Dataiku exercice : US Census Data Set

Bénédicte RALLET

Programming Langage : R

Contents

1 Statistic based and univariate audit

The data set from the US Census contains 42 variables but one of them is not used (”instance weight”) and one corresponds to the variable that we want to predict. Thus, we have a total of 40 available predictors to fit a model. 7 are continuous variables and 33 are nominal ones.

1.1 General information

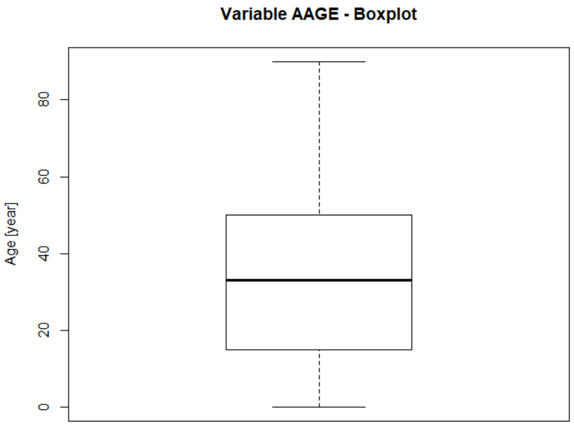
Variable	Code	Type	Missing values ”?”	NIU
age	AAGE	Continuous		
class of worker	ACLSWKR	Nominal (9 levels)		100245
industry code	ADTIND	Nominal (52 levels)		
occupation code	ADTOCC	Nominal (47 levels)		
education	AHGA	Nominal (17 levels)		
wage per hour	AHRSPAY	Continuous		
enrolled in edu inst last wk	AHSCOL	Nominal (yes/no/NIU)		186943
marital status	AMARITL	Nominal (7 levels)		
major industry code	AMJIND	Nominal (24 levels)		100684
major occupation code	AMJOCC	Nominal (15 levels)		100684
race	ARACE	Nominal (5 levels)		
hispanic origin	AREORGN	Nominal (10 levels)		
sex	ASEX	Binary		
member of a labor union	AUNMEM	Nominal (yes/no/NIU)		180459
reason for unemployment	AUNTYPE	Nominal (6 levels)		193453
full or part time employment stat	AWKSTAT	Nominal (8 levels)		
capital gains	CAPGAIN	Continuous		
capital losses	CAPLOSS	Continuous		
divdends from stocks	DIVVAL	Continuous		
tax filer status	FILESTAT	Nominal (6 levels)		
region of previous residence	GRINREG	Nominal (6 levels)		183750
state of previous residence	GRINST	Nominal (51 levels)	708	183750
detailed household and family stat	HHDFMX	Nominal (38 levels)		
detailed household summary in household	HHDREL	Nominal (8 levels)		
migration code-change in msa	MIGMTR1	Nominal (10 levels)	99696	1516
migration code-change in reg	MIGMTR3	Nominal (9 levels)	99696	1516
migration code-move within reg	MIGMTR4	Nominal (10 levels)	99696	1516
live in this house 1 year ago	MIGSAME	Nominal (yes/no/NIU)		101212
migration prev res in sunbelt	MIGSUN	Nominal (4 levels)	99696	84054
num persons worked for employer	NOEMP	Continuous		
family members under 18	PARENT	Nominal (5 levels)		144232
country of birth father	PEFNTVTY	Nominal (43 levels)	6713	
country of birth mother	PEMNTVTY	Nominal (43 levels)	6119	
country of birth self	PENATVTY	Nominal (43 levels)	3393	
citizenship	PRCITSHP	Nominal (5 levels)		
own business or self employed	SEOTR	Nominal (3 levels)		
fill veteran’s admin	VETQVA	Nominal (3 levels)		197539
veterans benefits	VETYN	Nominal (3 levels)		
weeks worked in year	WKSWORK	Continuous		
year	YEAR	Nominal (94 or 95)		

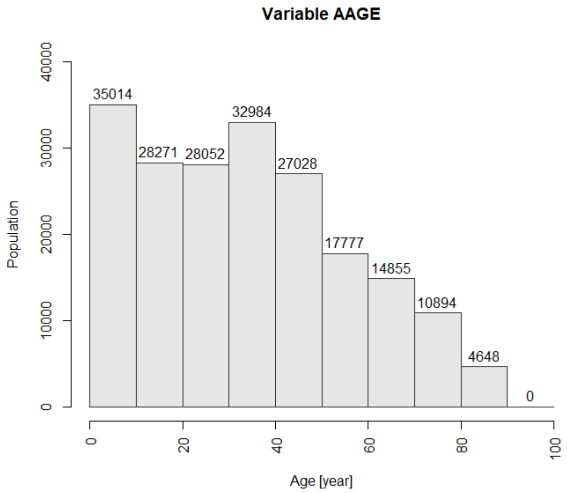
Note : ”NIU” stands for ”Not In Universe”

1.2 Continuous variables

Age

The variable *AAGE* is a continuous variable witch takes its values from 0 to 90. We can represent the variable using a boxplot to see the general distribution of the variable. We can see that the median is around 30 years and half of the population has an age between 15 and 50 years old.
In addition, we can plot an histogram, where the interval is 10 years. We can notice that the most represented category is the one between 0 and 10 years. This is an information that we didn't see with the boxplot.



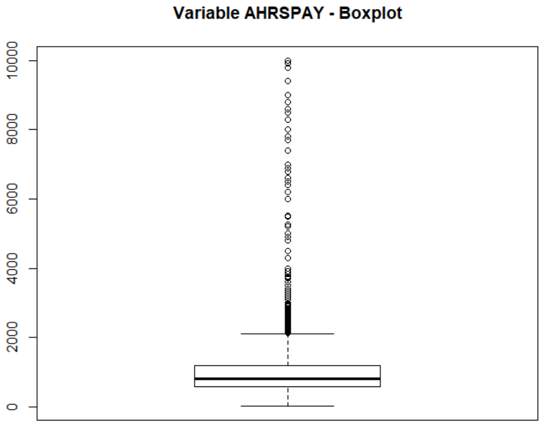


Wage per hour

With R, we can obtain a statistical summary of the variable *AHRSPAY*. The result shows that more than 75% of the values are 0. There is indeed 94.44% of values equal to 0 in the data set. Plotting a boxplot with all the values will not show anything.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.00	0.00	0.00	55.43	0.00	9999.00

A solution could be to plot only the non-zero value to have an idea of the distribution of the 5.66% values different from 0. The visual result is still not really good because there is a lot of extreme values.



Capital gains, losses and dividends from stocks

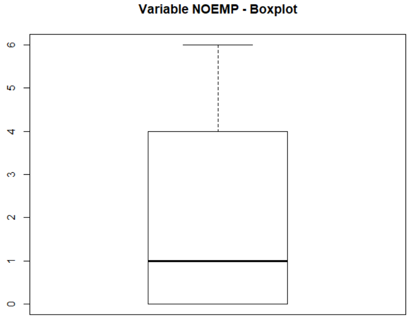
As previously, these variables have a majority of zero-values which make difficult to have a graphical representation of the distribution of the variable. The percentages of non-zero values are the following :

- *CAPGAIN* : 3.70%
- *CAPLOSS* : 1.96%
- *DIVVAL* : 10.96%

Persons work for employer

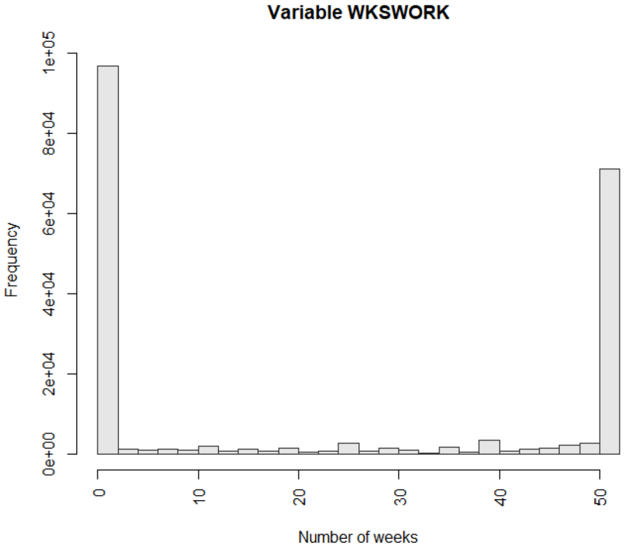
The summary of the variable *NOEMP* shows that the variable takes its value between 0 and 6, with a mean of almost 2 persons employed.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.000	0.000	1.000	1.956	4.000	6.000



Weeks worked in year

With the histogram below, we clearly see that most of the people are either working 0 week (not working tough) or 52 per year. 96.3% of the population of the data set is in one of these cases.

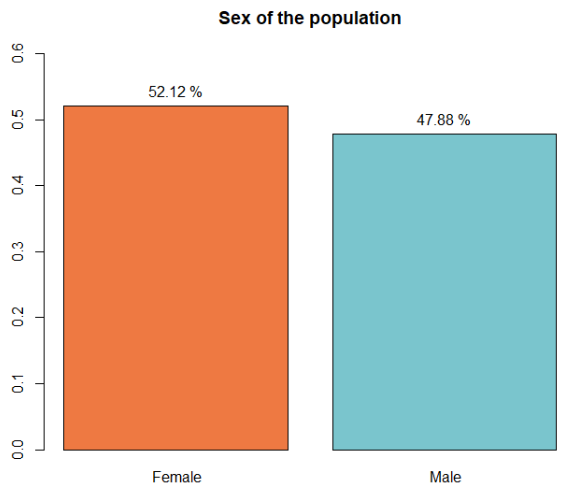


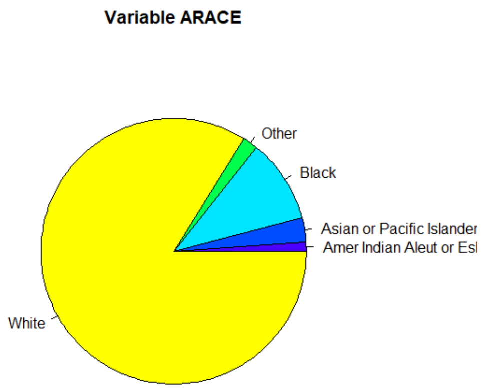
1.3 Nominal variables

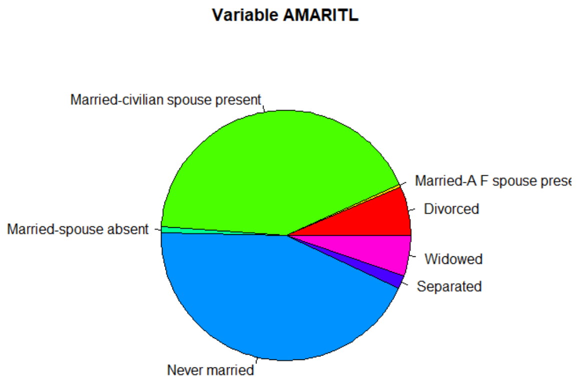
The other variables are quantitative variables, interpreted as ”factor” by R.
There are different ways to show the distribution of this type of variables : barplot, pie chart, histogram... Nevertheless, it can be tricky sometimes to have a nice visual chart when there are too many categories. The chart become messy and it is hard to distinguish the different values.

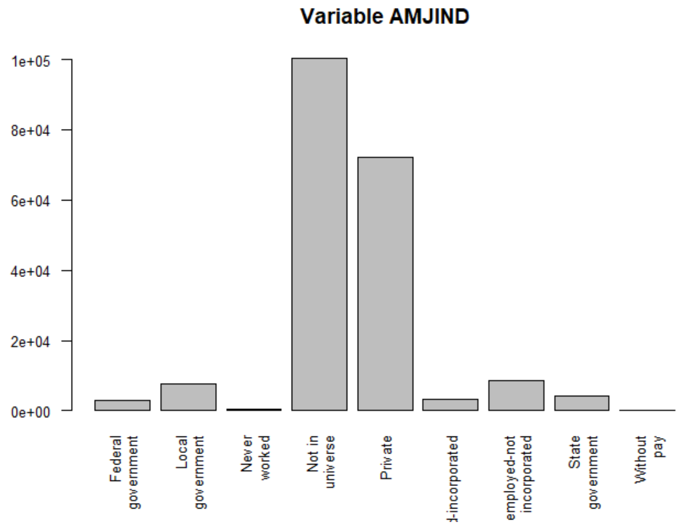
The following charts are representations for the variables *ASEX*, *ARACE*, *AMARITL*, *AMJIND*, *GRINREG* and *AHGA* (that shows how it is difficult to visualise a chart with a high number of categories).

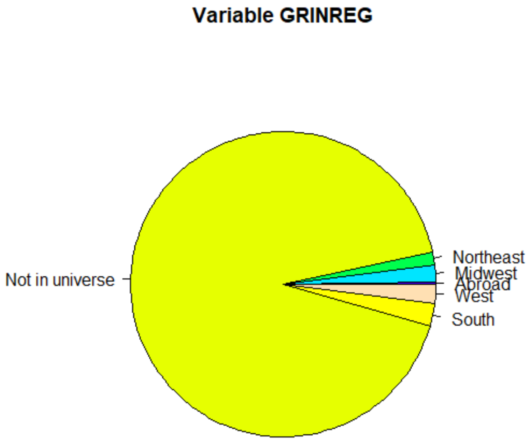
This list is not exhaustive and other descriptive charts can be found at `https://github.com/benrallet/Dataiku_US_Census/tree/master/Charts`.

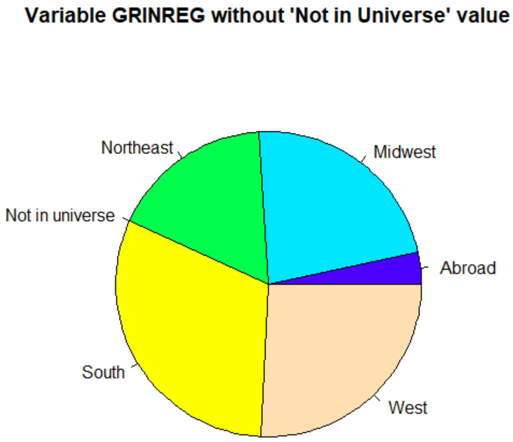


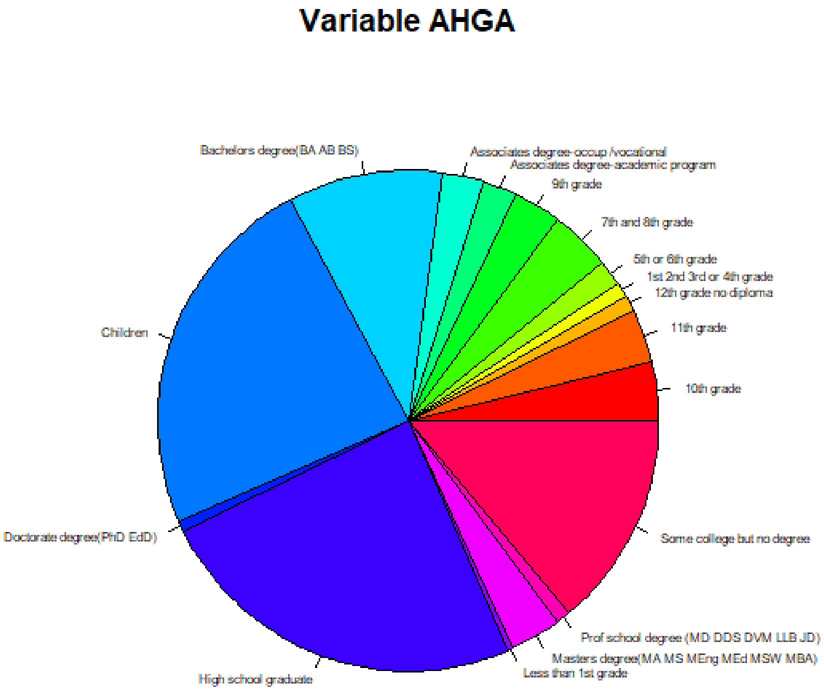







Variable GRINREG
Not in universe
Northeast
Midwest
Abroad
West
South


Variable GRINREG without 'Not in Universe' value
Northeast
Midwest
Not in universe
Abroad
South
West


Variable AHGA
Bachelors degree(BA AB BS)
Associates degree-occup /vocational
Associates degree-academic program
9th grade
7th and 8th grade
5th or 6th grade
1st 2nd 3rd or 4th grade
12th grade no diploma
11th grade
Children
10th grade
Doctorate degree(PhD EdD)
Some college but no degree
Prof school degree (MD DDS DVM LLB JD)
Masters degree(MA MS MEng MEd MSW MBA)
Less than 1st grade
High school graduate

2 Creation of a model

In this part, we will test two algorithms on our training set : decision tree and logistic regression.
After pre-processing the data, the algorithms will be implemented and their performance (error rate) will be computed using 10-Fold Cross-Validation. According to these results, one model will be chosen to model wining more or less than $50,000/year for the test data.

2.1 Decision Tree

Presentation of the steps followed to create a decision tree :

Pre-processing data

To model the decision tree, I will be using the library *tree* available as a package with R.

From an implementation point of view, one big advantage of the decision trees is that they can deal with both categorical and continuous variables. However, one limit with the library used is that it can't handle categorical variables with more than 32 levels.

In our case, seven variables don't fill this condition.

- Industry and Occupation code : these two variables have, respectively, 52 and 47 different values. However, a "major" category exists for each of these variables (Major industry code and Major occupation code) that regroup some categories to have a lower number of levels. For the aim of this exercise, I admit that the major variables will be enough to represent an observation.
- State of previous residence : this variable has 51 different values. A variable representing the region of previous residence is also in the data set. Regions can be seen as an aggregation of states. Thus, I made the choice to simply remove this variable to fit the model.
- Detailed Household and Family Stat : as the previous variables, it exists a predictor called "detailed household summary" that has only 8 levels (against 38 for the other one). I chose to remove also this variable from our model.
- Country of birth of the father, the mother and self (three variables) : these three variable have each one 43 categories corresponding to countries. To reduce this number, I thought about different solutions : grouping by continent, one-hot encoding, USA/not USA. I finally chose the last option as it does not increase the number of predictor and it was quite easy to set up. The three variables will now be used as binary predictors, 1 corresponding to United-States and 0 to others.

In this four cases, there is a loss of information but the information does not completely disappear (trade-off).

Construction of the tree

To create the decision tree with the *tree* library,

1. function *tree* to fit the model. The parameter *tree.control(nobs=dim(trainData)[1],mindev = 0.0001)* is used to have a complete tree.
2. *cv.tree* construct the sequence of the sub-trees embedded in the one obtained with function *tree* and to estimate the error rate with cross-validation
3. prune.misclass is used to do pruning based on the error rate estimation from the previous step
4. finally, *predict* gives the responses for a data set based on the decision tree constructed

Computation of the error rate

Once we have the responses for a data set (all variables but the output variables), we have to compare them to the "real" response, contained in the last column of the data set. Let *pred* be the array of predictions from the model and z the array of reality predicitions. The error rate can be computed as follow :

$$error = \frac{1}{n} \sum_{i=1}^{n} I(pred_i \neq z_i)$$

where n is the number of observations and I is a function that returns 1 if $pred_i \neq z_i$ and 0 otherwise.

2.2 Logistic Regression

Pre-processing data

For the logistic regression model, more pre-processing had to be done. Indeed, the algorithm can't handle categorical variables. Most of the predictors being categorical variables, a solution had to be implemented.

First, I applied the same transformation and removed the same variables than for the decision tree model. I simplified the model and removed the variable with a lot of factors than could lead to a big number of dummy variables (see the section corresponding).

Then, I transformed the variables $ASEX$ and $YEAR$ into binary variable (0/1). The transformation was immediate because they had only two categories.
To transform the other categorical variables (with more than two levels) to a binary variable, I used one-hot encoding. Integer encoding couldn't work because there were no ordinal relationships between the categories.

Fit the logistic regression model

First, I used the *glm()* function of R, with *family="binomial"* to fit a logistic regression model. The one-hot encoding strategy increased a lot the number of variables of our model and, because of the curse of dimensionality, the algorithm didn't converge.

To fix this issue, I used a shrinkage method, called **lasso** to perform both variable selection and regularization. The aim of this method is to improve the prediction accuracy as well as the interpretability of the model. Contrary to the ridge regression method, it does not necessary keep all the predictors in the final model. Thus, we can easily see which predictors don't have much importance in the prediction and, on the other side, the ones which play the biggest roles to predict the output.

This method is available with the R library *glmnet*. I used the function called *cv.glmnet*. I set α, the elasticnet parameter, to 1 (the lasso penalty) and *nfolds*, the cross-validation parameter to find the optimal λ, to 5.
Once the cross-validation for *glmnet* is done, we can predict the income level by using a specified λ. I chose to use the λ that minimizes the mean cross-validated error.

With this method, we can get the coefficients for each variable with *coef()*. The variables that are assigned to 0 are not used to construct the model. For the others, if the variables are standardized before fitting the model, we can compare the variable coefficients to evaluate their importance.

2.3 Performance comparison and choice of the model

To estimate the error rate of the two models, I used **cross-validation**.

In my case, I chose to implement a 10-Fold Cross-Validation. It means that we separate randomly our training data set into 10 different groups of approximately the same size. We use the first fold as a validation set and the nine others to fit the model. We compute an error rate that we will call MSE_i. We repeat this procedure $k-1$ using the $k-1$ other folds, one after one, as the validation set and we fit a new model.

At the end, we compute the CV Error Rate :

$$CV = \frac{1}{k}\sum_{i=1} kMSE_i$$

The results obtained with the two algorithms are the following :

Model	CV Error Rate	95% CI
Logistic regression	0.04798946	$[0.04730451, 0.04867440]$
Decision Tree	0.04864101	$[0.04777509, 0.04950693]$

According to these results, there isn't a significant difference between the two models. Indeed, the 95% confidence interval are overlapped.

Thus, I chose to keep the decision tree model for different reasons. First, there is less pre-processing to do with the data set. Then, as there is no one-hot encoding data, it is easier to interpret the results. The variables have kept their original names.

3 Application of the model on the test file

We will now train our decision tree model with all the training set.
We have to apply the same modification to the test data that we did to the training data to fit the model.

Error rate

We obtain a test error rate of **4.81%** with our decision tree model.

Less than 5% of error rate is a good result in absolute terms. Nevertheless, we have to be careful with this number. Indeed, there is a very low number of observations with the label "+50000". For the training set, 6.2058% of the population has this label and 6.2008% for the test set. It means that a classifier that would always predict the response "-50000" would have an error rate of around 6.20%.

Confusion Matrix

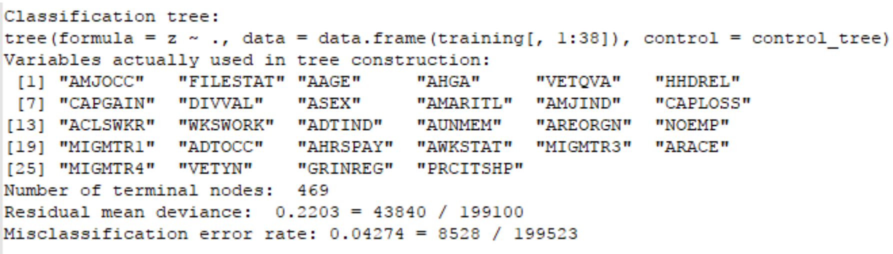
To see where the classifier made mistakes in its prediction, we can show the confusion matrix.

		Predicted	
		Positive	Negative
Observed	Positive	92289	1287
	Negative	3517	2669

We can also compute other indicators as the sensitivity, equal to 0.9633, and the specificity, equal to 0.6747.

Importance of predictors

With our library, we can get the variables that were used in the decision tree :

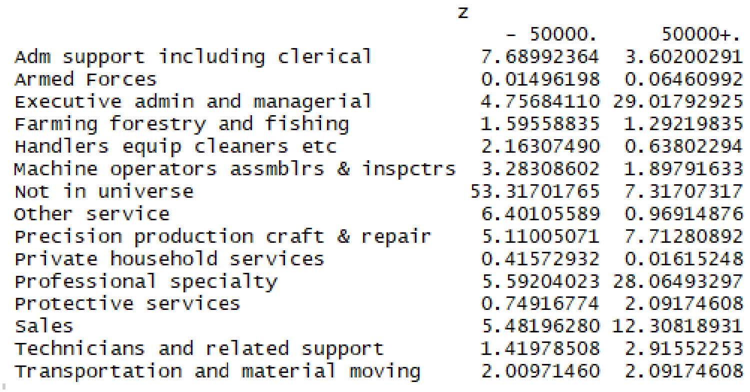


```
Classification tree:
tree(formula = z ~ ., data = data.frame(training[, 1:38]), control = control_tree)
Variables actually used in tree construction:
 [1] "AMJOCC"   "FILESTAT" "AAGE"     "AHGA"     "VETQVA"   "HHDREL"
 [7] "CAPGAIN"  "DIVVAL"   "ASEX"     "AMARITL"  "AMJIND"   "CAPLOSS"
[13] "ACLSWKR"  "WKSWORK"  "ADTIND"   "AUNMEM"   "AREORGN"  "NOEMP"
[19] "MIGMTR1"  "ADTOCC"   "AHRSPAY"  "AWKSTAT"  "MIGMTR3"  "ARACE"
[25] "MIGMTR4"  "VETYN"    "GRINREG"  "PRCITSHP"
Number of terminal nodes:  469
Residual mean deviance:  0.2203 = 43840 / 199100
Misclassification error rate: 0.04274 = 8528 / 199523
```

Using this result, we can do more in-depth analysis of some of these variables, especially the ones used at the top of the decision tree. It can help us to draw a profile of the people that make more than $50,000 per year.

1. Major Occupation Code

   This variable represents the field in which the person is working. The following table is a table that shows, for each level, the proportion of person in each category of the *AMJOCC* variable. I chose to show the proportion because the two levels of income don't have the same number of observations. In this way it is easier to compare both repartition.

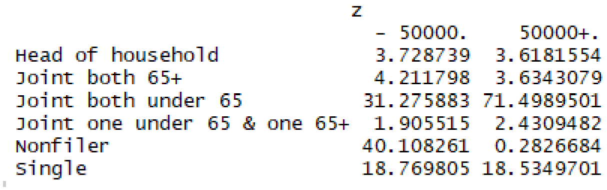


```
                                        z
                                           - 50000.     50000+.
Adm support including clerical          7.68992364  3.60200291
Armed Forces                            0.01496198  0.06460992
Executive admin and managerial          4.75684110 29.01792925
Farming forestry and fishing            1.59558835  1.29219835
Handlers equip cleaners etc             2.16307490  0.63802294
Machine operators assmblrs & inspctrs   3.28308602  1.89791633
Not in universe                        53.31701765  7.31707317
Other service                           6.40105589  0.96914876
Precision production craft & repair     5.11005071  7.71280892
Private household services              0.41572932  0.01615248
Professional specialty                  5.59204023 28.06493297
Protective services                     0.74916774  2.09174608
Sales                                   5.48196280 12.30818931
Technicians and related support         1.41978508  2.91552253
Transportation and material moving      2.00971460  2.09174608
```

   We can see that almost 70% of the "50000+" people are in three categories: Executive admin and managerial, Professional specialty and Sales.

2. Tax Filer Status

   We present the same type of table for this variable.

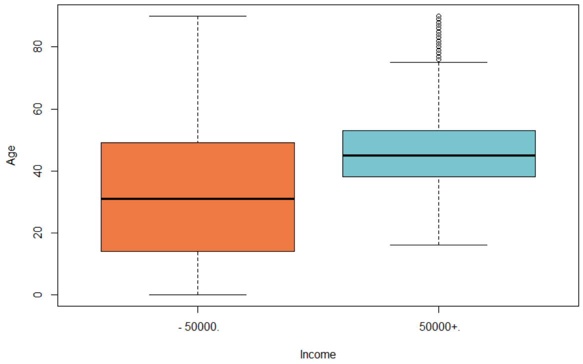


```
                                  z
                                     - 50000.    50000+.
Head of household                 3.728739  3.6181554
Joint both 65+                    4.211798  3.6343079
Joint both under 65              31.275883 71.4989501
Joint one under 65 & one 65+      1.905515  2.4309482
Nonfiler                         40.108261  0.2826684
Single                           18.769805 18.5349701
```

   90% of people that make more than $50000 per year are either "Joint both under 65" or "Single". However, there is the same proportion of single people in both categories of income level whereas the proportion of "Joint both under 65" is more than the double in the case of "+50000". On the other side, we see that it is really rare for a "Nonfiler" to earn more than $50000 a year.

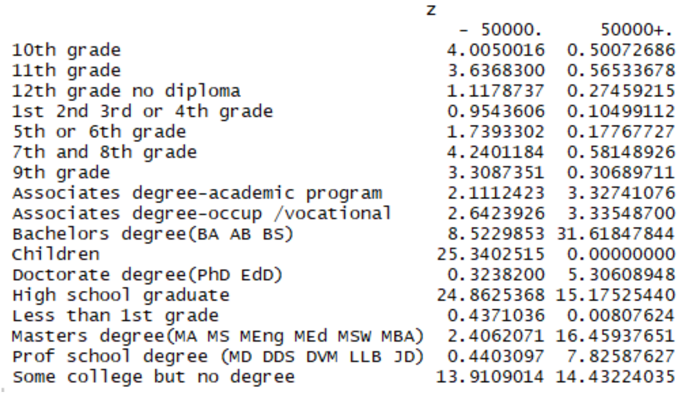
3. Age

   The variable *AAGE* is continuous so we can plot a boxplot for each income level.



We can clearly see that the mean of ”50000+” people is higher than the mean of the others. The age of 75% of the observations for this category is between 38 and 53 years old.

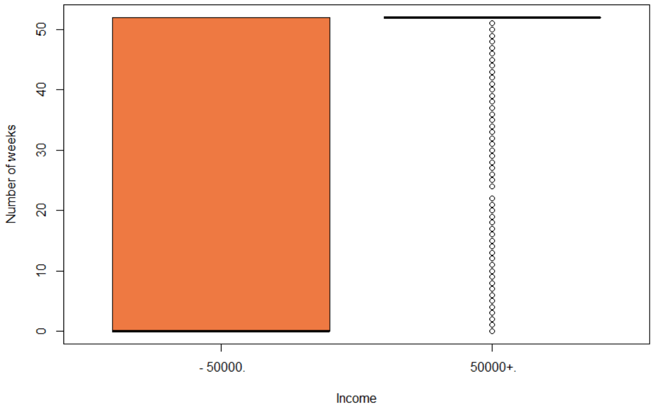
4. Education

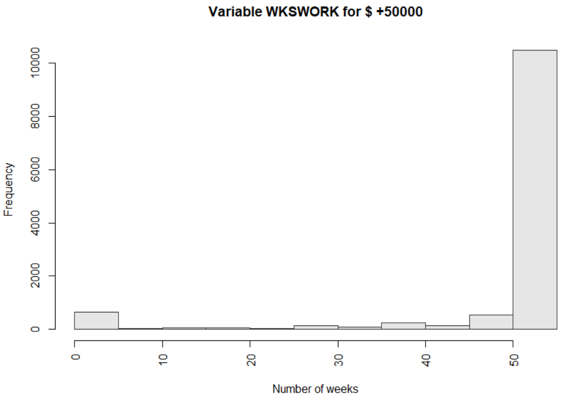


```
                                                z
                                                  - 50000.     50000+.
10th grade                                    4.0050016  0.50072686
11th grade                                    3.6368300  0.56533678
12th grade no diploma                         1.1178737  0.27459215
1st 2nd 3rd or 4th grade                      0.9543606  0.10499112
5th or 6th grade                              1.7393302  0.17767727
7th and 8th grade                             4.2401184  0.58148926
9th grade                                     3.3087351  0.30689711
Associates degree-academic program            2.1112423  3.32741076
Associates degree-occup /vocational           2.6423926  3.33548700
Bachelors degree(BA AB BS)                    8.5229853 31.61847844
Children                                     25.3402515  0.00000000
Doctorate degree(PhD EdD)                     0.3238200  5.30608948
High school graduate                         24.8625368 15.17525440
Less than 1st grade                           0.4371036  0.00807624
Masters degree(MA MS MEng MEd MSW MBA)        2.4062071 16.45937651
Prof school degree (MD DDS DVM LLB JD)        0.4403097  7.82587627
Some college but no degree                   13.9109014 14.43224035
```

The education level the most represented for the person that earn more than $50000/year is ”Bachelor degrees” (around 30%), followed by ”High School Graduate”, ”Master Degree” and ”Some college but no degree”.

5. Weeks worked per year





With the chart on the left, we can clearly see a difference of distribution between the two categories of income. If we look deeper at the second category ”50000+”, we notice that a large majority of these persons is working 52 weeks a year.

4 Conclusion

4.1 Challenging parts

I spent a lot of time exploring the data set. Indeed, to be able to implement a good model, you have to know your data. I went through all the variables, check their type, the missing values and I tried to anticipate which variables will have to be pre-processed to apply one of the algorithms. Moreover, I had to make sure that R was detecting the good type for all the variables. For example, the variables "ADTIND" and "ADTOCC" are quantitative variable (they represent a code) but they have a integer as value. R was detecting these variables as "int", that is false because there is no ordinal relationship between the different codes and it could lead to a bad model.

Then, I had to find good alternatives to quantitative variables to implement the logistic regression algorithm. Integer encoding was not a good solution, I chose consequently one-hot encoding alternative. Nevertheless, it increased a lot the number of predictors, which means more parameters to compute. I removed arbitrarily some variables that had a high number of categories and that were summarised in a way by other variables.

One other challenging part was to handle the curse of dimensionality for the logistic regression model. Indeed, the algorithm didn't converge at first. To reduce the number of variables, I though about different solutions as stepwise variable selection or dimension reduction. I finally chose the lasso method, which turned out to be an efficient solution for our problem.

4.2 Improvements or other ideas

One other method that could be explored is Random Forests. I tried to implement it quickly but there was an issue with the size of our data with R. However, it could be also a good method to study deeper for the purpose of the exercise.

To reduce the number of variables, I could have used also the Principal Component Analysis procedure. This is a dimension reduction method. The inconvenient with this method (compared to the lasso) is that we don't used the initial variables anymore to build the model and, thus, the interpretation of the importance of the initial predictors is not possible.