have child is 133.18%.

Rule5:

Support—the probability that a male adult who has a part-time job and have child among all the dataset is 2.053633%.

Confidence—for a male adult who has a part-time job and have child, the probability that this adult earning less than $50K a year is 100%.

Lift— the increase in probability that an adult earning less than $50K a year, given by he is a adult who has a part-time job and have child is 133.18%.

Rule6:

Support—the probability that a never-married male adult, who work as handlers-cleaner and have child among all the dataset is 1.100160%.

Confidence—for a never-married male adult, who work as handlers-cleaner and have child, the probability that this adult earning less than $50K a year is 100%.

Lift— the increase in probability that an adult earning less than $50K a year, given by he is an never-married adult, who work as handlers-cleaner and have child is 133.18%.

Rule7:

Support—the probability that a never-married adult, who work as private handlers-cleaner and have child among all the dataset is 1.184964%.

Confidence—for a never-married adult, who work as private handlers-cleaner and have child, the probability that this adult earning less than $50K a year is 100%.

Lift— the increase in probability that an adult earning less than $50K a year, given by he/she is a never-married adult, who work as private handlers-cleaner and have child is 133.18%.

Rule8:

Support—the probability that a never-married white adult, who work as handlers-cleaner and have child among all the dataset is 1.061196%.

Confidence—for a never-married white adult, who work as handlers-cleaner and have child, the probability that this adult earning less than $50K a year is 100%.

Lift— the increase in probability that an adult earning less than $50K a year, given by he/she is a never-married white adult, who work as handlers-cleaner and have child is 133.18%.

Rule9:

Support—the probability that a never-married young adult, who work as private handlers-cleaner among all the dataset is 1.375201%.

Confidence—for a never-married young adult, who work as private handlers-cleaner, the probability that this adult earning less than $50K a year is 100%.

Lift— the increase in probability that an adult earning less than $50K a year, given by he/she is a never-married young adult, who work as private handlers-cleaner is 133.18%.

Rule10:

Support—the probability that a young adult, who work as part-time other-service and have child among all the dataset is 1.276645%.

Confidence—for a young adult, who work as part-time other-service and have child , the probability that this adult earning less than $50K a year is 100%.

Lift— the increase in probability that an adult earning less than $50K a year, given by he/she is a young adult, who work as part-time other-service and have child is 133.18%.

Interpretation of the top ten rules that earing >$50K a year:

Rule1 - Rule5 (exactly the same):

Support—the probability that a married male adult who graduated from Prof-school among all the dataset is 1.006188%.

Confidence—for a married male adult who graduated from Prof-school, the probability that this adult earning more than $50K a year is 86.25%.

Lift— the increase in probability that an adult earning more than $50K a year, given by he is a married adult who graduated from Prof-school is 346.21%.

Rule6:

Support—the probability that a married adult who graduated from Prof-school among all the dataset is 1.074948%.

Confidence—for a married adult who graduated from Prof-school, the probability that this adult earning more than $50K a year is 86.21%.

Lift— the increase in probability that an adult earning more than $50K a year, given by he/she is a married adult who graduated from Prof-school is 346.07%.

Rule7 - Rule10 (exactly the same):

Support—the probability that a married male white adult who has Bachelor degree, works over-time as exec-managerial among all the dataset is 1.086408%.

Confidence—for a married male white adult who has Bachelor degree, works over-time as exec-managerial, the probability that this adult earning more than $50K a year is 84.29%.

Lift— the increase in probability that an adult earning more than $50K a year, given by he is a married white adult who has Bachelor degree, works over-time as exec-managerial is 337.96%.

Results:

From the rules we see that a never-married adult who has an part-time low skill job (hand-cleaner or other-service) tend to have a small income (less than $50K) a year; while an married male adult who has high education level, also own a high skill job and works over-time tend to have a large income (more than $50K) a year.

Furthermore, we can conclude that the attributes 'hours.per.week', 'marital status/relationship', ' occupation', 'education/education.num', 'sex', all these attributes would have impact on the income level.

Conclusion

Based on the analysis above, on the one hand, we realized that both 'fnlwgt' and 'native.country' do not influence people's income. On the other hand, we can see that several attributes will influence an individual's income, such as 'age', 'education', 'education.num', 'hours.per.week', 'occupation', 'marital.status' and 'relationship'.

- Generally speaking, middle-aged people make more money than other age groups.
- Higher education may result in a higher possibility to earn more. For example, Ph.D and Master's degrees lead to increased income.
- People who spend more hours at work per week are more likely getting a higher income.
- In most situations, individuals who are employed in high-skilled jobs tend to earn more than 50K dollars a year.
- The marital status of an individual will influence one's income, compared to unmarried people, married people make more money in general.

In addition, we also realized that the correlation between 'education' and 'education.num' is high. Since these two variables provide almost the same information but in different pattern: 'education.num' is the numeric representation of "education". Hence, we can remove one of them from our prediction models.

Supervised Learning

Logistic Regression

A logistic regression using income as the response variable, and all other 9 variables as predictors to build a model that predicts the income level of an adult to be greater than $50K or less than $50K using Census data. the output is as below:

```
Call:
glm(formula = income ~ ., family = binomial(link = "logit"),
    data = adulttrain)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.4395  -0.5395  -0.1987   0.0000   3.7754

Coefficients:
                                  Estimate Std. Error z value Pr(>|z|)
(Intercept)                     -8.992e+00  4.081e-01 -22.031  < 2e-16 ***
age                              3.004e-02  1.732e-03  17.341  < 2e-16 ***
workclass Local-gov             -6.963e-01  1.125e-01  -6.192 5.94e-10 ***
workclass Private               -5.083e-01  9.358e-02  -5.432 5.58e-08 ***
workclass Self-emp-inc          -3.509e-01  1.235e-01  -2.840 0.004505 **
workclass Self-emp-not-inc      -9.958e-01  1.094e-01  -9.103  < 2e-16 ***
workclass State-gov             -8.507e-01  1.248e-01  -6.817 9.29e-12 ***
workclass Without-pay           -1.244e+01  1.179e+02  -0.105 0.915994
education.num                    2.911e-01  9.635e-03  30.210  < 2e-16 ***
marital.status Married-AF-spouse 2.515e+00  6.093e-01   4.127 3.67e-05 ***
marital.status Married-civ-spouse 1.939e+00 2.808e-01   6.906 4.98e-12 ***
marital.status Married-spouse-absent -1.251e-01 2.373e-01 -0.527 0.597894
marital.status Never-married    -4.433e-01  8.766e-02  -5.058 4.25e-07 ***

occupation Farming-fishing      -9.751e-01  1.393e-01  -7.000 2.57e-12 ***
occupation Handlers-cleaners    -6.705e-01  1.437e-01  -4.667 3.06e-06 ***
occupation Machine-op-inspct    -2.610e-01  1.017e-01  -2.567 0.010260 *
occupation Other-service        -8.664e-01  1.184e-01  -7.320 2.48e-13 ***
occupation Priv-house-serv      -3.435e+00  1.280e+00  -2.683 0.007299 **
occupation Prof-specialty        5.887e-01  7.999e-02   7.360 1.84e-13 ***
occupation Protective-serv       5.946e-01  1.262e-01   4.710 2.48e-06 ***
occupation Sales                 2.937e-01  8.251e-02   3.560 0.000372 ***
occupation Tech-support          6.345e-01  1.111e-01   5.711 1.12e-08 ***
occupation Transport-moving     -9.667e-02  9.965e-02  -0.970 0.332017
relationship Not-in-family       3.040e-01  2.781e-01   1.093 0.274274
relationship Other-relative     -5.098e-01  2.510e-01  -2.031 0.042281 *
relationship Own-child          -8.645e-01  2.798e-01  -3.090 0.002004 **
relationship Unmarried           1.433e-01  2.927e-01   0.489 0.624497
relationship Wife                1.309e+00  1.045e-01  12.528  < 2e-16 ***
race Asian-Pac-Islander          5.221e-01  2.451e-01   2.130 0.033166 *
race Black                       4.869e-01  2.344e-01   2.078 0.037746 *
race Other                      -7.807e-02  3.616e-01  -0.216 0.829067
race White                       6.218e-01  2.229e-01   2.789 0.005283 **
sex Male                         8.596e-01  7.953e-02  10.807  < 2e-16 ***
hours.per.week                   2.925e-02  1.698e-03  17.232  < 2e-16 ***
capital.change                   2.413e-04  8.890e-06  27.138  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 32730  on 29095  degrees of freedom
Residual deviance: 19551  on 29056  degrees of freedom
AIC: 19631
```

From the table above, we can find that ‘age’, ‘education.num’, ‘hours.per.week’, ‘occupation’ ‘’relationship’’ are some top attributes that relevant to the income level. This is quite similar as the result we had through association rule.

Most of the explanatory variables are essential in explaining incomes. Only some of them are less important due to the smaller amount of observations, such as work class ‘without-pay’, some marital status, and some occupations such as ‘armed-forces’ or ‘craft-repair’. The example of interpreting the coefficient are as the followings.

The coefficient for ‘age’ is 3.043e-02, which means holding other variables fixed, we will see a 3% ($e^{0.03043}$) increase in the odds of a person earning an income more than 50K for a one-unit increase in age. Also, we will see a 33% ($e^{0.2907}$) increase in the odds of earning an income more than 50K for a one-unit increase in education. The odds of earning an income more than 50K for

people who work at the local government over the odds of that for people working at a place other than the local government is $e^{-0.681}$, which is around 50%.

Since we have many explanatory variables in our data, we want to see if the forward and backward selection can be used in terms of AIC to articulate a simpler model. By looking at the output below, it gives the same model as before. None of the explanatory variables should be removed from the model.

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 10120  1470
         1   864  2128

               Accuracy : 0.8399
                 95% CI : (0.8339, 0.8459)
    No Information Rate : 0.7533
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5436
 Mcnemar's Test P-Value : < 2.2e-16
```

Next, we used Cross Validation (CV) by apply the created prediction model to the test data to validate the performance. The probabilities that are closer to '1' means they have higher probabilities to earn more than 50K and closer to '0' means there are fewer probabilities to earn more than 50K. Therefore, we want to use a threshold of 0.5 to determine if an individual can earn more than 50K and get the prediction values. By looking at the confusion matrix, the accuracy is around 84%.

Regression Tree

```
# 1. Trees
tree <- rpart(income ~ ., data = adulttrain, method = 'class')
pred.tree <- predict(tree, newdata =adulttest, type = 'class')
confusionMatrix(pred.tree ,income)
```

For non-parametric modeling methods, we first want to build a regression tree model by using the 'CART' package. Since the response variable is binary, CART will return a classification tree based on the training dataset.

We construct the classification tree model using rpart() function and fit the model to the test

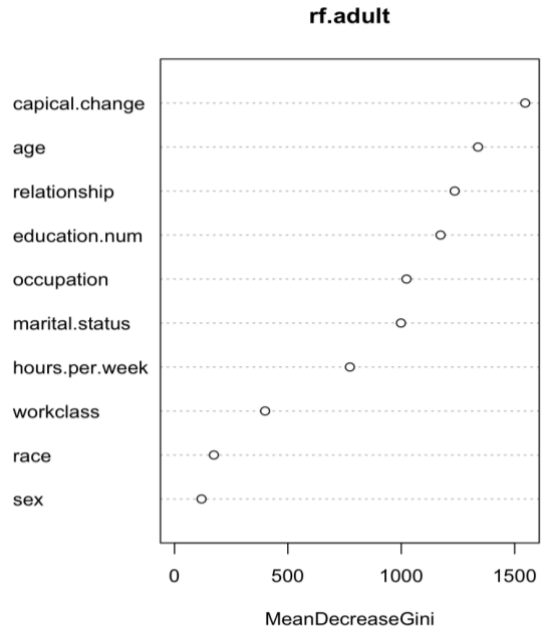
```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 10411  1788
         1   573  1810

               Accuracy : 0.8381
```

dataset. By looking at the confusion matrix function, the prediction accuracy is 83.8%.

Random Forest

We applied Random Forest by using the 'randomForest' function. Random Forest only chooses one from around $m = \sqrt{p}$ variables at each node, in this case, we will choose one out of three variables. As we can see the output, the OOB error rate for the training dataset is 13.92%. To validate our model, we predicted the test dataset by using our Random Forest model, the prediction accuracy is 0.855. The confusion matrix is as the following

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 10203  1333
         1   781  2265

               Accuracy : 0.855
                 95% CI : (0.8492, 0.8607)
    No Information Rate : 0.7533
    P-Value [Acc > NIR] : < 2.2e-16
```



We also examined the variance importance plot. As we can see above, the most important variables are 'capital.change', 'age', 'relationship' and 'education.num'.

Neural Network

As a similar manner as logistic regression, we still split the original data set into a training set and a test set. Models are trained on the training set and validated on the testing set.

Neural network (NN) added the interactions that were missing from logistic and hence has the best performance. For the 9 input variables, many of them are categorical and would result several dummy variables. After trying different number of hidden layers, we found that number of hidden layers as 40 gave us the best training accuracy considering model complexity as well. Maximum number of iterations are set to 2,000.

```
> nn1 <- nnet(income~age + educationnum + hoursperweek + workclass + maritalstatus
+         + occupation + relationship + race + sex, data = training, size = 40,
+         maxit = 500, MaxNWts=2000)
# weights:  1721
initial  value 35845.212136
iter  10 value 15730.799694
iter  20 value 15050.154702
iter  30 value 13229.100050
iter  40 value 11755.304784
iter  50 value 11057.340122
... ...
iter 480 value 9664.577813
iter 490 value 9642.475993
iter 500 value 9626.808594
final  value 9626.808594
stopped after 500 iterations
```

Next, we used Cross Validation (CV) by apply the created prediction model to the test data to validate the true performance.

The confusion matrix shows below evaluate how well the model predicts income:

```
> confusionMatrix(nn1.pred,income)
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 10184  1354
         1   800  2244

               Accuracy : 0.8523
                 95% CI : (0.8464, 0.858)
    No Information Rate : 0.7533
    P-Value [Acc > NIR] : < 2.2e-16
```

The prediction result has a misclassification rate of 14.77% and an accuracy of 85.23%. The neural network (NN) model has good accuracy and make a little improve compare with logistic regression.

Naïve Bayes

As a generative model and big data classifier, the Naïve Bayes algorithm can be used with continuous features but is more suited to categorical variables. While, for numeric features, it also makes a strong assumption which is that the numerical variable is normally distributed. Therefore, it is possible to mix different variable types (categorical and numerical features in our case) in Naïve Bayes.

Below is the Naïve Bayes' Theorem:

$$P(A \mid B) = P(A) * P(B \mid A) / P(B)$$

There is a package in R called 'e1071' which provides the Naïve Bayes training function. The 'naiveBayes' function supports a numeric matrix or a data frame of categorical and/or numeric variables. We used the Naïve Bayes to conduct the prediction, the confusion matrix we got is as follows:

```
Confusion Matrix and Statistics

          Reference
Prediction  <=50K  >50K
     <=50K  10368   616
     >50K    1912  1686

               Accuracy : 0.8266
                 95% CI : (0.8204, 0.8327)
```

The accuracy of classification is 82.66%. The Naïve Bayes classifier performed not as good as the logistic regression model. This may be due to the fact that the algorithm makes a very strong assumption about the data having variables independent of each other, and if this assumption of independence holds, Naïve Bayes usually performs extremely well and often better than other. However, we already known that in our case, this conditional independence assumption does not hold.

SVM

SVM (Support Vector Machine) is another useful technique for data classification. In general, the SVM classifier works well with nonlinear data. We noticed nonlinearity in some of our variables, hence we also used SVM to handle the classification task.

Data preprocessing

SVM requires that each data instance is represented as a vector of real numbers. Hence, if there are categorical variables, we first have to convert them into numerical data. There are no ordinal variables in our dataset (all categorical variables are nominal), so we can transform these factors to dummy variables. A dummy variable is a binary variable (coded as 1 or 0) to reflect the presence or absence of a particular categorical code in a given variable. And adding dummy variables to the analysis will help to create a better fit to the model.

Besides, scaling before applying SVM is very important and the main advantage of scaling (normalize or standardize data) is to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges. Therefore, scaling of the data usually drastically improves the results.

The 'svm()' function in R package 'e1071' constructs dummy variables then scales the data by default. Therefore, in this case, we do not need transform categorical attributes in advance.

Model Selection

There are several common kernel functions in SVM and choosing the proper kernel function can significantly improve the SVM performance. In general, the RBF kernel (i.e, Radial basis function kernel) is a reasonable first choice. Unlike the linear kernel, this kernel nonlinearly maps samples into a higher dimensional space, so it can handle the case when the relation between class labels and variables is nonlinear.

Therefore, the kernel used in training and predicting is the radial basis kernel, and the cost and gamma parameters we defined are 1 and 0.1, respectively.

```
Confusion Matrix and Statistics

          Reference
Prediction  <=50K  >50K
     <=50K  10273   711
     >50K    1483  2115

               Accuracy : 0.8495
                 95% CI : (0.8436, 0.8553)
```

The confusion matrix above displays that the classification accuracy of the SVM model increases by almost 3 percent to 84.95%, when compared to the Naïve Bayes model.

KNN

K-nearest neighbors algorithm can be used for both classification and regression tasks. It is a non-parametric machine learning algorithm, which means that it does not make any assumptions on the underlying data distribution. For classification, the algorithm stores all available cases and classifies new objects by a majority vote of its k neighbors. By doing this, this algorithm can separate unlabeled data points into well-defined groups.

Be similar with the SVM algorithm, the KNN classifier works naturally with numerical attributes, so we need to convert categorical values into numbers. And again, all categorical attributes in our dataset are nominal so we need dummy variables. However, in contrast to the 'svm()' function we used, the 'knn()' function in the 'class' package requires all the independent/predictor variables to be numeric and the dependent or target variable to be categorical. In other words, the function itself could not handle categorical variables or mix type data without encoding them, so we have to create dummy variables by hand.

There are two classic ways to generate dummy variables, one-hot-encoding and dummy coding. Assuming one categorical variable has n categories, one-hot-encoding converts it into n variables, while dummy coding converts it into $n-1$ dummy variables. For instance, the dummy coding approach creates four dummy variables to represent the five-category attribute 'race'.

We decided to use the dummy coding approach since we do not want to give the regression redundant information and we want to avoid the multicollinearity issues in models. Or one can leave one category out when using one-hot encoding approach in order to prevent multicollinearity. After doing this, we obtained a new dataset with 40 columns. Then before applying the 'knn' function, we standardized our data manually.

We tried different k values, different k values gave us slightly different result:

	Prediction Accuracy
k=5	81.81%
k=10	82.71%
k=15	82.64%
k=sqrt(n)	82.66%

We can see that K-nearest neighbors with k = 10 yielded the highest accuracy of 82.71%.

Conclusion

Performance Evaluation: ROC Curves and Area Under the Curve

For binary-response datasets, Receiver Operating Characteristic (ROC) Curve is an essential way of performance evaluation. Since both the true response values and predicted values are either '0' (negative) or '1' (positive). We will have the following four possible combinations:

		True Outcome	
		Positive	Negative
Predicted Outcome	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

The True Positive Rate (TPR) is the proportion of true positive ones, which is calculated by TP/(TP+FN), this ratio is also known as the sensitivity of the test. It describes the ability that a certain model could diagnose a positive case correctly. The True Negative Rate (TNR) is the proportion of true negative ones, which is calculated by TN/(TN+FP). This ratio is known as the specificity of the test, and it describes the ability that a certain model could diagnose a negative case correctly. Its complement, FP/(TN+FP), is called the False Positive Rate (FPR). Ideally, we are seeking a model that could maximize both sensitivity and specificity, that means sensitivity = specificity = 1. Or equivalently,

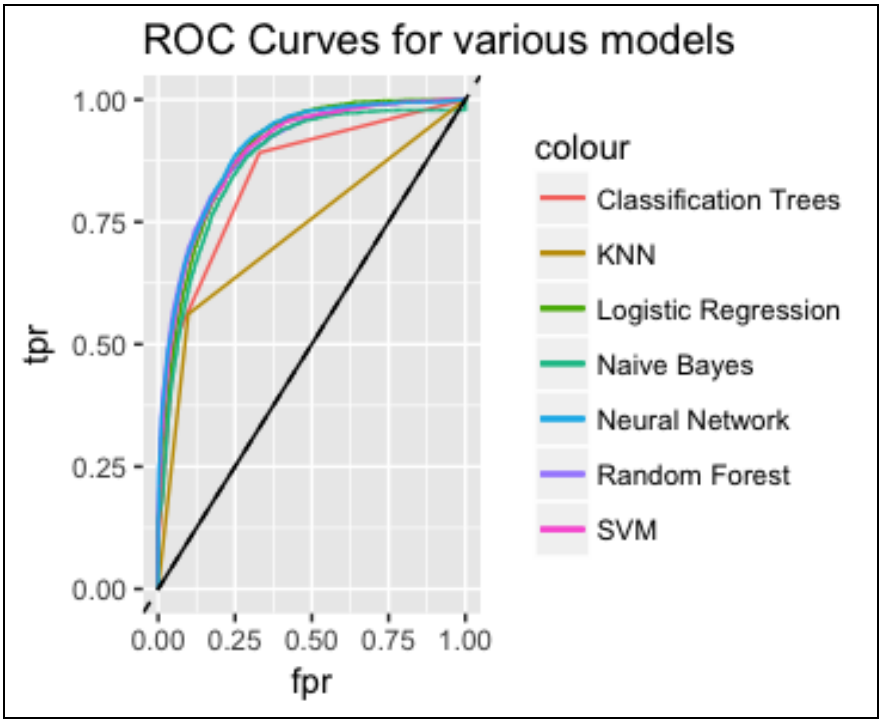
$$\begin{cases} TPR = 1 \\ FPR = 0 \end{cases}$$

Models tend to predict in probabilities rather than classifying automatically, especially for parametric models such as logistic regression model. Thresholds are required to classify a test observation into other class '0' or '1'. Lower thresholds will lead to more positive cases (higher True Positive cases and False Positive cases) and higher thresholds will lead to more negative cases (lower True Positive cases and False Positive cases). ROC Curve is a visual representation that plots TPR against FPR and connects all possible thresholds' TPRs and FPRs in a smooth curve. Since we want the TPR as close to 1 as possible, and FPR as close to 0 as possible, the optimal point will be the one that is closest to the upper left corner.

In our case, we so far have done five supervised learning methods: Naïve Bayes, Classification Tree, Random Forest, Logistic Regression, and Neural Network.

```
treepreds <- predict(tree,newdata=adulttest,type="prob")
posteriorYes.tree <- treepreds[,2]
ROCres.tree <- roc(posteriorYes.tree,trueYes)
tidyROCres.tree <- tidy(ROCres.tree)
```

We finally got these following curves:



As we can see, these five methods produce similar results: all of them are close to the upper-left corner. When a classifier cannot distinguish between two groups, the probability for each class will be both 0.5 (plain guessing), thus both TPR = FPR, this is the black diagonal line shown on the plot.

To distinguish more carefully between all these methods, we will be using the concept of Area Under the Curve (AUC). The idea is the calculate the area under the 'curve', including the area that is below the diagonal line. After calculating, the results are the following:

models	aucs
Neural Network	0.9029529
Random Forest	0.8975731
Logistic Regression	0.8952478
SVM	0.8951504
Naïve Bayes	0.8707187
Classification Trees	0.8429788
KNN	0.7318701

Models are ranked in descending order, and as we can see, the best model is Random Forest, which yields almost 90% of AUC. That means, by using the Random Forest model, a randomly selected case from the test dataset with the predicted value equals to '1' has a score larger than that for a randomly chosen case from the group with the predicted value equals to '0' in around 90% of the time.

R code

```
install.packages(c('caret','dplyr','rpart','MASS','AUC','broom','ggplot2','randomForest','clustMixType'))
library(caret)
library(dplyr)
library(rpart)
library(MASS)
library(AUC)
library(broom)
library(ggplot2) ## package for producing graphical plots
library(randomForest)
library(clustMixType)
library(arules)
library(klaR)
library(e1071)
library(dummies)
library(gridExtra) ## package for drawing multiple graphs on a grid


################################################
################################################
############## import data  ###################
################################################
################################################

datatrainurl <- 'http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data'
datatesturl <-'http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.test'
datanameurl <- 'http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.names'
adulttrain <- read.table(datatrainurl,sep = ',',stringsAsFactors = FALSE)
adulttest <- readLines(datatesturl)[-1]
adulttest <- read.table(textConnection(adulttest),sep = ',',stringsAsFactors = FALSE)

adultnames <- readLines(datanameurl)[97:110]
adultnames <- as.character(lapply(strsplit(adultnames,':'), function(x) x[1]))
adultnames <- c(adultnames,'income')
colnames(adulttrain) <- adultnames
colnames(adulttest) <- adultnames

str(adulttrain)

# We first remove missing value (ones with ' ?')
no.question.mark <- apply(adulttrain, 1, function(r) !any(r %in% ' ?'))
```

```
adulttrain <- adulttrain[no.question.mark,]

no.question.mark <- apply(adulttest, 1, function(r) !any(r %in% ' ?'))
adulttest <- adulttest[no.question.mark,]

adulttrain <- as.data.frame(unclass(adulttrain),stringsAsFactors = T)
adulttest <- as.data.frame(unclass(adulttest),stringsAsFactors = T)

################################################################################
##################
################################################################################
##################
#############code for: Visulization, Association Rule, Dimension Reduction
####################
################################################################################
##################
################################################################################
##################

Adultdata<-rbind(adulttrain,adulttest)
Adultdata$income <- gsub(".","",as.character(Adultdata$income),fixed=TRUE)
# remove outliers #
no.outlier<-function(data,x)
{
 for (i in x)
 {
  a<-boxplot.stats(data[,i])$out
  data<-data[!data[,i]%in%a,]
  print(length(a))
 }
 return(data)
}
str(adulttrain)
Adultdata<- no.outlier(Adultdata,c(1,3))
adulttrain <- no.outlier(adulttrain,c(1,3))
adulttest <- no.outlier(adulttest,c(1,3))

#------------------------------------------------------------------
### Visualization
## Detecting skewed variables
skewedVars<- NA
library(moments) # for skewness()
for(i in names(Adultdata)){
 if(is.numeric(Adultdata[,i])){
```

```
    if(i != "income"){
      # Enters this block if variable is non-categorical
      skewVal <- skewness(Adultdata[,i])
      print(paste(i, skewVal, sep = ": "))
      if(abs(skewVal) > 0.5){
        skewedVars <- c(skewedVars, i)
      }
    }
  }
}
N.obs<-dim(Adultdata)[1]
N.var<-dim(Adultdata)[2]
## Explore Numerical Variable
## Correlation between numerical variables
numeric.var <- sapply(Adultdata, is.numeric)
## Calculate the correlation matrix
corr <- cor(Adultdata[,numeric.var])
corr

p1 <- ggplot(Adultdata, aes(x=age)) + ggtitle("Histogram of Age") +
  geom_histogram(aes(y = 100*(..count..)/sum(..count..)), binwidth=5, colour="black",
fill="salmon") + ylab("Percentage")
grid.arrange(p1)
p2 <- ggplot(Adultdata, aes(x=log10(fnlwgt))) + ggtitle("Histogram of Log(fnlwgt)") +
  geom_histogram(aes(y = 100*(..count..)/sum(..count..)), colour="black", fill="salmon") +
ylab("Percentage")
grid.arrange(p2)
p3 <- ggplot(Adultdata, aes(x=education.num)) + ggtitle("Histogram of Educationnum") +
  geom_histogram(aes(y = 100*(..count..)/sum(..count..)), binwidth=1, colour="black",
fill="salmon") + ylab("Percentage")
grid.arrange(p3)
p4 <- ggplot(Adultdata, aes(x=hours.per.week)) + ggtitle("Histogram of Hours per Week") +
  geom_histogram(aes(y = 100*(..count..)/sum(..count..)), colour="black", fill="salmon") +
ylab("Percentage")
grid.arrange(p4)
p5 <- ggplot(Adultdata, aes(x=log10(capital.gain+1))) + ggtitle("Histogram Log(Capital Gain)") +
  geom_histogram(aes(y = 100*(..count..)/sum(..count..)), colour="black", fill="salmon") +
ylab("Percentage")
grid.arrange(p5)
# numbers of data with zero Capital Gain
(CG <- sum(Adultdata$capital.gain==0)/N.obs*100 )
p6 <- ggplot(Adultdata, aes(x=log10(capital.loss+1))) + ggtitle("Histogram of log(Capital Loss)") +
  geom_histogram(aes(y = 100*(..count..)/sum(..count..)), colour="black", fill="salmon") +
ylab("Percentage")
```

```
grid.arrange(p6)
# number of data with zero Capital Loss
(CL <- sum(Adultdata$capital.loss==0)/N.obs*100)

# remove the variable: capital.gain and capital.loss
Adultdata[["capital.gain"]] <- NULL
Adultdata[["capital.loss"]] <- NULL

## Explore Categorical Data
# Sort categorical variables in descending order
categ.sort <- function(x){reorder(x,x,function(y){-length(y)})} ## Sorting function for categorical
variables
categ.var <- which(sapply(Adultdata, is.factor)) ## Find the categorical variables
for (c in categ.var){  ## Apply the sort function on each categorical variable
  Adultdata[,c] <- categ.sort(Adultdata[,c])
}
attach(Adultdata)
p1 <- ggplot(Adultdata, aes(x=Adultdata$workclass)) + ggtitle("Histogram of Work Class") +
xlab("Work Class") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)),colour="black", fill="salmon") +
ylab("Percentage")  +
  scale_x_discrete(limits = levels(Adultdata$workclass))
grid.arrange(p1)
p2 <- ggplot(Adultdata, aes(x=Adultdata$education)) + ggtitle("Histogram of Education") +
xlab("Education") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)),colour="black", fill="salmon") +
ylab("Percentage") +
  scale_x_discrete(limits = levels(Adultdata$education))
grid.arrange(p2)
p3 <- ggplot(Adultdata, aes(x=Adultdata$marital.status)) + ggtitle("Histogram of Marital Status")
+ xlab("Marital Status") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)),colour="black", fill="salmon") +
ylab("Percentage") +
  scale_x_discrete(limits = levels(Adultdata$marital.status))
grid.arrange(p3)
p4 <- ggplot(Adultdata, aes(x=occupation)) + ggtitle("Histogram of Occupation") +
xlab("Occupation") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)),colour="black", fill="salmon") +
ylab("Percentage") +
  scale_x_discrete(limits = levels(Adultdata$occupation))
grid.arrange(p4)
p5 <- ggplot(Adultdata, aes(x=relationship)) + ggtitle("Histogram of Relationship") +
xlab("Relationship") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)),colour="black", fill="salmon") +
```

```
ylab("Percentage") +
  scale_x_discrete(limits = levels(Adultdata$relationship))
grid.arrange(p5)
p6 <- ggplot(Adultdata, aes(x=race)) + ggtitle("Histogram of Race") + xlab("Race") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)),colour="black", fill="salmon") +
ylab("Percentage") +
  scale_x_discrete(limits = levels(Adultdata$race))
grid.arrange(p6)
p7 <- ggplot(Adultdata, aes(x=sex)) + ggtitle("Histogram of sex") + xlab("sex") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)),colour="black", fill="salmon") +
ylab("Percentage") +
  scale_x_discrete(limits = levels(sex))
grid.arrange(p7)
p8 <- ggplot(Adultdata, aes(x=native.country)) + ggtitle("Histogram of Native Country") +
xlab("Native Country") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)),colour="black", fill="salmon") +
ylab("Percentage") +
  scale_x_discrete(limits = levels(native.country))
grid.arrange(p8)

# remove the variable: nativecountry
Adultdata[["native.country"]] <- NULL

#----------------------------------------
# Correlation between numerical variables and income class
boxplot (log(fnlwgt)~income, data =Adultdata, main = "Log(fnlwgt) for different income levels",
      xlab = "Income Levels", ylab = "Fnlwgt", col = "salmon")
boxplot (age~income, data =Adultdata, main = "Age for different income levels",
      xlab = "Income Levels", ylab = "Age", col = "salmon")
boxplot (education.num~income, data =Adultdata, main = "Years of Eduction for different
income levels",
      xlab = "Income Levels", ylab = "Years of Eduction", col = "salmon")
boxplot (hours.per.week~income, data =Adultdata, main = "hoursperweek for different income
levels",
      xlab = "Income Levels", ylab = "Hours per week", col = "salmon")
# remove the variable: fnlwgt
Adultdata[["fnlwgt"]] <- NULL

#----------------------------------------
# Correlation between categorical variables and income class
qplot(income, data = Adultdata, fill = workclass) + facet_grid (. ~ workclass)
qplot(income, data = Adultdata, fill = education) + facet_grid (. ~ education)
qplot(income, data = Adultdata, fill = occupation) + facet_grid (. ~ occupation)
qplot(income, data = Adultdata, fill = marital.status) + facet_grid (. ~ marital.status)
```

```
qplot(income, data = Adultdata, fill = relationship) + facet_grid (. ~ relationship)
qplot(income, data = Adultdata, fill = race) + facet_grid (. ~ race)
qplot(income, data = Adultdata, fill = sex) + facet_grid (. ~ sex)

dim(Adultdata)
############################################################

#---------------------------------------------------------------------
### association rule
colnames(Adultdata)[5] <- "maritalstatus"
colnames(Adultdata)[10] <- "hoursperweek"
Adultdata[["education.num"]] <- NULL
Adultdata[["age"]] <- ordered(cut(Adultdata[["age"]], c(15, 25, 45, 65, 100)), labels = c("Young",
"Middle-aged","Senior", "Old"))
Adultdata[["hoursperweek"]] <- ordered(cut(Adultdata[["hoursperweek"]],c(0, 25, 40, 60, 168)),
labels = c("Part-time", "Full-time","Over-time", "Workaholic"))
Adultdata$income=as.factor(as.character(Adultdata$income))
Adult0 <- as(Adultdata, "transactions")
summary(Adult0)
itemLabels(Adult0)
rules <- apriori(Adult0, parameter = list(support = 0.01, confidence = 0.7))
summary(rules)
rulesIncomeSmall <- subset(rules, subset = rhs %in% "income= <=50K" &lift > 1.2)
rulesIncomeLarge <- subset(rules, subset = rhs %in% "income= >50K" &lift > 1.2)
inspect(sort(rulesIncomeSmall, by = "confidence")[1:10])
inspect(sort(rulesIncomeLarge, by = "confidence")[1:10])


#---------------------------------------------------------------------
########## Dimension Reduction ##########
census<-adulttrain
Adult<-adulttrain
Adult_test<-adulttest
Adult_test$income<-as.factor(Adult_test$income)
str(Adult_test)
summary(Adult)
#------- MCA ---------#
library(gtools)
cont<-c(3,5,11,12,13) #the continuous variables that will be split into quartiles
for (i in cont){Adult[,i]<-quantcut(Adult[,i])}
for(i in cont) levels(Adult[,i]) <- paste(colnames(Adult)[i],levels(Adult[,i]))

# Age variable
Adult[,1]<- cut(Adult$age, breaks = c(0,25,35,49,64,75))
```

```
levels(Adult[,1])<-c("age:25 and under", "age:26 to 35", "age:36 to 49", "age:50 to 64", "age:65
and up")
summary(Adult)
str(Adult)

library(PCAmixdata)
x.quanti<-census[,c(1,3,5,11,12,13)]
x.quali<-census[,-c(1,3,5,11,12,13,15)]
PCAmix(x.quanti,x.quali)

library(FactoMineR)
res<-MCA(Adult,quali.sup = c(15))
summary(res,ncp=3,nbelements=Inf)
plot(res,label=c("var","quali.sup")) # too loaded
plot(res,invisible=c("ind","quali.sup"),autoLab="y",cex=0.7) # plot active variables (categories)
plot(res,invisible=c("ind"),cex=0.7,selectMod="contrib 20",unselect="grey90") # 20 variables
contributed most
plot(res,invisible=c("ind"),autoLab="y",cex=0.7,selectMod="cos2 20",unselect="grey90") # 20
variables most correlated
plot(res, invisible=c("ind","var")) # illustrative variable (target modalities)
dimdesc(res)

# Visualize eigenvalues (scree plot) #
eig.val <- res$eig
barplot(eig.val[, 2],
       names.arg = 1:nrow(eig.val),
       main = "Variances Explained by Dimensions (%)",
       xlab = "Principal Dimensions",
       ylab = "Percentage of variances",
       col ="steelblue")
lines(x = 1:nrow(eig.val), eig.val[, 2], # Add connected line segments to the plot
     type = "l",pch = 19,col = "red")

#-------- Variable clustering --------#
x.quanti<-census[,c(1,3,5,11,12,13)]
x.quali<-census[,-c(1,3,5,11,12,13,15)]
library(ClustOfVar)
#run variable clustering excluding the target variable (income)
variable_tree <- hclustvar(X.quali = x.quali, X.quanti = x.quanti)
#plot the dendrogram of variable groups
plot(variable_tree)
stability(variable_tree, B=25)
##############################################################################
######################
```

```
##############################################################################
######################
############Back to introduction: data understanding,supervised and unsupervised
Learning#############
##############################################################################
######################
##############################################################################
######################
# Display the histogram
par(mfrow=c(2,3))
hist(adulttrain$age)
hist(adulttrain$education.num)
hist(adulttrain$hours.per.week)
hist(adulttrain$capital.gain)
hist(adulttrain$capital.loss)

par(mfrow=c(2,3))
hist(adulttest$age)
hist(adulttest$education.num)
hist(adulttest$hours.per.week)
hist(adulttest$capital.gain)
hist(adulttest$capital.loss)

# Display barplots
par(mfrow=c(2,3))
barplot(table(adulttrain$marital.status),main='marital.status')
barplot(table(adulttrain$occupation),main='occupation')
barplot(table(adulttrain$relationship),main='relationship')
barplot(table(adulttrain$race),main='race')
barplot(table(adulttrain$sex),main='sex')
barplot(table(adulttrain$native.country),main='native.country')

par(mfrow=c(2,3))
barplot(table(adulttest$marital.status),main='marital.status')
barplot(table(adulttest$occupation),main='occupation')
barplot(table(adulttest$relationship),main='relationship')
barplot(table(adulttest$race),main='race')
barplot(table(adulttest$sex),main='sex')
barplot(table(adulttest$native.country),main='native.country')

length(adulttrain$capital.gain[adulttrain$capital.gain !=
0])/length(adulttrain$capital.gain)+length(adulttrain$capital.loss[adulttrain$capital.loss !=
0])/length(adulttrain$capital.loss)
length(adulttest$capital.gain[adulttest$capital.gain !=
```

```
0])/length(adulttest$capital.gain)+length(adulttest$capital.loss[adulttest$capital.loss !=
0])/length(adulttest$capital.loss)

# Delete variables
adulttrain$education <- NULL
adulttrain$native.country <- NULL
adulttest$education <- NULL
adulttest$native.country <- NULL

adulttrain$capital.change <- adulttrain$capital.gain - adulttrain$capital.loss
adulttest$capital.change <- adulttest$capital.gain - adulttest$capital.loss
adulttrain$capital.gain <- NULL
adulttrain$capital.loss<-NULL
adulttest$capital.gain <- NULL
adulttest$capital.loss<-NULL

# swap capital.change and income
adulttrain[c(11,12)] <- adulttrain[c(12,11)]
colnames(adulttrain)[11:12] <- colnames(adulttrain)[12:11]
adulttest[c(11,12)] <- adulttest[c(12,11)]
colnames(adulttest)[11:12] <- colnames(adulttest)[12:11]

adulttrain$income <- as.factor(ifelse(adulttrain$income == ' <=50K',0,1))
adulttest$income <- as.factor(ifelse(adulttest$income == ' <=50K.',0,1))
str(adulttrain)
str(adulttest)

no.outlier<-function(data,x)
{
 for (i in x)
 {
  a<-boxplot.stats(data[,i])$out
  data<-data[!data[,i]%in%a,]
  print(length(a))
 }
 return(data)
}
str(adulttrain)

adulttrain <- no.outlier(adulttrain,c(1,3))
adulttest <- no.outlier(adulttest,c(1,3))

adulttrain$fnlwgt <- NULL
adulttest$fnlwgt <- NULL
```

```
dim(adulttrain)
dim(adulttest)
income <- adulttest$income
adulttest <- adulttest[,-11]


############### Unsupervised Learning##############################
# 1.---------- K-Prototype Clustering ------------#
# Check for the optimal number of clusters given the data
set.seed(123)
wss<-vector()
for (i in 2:15){ wss[i] <- sum(kproto(adulttrain, i)$withinss)}
plot(1:15, wss, type="b", xlab="number of clusters",
    ylab="withinss",
    main="Optimal Number of Clusters",
    pch=20, cex=2)
# From the plot we conclude that 6 is the best number.

proto<-kproto(adulttrain, k=6)
adulttrain$cluster = as.factor(proto$cluster)
clprofiles(proto, adulttrain)
plot(adulttrain$age,adulttrain$education.num,col=rainbow(6)[adulttrain$cluster],main='age vs
education.num',pch=19)
plot(adulttrain$age,adulttrain$hours.per.week,col=rainbow(6)[adulttrain$cluster],main='age vs
hours.per.week',pch=19)
plot(adulttrain$age,adulttrain$capital.change,col=rainbow(6)[adulttrain$cluster],main='age vs
capital.change',pch=19)
plot(adulttrain$education.num,adulttrain$hours.per.week,col=rainbow(6)[adulttrain$cluster],
main='education.num vs hours.per.week',pch=19)
plot(adulttrain$education.num,adulttrain$capital.change,col=rainbow(6)[adulttrain$cluster],m
ain='education.num vs capital.change',pch=19)
plot(adulttrain$hours.per.week,adulttrain$capital.change,col=rainbow(6)[adulttrain$cluster],m
ain='hours.per.week vs capital.change',pch=19)

########################## Supervised Learning ##########################
adulttrain$cluster<-NULL
set.seed(1)
# 1.---------- Regression Tree------------#
tree <- rpart(income ~ ., data = adulttrain, method = 'class')
pred.tree <- predict(tree, newdata =adulttest, type = 'class')
confusionMatrix(pred.tree ,income)


# 2.---------- Random Forest ------------#
```

```
set.seed(123)
rf.adult <- randomForest(adulttrain$income ~ .,
data=adulttrain,mtry=sqrt(10),importance=TRUE)
rf.adult
rf.hpred <- predict(rf.adult,newdata=adulttest,type="class")
confusionMatrix(rf.hpred,income)
varImpPlot(rf.adult,type=2)

# 3. #---------- Logistic Regression ----------#
logistfit<- glm(income ~ .,data=adulttrain,family=binomial(link='logit'))
summary(logistfit)

null_model<- glm(income ~ 1, data = adulttrain, family = binomial('logit'))

# backward selection
fwd_aic <- step(logistfit, trace = F, scope = list(lower=formula(null_model),
upper=formula(logistfit)),
            direction = 'forward')
fwd_aic
# Logistic Regression Prediction
preds <- predict(logistfit,newdata=adulttest,type='response')
preds <- ifelse(preds > 0.5,1,0)
# Accuracy

confusionMatrix(as.factor(preds),income) # 0.8402

#census$capital.change <- census$capital.gain-census$capital.loss
#adulttest$capital.change <- adulttest$capital.gain-adulttest$capital.loss
#adulttrain<-census[,-c(3,4,11,12,14)]
#adulttest<-adulttest[,-c(3,4,11,12,14)]

# 4. ---------- Naive Bayes ----------#
set.seed(2)
NB_model<-naiveBayes(income~.,data = adulttrain)
NB_prediction<-predict(NB_model,adulttest)
confusionMatrix(NB_prediction,income )
# 5. ---------- SVM ----------#
set.seed(3)
svm.model<- svm(income~., data = adulttrain,kernel = "radial", cost = 1, gamma =
0.1,scale=TRUE)
svm.predict <- predict(svm.model, adulttest)
confusionMatrix(svm.predict,income)
#table(svm.predict,Adult.tdummy[,10])
```

```
# 6. ---------- KNN ------------#
## Convert categorical variables to numerical data (Generate dummy variables) ##
## training set ##
# create dummy variables (training set) #
#set.seed()
library(dummies)
Adult.dummy <- dummy.data.frame(adulttrain[,-11], sep = ".")
Adult.dummy<-cbind(Adult.dummy,adulttrain$income)
colnames(Adult.dummy)[46] <- "income"
names(Adult.dummy)
Adult.dummy<-Adult.dummy[,-c(2,10,17,31,37,42)] # remove first dummy #

# scale numerical variables #
## 2 sd ##
#Adult.dummy$age<-(Adult.dummy$age-mean(adulttrain$age))/(sd(adulttrain$age))/2
#Adult.dummy$education.num<-(Adult.dummy$education.num-
mean(adulttrain$education.num))/(sd(adulttrain$education.num))/2
#Adult.dummy$hours.per.week<-(Adult.dummy$hours.per.week-
mean(adulttrain$hours.per.week))/(sd(adulttrain$hours.per.week))/2
## 1 sd ##
#Adult.dummy$age<-(Adult.dummy$age-mean(adulttrain$age))/sd(adulttrain$age)
#Adult.dummy$education.num<-(Adult.dummy$education.num-
mean(adulttrain$education.num))/sd(adulttrain$education.num)
#Adult.dummy$hours.per.week<-(Adult.dummy$hours.per.week-
mean(adulttrain$hours.per.week))/sd(adulttrain$hours.per.week)
## max-min ##
Adult.dummy$age<-(Adult.dummy$age-min(adulttrain$age))/(max(adulttrain$age)-
min(adulttrain$age))
Adult.dummy$education.num<-(Adult.dummy$education.num-
min(adulttrain$education.num))/(max(adulttrain$education.num)-
min(adulttrain$education.num))
Adult.dummy$hours.per.week<-(Adult.dummy$hours.per.week-
min(adulttrain$hours.per.week))/(max(adulttrain$hours.per.week)-
min(adulttrain$hours.per.week))
Adult.dummy$capital.change<-(Adult.dummy$capital.change-
min(adulttrain$capital.change))/(max(adulttrain$capital.change)-
min(adulttrain$capital.change))

## test set ##
# create dummy variables (test set) #
Adult.tdummy <- dummy.data.frame(adulttest, sep = ".")
#Adult.tdummy<-cbind(Adult.tdummy,income)
str(Adult.tdummy)
#colnames(Adult.tdummy)[46] <- "income" income==40
```

```
names(Adult.tdummy)
Adult.tdummy<-Adult.tdummy[,-c(2,10,17,31,37,42)]  # remove first dummy #

# scale numerical variables #
## 2 sd ##
#Adult.tdummy$age<-(Adult.tdummy$age-mean(adulttrain$age))/(sd(adulttrain$age))/2
#Adult.tdummy$education.num<-(Adult.tdummy$education.num-
mean(adulttrain$education.num))/(sd(adulttrain$education.num))/2
#Adult.tdummy$hours.per.week<-(Adult.tdummy$hours.per.week-
mean(adulttrain$hours.per.week))/(sd(adulttrain$hours.per.week))/2
## 1 sd ##
#Adult.tdummy$age<-(Adult.tdummy$age-mean(adulttrain$age))/sd(adulttrain$age)
#Adult.tdummy$education.num<-(Adult.tdummy$education.num-
mean(adulttrain$education.num))/sd(adulttrain$education.num)
#Adult.tdummy$hours.per.week<-(Adult.tdummy$hours.per.week-
mean(adulttrain$hours.per.week))/sd(adulttrain$hours.per.week)
## max-min ##
Adult.tdummy$age<-(Adult.tdummy$age-min(adulttrain$age))/(max(adulttrain$age)-
min(adulttrain$age))
Adult.tdummy$education.num<-(Adult.tdummy$education.num-
min(adulttrain$education.num))/(max(adulttrain$education.num)-
min(adulttrain$education.num))
Adult.tdummy$hours.per.week<-(Adult.tdummy$hours.per.week-
min(adulttrain$hours.per.week))/(max(adulttrain$hours.per.week)-
min(adulttrain$hours.per.week))
Adult.tdummy$capital.change<-(Adult.tdummy$capital.change-
min(adulttrain$capital.change))/(max(adulttrain$capital.change)-
min(adulttrain$capital.change))
# K=5 #
library(class)
set.seed(5)
knnpred <- knn(Adult.dummy[, -40],Adult.tdummy,Adult.dummy[,40], k = 5)
confusionMatrix(knnpred,income)
# k=10 #
set.seed(6)
knnpred <- knn(Adult.dummy[,-40],Adult.tdummy,Adult.dummy[,40], k = 10)
confusionMatrix(knnpred,income)
# k=15 #
set.seed(7)
knnpred <- knn(Adult.dummy[,-40],Adult.tdummy,Adult.dummy[,40], k = 15)
confusionMatrix(knnpred,income)
# k=the square-root of the number of observations #
set.seed(8)
knnpred <- knn(Adult.dummy[,-40],Adult.tdummy,Adult.dummy[,40], k =
```

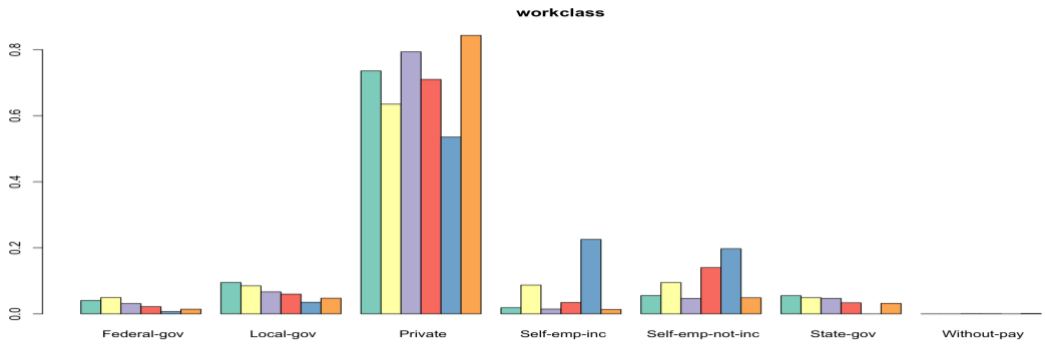
```
round(sqrt(nrow(Adult.dummy))))
confusionMatrix(knnpred,income)

# 7. ---------- Neural Network ----------#
set.seed(4)
library(nnet)
nn1 <- nnet(income~., data = adulttrain, size = 40,
        maxit = 500, MaxNWts=2000)
summary(nn1)
nn1.pred <- predict(nn1, newdata = adulttest, type = 'class')
nn1.pred <- as.factor(nn1.pred)
confusionMatrix(nn1.pred,income)

################################################################################
#####
################################################################################
#####
##############Conclusion: Evaluate the performance using ROC Curve ##################
################################################################################
#####
################################################################################
#####

trueYes <- income

# Classification Trees
treepreds <- predict(tree,newdata=adulttest,type="prob")
posteriorYes.tree <- treepreds[,2]
ROCres.tree <- roc(posteriorYes.tree,trueYes)
tidyROCres.tree <- tidy(ROCres.tree)

# Random Forest
rfpreds <- predict(rf.adult,newdata=adulttest,type="prob")
posteriorYes.rf <- rfpreds[,2]
ROCres.rf <- roc(posteriorYes.rf,trueYes)
tidyROCres.rf <- tidy(ROCres.rf)

# Logistic Regression
logitpreds <- predict(logistfit,newdata=adulttest,type='response')
posteriorYes.logit <- logitpreds
ROCres.logit <- roc(posteriorYes.logit,trueYes)
tidyROCres.logit <- tidy(ROCres.logit)

# Naive Bayes
```

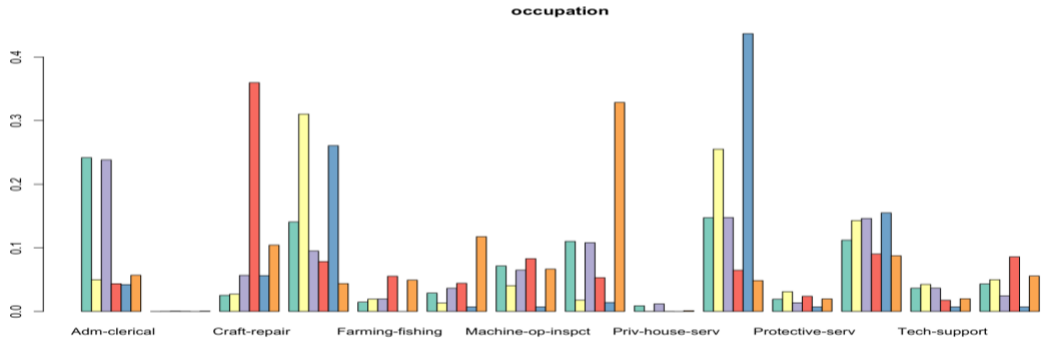
```
nbpreds <-predict(NB_model,adulttest,type='raw')
posteriorYes.nb <- nbpreds[,2]
ROCres.nb <- roc(posteriorYes.nb,trueYes)
tidyROCres.nb <- tidy(ROCres.nb)

# SVM
set.seed(3)
svm.model1 <- svm(income~., data = adulttrain,kernel = "radial", cost = 1, gamma =
0.1,scale=TRUE,probability=T)
svmpreds <- predict(svm.model1, adulttest,probability = T)
posteriorYes.svm <- attr(svmpreds, "probabilities")[,2]
ROCres.svm <- roc(posteriorYes.svm,trueYes)
tidyROCres.svm <- tidy(ROCres.svm)

# KNN
set.seed(6)
knnpreds <- knn(Adult.dummy[, -40],Adult.tdummy,Adult.dummy[,40], k = 5,prob = T)
posteriorYes.knn <- knnpreds
ROCres.knn <- roc(posteriorYes.knn,trueYes)
tidyROCres.knn <- tidy(ROCres.knn)

# Neural Network
nnpreds <- predict(nn1, newdata = adulttest, type = 'raw')
posteriorYes.nn <- nnpreds
ROCres.nn <- roc(posteriorYes.nn,trueYes)
tidyROCres.nn <- tidy(ROCres.nn)

# ROC Curve

ggplot() +
  geom_line(data=tidyROCres.tree,aes(x=fpr,y=tpr,color='Classification Trees')) +
  geom_line(data=tidyROCres.rf,aes(x=fpr,y=tpr,color='Random Forest')) +
  geom_line(data=tidyROCres.logit,aes(x=fpr,y=tpr,color='Logistic Regression')) +
  geom_line(data=tidyROCres.nb,aes(x=fpr,y=tpr,color='Naive Bayes')) +
  geom_line(data=tidyROCres.svm,aes(x=fpr,y=tpr,color='SVM')) +
  geom_line(data=tidyROCres.knn,aes(x=fpr,y=tpr,color='KNN')) +
  geom_line(data=tidyROCres.nn,aes(x=fpr,y=tpr,color='Neural Network')) +
  geom_segment(aes(x = 0, xend = 1, y = 0, yend = 1)) +
  geom_abline(slope=1, intercept=0, linetype=2) +
  ggtitle('ROC Curves for various models')

########################## ROC Curve END ######################

########################## AUC ##########################
```

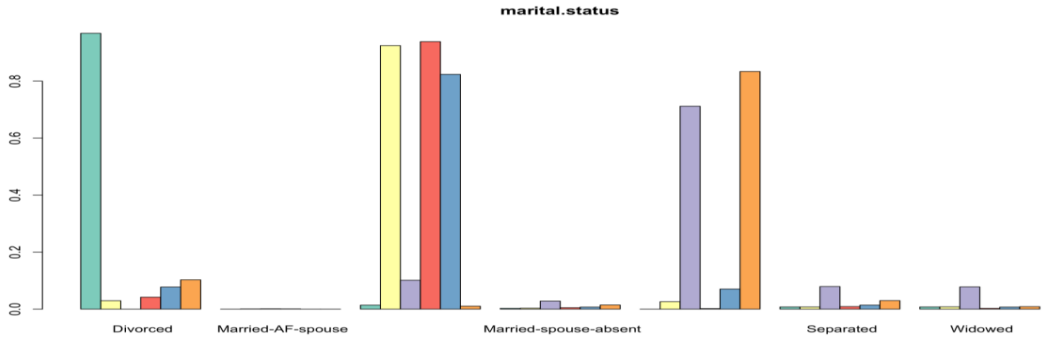
```
require(AUC)
aucs <-
c(auc(ROCres.tree),auc(ROCres.rf),auc(ROCres.logit),auc(ROCres.nb),auc(ROCres.svm),auc(ROCr
es.knn),auc(ROCres.nn))
aucs.df <- data.frame(models = c('Classification Trees','Random Forest','Logistic Regression',
                                 'Naive Bayes','SVM','KNN','Neural Network'),aucs)
aucs.df[order(aucs.df$aucs,decreasing = T),]
########################## AUC END ##########################
```

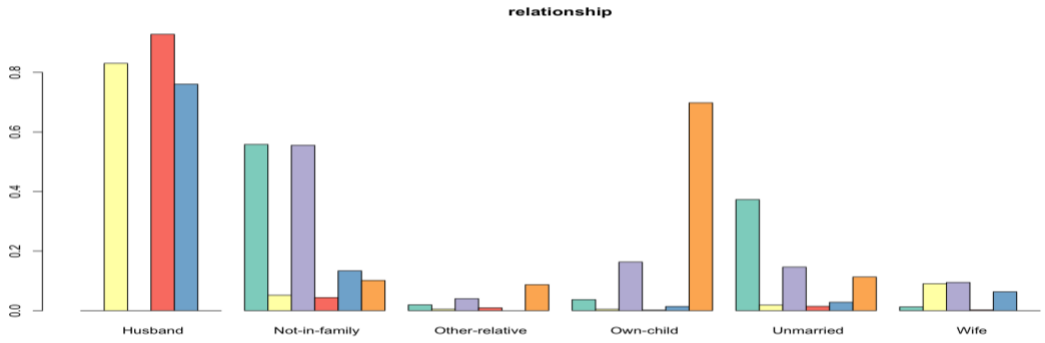
Appendix

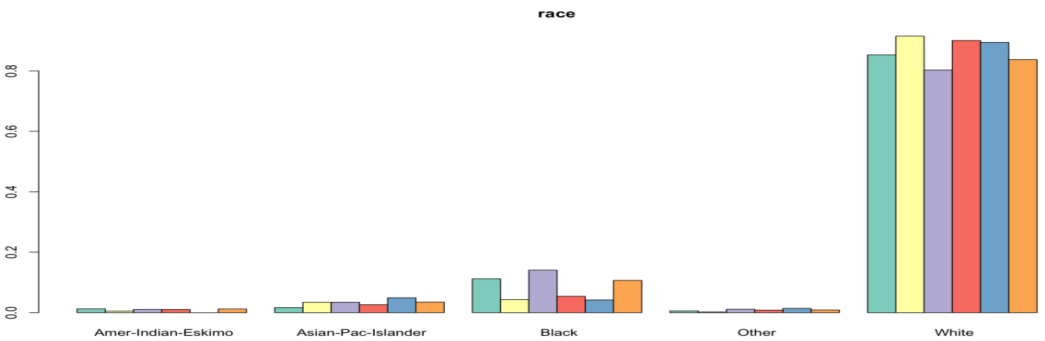
Appendix1: Graphical results for K-prototype clustering

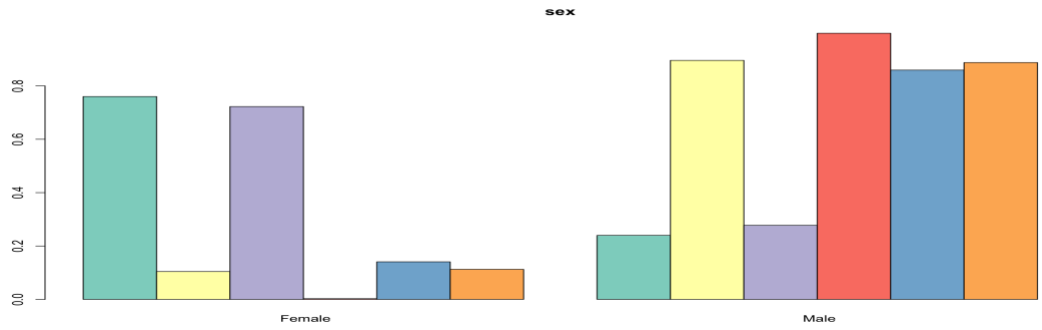


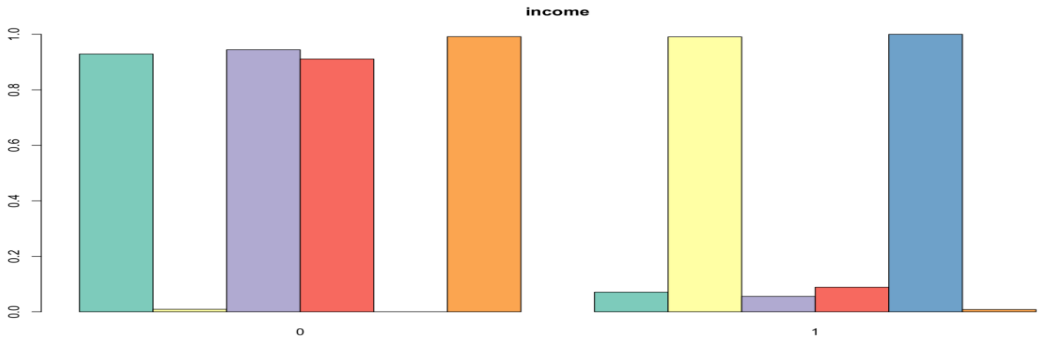


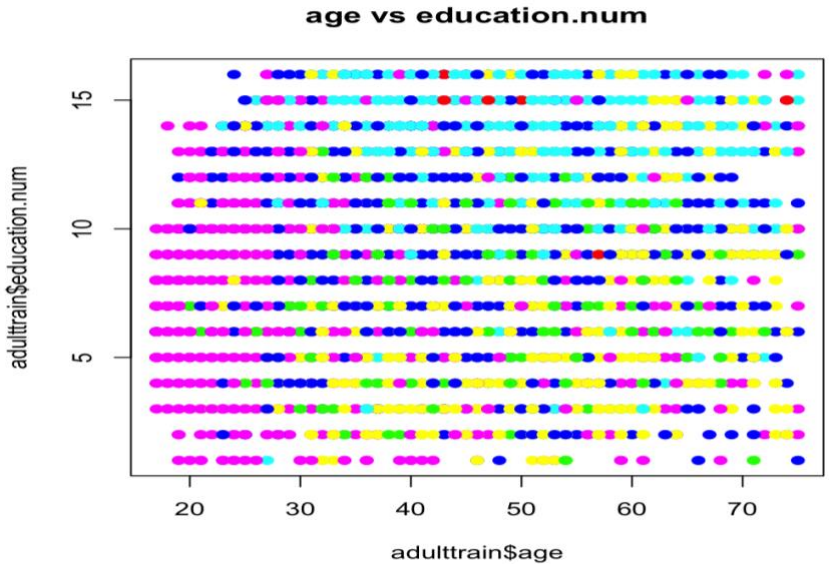

marital.status
0.0
0.2
0.4
0.6
0.8
Divorced
Married-AF-spouse
Married-spouse-absent
Separated
Widowed

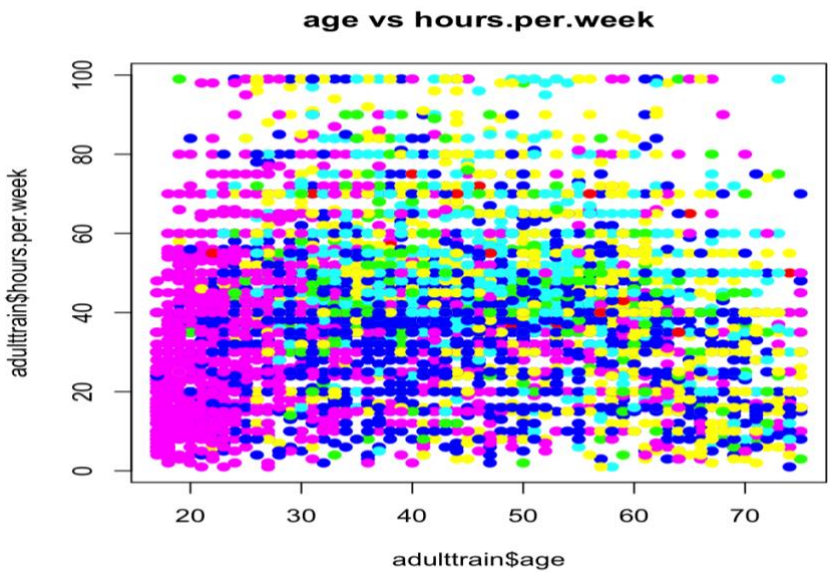

relationship
0.0
0.2
0.4
0.6
0.8
Husband
Not-in-family
Other-relative
Own-child
Unmarried
Wife

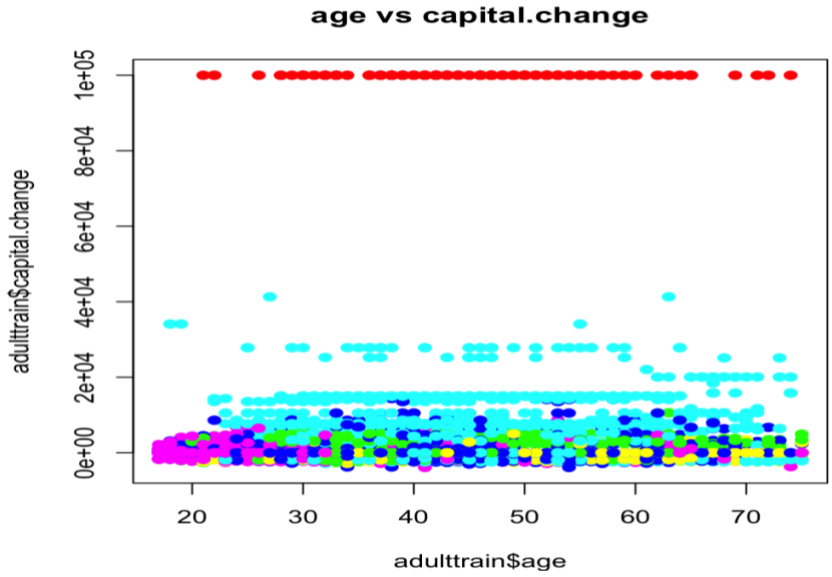

race
0.0
0.2
0.4
0.6
0.8
Amer-Indian-Eskimo
Asian-Pac-Islander
Black
Other
White

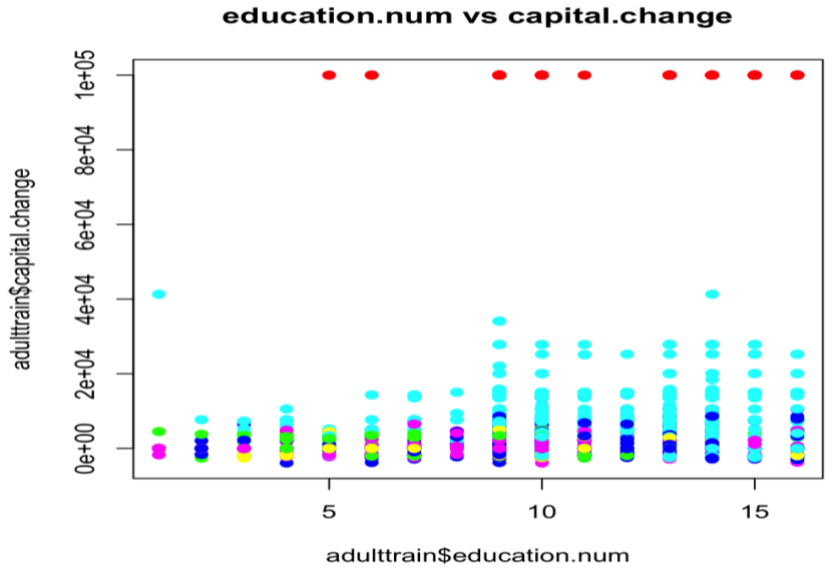

sex
0.0
0.2
0.4
0.6
0.8
Female
Male

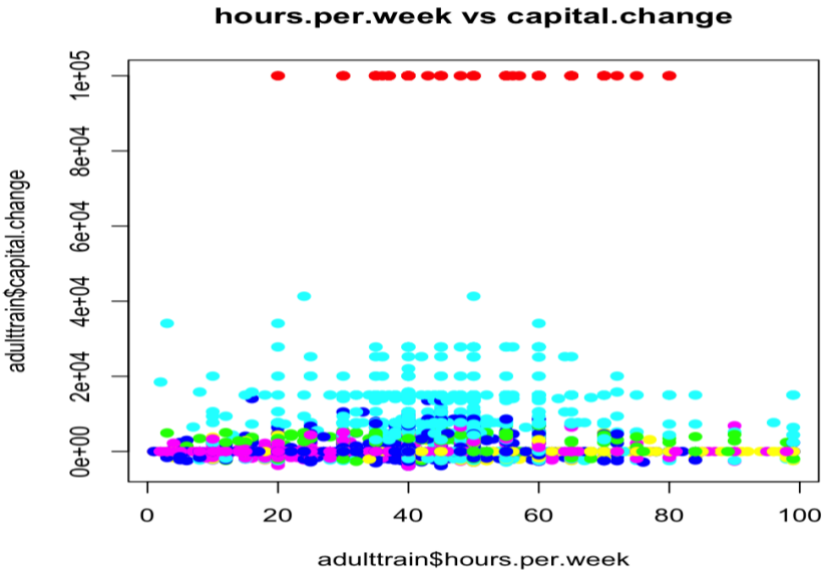

income
0.0
0.2
0.4
0.6
0.8
1.0
0
1

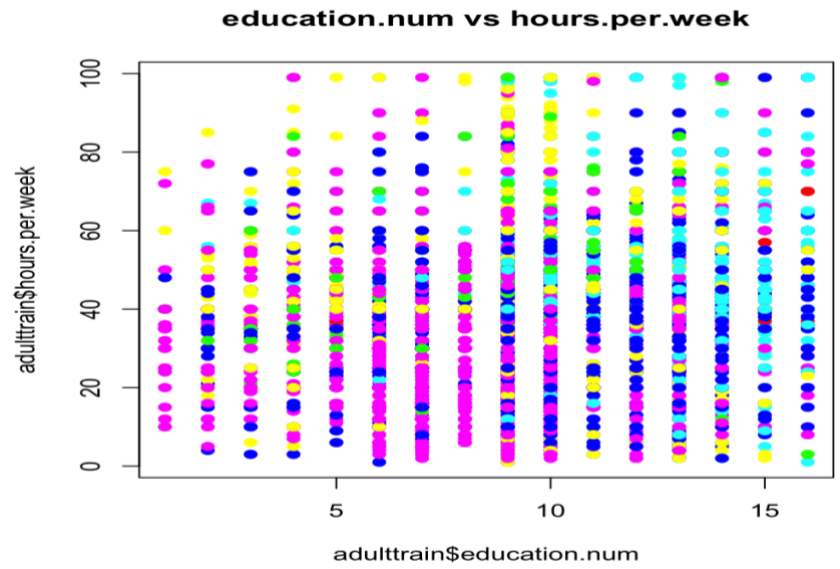

age vs education.num
adulttrain$education.num
5
10
15
20
30
40
50
60
70
adulttrain$age


age vs hours.per.week
adulttrain$hours.per.week
0
20
40
60
80
100
20
30
40
50
60
70
adulttrain$age


age vs capital.change
adulttrain$capital.change
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
20
30
40
50
60
70
adulttrain$age


education.num vs capital.change
adulttrain$capital.change
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
5
10
15
adulttrain$education.num


hours.per.week vs capital.change
adulttrain$capital.change
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
0
20
40
60
80
100
adulttrain$hours.per.week


education.num vs hours.per.week
adulttrain$hours.per.week
0
20
40
60
80
100
5
10
15
adulttrain$education.num

Responsibilities

Introduction (Vicky Xu)

Visualization (Xue Cao)

Dimension Reduction

MCA (Xiyu Liang)

Variable Clustering (Xiyu Liang)

Data Reduction &Unsupervised Learning

Clustering (Vicky Xu)

Association Rules (Xue Cao)

Conclusion

The influence of predictors on the target variable (Xue Cao, Xiyu Liang)

Supervised Learning

Logistic (Xue Cao, Vicky Xu)

Neural Network (Xue Cao)

K-nearest neighbors (Xiyu Liang)

Naïve Bayes (Xiyu Liang)

SVM (Xiyu Liang)

Regression Tree (Vicky Xu)

Random Forest (Vicky Xu)

Conclusion

Performance Evaluation: ROC Curves and AUC (Vicky Xu)

Reference

Z.Huang (1998): Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Variables, Data Mining and Knowledge Discovery 2, 283-304.

Olano, Diego Garcia, and Elsa Mullor. 27 June 2014, diegoolano.com/reports/US-census-wealth.pdf.