Income Prediction. Classification Predictive Modeling

by Anupama r.k, Queenie Tsang, Crystal (Yunan) Zhu

12/02/2021

Abstract

Income Classifier is an application developed for ABC Handbags LLC to classify target population for a marketing campaign. The demographic information of the population is obtained from Adult dataset. The application is build using R and shinyapp following a CRISP-DM framework.

Background

Our client ABC Handbags LLC is looking to open a new retail location exclusively for luxury handbags at XYZ square in New York City. To market the store's handbags, the client wants to target customers with an annual income above 50K in New York City. The dataset of people living in New York City available to us has incomplete income data. To identify customers with income above 50K, the client wants us to predict the income data for customers using the available data points.

Analytical Objective

The objective of the project is to develop a supervised learning model and evaluate the effectiveness in predicting income above 50K USD. The model will classify income above/below 50K output against each input demographic item.Supervised learning model will be fitted for algorithms like Logistic Regression, Random Forest and KNN. The most accurate and precise model will chosen for deployment.

Assumptions and Success Criteria

1. Marketing team has sufficient demographic information on the population that the model needs to work with reasonable accuracy.
2. The dataset is an accurate representation of the current demographic to be used as a train dataset.
3. Attributes like age, work hours,job type, education are more significant than other attributes in the dataset
4. Achieving a classification accuracy above 80% and precision of 60% is ideal.

Ethics & Privacy

The attributes used in the application has no significant sensitivity. If any sensitivity outcomes occur from usage of the application may be contextual. Personally Identifiable Information is not collected nor is it an output of the application. Some of the attributes are anonymized to a single level. The hosting environment has implemented security best practices. The data and application logic is not exposed in the application front. Authentication is not currently set for the deployed application.

Data Understanding

The data comes from UCI Machine Learning repository for Adult Dataset. The owner of the dataset is Ronny Kohavi and Barry Becker.The dataset is based on 1994 U.S Census dataset. The dataset has 16 attributes and 32561 records.An individual's annual income is dependent on various factors. Some of these factors

include individual's education level, age, gender, occupation, and etc. Adult dataset is an ideal representation for our income prediction model. The target variable has two classes, values '>50K' and '<=50K', meaning it is a binary classification task.

Reformulate a problem statement as an analytics problem

Our client is looking to open a business in a new location. The client is looking to open a store that sells products of one of their luxury brands. The luxury brand is trying to target people with income of above 50K. The current business problem we are trying to solve is how to predict the income of a given customer into 2 classes: less than or equal to $50 thousand USD, or greater than $50 thousand USD. This is a business problem, because given some demographic information such as age, sex, education, marital status, occupation, we want to be able to predict the customer's income into the =<50K category or >50K category.

If we can predict this income accurately, the company can use this information to determine whether they should allocate resources to market some premium grade products to the customer. The marketing team can use this tool to find the audience for our marketing pitch in anticipation of the branch opening and improve targeted advertising to people who have income above 50K. The tool allows a true/false output against each demographic item.

Develop a proposed set of drivers and relationships to inputs

The output function is the prediction of income, and whether it belongs to the =< $50K class or to the >$50K class. The input variables are the age, sex, occupation, workclass, education level, education number, relationship, marital status, final weight(refering to the weight of that demographic class within the current population survey), the capital gain, capital loss, hours per week (of work) and the native country.
- How does age affect the income class of a customer? - How does education level affect the income class of a customer? - What types of occupation is associated with income greater than $50K or with income less than or equal to $50K?

State the set of assumptions related to the problem

One assumption related to this problem is that the relationships between the input variables (such as age, occupation, workclass, marital status) to the target variable income obtained through the 1994 census data will hold true to what is observed today in 2021.

Define key metrics of success

One key metric of success is that the prediction model can accurately predict the income class, given the input information.

Describe how you have applied the ethical ML framework

One of the considerations of the Machine Learning Ethics framework is how machine learning models have the potential to propagate biases through feedback loops. Our model aims to predict the income category for customers based on certain demographic information collected in the 1994 US census, and does have some potential to negatively impact individuals. For example, if the model predicts people of certain work class, race, gender, age or native country has a greater probability to have income less than $50 thousand, this information may perpetuate historic biases against people of that demographic. Based on the income predictions produced by this tool, companies may treat different groups of people differently, in terms of marketing resources targeted towards different groups of people. In terms of privacy, all the data in the census dataset is anonymous and location data is not available, which helps prevent identification of any individual who contributed to the census dataset. The Income Prediction Shiny App tool does not collect any personal information of any individual so privacy is the default. It also does not track users who use the tool so privacy is embedded by design. User inputs to the Income Prediction tool is not retained.

Identify and prioritize means of data acquisition

The means of data acquisition is through downloading the US census adult data set.

Define and prepare your target variables. Use proper variable representations (int, float, one-hot, etc.).

The target variable is income.

Modeling and Evaluation

Describe the data

Data Dictionary

```
## The dimension of the dataset is 32561 by 15 .
```

There are 32,561 records and 15 columns in the original data set.

There are 6 numeric and 9 categorical variables shown as follows:

Column Name	Data Type	Column Description
age	Integer	The age of the adult (e.g., 39, 50, 38, etc.)
workclass	Factor	The work class of the adult (e.g., Private, Self-emp-not-inc, Federal-gov, etc.)
fnl_wgt	Integer	The weights on the Current Population Survey (CPS) files are controlled to independent estimates of the civilian noninstitutional population of the US (e.g., 77516, 83311, etc.)
education	Factor	The education of the adult (e.g., Bachelors, Some-college, 10th, etc.)
education_num	Integer	The number years of the adult's education (e.g., 13, 9, 7, etc.)
marital_status	Factor	The marital status of the adult (e.g., Divorced, Never-married, Separated, etc.)
occupation	Factor	The occupation of the adult (e.g., Tech-support, Craft-repair, Sales, etc.)
relationship	Factor	The relationship of the adult in a family (e.g., Wife, Own-child, Husband, etc.)
race	Factor	The race of the adult (e.g., White, Asian-Pac-Islander, Amer-Indian-Eskimo, etc.)
sex	Factor	The gender of the adult.(Female, Male)
capital_gain	Integer	The capital gain of the adult (e.g., 0, 2174, 14084, etc.)
capital_loss	Integer	The capital loss of the adult (e.g., 0, 1408,2042, etc.)
hours_per_week	Integer	The number of working hours each week for the adult (e.g. 40, 13, 16, etc.)

Column Name	Data Type	Column Description
native_country	Factor	The native country of the adult (e.g. Cambodia, Canada, Mexico, etc.)
income	Factor	The yearly income of the adult at 2 levels: <=50K and >50K.

Data Description

First, let's check whether there are duplicates in the dataset.

```
## The number of duplicated records in the dataset is 24 .
```

For the benefit of this report's length, let's look at a sample of duplicated records:

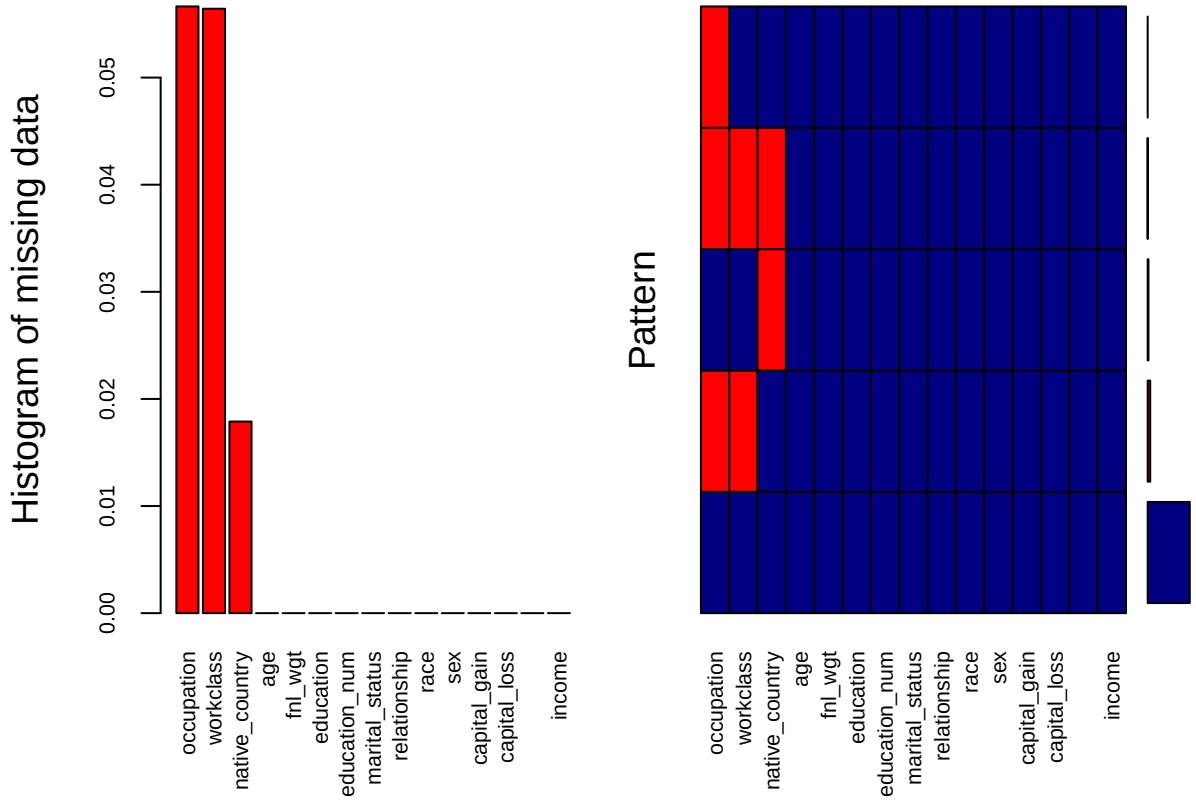
	age	workclass	fnl_wgt	education	education_num	marital_status	occupation	relationship
4768	21	Private	250051	Some-college	10	Never-married	Prof-specialty	Own-child
9172	21	Private	250051	Some-college	10	Never-married	Prof-specialty	Own-child
4326	25	Private	308144	Bachelors	13	Never-married	Craft-repair	Not-in-family
4882	25	Private	308144	Bachelors	13	Never-married	Craft-repair	Not-in-family

	race	sex	capital_gain	capital_loss	hours_per_week	native_country	income
4768	White	Female	0	0	10	United-States	<=50K
9172	White	Female	0	0	10	United-States	<=50K
4326	White	Male	0	0	40	Mexico	<=50K
4882	White	Male	0	0	40	Mexico	<=50K

The 24 duplicated rows will be removed from all later analysis.

Then let's check whether there are any missing values in the dataset.

```
## Warning in plot.aggr(res, ...): not enough horizontal space to display
## frequencies
```



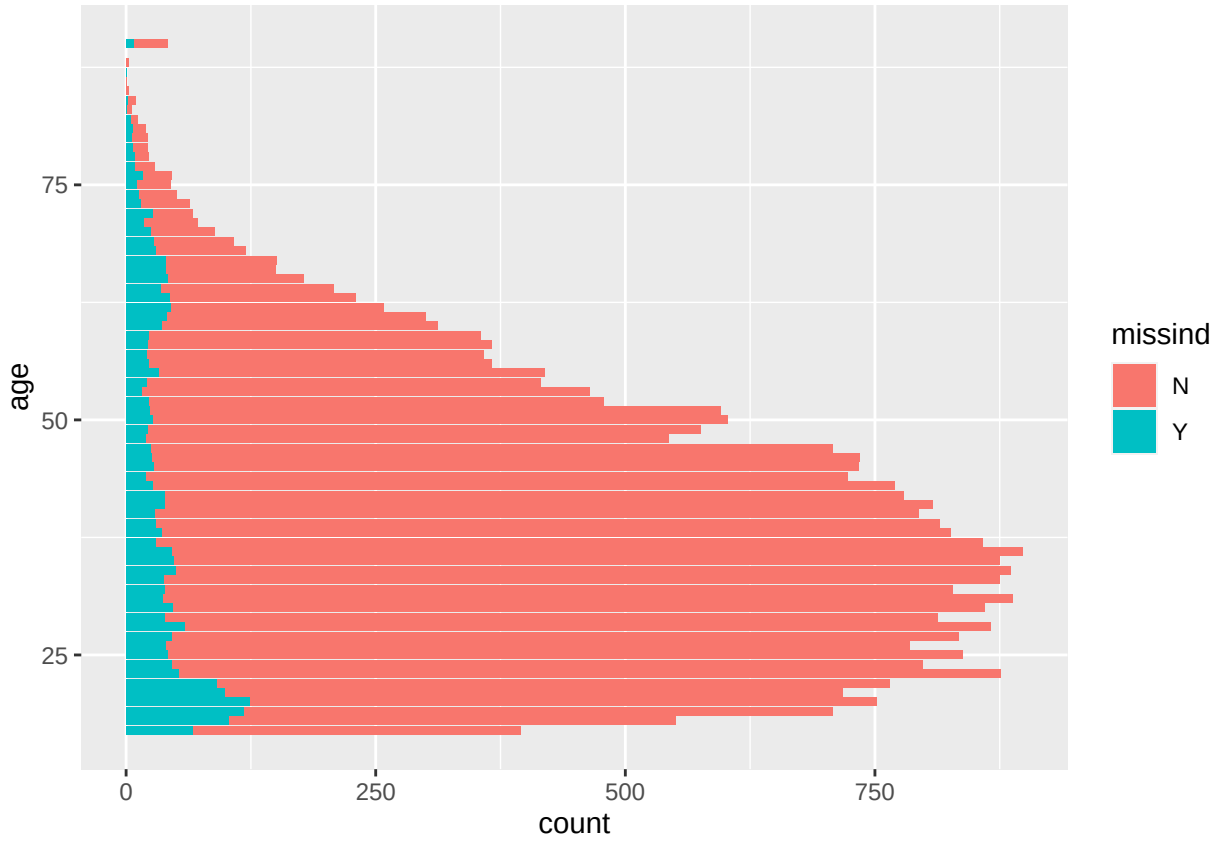

```
##
##  Variables sorted by number of missings:
##        Variable      Count
##      occupation 0.05664321
##       workclass 0.05642807
##  native_country 0.01788733
##             age 0.00000000
##         fnl_wgt 0.00000000
##       education 0.00000000
##   education_num 0.00000000
##  marital_status 0.00000000
##    relationship 0.00000000
##            race 0.00000000
##             sex 0.00000000
##    capital_gain 0.00000000
##    capital_loss 0.00000000
##  hours_per_week 0.00000000
##          income 0.00000000
```

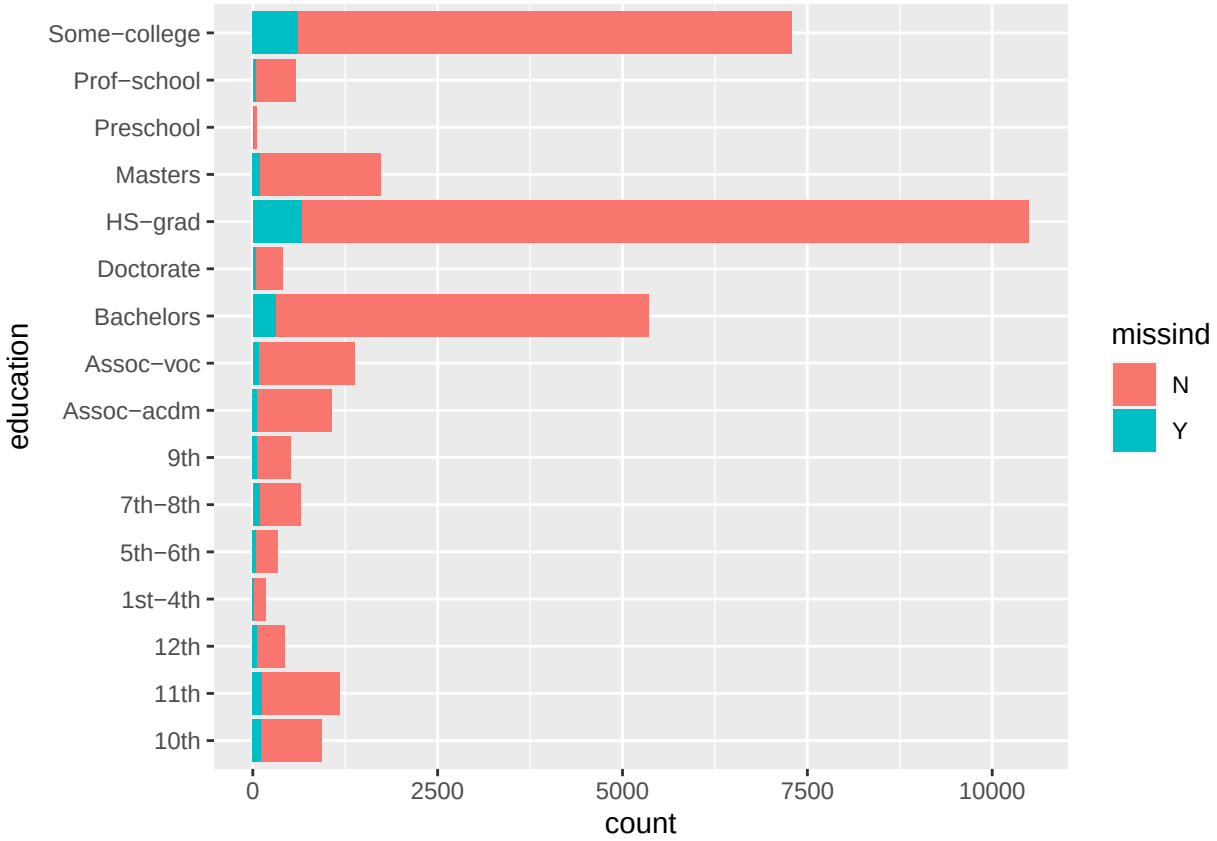
Note: we didn't find a way to remove the warning message and accommodate the entire plot in the page due to the limitation of time.

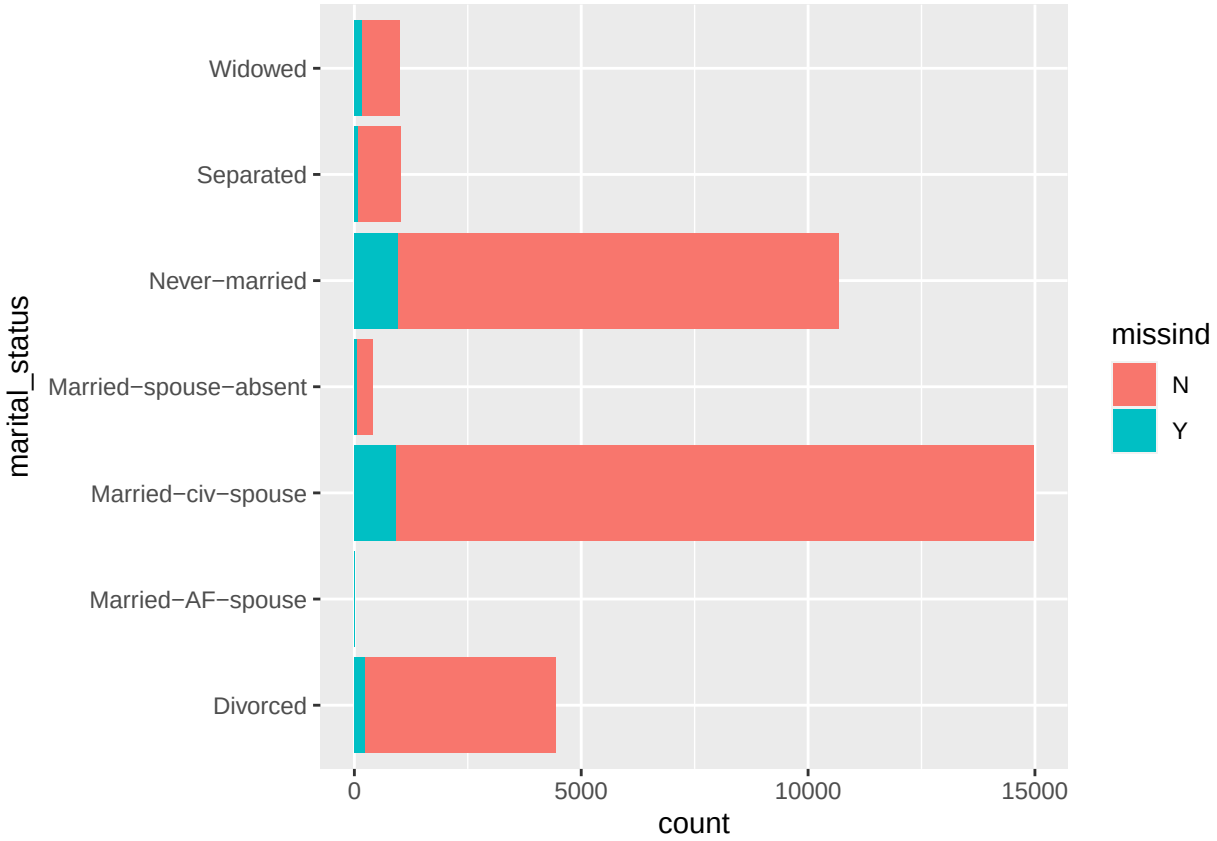
From the above, there are missing values in this data set and all the missing values are from categorical variables.

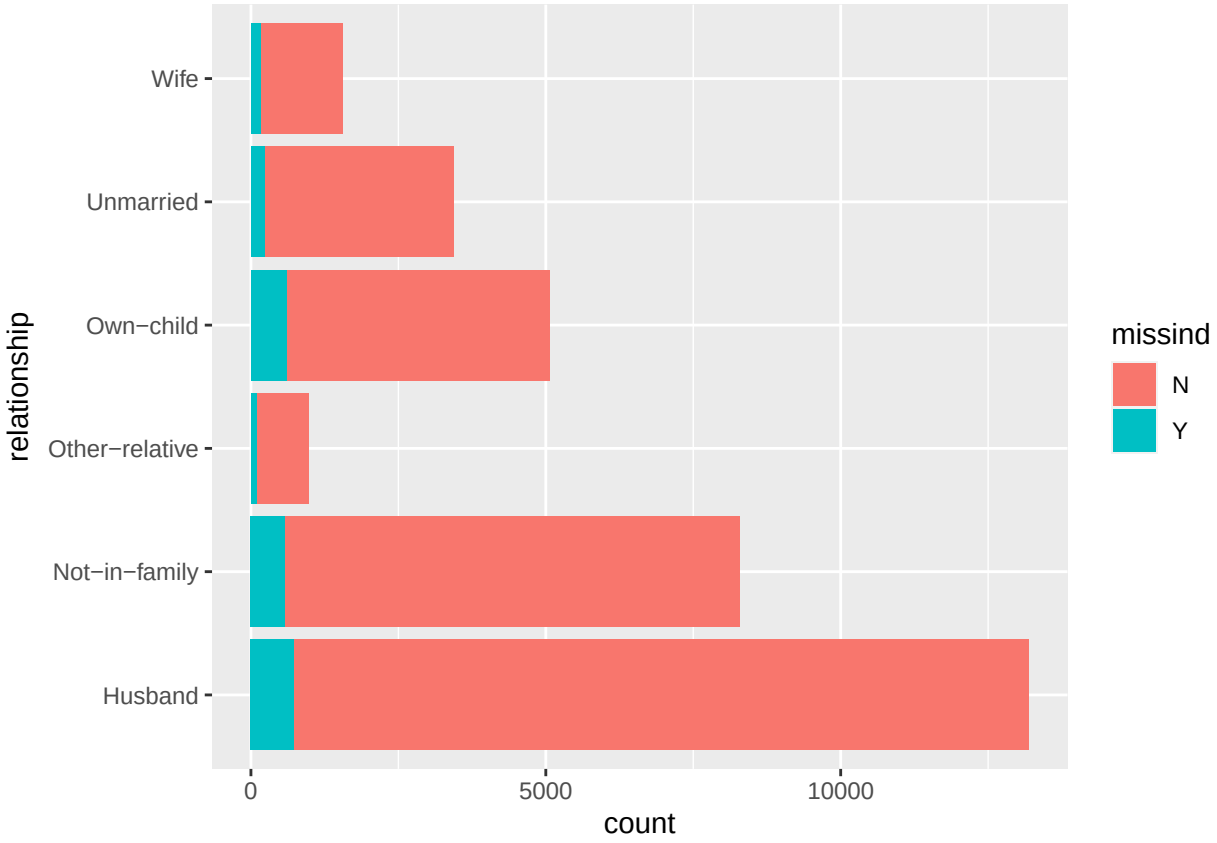
Comparing records with at least one missing value to those without any missing values.

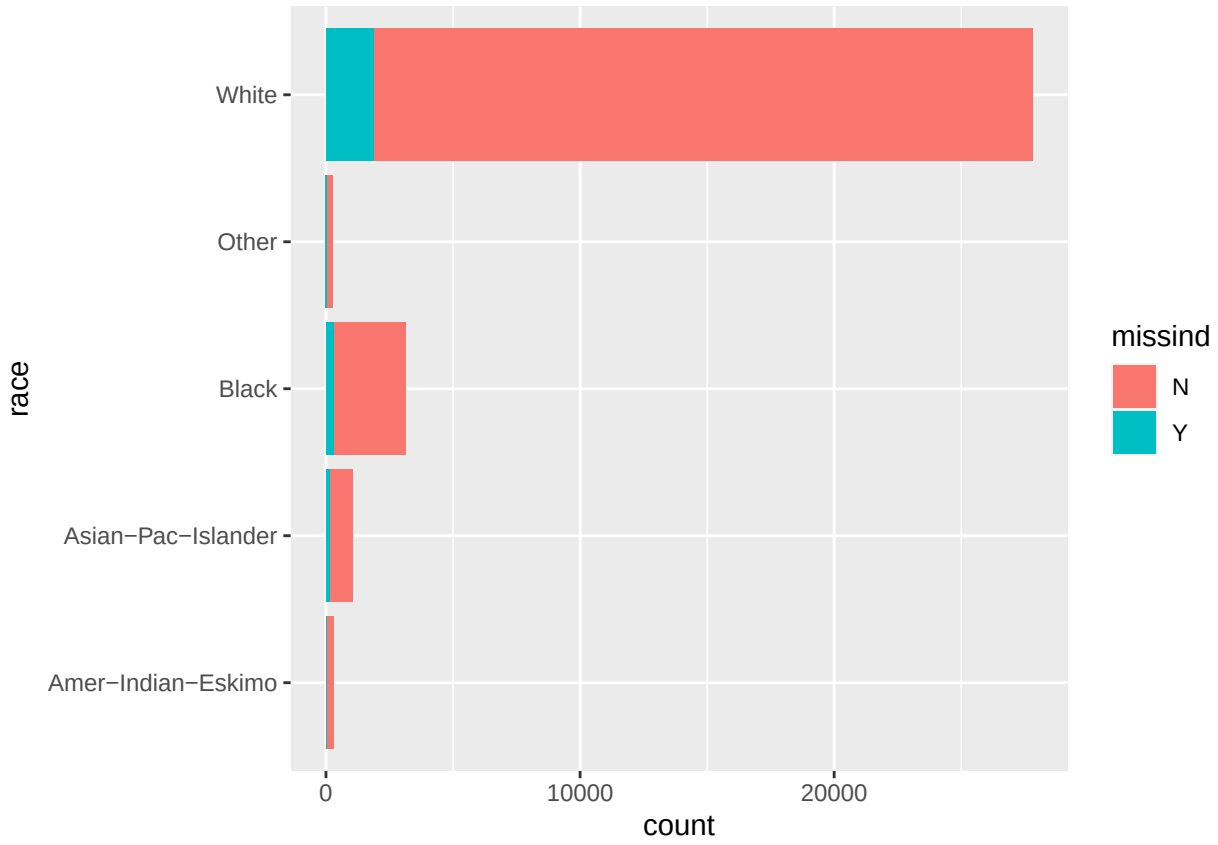
In order to better understand the patterns of the missing values, let's look at some descriptions of the records with missing values.

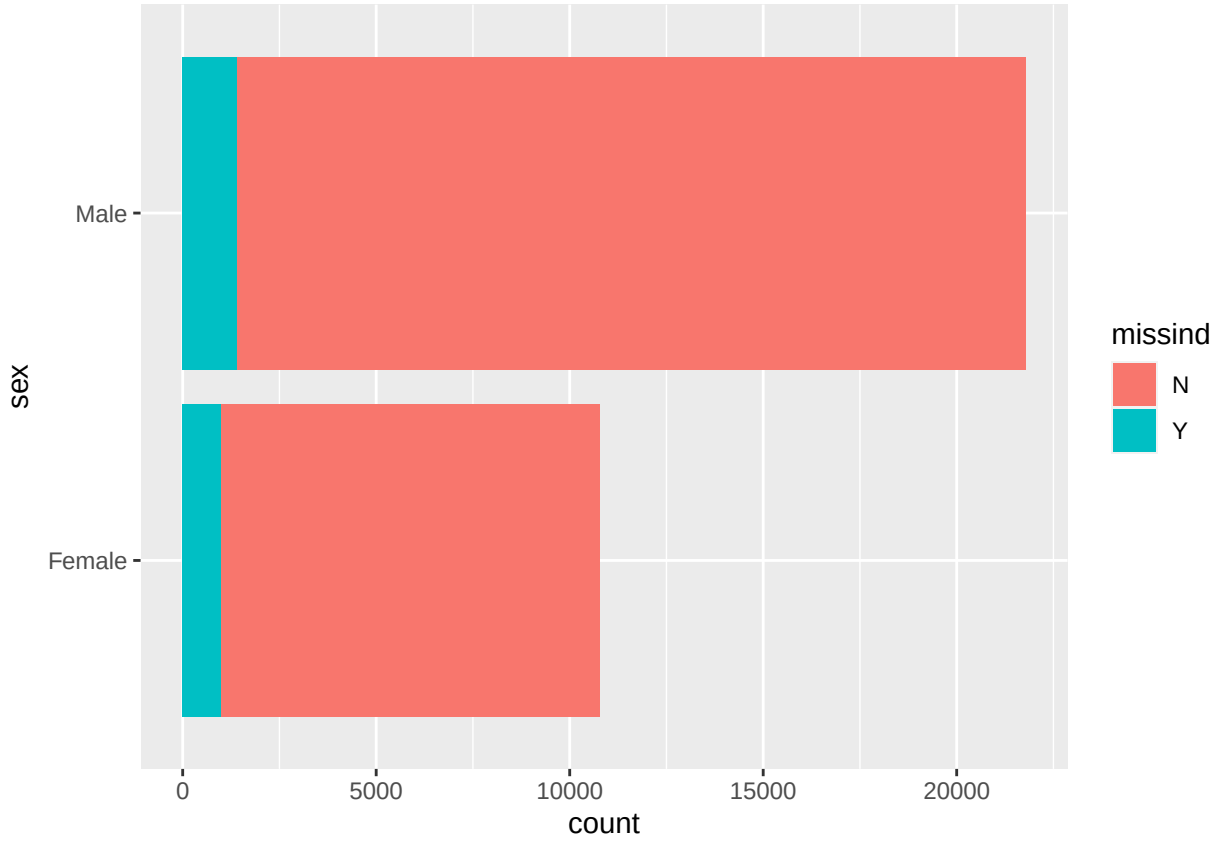


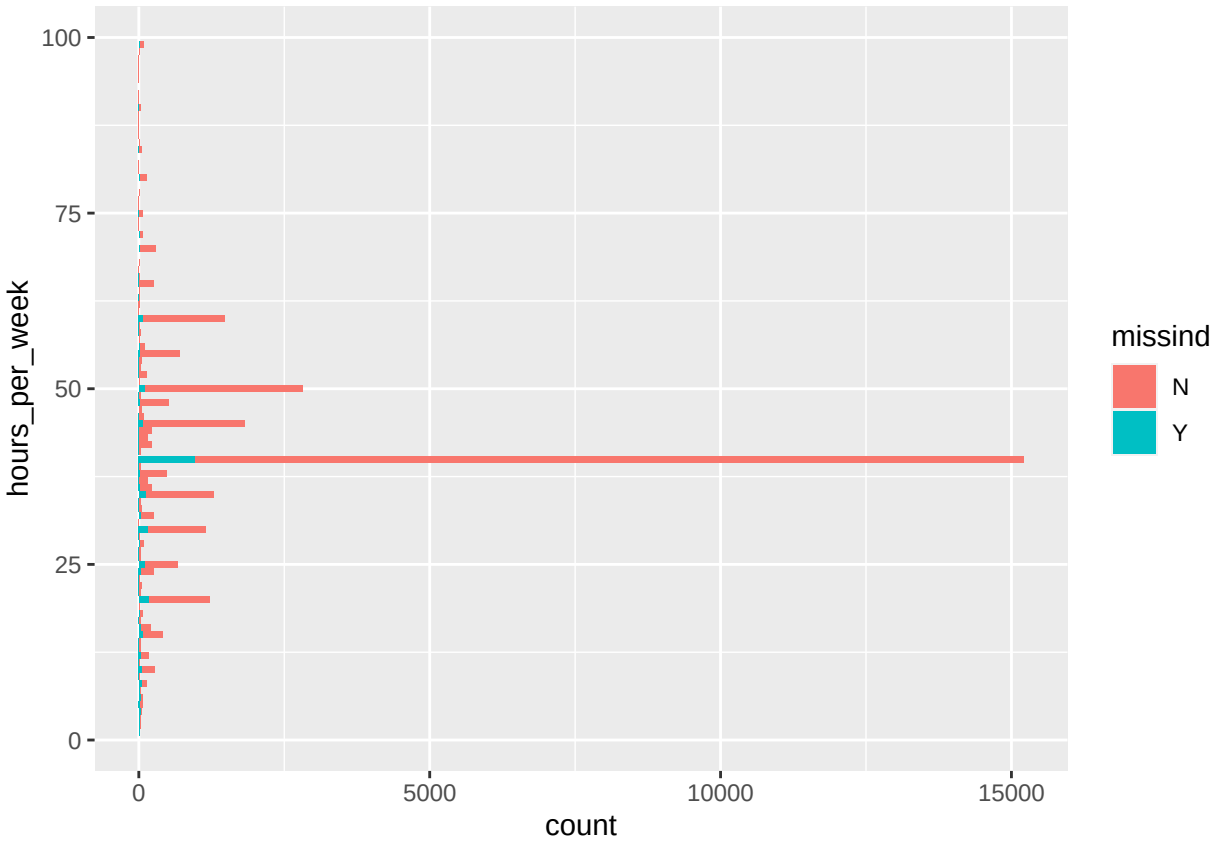

education
Some−college
Prof−school
Preschool
Masters
HS−grad
Doctorate
Bachelors
Assoc−voc
Assoc−acdm
9th
7th−8th
5th−6th
1st−4th
12th
11th
10th
0
2500
5000
7500
10000
count
missind
N
Y

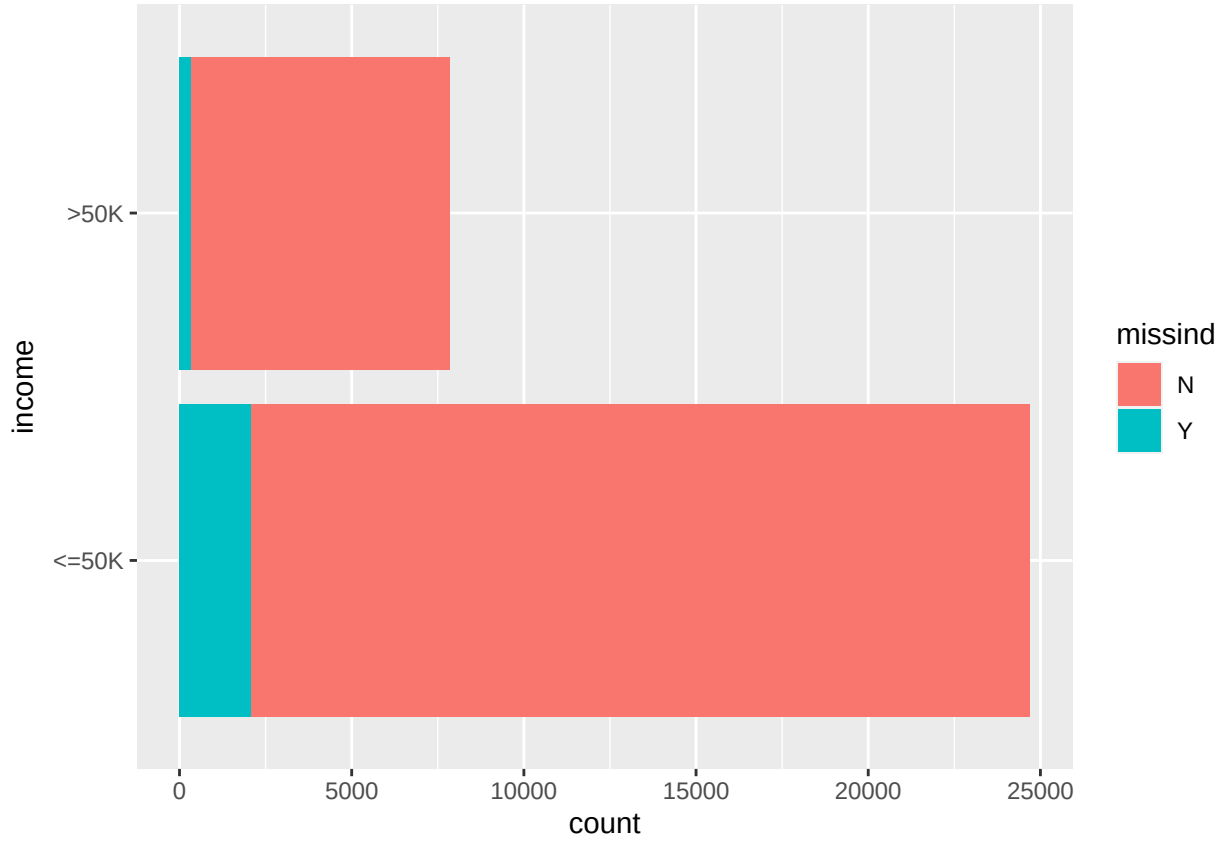

marital_status
Widowed
Separated
Never−married
Married−spouse−absent
Married−civ−spouse
Married−AF−spouse
Divorced
0
5000
10000
15000
count
missind
N
Y


relationship
Wife
Unmarried
Own−child
Other−relative
Not−in−family
Husband
0
5000
10000
count
missind
N
Y


White
Other
Black
Asian−Pac−Islander
Amer−Indian−Eskimo
race
0
10000
20000
count
missind
N
Y


Male
Female
sex
0
5000
10000
15000
20000
count
missind
N
Y


hours_per_week
100
75
50
25
0
0
5000
10000
15000
count
missind
N
Y



From the above bar charts comparing the distributions of 7 variables of the group that do not have missing values and the group that have at least one missing record, we can see that the missing records are generally evenly distributed across all ages, education level, marital status, family relationship, race, working hours per week and the target variable income. When compared with the whole population in the census, the percentages of people not answering all the survey questions are slightly lower in the age group between 20-50, Married civ spouse marital status and husbands, and 60-70 years old and never-married people tend to omit some questions to answer. Males tend to have fewer missing records than females.

Since the proportion of missing values is relatively small (7%) where we would still have 30K records left, and it's generally the same for people with income higher and lower than 50K USD, we think it would be reasonable to remove the records with missing values for our analysis in this report. If we had more time, we'd recommend fitting models separately for female and male since they have different willingness to answer occupation, work class or native country related questions, which could be strong predictors for adult income.

Remove rows with missing values

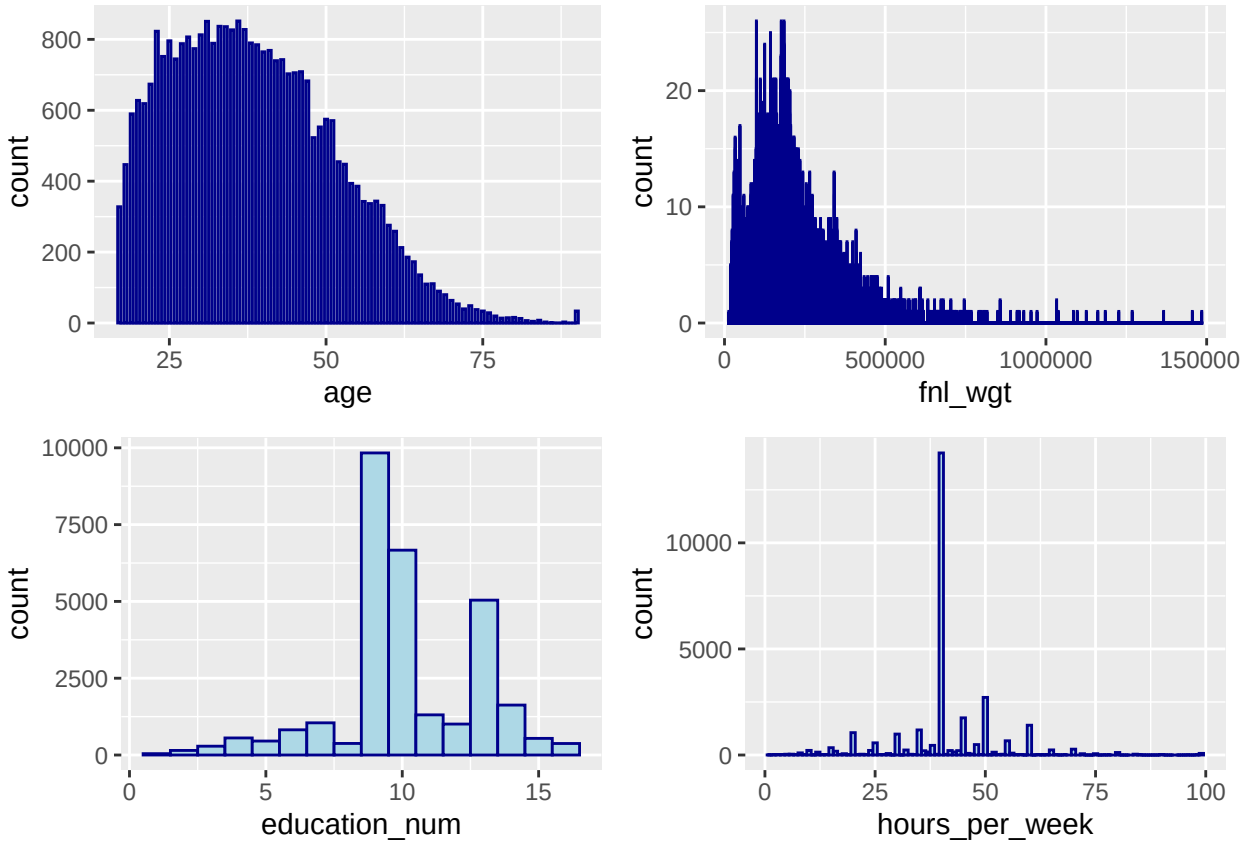
```
## Now the dimension of the dataset becomes:
```

```
## [1] 30139    16
```

Now let's view the summary of the 6 numeric columns:

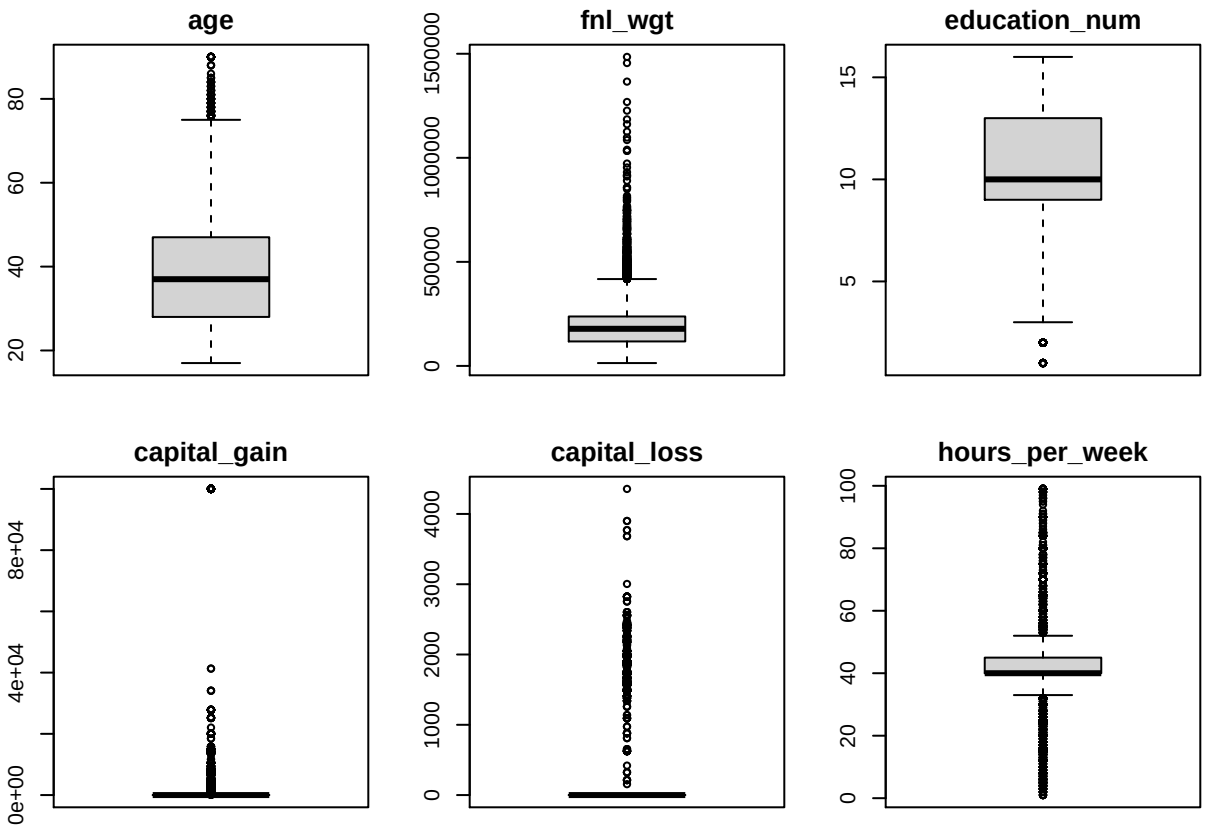
	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
age	17.00	28.00	37.00	38.44	47.00	90.00
fnl_wgt	13769.00	117627.50	178417.00	189795.03	237604.50	1484705.00
education_num	1.00	9.00	10.00	10.12	13.00	16.00
capital_gain	0.00	0.00	0.00	1092.84	0.00	99999.00
capital_loss	0.00	0.00	0.00	88.44	0.00	4356.00
hours_per_week	1.00	40.00	40.00	40.93	45.00	99.00

Let's take a clearer look at the numeric values by visualizing their distributions using histograms, except for capital gain and capital loss.



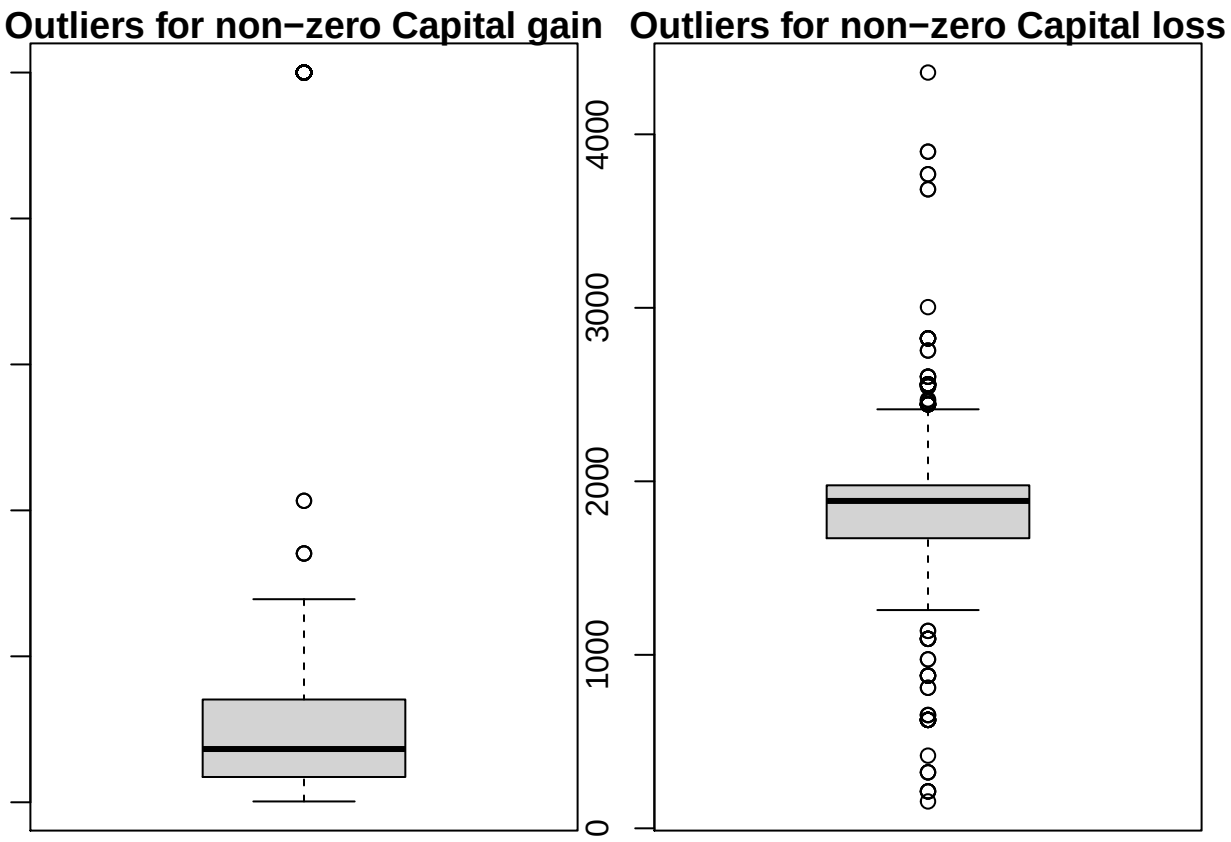
Both age and fnl_wght variables are skewed to the right, where log-normal transformation could help if modeling techniques with normality assumptions were to be used. The working hours per week variable is heavy-tailed distributed, meaning there are still quite a number of people working extremely small or large number of hours each week.

Boxplots to discover outliers for each numeric variable.



There are outliers for all numeric variables. There are large amount of records for ages and fnl_wght that are larger than their upper quantiles, which are consistent with the histograms showing their distributions are skewed to the right.

Since there are large number of zeros in capital_gain & capital_loss variables, let's check if there are still outliers for non-zero values.

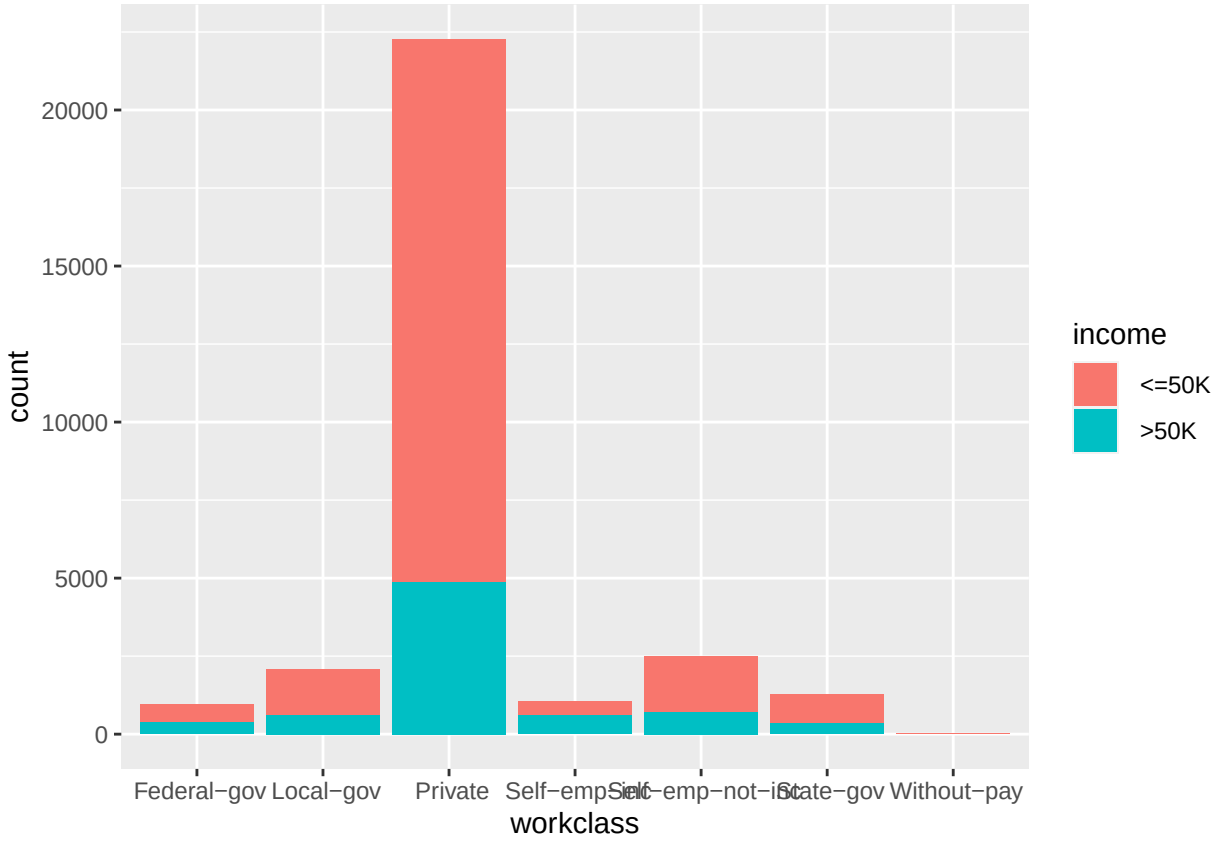


We can see there are still outliers even excluding zeros for capital gain and capital loss variables.

Distributions of categorical variables by target variable.

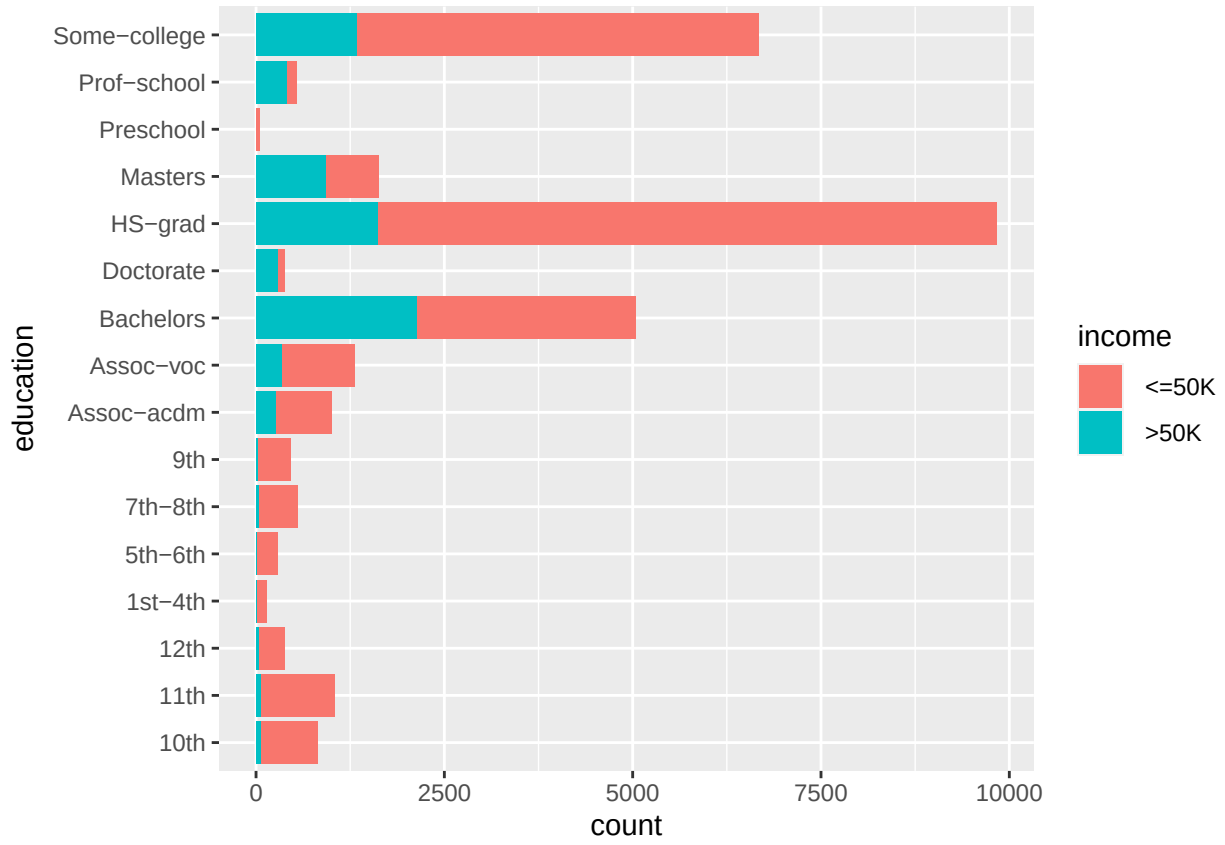
```
par(mar=c(1,1,1,1))

ggplot(X, aes(workclass, fill = income)) + geom_bar()
```



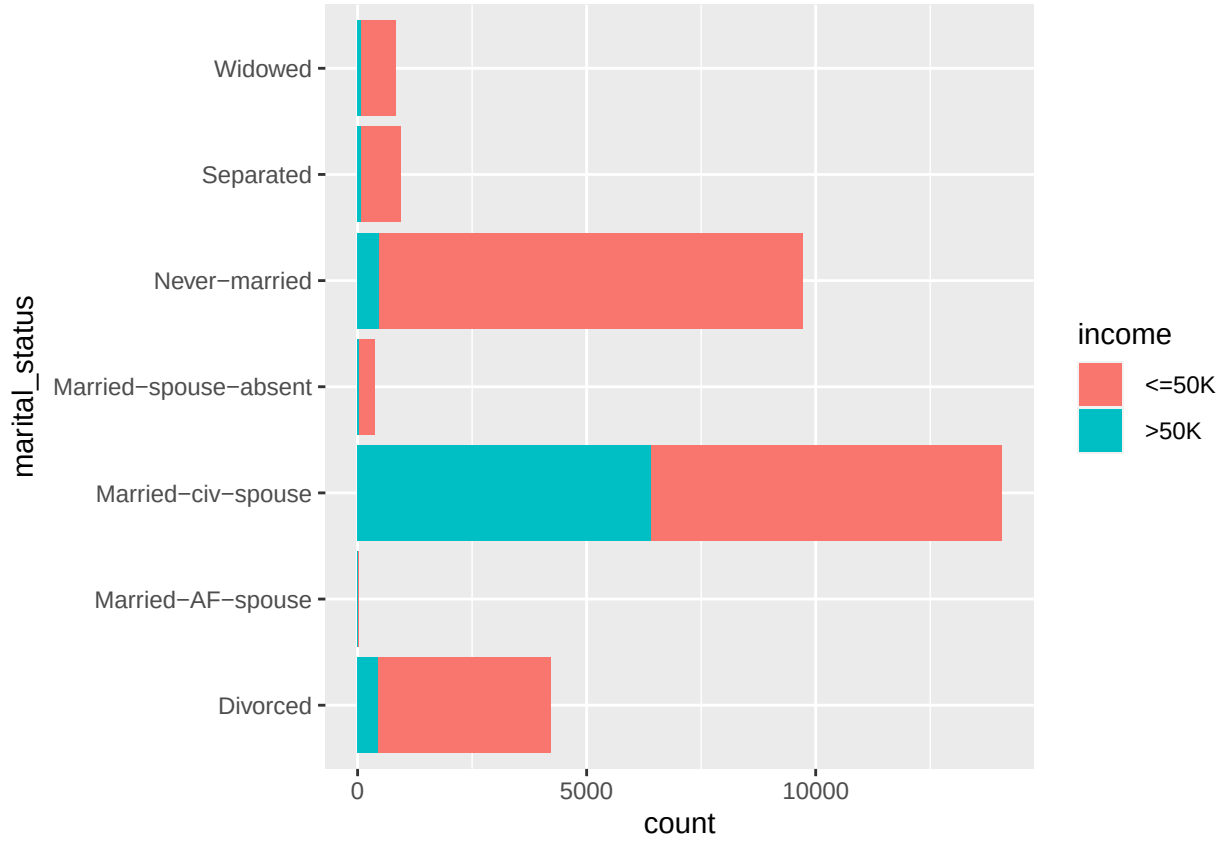
From the above bar chart we can see the majority of adults in the census were working in private sectors.

```
#plotting education vs income
ggplot(data = X, aes(y = education, fill = income)) +
  geom_bar(position = "stack")
```



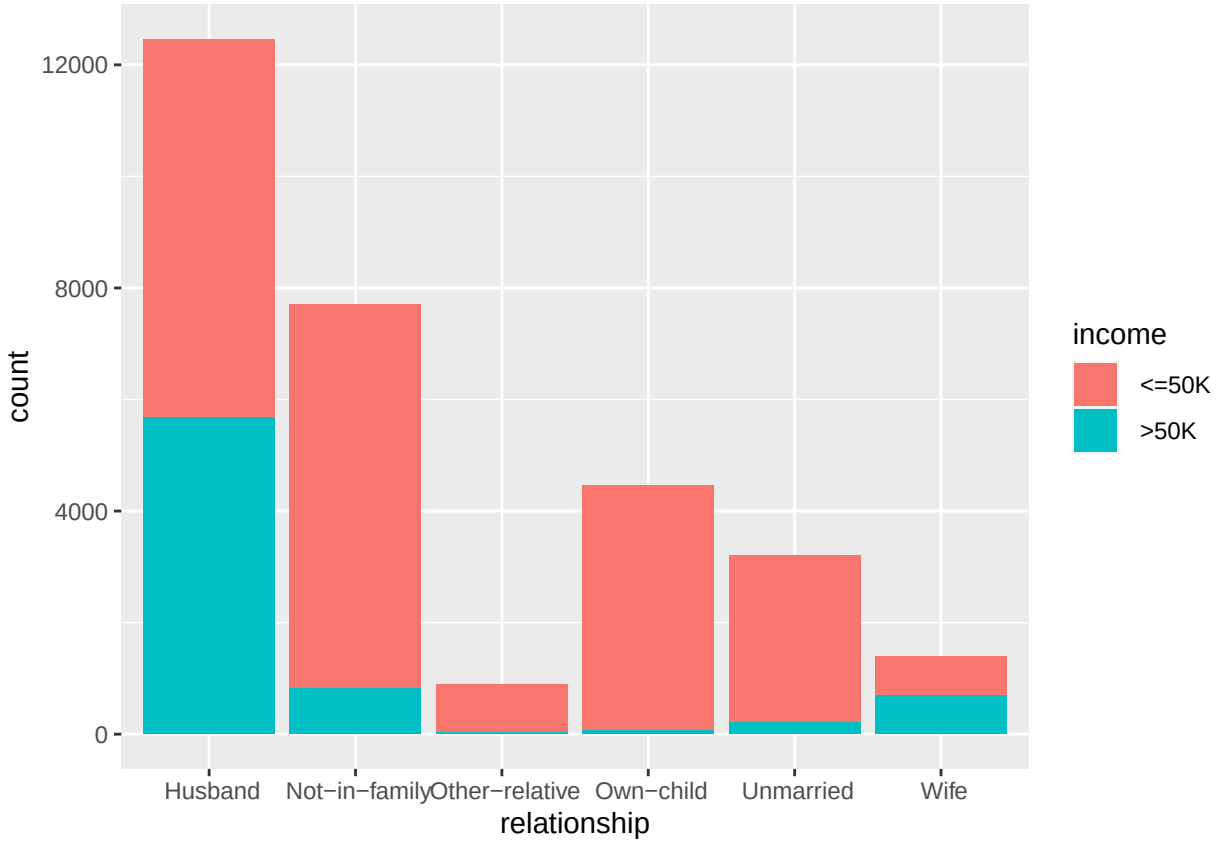
The majority of people earning less than $50K are high school graduates. The next largest education group is some college, and the third largest education group is Bachelors.

```
#plotting marital status vs. income
ggplot(data = X, aes(y = marital_status, fill = income))+
  geom_bar(position = "stack")
```



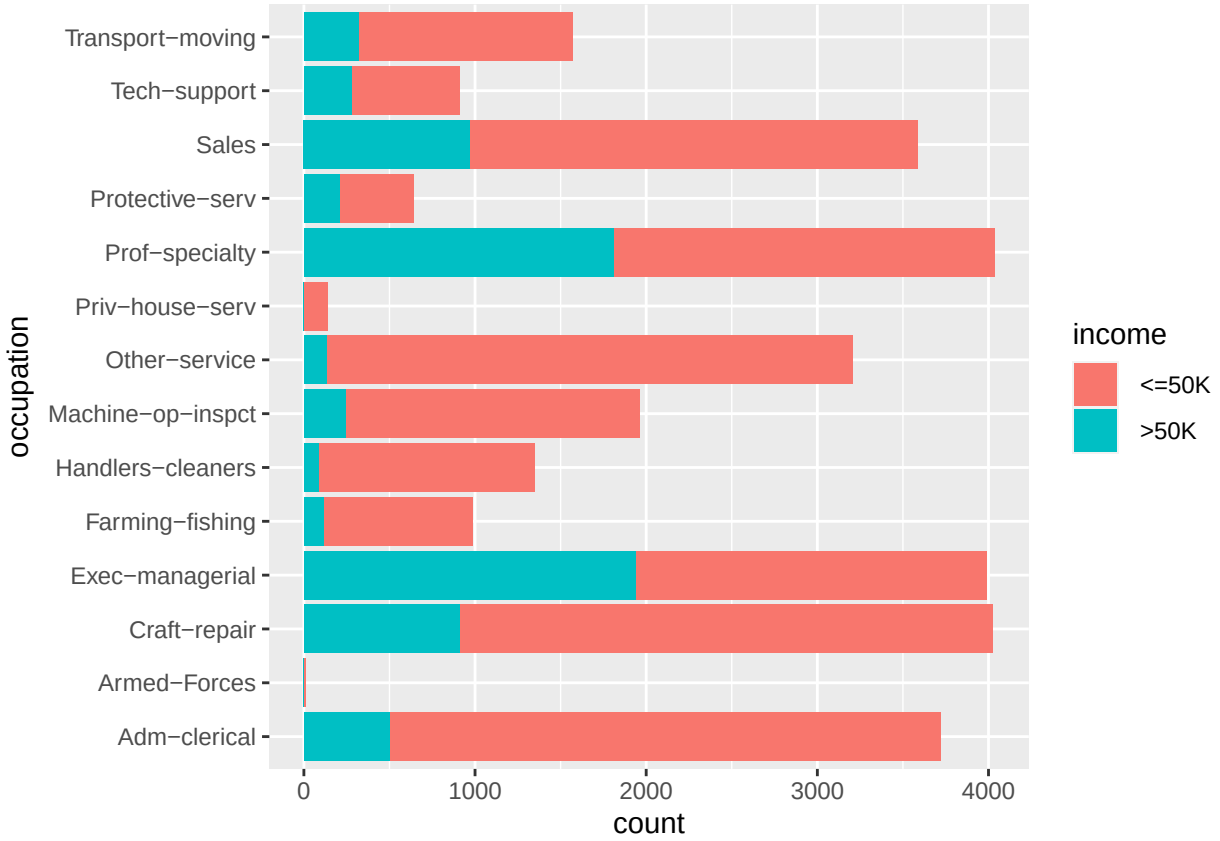
The majority of people surveyed are Married civ spouse, and in this marital status category, the income is roughly equally divided between <=50K or >50K. The second largest category is Never-married, with the majority of people earning <=50K.

```
ggplot(X, aes(relationship, fill = income)) + geom_bar()
```



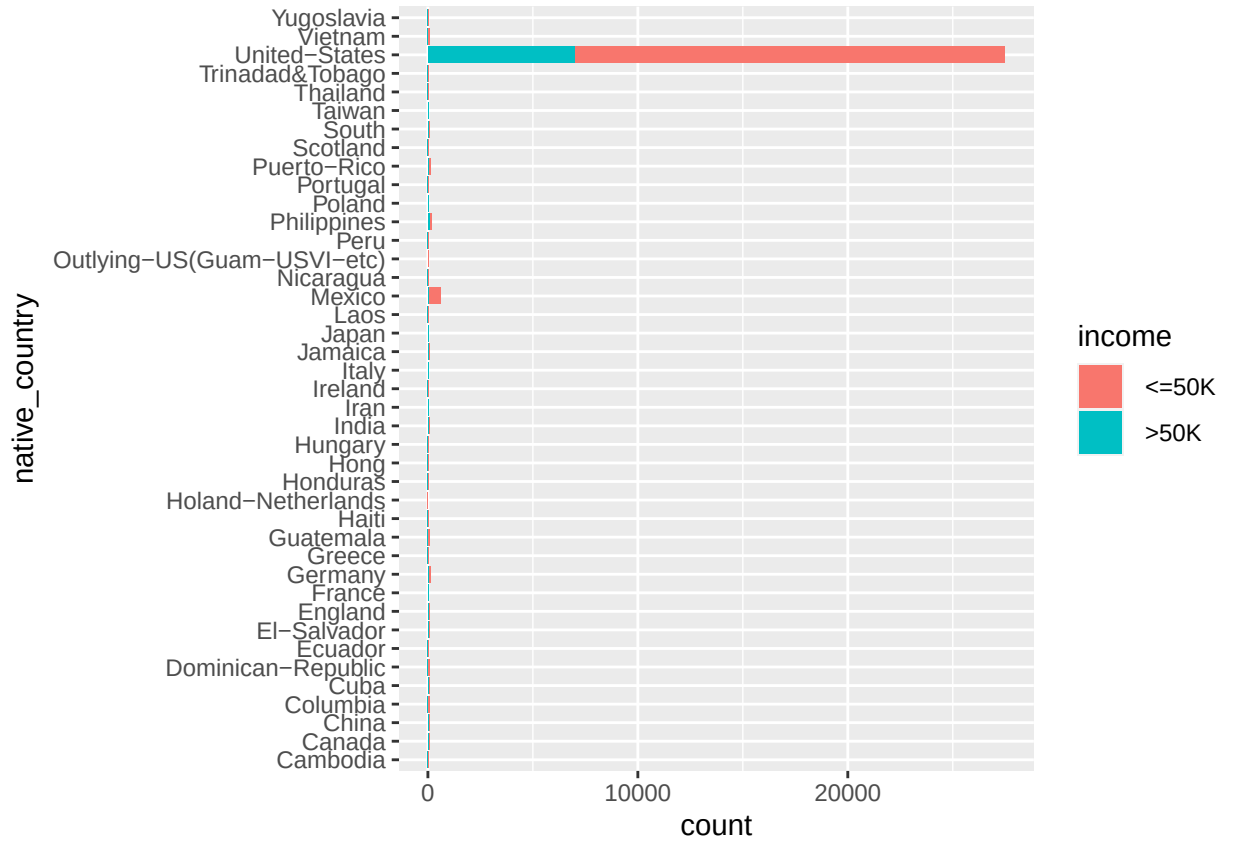
Most people surveyed in the census belong to the Husband category of relationships, with slightly more people earning less than or equal to 50K. However, in the Husband category, there is almost an even split between the 2 target income classes. Not-in-family is the second largest category for relationships and the majority people in this category have income <=50K.

```
#plotting occupation vs income
ggplot(data = X, aes(y = occupation, fill = income))+
  geom_bar(position = "stack")
```



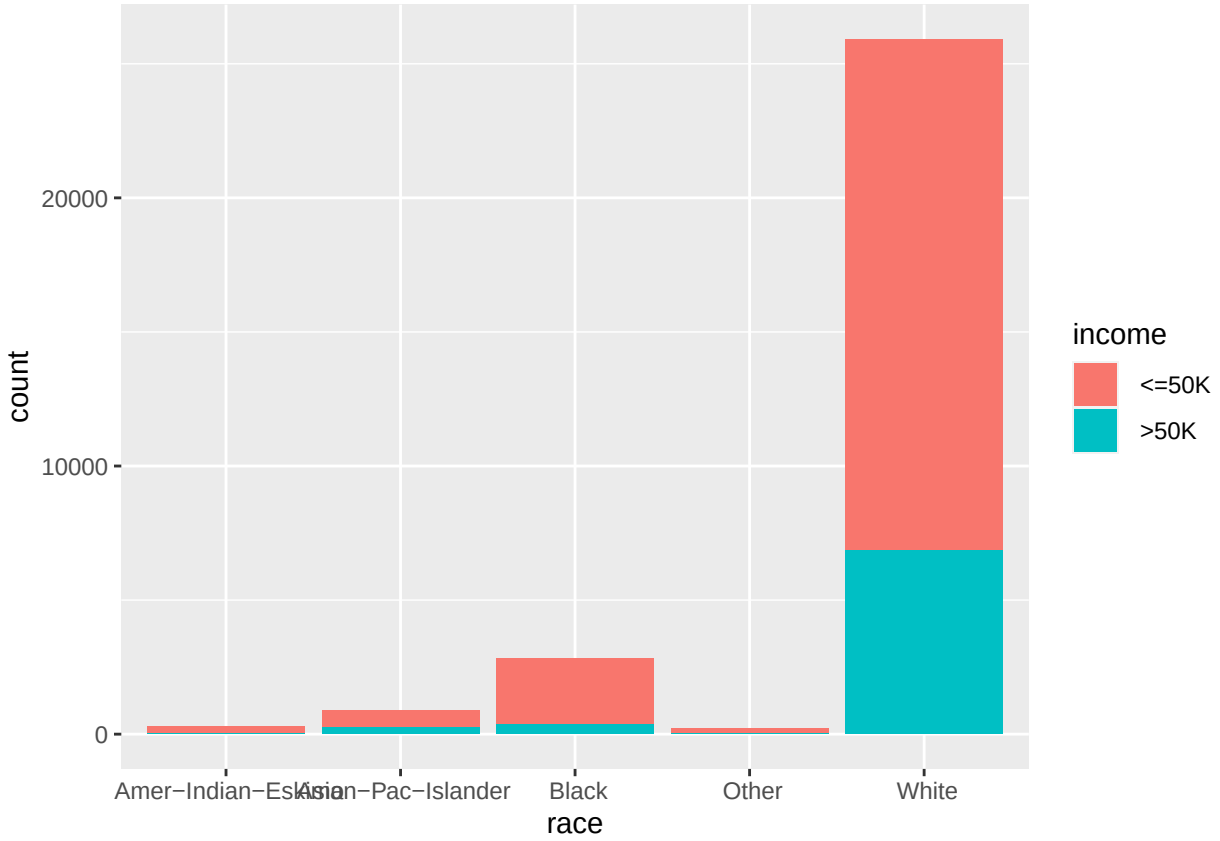
Most common occupations are Prof-specialty, Exec-managerial, Craft-repair, Sales, and Adm-clerical. For Exec-managerial, and Prof-specialty, there is an even number of people earning <=50K and >50K. For Craft-repair, Adm-clerical, and Sales, the majority of people earn <=50K.

```
#plotting native country vs. income
ggplot(data = X, aes(y = native_country, fill = income))+
  geom_bar(position = "stack")
```



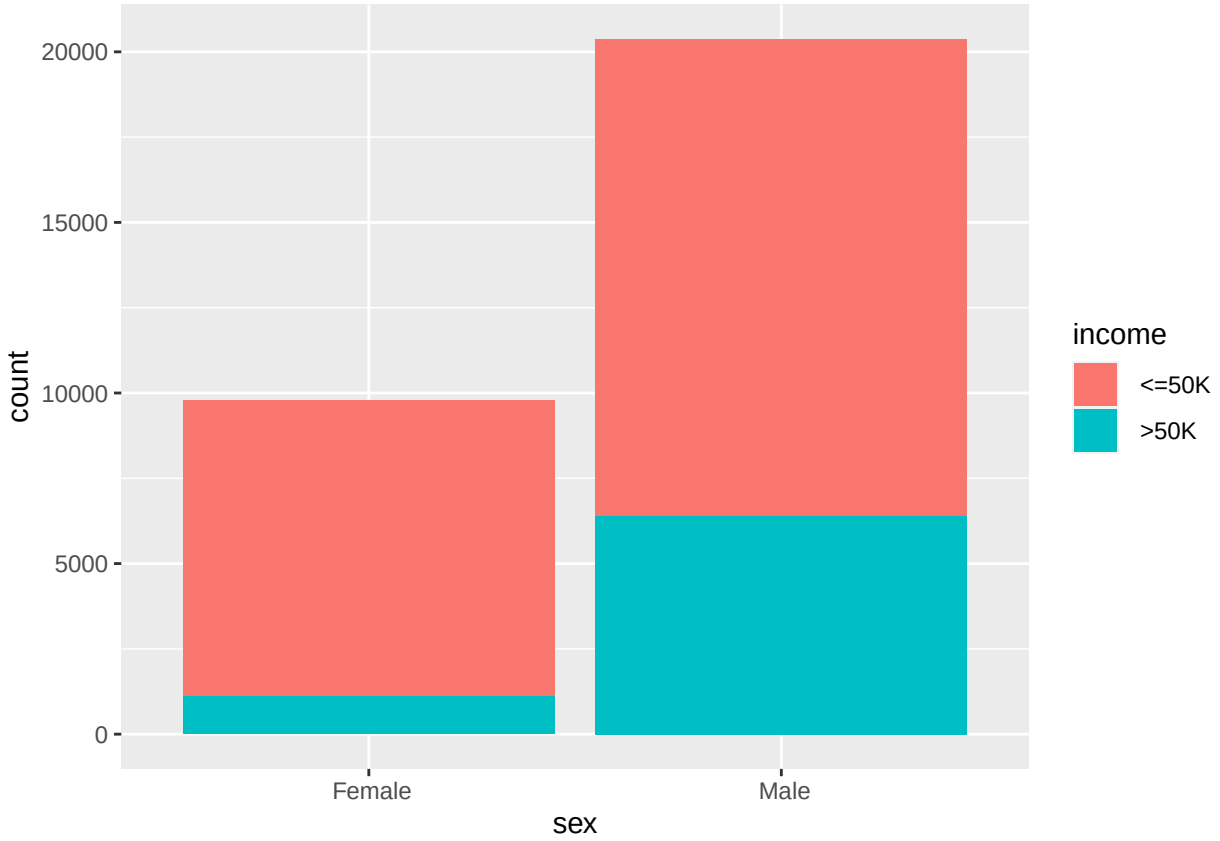
Most people surveyed come from the United States. This makes sense as the census was conducted in the US. Other than the United States, the second highest number of people come from Mexico.

```
ggplot(X, aes(race, fill = income)) + geom_bar()
```



Most people surveyed are White, and earn <=50K. The second highest race category is Black.

```
ggplot(X, aes(sex, fill = income)) + geom_bar()
```

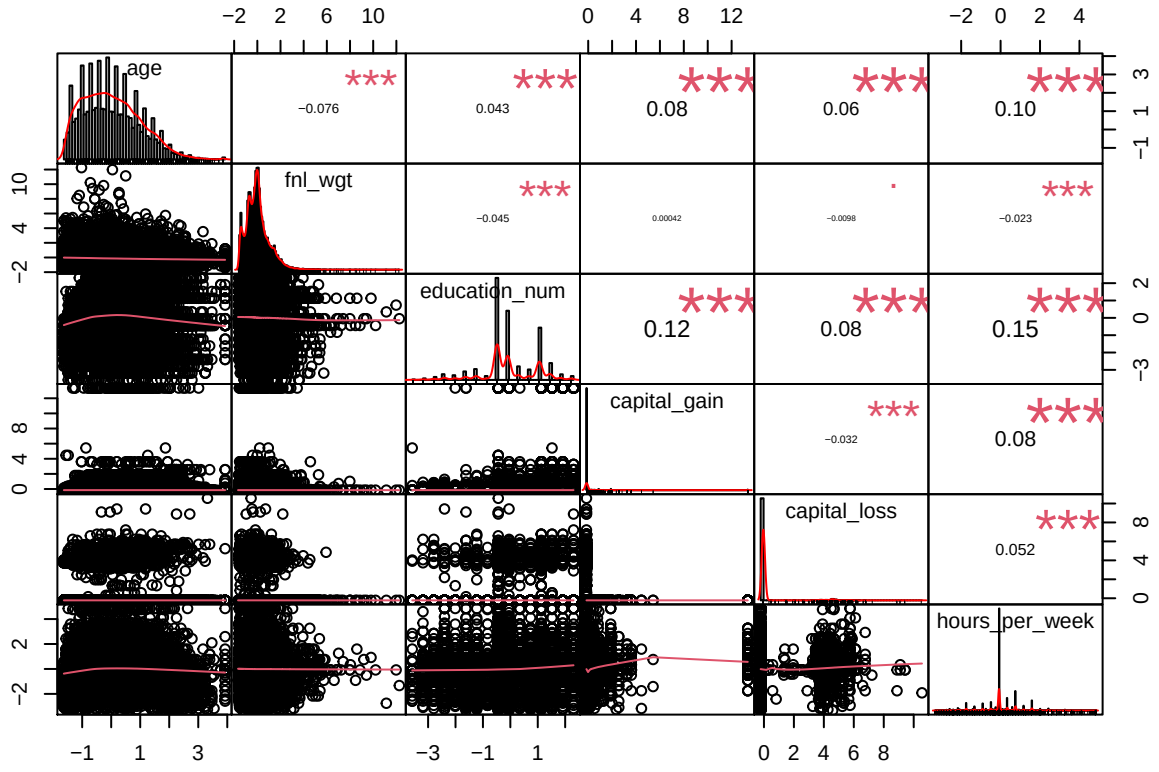


There are more than twice as many males surveyed in this census compared to females.

Explore relationships between attributes

Pearson's correlation between numeric variables.

```
#Display the chart of a correlation matrix
library(PerformanceAnalytics)
numindex=datatype=="integer"
chart.Correlation(scale(X[,numindex]), histogram=TRUE, pch=19)
```



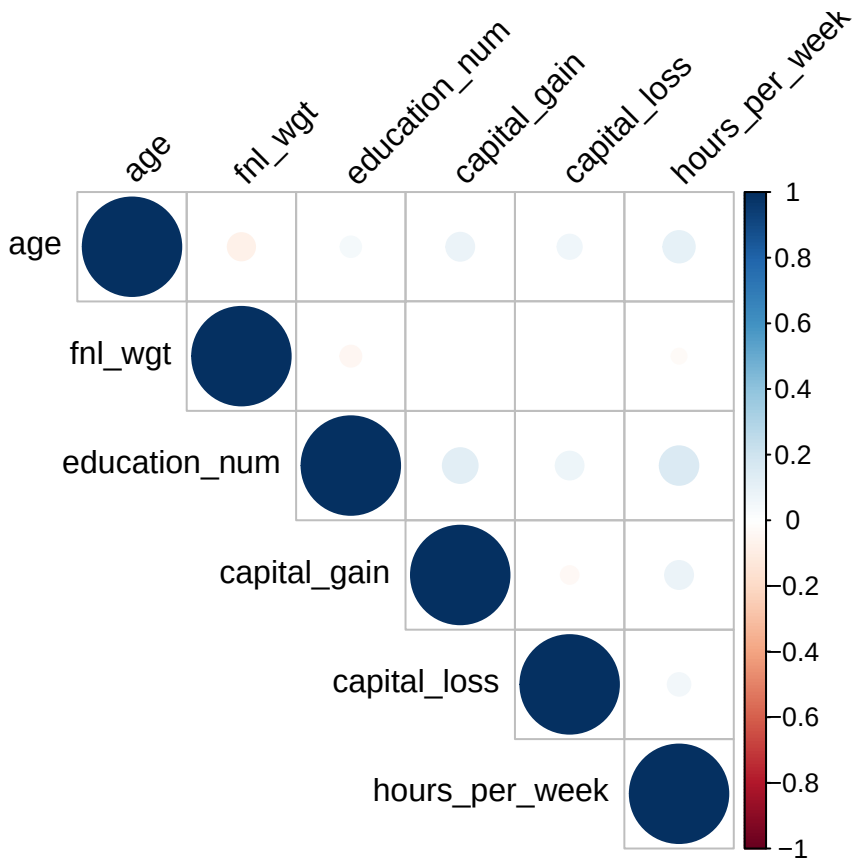
From the chart of the correlation matrix, we can see that while the magnitude of the correlations are small, all of them are statistically significant. For example, age and education_num are significantly correlated. Fnl_wgt and education num are also significantly correlated.

The following correlogram confirms that the correlations between numeric variables are very small, while they are significantly different from zero, probably due to large sample size.

```
library("Hmisc")

cormat <- rcorr(as.matrix(X[,numindex]))

#Draw a correlogram
library(corrplot)
corrplot(cormat$r, type = "upper",
         tl.col = "black", tl.srt = 45, p.mat = cormat$P, sig.level = 0.01, insig = "blank")
```



Chi-square test & Cramer's V to show associations between categorical variables

The test statistics from Chiq-Square Test between each pair of the categorical variables.

	workclass	education	marital_status	occupation	relationship	race	sex	native_country	income
workclass		2445.29	1720534.10	9314.40	9314.40	14712.97	4214.70	12388.39	357.31
education			127730.72	2182.09	2182.09	1078.26	8534.11	1198.45	398.14
marital_status				321142.94	321142.94	137835.19	269434.63	115466.27	111620.41
occupation					452085.00	1364.78	15299.12	2088.27	689.73
relationship						1364.78	15299.12	2088.27	689.73
race							3154.45	35767.89	844.68
sex								4814.17	846.78
native_country									1136.52
income									

The p-values from the Chi-Square Test between each pair of the categorical variables.

	workclass	education	marital_status	occupation	relationship	race	sex	native_country	income
workclass		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
education			0.00	0.00	0.00	0.00	0.00	0.00	0.00
marital_status				0.00	0.00	0.00	0.00	0.00	0.00
occupation					0.00	0.00	0.00	0.00	0.00
relationship						0.00	0.00	0.00	0.00
race							0.00	0.00	0.00
sex								0.00	0.00
native_country									0.00
income									

The Cramer's V statistics between each pair of categorical variables to measure their associations.

	workclass	education	marital_status	occupation	relationship	race	sex	native_country	income
workclass			0.90	0.14	0.14	0.29	0.10	0.29	0.05
education									
marital_status				0.84	0.84	0.87	0.83	0.88	0.96
occupation					1.00	0.09	0.20	0.12	0.08
relationship						0.09	0.20	0.12	0.08
race							0.13	0.49	0.08
sex								0.18	0.08
native_country									0.10
income									

From the Cramer's V, we can see that occupation and family relationship, marital_status and income are two of the pairs that have strong associations.

Feature Engineering

From the above exploratory analysis on the numeric and categorical variables, we think the following transformations can be adopted to build good-quality predictive models.

1.Education and education number are redundant.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
10th	0	0	0	0	0	820	0	0	0	0	0	0	0	0	0	0
11th	0	0	0	0	0	0	1048	0	0	0	0	0	0	0	0	0
12th	0	0	0	0	0	0	0	377	0	0	0	0	0	0	0	0
1st-4th	0	149	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5th-6th	0	0	287	0	0	0	0	0	0	0	0	0	0	0	0	0
7th-8th	0	0	0	556	0	0	0	0	0	0	0	0	0	0	0	0
9th	0	0	0	0	455	0	0	0	0	0	0	0	0	0	0	0
Assoc-acdm	0	0	0	0	0	0	0	0	0	0	0	1008	0	0	0	0
Assoc-voc	0	0	0	0	0	0	0	0	0	0	1307	0	0	0	0	0
Bachelors	0	0	0	0	0	0	0	0	0	0	0	0	5042	0	0	0
Doctorate	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	375
HS-grad	0	0	0	0	0	0	0	0	9834	0	0	0	0	0	0	0
Masters	0	0	0	0	0	0	0	0	0	0	0	0	0	1626	0	0
Preschool	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Prof-school	0	0	0	0	0	0	0	0	0	0	0	0	0	0	542	0
Some-college	0	0	0	0	0	0	0	0	0	6669	0	0	0	0	0	0

From the above perfectly 1-1 relationship, we can see these two variables are essentially exact the same. So we decide to remove the education_num variable.

```
#drop educationnum variable
X=subset(X,select = -education_num)
```

2. Re-group Native countries.

There are 41 levels, namely 41 different countries, in the dataset. Since too many levels for a categorical variable could lead to over-fitting, we decide to regroup the native countries into regions.

```
##
##  Amer-Indian-Eskimo  Asian-Pac-Islander               Black               Other
##                   2                  68                   0                   0
##               White
##                   1
```

Since the race of almost all records with Native country "South" is "Asian-Pac-Islander", we think the country "South" is very likely to be South Korea. So we decided to group the country "South" into "Asia_East".

```
Asia_East <- c(" Cambodia", " China", " Hong", " Laos", " Thailand",
               " Japan", " Taiwan", " Vietnam", " South", " Philippines")
```

```
Asia_Central <- c(" India", " Iran")

Central_America <- c(" Cuba", " Guatemala", " Jamaica", " Nicaragua",
                     " Puerto-Rico",  " Dominican-Republic", " El-Salvador",
                     " Haiti", " Honduras",  " Trinadad&Tobago")

South_America <- c(" Ecuador", " Peru", " Columbia")

North_America <- c(" Canada", " United-States"," Mexico", " Outlying-US(Guam-USVI-etc)")

Europe_West <- c(" England", " Germany", " Holand-Netherlands", " Ireland",
                 " France", " Greece", " Italy", " Portugal", " Scotland")

Europe_East <- c(" Poland", " Yugoslavia", " Hungary")


X <- mutate(X,
       native_region = ifelse(native_country %in% Asia_East, " East-Asia",
                ifelse(native_country %in% Asia_Central, " Central-Asia",
                ifelse(native_country %in% Central_America, " Central-America",
                ifelse(native_country %in% South_America, " South-America",
                ifelse(native_country %in% Europe_West, " Europe-West",
                ifelse(native_country %in% Europe_East, " Europe-East",
                ifelse(native_country %in% North_America, "North America", "Other")))))))

#convert native_region to a factor
#7 regions now
X$native_region=as.factor(X$native_region)
```

3. Re-group Capital Gain and Capital Loss.

First, let's look at how many zeros are in each of these two variables.

```
## the proportion of zeros in Capital Gain is 91.57902%
## the proportion of zeros in Capital Loss is 95.26527%

## The summary of non-zeros for thes two variables:

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     114    3464    7298   12978   14084   99999

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     155    1672    1887    1868    1977    4356
```

Based on the boxplots and summary of Capital Gain and Capital Loss variables in the Data Description section, and the fact that over 90% of the two variables are 0, we decide to categorize these two variables into groups in the following way:

- the first group is for the zeros;
- the other groups are based on quantiles of non-zeros. More specifically, if a value is larger than zero and lower than the 1st quantile, it's grouped as "Low".
- the values between 1st and 3rd quantitles are grouped into "Medium" and those higher than the 3rd quantile are categorized as "High".

```
X=mutate(X, cap_gain = ifelse(X$capital_gain==0, "Zero",
                     ifelse(X$capital_gain>0 & X$capital_gain<3464,"Low",
                     ifelse(X$capital_gain>=3464 & X$capital_gain<14084,"Medium","High"))),
```

```
       cap_loss = ifelse(X$capital_loss==0,"Zero",
                  ifelse(X$capital_loss>0  & X$capital_loss<1672,"Low",
                  ifelse(X$capital_loss>=1672 & X$capital_loss<1977,"Medium","High")))

           )

X$cap_gain=as.factor(X$cap_gain)
X$cap_loss=as.factor(X$cap_loss)
```

4. Hours per week

From the boxplot and histogram in the Data Description section before, we've known there are large number of outliers in the hours_per_week variable.

So We decide to group this variable in the following way:

- if the value is lower than the 1st quantile (40), it's called "less than 40 hours";
- if a value is between the 1st and 3rd quantitle (40 to 45), it's called "40 to 45 hours";
- if the value is higher than 3rd quantile but lower than 50 , it's called "45 to 50 hours";
- if the value is between 50 and 60, it's called "50 to 60 hours";
- if the value is between 60 and 70, it's called "60 to 70 hours";
- if the value is between 70 and 80, it's called "70 to 80 hours";
- if the value is more than 80, it's called "greater than 80 hours";

```
#summary(X$hours_per_week)

X=mutate(X,
         weekly_hours=ifelse(X$hours_per_week<40,"less than 40 hrs",
                         ifelse(X$hours_per_week<=45, "40 to 45 hrs",
                         ifelse(X$hours_per_week<=50, "45 to 50 hrs",
                         ifelse(X$hours_per_week<=60, "50 to 60 hrs",
                         ifelse(X$hours_per_week<=70, "60 to 70 hrs",
                         ifelse(X$hours_per_week<=80, "70 to 80 hours",
                                "greater than 80 hrs")))))))


X$weekly_hours=as.factor(X$weekly_hours)

table4=as.matrix(table(X$weekly_hours))
print(xtable(table4, caption = "Frequency table of weekly_hours" ) )
```

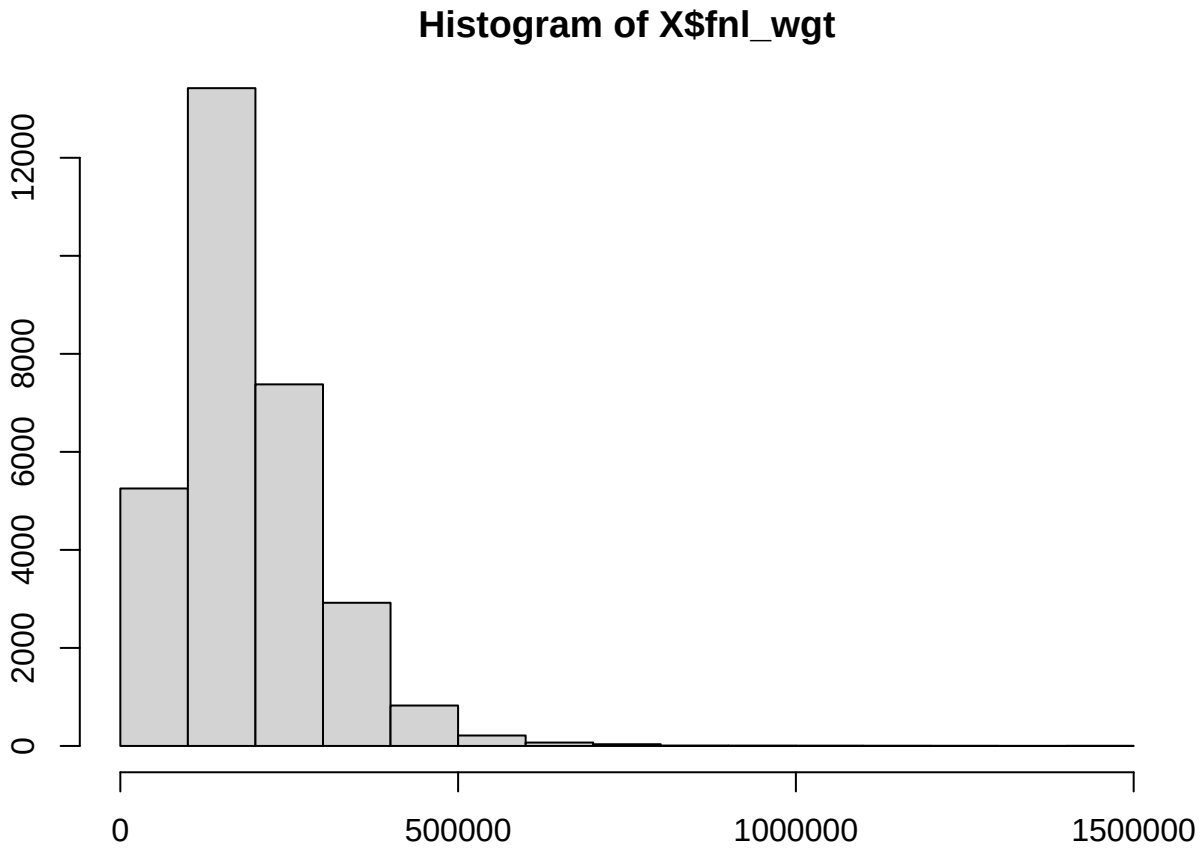
	x
40 to 45 hrs	16593
45 to 50 hrs	3364
50 to 60 hrs	2422
60 to 70 hrs	592
70 to 80 hours	265
greater than 80 hrs	195
less than 40 hrs	6708

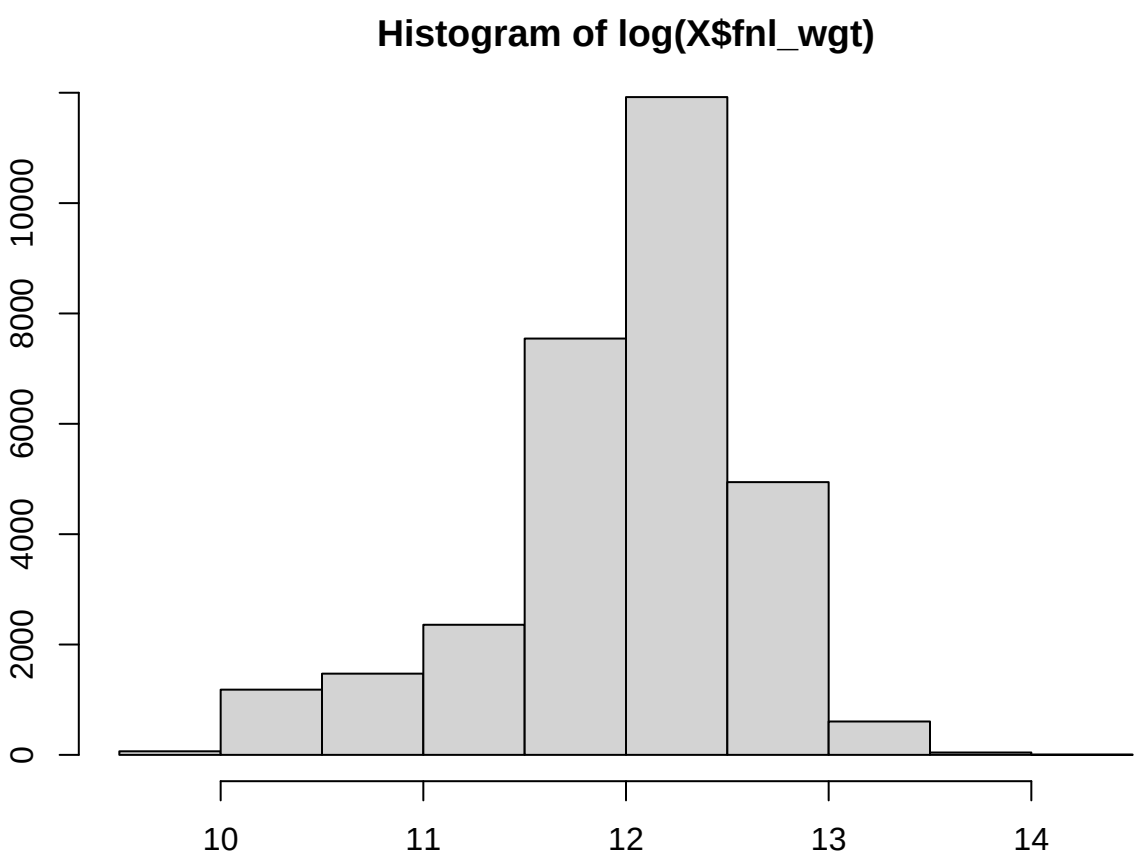
5. Drop empty level of work class.

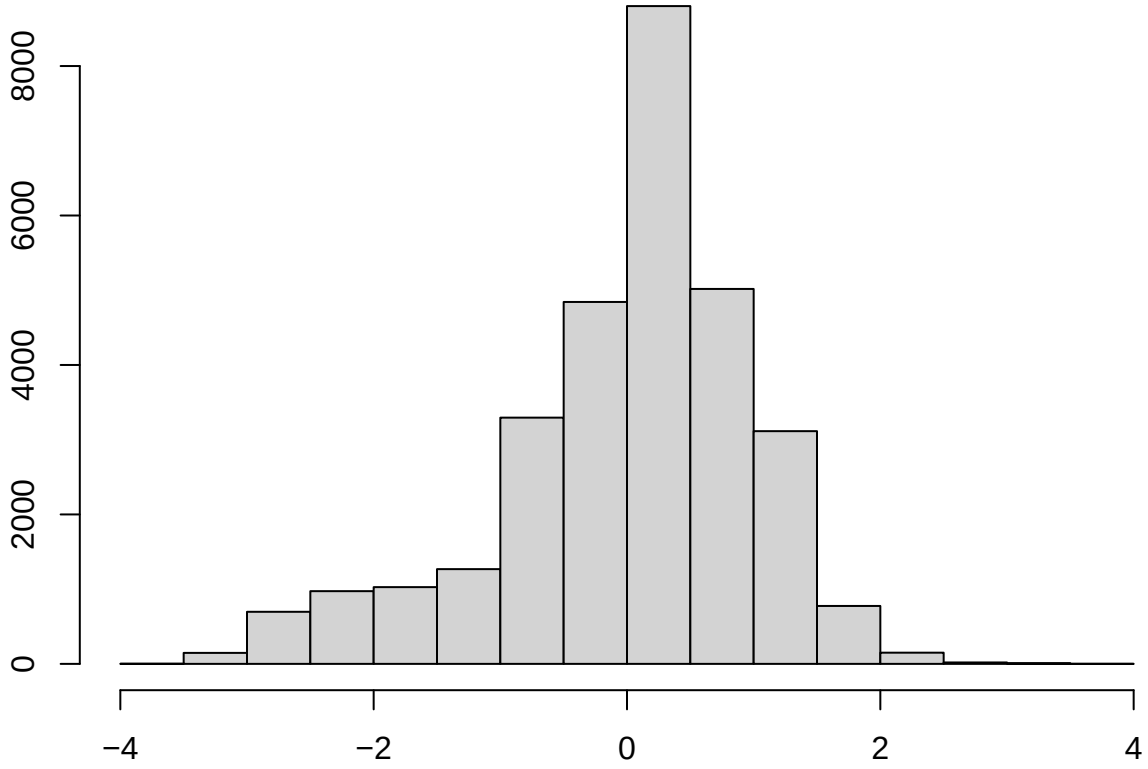
We noticed one of the levels of the work class variable does not have any records in it, so we decide to drop that level to avoid potential issues caused by empty cells.

	x
Federal-gov	943
Local-gov	2067
Never-worked	0
Private	22264
Self-emp-inc	1074
Self-emp-not-inc	2498
State-gov	1279
Without-pay	14

6. Log transformation and standardization of fnl_wgt




Histogram of log(X$fnl_wgt)
0
2000
4000
6000
8000
10000
10
11
12
13
14



From the above histograms, we can see that the fnl_wgt generally follows a log-normal distribution and the values are generally large. So we decide to perform a log transformation and then standardize it, after which the variable is generally normally distributed with small values.

```
X=mutate(X,
         fnlwgt_logstand=scale(log(X$fnl_wgt)))
```

6. Age standardization.

```
X=mutate(X,
        age_stand=scale(X$age))
```

To make age in the same range of the standardized fnl_wgt, we decide to standardize age as well.

7. Group marital status.

```
X=mutate(X,
         marital_status_group=ifelse(marital_status %in%
      c(" Married-AF-spouse"," Married-civ-spouse"," Married-spouse-absent"), "Married", as.character(ma

X$marital_status_group=as.factor(X$marital_status_group)
```

8. Drop the variables that would not be used for building predictive models.

```
X=subset(X, select = -c(capital_gain, capital_loss,hours_per_week,native_country,marital_status))
cat("The current structure of the dataset is:")
```

```
## The current structure of the dataset is:
```

```
str(X)
```

```
## 'data.frame':    30139 obs. of  16 variables:
##  $ age                 : int  39 50 38 53 28 37 49 52 31 42 ...
##  $ workclass           : Factor w/ 7 levels " Federal-gov",..: 6 5 3 3 3 3 3 5 3 3 ...
##  $ fnl_wgt             : int  77516 83311 215646 234721 338409 284582 160187 209642 45781 159449 ...
##  $ education           : Factor w/ 16 levels " 10th"," 11th",..: 10 10 12 2 10 13 7 12 13 10 ...
##  $ occupation          : Factor w/ 14 levels " Adm-clerical",..: 1 4 6 6 10 4 8 4 10 4 ...
##  $ relationship        : Factor w/ 6 levels " Husband"," Not-in-family",..: 2 1 2 1 6 6 2 1 2 1 ...
##  $ race                : Factor w/ 5 levels " Amer-Indian-Eskimo",..: 5 5 5 3 3 5 3 5 5 5 ...
##  $ sex                 : Factor w/ 2 levels " Female"," Male": 2 2 2 2 1 1 1 2 1 2 ...
##  $ income              : Factor w/ 2 levels " <=50K"," >50K": 1 1 1 1 1 1 1 2 2 2 ...
##  $ native_region       : Factor w/ 7 levels " Central-America",..: 7 7 7 7 1 7 1 7 7 7 ...
##  $ cap_gain            : Factor w/ 4 levels "High","Low","Medium",..: 2 4 4 4 4 4 4 4 1 3 ...
##  $ cap_loss            : Factor w/ 4 levels "High","Low","Medium",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ weekly_hours        : Factor w/ 7 levels "40 to 45 hrs",..: 1 7 1 1 1 1 7 1 2 1 ...
##  $ fnlwgt_logstand     : num [1:30139, 1] -1.151 -1.036 0.472 0.606 1.186 ...
##  $ age_stand           : num [1:30139, 1] 0.0425 0.8802 -0.0336 1.1087 -0.7952 ...
##  $ marital_status_group: Factor w/ 5 levels " Divorced"," Never-married",..: 2 5 1 5 5 5 5 5 2 5 ...
```

Modeling

###Recode target variable.

First, for the simplicity in coding and to avoid potential issues caused by the characters in the target variable, let's re-code the values of target variable to be "Y", meaning yearly income is higher than 50K USD, and "N", indicating the income is no more than 50K.

```
X$income=ifelse(X$income==" <=50K","N","Y")
X$income=as.factor(X$income)
```

```
## Now the levels of the target variable are:
```

```
##     N     Y
## 22633  7506
```

1. Train-test split.

Before training supervised learning models, we first split the dataset into training and testing sets.

```
## The number of the two levels of the target variable are:
```

```
##     N     Y
## 18107  6005
```

2. Down-sample the majority group to balance the training data.

We've seen from Data Exploratory analysis that there are about 3 times of people making no more than 50K annually (the majority group) than those who make over 50K a year (the minority group), so the dataset is imbalanced, which could lead to over-fitting issues. So we decide to down-sample the majority group. More specifically, we will randomly select 34% of the records that have annual income no more than 50K, and then combine them with all the records in the minority group.

- First, randomly select 34% of records from the majority group.

```
## The number of the two levels of the target variable at the randomly selected majority sample are:
```

```
##    N    Y
## 6156    0
```

- Then, combine the randomly down sampled majority group with the original minority group.

```
## The number of the two levels of the target variable at the combined balanced sample are:
```

```
##    N    Y
## 6156 6005
```

- Next, randomly shuffle the rows so that not all "N" records are on the top and "Y"s are in the end.

```
## The number of the two levels of the target variable at the final training dataset are not changed!
```

```
##    N    Y
## 6156 6005
```

Create different sets of features to train supervised learning models.

Based on different ways of cleaning the variables, we have different sets of features to train supervised learning models on.

####Training scenario 1: apply Random Forest, KNN and Logistic Regresstion to transformed variables.

Random Forest Model.

We will use 5-fold cross-validation for the re-sampling to tune model parameters.

```
cv_5=trainControl(method = "cv", number = 5,
                  allowParallel = TRUE,
                  summaryFunction = twoClassSummary ,
                  classProbs = TRUE)
```

```
#random forest
library(randomForest)
set.seed(123)
t1_rf=proc.time()
model_rf <- randomForest(income ~ age_stand + workclass + fnlwgt_logstand +            education + mari
            data = X_train_bal,
            trControl=cv_5 )
t2_rf=proc.time()
```

```
## The computational time for training a random forest model is 8.37 s.
```

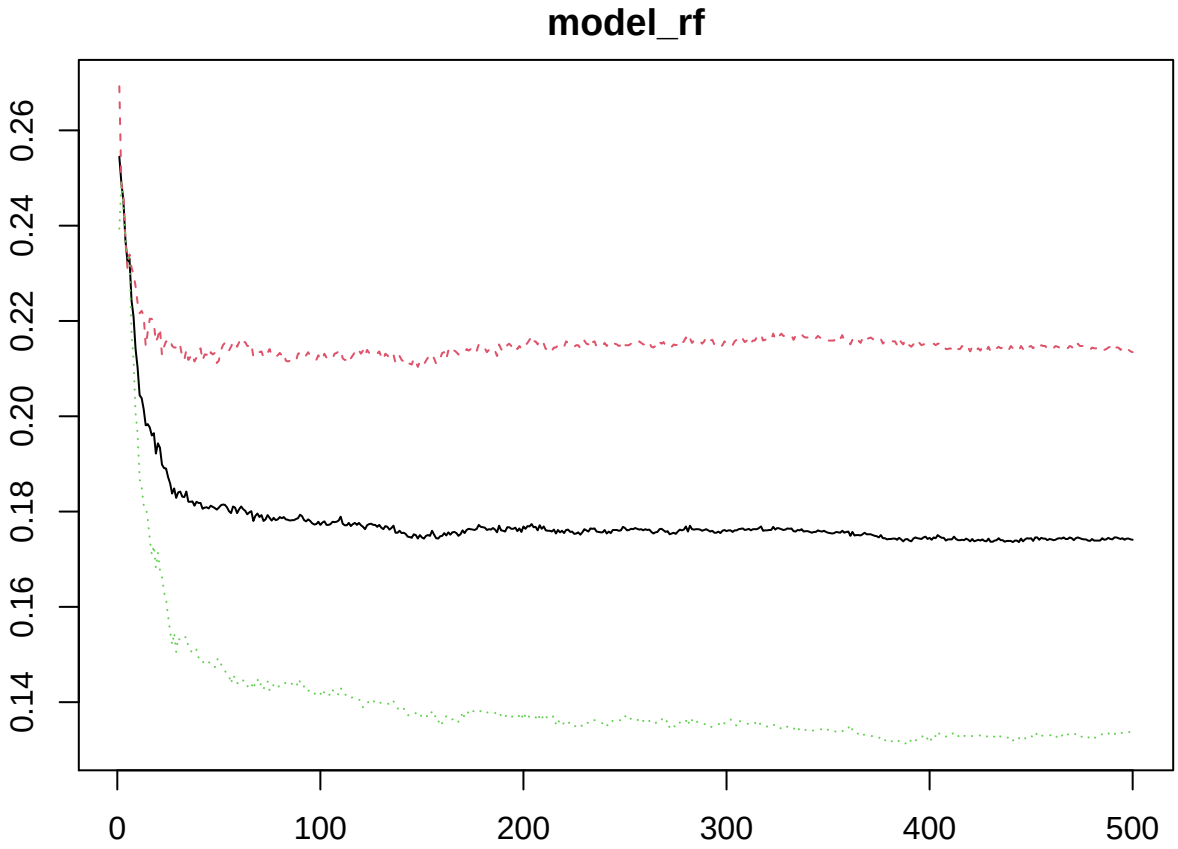
```
## The fitted random forest model:
```

```
##
## Call:
##  randomForest(formula = income ~ age_stand + workclass + fnlwgt_logstand +      education + marital_
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 17.41%
## Confusion matrix:
##      N    Y class.error
## N 4842 1314   0.2134503
## Y  803 5202   0.1337219
```

```
## Importance of features based on the random forest model:
```

```
##                      MeanDecreaseGini
## age_stand                  780.51237
## workclass                  202.16609
## fnlwgt_logstand            555.73061
## education                  586.81450
```

```
## marital_status_group        671.41456
## occupation                  658.79789
## relationship                820.14554
## race                         72.90622
## sex                         101.63749
## native_region                63.96797
## cap_loss                     95.95992
## cap_gain                    342.74626
## weekly_hours                284.87383
```

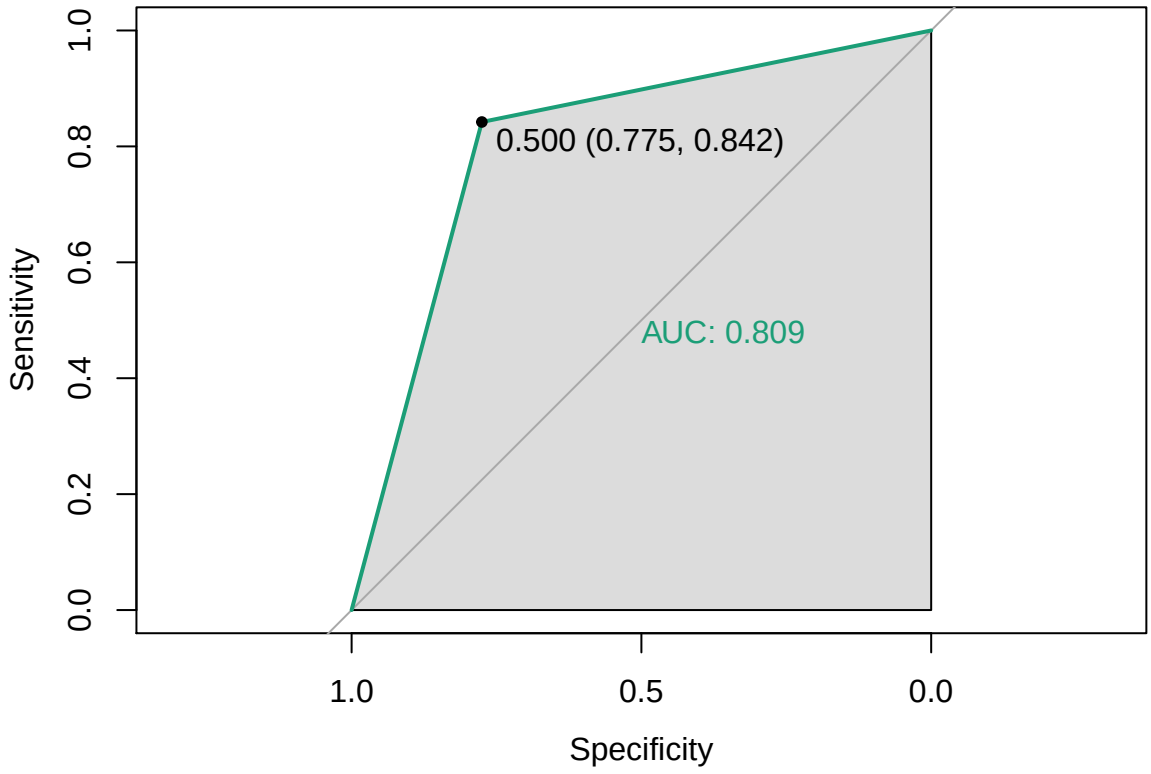



```
## The confusion matrix of random forest model on test data:

## Confusion Matrix and Statistics
##
##           Reference
## Prediction    N    Y
##          N 3504  237
##          Y 1022 1264
##
##                Accuracy : 0.7911
##                  95% CI : (0.7806, 0.8013)
##     No Information Rate : 0.751
##     P-Value [Acc > NIR] : 1.139e-13
##
##                   Kappa : 0.5246
##
##  Mcnemar's Test P-Value : < 2.2e-16
```

```
##
##             Sensitivity : 0.8421
##             Specificity : 0.7742
##          Pos Pred Value : 0.5529
##          Neg Pred Value : 0.9366
##              Prevalence : 0.2490
##          Detection Rate : 0.2097
##    Detection Prevalence : 0.3793
##       Balanced Accuracy : 0.8081
##
##        'Positive' Class : Y
##
```

Then let's look at the ROC curve and AUC of the random forest model.



KNN Model.

K-nearest neighbors (KNN) algorithm is one of the supervised learning algorithms that can be used for classification problems. It is a lazy learning algorithm because it does not have a specialized training phase and uses all the data for training while classification. KNN is also a non-parametric learning algorithm because it does not have assumptions of the data set.

```
#knn
set.seed(123)
t1_knn=proc.time()
model_knn=train(income ~ age_stand + workclass + fnlwgt_logstand +             education + marital_statu
                data = X_train_bal,
                 metric="ROC",
```

```
                trControl=cv_5, method="knn")
t2_knn=proc.time()
cat("The computational time of training a knn model is",(t2_knn-t1_knn)[3],"s.","\n")
```

```
## The computational time of training a knn model is 101.25 s.
```

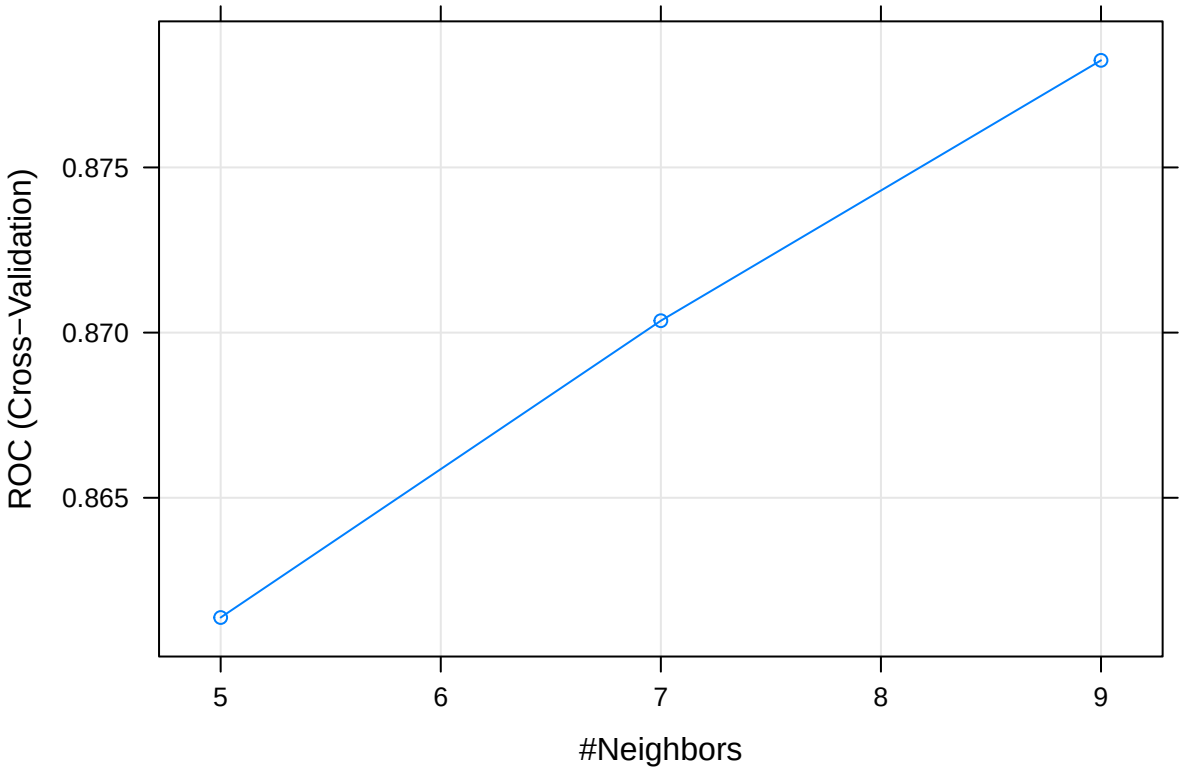
```
cat("The fitted knn model:","\n")
```

```
## The fitted knn model:
```

```
model_knn
```

```
## k-Nearest Neighbors
##
## 12161 samples
##    13 predictor
##     2 classes: 'N', 'Y'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 9729, 9729, 9728, 9729, 9729
## Resampling results across tuning parameters:
##
##   k  ROC        Sens       Spec
##   5  0.8613749  0.7621836  0.8231474
##   7  0.8703603  0.7600706  0.8353039
##   9  0.8782422  0.7668922  0.8404663
##
## ROC was used to select the optimal model using the largest value.
## The final value used for the model was k = 9.
```

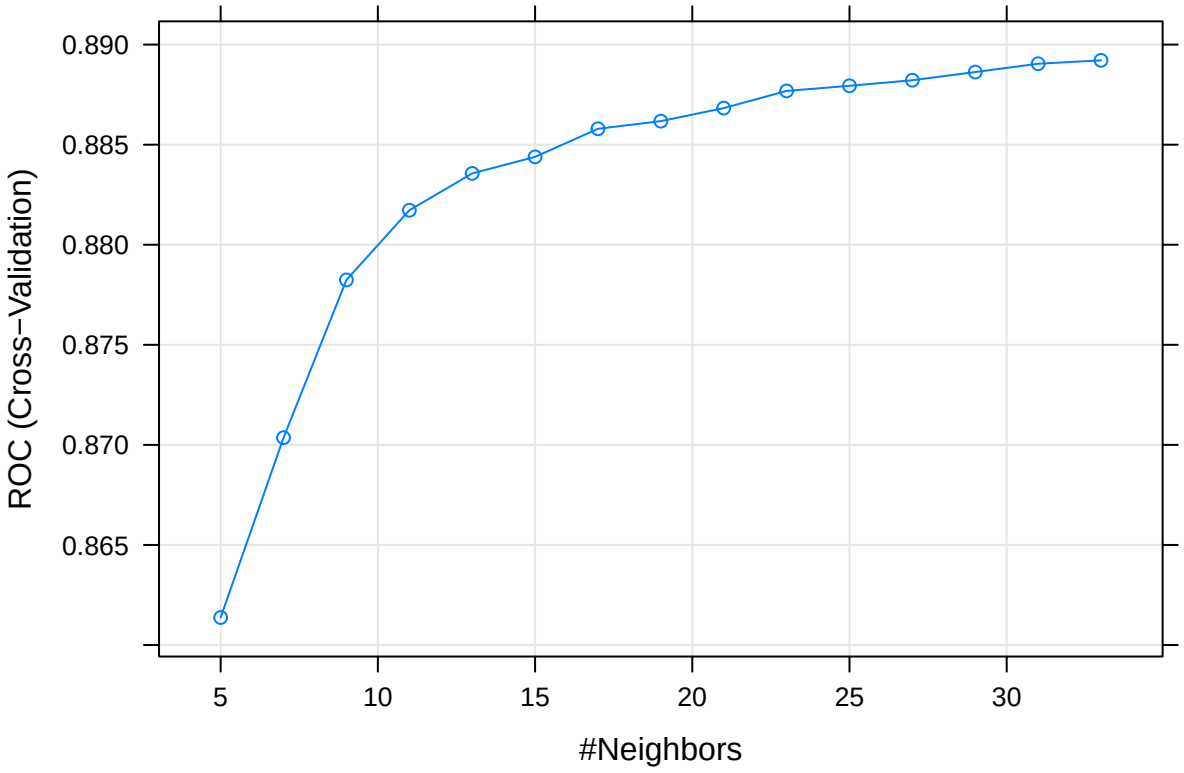
```
plot(model_knn)
```



From the above plot of accuracy and number of neighbors, we can see that the accuracy still increases after training on the default number of ks. So we decide to increasing the number of k values to 15 to try to locate a better model.

```
set.seed(123)
t1_knn=proc.time()
model_knn=train(income ~ age_stand + workclass + fnlwgt_logstand +             education + marital_statu
                data = X_train_bal,
                metric="ROC",
                trControl=cv_5,
                method="knn",
                tuneLength = 15)
t2_knn=proc.time()
```

```
## The computational time of training a knn model is 479.58 s.
```



Now we can see the accuracy does not increase much once the number of neighbors reaches about 15.

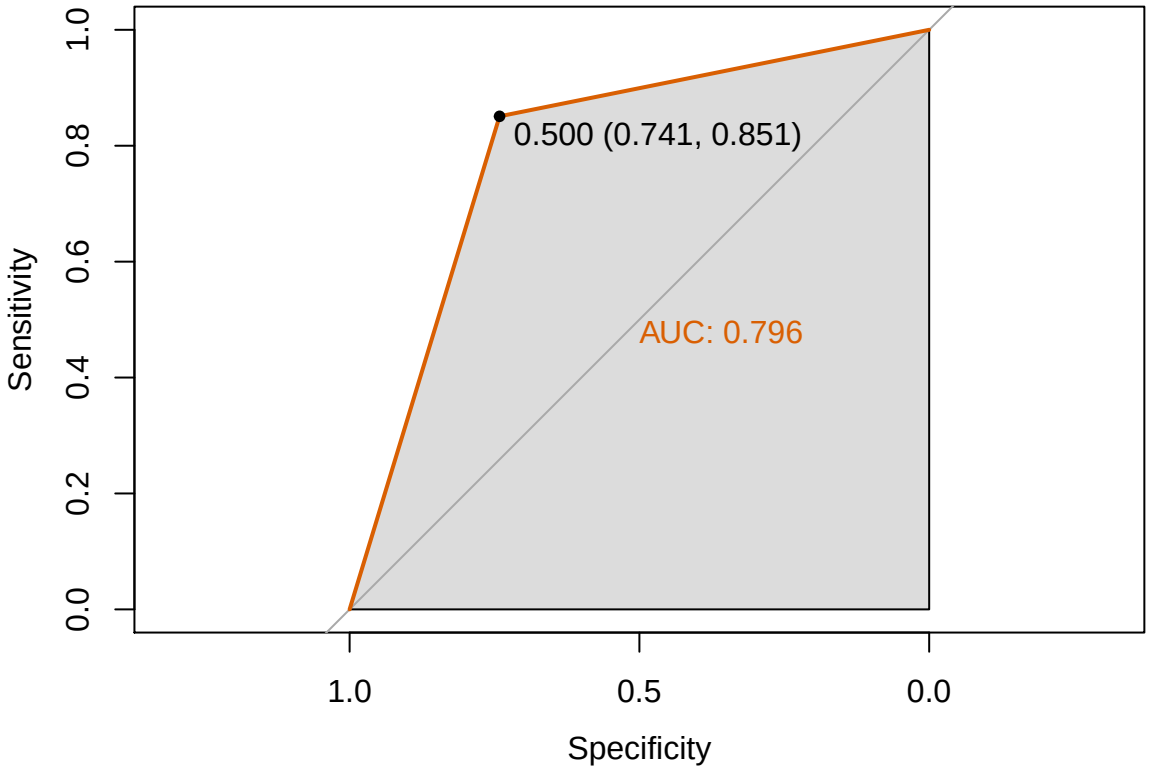
```
## The current fitted KNN models trying more tuning parameters:
```

```
## k-Nearest Neighbors
##
## 12161 samples
##    13 predictor
##     2 classes: 'N', 'Y'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 9729, 9729, 9728, 9729, 9729
## Resampling results across tuning parameters:
##
##   k   ROC        Sens       Spec
##    5  0.8613749  0.7623461  0.8231474
##    7  0.8703603  0.7600707  0.8353039
##    9  0.8782422  0.7668922  0.8404663
##   11  0.8817225  0.7616937  0.8459617
##   13  0.8835633  0.7628319  0.8459617
##   15  0.8843907  0.7597458  0.8504580
##   17  0.8857936  0.7589346  0.8544546
##   19  0.8861756  0.7540612  0.8551207
##   21  0.8868277  0.7548743  0.8564530
##   23  0.8876855  0.7530869  0.8559534
##   25  0.8879434  0.7511372  0.8566195
```

```
##   27  0.8882202  0.7501625  0.8569525
##   29  0.8886291  0.7514625  0.8607827
##   31  0.8890439  0.7498378  0.8591174
##   33  0.8892134  0.7483756  0.8574521
##
## ROC was used to select the optimal model using the largest value.
## The final value used for the model was k = 33.
```

```
## The confusion matrix of knn model on test data:
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    N    Y
##          N 3355  224
##          Y 1171 1277
##
##                Accuracy : 0.7685
##                  95% CI : (0.7577, 0.7791)
##     No Information Rate : 0.751
##     P-Value [Acc > NIR] : 0.0007745
##
##                   Kappa : 0.4889
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.8508
##             Specificity : 0.7413
##          Pos Pred Value : 0.5217
##          Neg Pred Value : 0.9374
##              Prevalence : 0.2490
##          Detection Rate : 0.2119
##    Detection Prevalence : 0.4062
##       Balanced Accuracy : 0.7960
##
##        'Positive' Class : Y
##
```



As there are mixed types of data in our dataset, we are supposed to use the gower distance metric in the KNN method. However, we are not able to specify "gower" in the train() function and the knngow() function which can run knn using the gower metric is not available to our personal version of R, so for future discussion, if time allows, please try to run the KNN method using gower distance metric on this data set.

Logistic Regression Model.

```
#logreg
set.seed(123)
t1_log=proc.time()
model_logreg=train(income ~age_stand + workclass + fnlwgt_logstand +            education + marital_sta
                   data = X_train_bal,
                   metric="ROC",
                   trControl=cv_5, method="regLogistic")

t2_log=proc.time()
```

```
## The computational time of training a logistic regression on transformed variables is 33.04 s.
```

```
## The fitted logistic regression model:
```

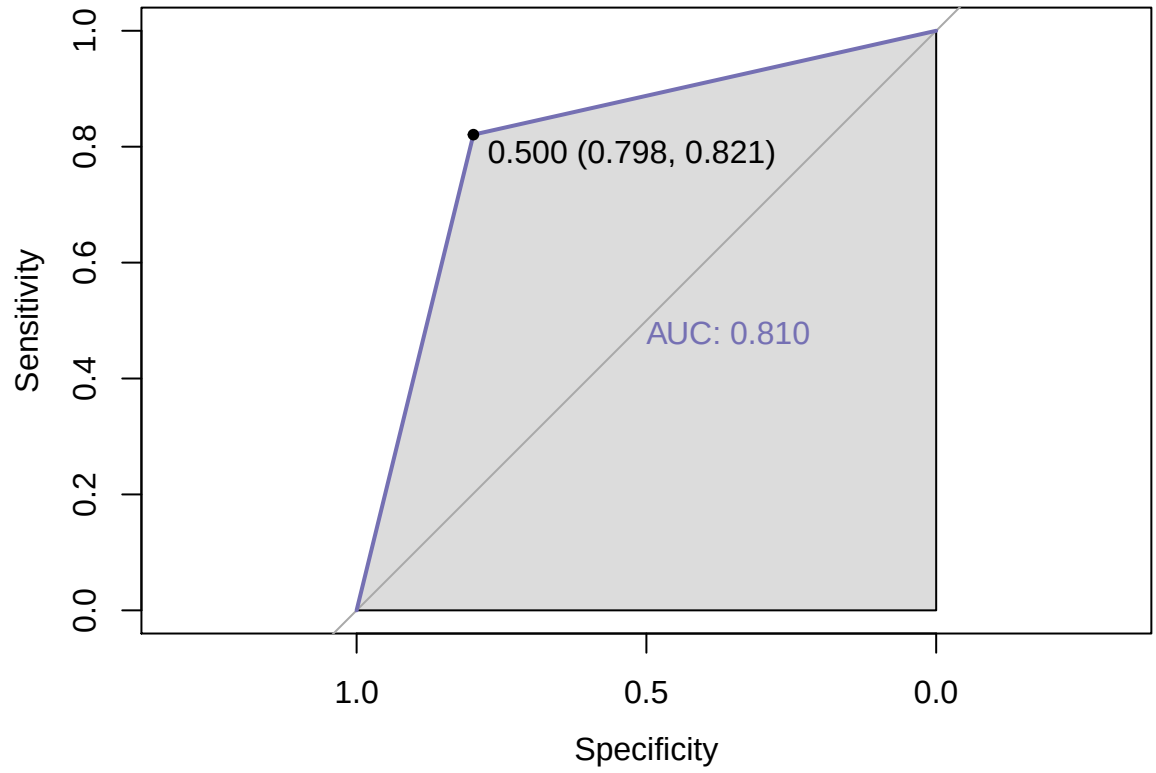
```
## Regularized Logistic Regression
##
## 12161 samples
##    13 predictor
##     2 classes: 'N', 'Y'
##
## No pre-processing
```

```
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 9729, 9729, 9728, 9729, 9729
## Resampling results across tuning parameters:
##
##   cost  loss       epsilon  ROC        Sens       Spec
##   0.5   L1         0.001    0.9045239  0.8019786  0.8373022
##   0.5   L1         0.010    0.9035459  0.7997049  0.8371357
##   0.5   L1         0.100    0.8967691  0.7901202  0.8394671
##   0.5   L2_dual    0.001    0.9043658  0.8008411  0.8388010
##   0.5   L2_dual    0.010    0.9043661  0.8010036  0.8388010
##   0.5   L2_dual    0.100    0.9043611  0.8011658  0.8386345
##   0.5   L2_primal  0.001    0.9043696  0.8010036  0.8388010
##   0.5   L2_primal  0.010    0.9042917  0.8003538  0.8384679
##   0.5   L2_primal  0.100    0.9037455  0.7997042  0.8388010
##   1.0   L1         0.001    0.9050671  0.8001919  0.8378018
##   1.0   L1         0.010    0.9040197  0.7995420  0.8379684
##   1.0   L1         0.100    0.8954953  0.7894694  0.8404663
##   1.0   L2_dual    0.001    0.9048735  0.7998668  0.8389675
##   1.0   L2_dual    0.010    0.9048747  0.7997043  0.8389675
##   1.0   L2_dual    0.100    0.9048792  0.7997045  0.8393006
##   1.0   L2_primal  0.001    0.9048562  0.8003541  0.8394671
##   1.0   L2_primal  0.010    0.9046635  0.7995419  0.8383014
##   1.0   L2_primal  0.100    0.9037334  0.8010042  0.8361366
##   2.0   L1         0.001    0.9052153  0.8006794  0.8381349
##   2.0   L1         0.010    0.9043255  0.8000294  0.8381349
##   2.0   L1         0.100    0.8963592  0.7910945  0.8374688
##   2.0   L2_dual    0.001    0.9051130  0.8003545  0.8401332
##   2.0   L2_dual    0.010    0.9051132  0.8005168  0.8401332
##   2.0   L2_dual    0.100    0.9051246  0.8000294  0.8409659
##   2.0   L2_primal  0.001    0.9051119  0.8001920  0.8401332
##   2.0   L2_primal  0.010    0.9048666  0.8010045  0.8391341
##   2.0   L2_primal  0.100    0.9035989  0.7988924  0.8381349
##
## ROC was used to select the optimal model using the largest value.
## The final values used for the model were cost = 2, loss = L1 and epsilon
##  = 0.001.
```

```
## The confusion matrix of logistic regression model:
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    N    Y
##          N 3614  269
##          Y  912 1232
##
##                Accuracy : 0.804
##                  95% CI : (0.7938, 0.814)
##     No Information Rate : 0.751
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5417
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
```

```
##             Sensitivity : 0.8208
##             Specificity : 0.7985
##          Pos Pred Value : 0.5746
##          Neg Pred Value : 0.9307
##              Prevalence : 0.2490
##          Detection Rate : 0.2044
##    Detection Prevalence : 0.3557
##       Balanced Accuracy : 0.8096
##
##        'Positive' Class : Y
##
```



Training scenario 2: Logistic Regression Model on unstandardized age and fnl_wgt.

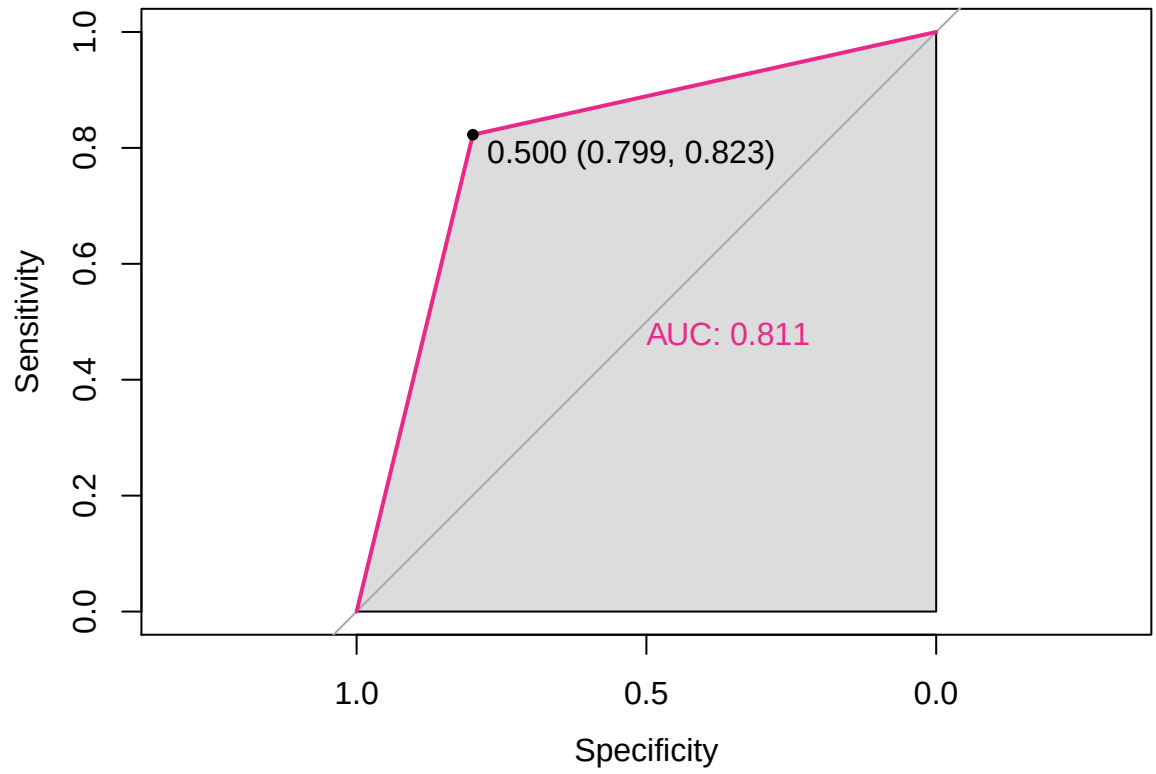
```
#logreg
set.seed(123)
t1_log_2=proc.time()
model_logreg_ori=train(income ~ age + workclass + fnl_wgt + education + marital_status_group + occupati
                   data = X_train_bal,
                   metric="ROC",
                   trControl=cv_5,
                   method="regLogistic")
t2_log_2=proc.time()
```

```
## The time of training a logistic regression model using unstandardized variables is 445.41 s.
```

```
## The confusion matrix of logistic regression model:
```

```
## Confusion Matrix and Statistics
```

```
##
##           Reference
## Prediction    N    Y
##          N 3618  266
##          Y  908 1235
##
##                Accuracy : 0.8052
##                  95% CI : (0.795, 0.8151)
##     No Information Rate : 0.751
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5444
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.8228
##             Specificity : 0.7994
##          Pos Pred Value : 0.5763
##          Neg Pred Value : 0.9315
##              Prevalence : 0.2490
##          Detection Rate : 0.2049
##    Detection Prevalence : 0.3556
##       Balanced Accuracy : 0.8111
##
##        'Positive' Class : Y
##
```



Training scenario 3: Logistic Regression Model on unstandardized age without fnl_wgt.

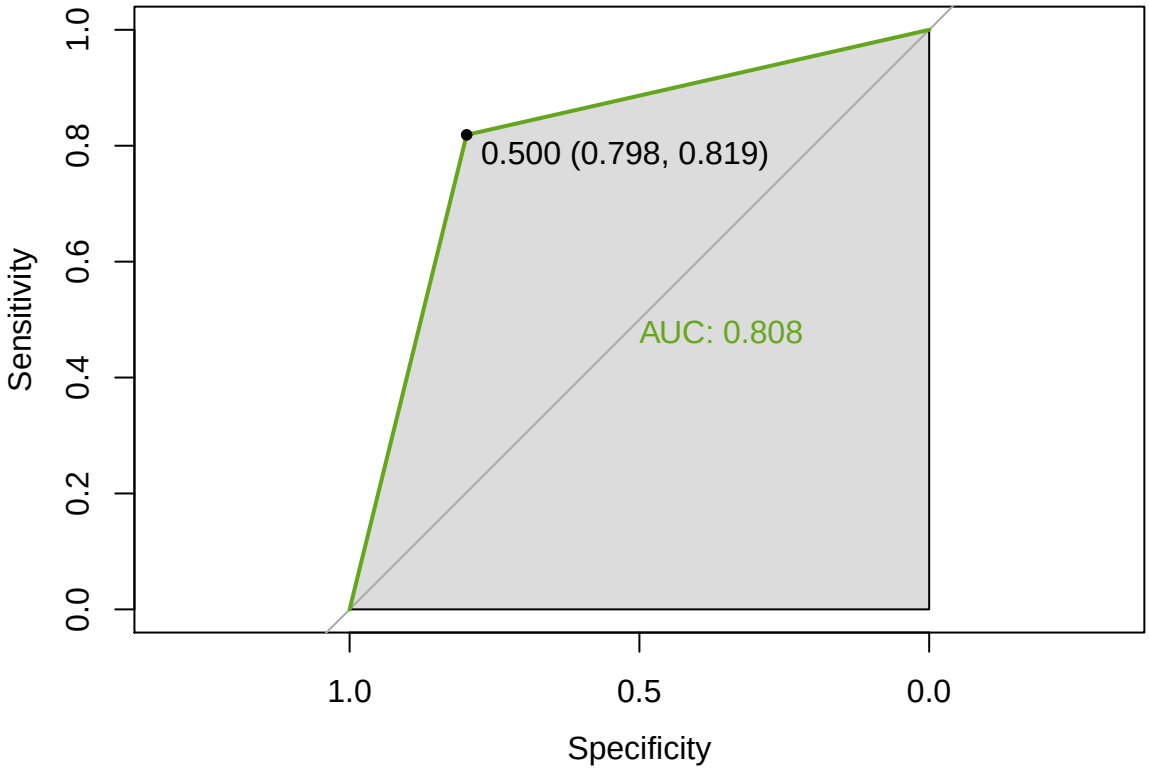
We are going to remove final weight for the model because it has to do with a weighted value for certain demographics when the census was conducted. However, it would not make sense as an input for individual users of the Shiny app

```
#logreg
set.seed(123)
t1_log_3=proc.time()
model_logreg_nofnlwgt=train(income ~ age + workclass + education + marital_status_group + occupation + r
                    data = X_train_bal,
                    trControl=cv_5,
                    metric="ROC",
                    method="regLogistic")
t2_log_3=proc.time()
```

```
## The computational time of training a logistic regression model is 507.29 s.
```

```
## The confusion matrix of logistic regression model:
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    N    Y
##          N 3612  272
##          Y  914 1229
##
##                Accuracy : 0.8032
##                  95% CI : (0.793, 0.8132)
##     No Information Rate : 0.751
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5397
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.8188
##             Specificity : 0.7981
##          Pos Pred Value : 0.5735
##          Neg Pred Value : 0.9300
##              Prevalence : 0.2490
##          Detection Rate : 0.2039
##    Detection Prevalence : 0.3556
##       Balanced Accuracy : 0.8084
##
##        'Positive' Class : Y
##
```




```
save(model_logreg_nofnlwgt , file = 'logmodel.rda')
saveRDS(model_logreg_nofnlwgt, "logmodel.rds")
```

The models give similar accuracy level, so for simplicity, we use the logistic model with un-transformed age and with fnl_wgt removed for deploying the shiny App.

Training scenario 4: Up-sampling

For imbalanced data, other than down-sample the majority group, we can also up-sample the minority group. So let's try to fit a logistic regression on an up-sampled training data set.

Check whether the distribution of the income classes is balanced in the training set.

```
##
##  0  1
## 50 50
```

Train the model

```
##
## Call:
## glm(formula = income ~ age_stand + workclass + education + marital_status_group +
##     occupation + relationship + race + sex + cap_gain + cap_loss +
##     weekly_hours + native_region, family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.9726  -0.5418   0.0000   0.5639   3.4923
```

```
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      26.96971  176.92119   0.152 0.878841
## age_stand                         0.41301    0.02479  16.658  < 2e-16 ***
## workclass Local-gov              -0.55493    0.11076  -5.010 5.44e-07 ***
## workclass Private                -0.80100    0.09322  -8.593  < 2e-16 ***
## workclass Self-emp-inc           -0.15122    0.12836  -1.178 0.238755
## workclass Self-emp-not-inc       -0.62839    0.10932  -5.748 9.02e-09 ***
## workclass State-gov              -0.75097    0.12579  -5.970 2.38e-09 ***
## workclass Without-pay           -18.50097 1839.52227  -0.010 0.991975
## education 11th                    0.50141    0.20510   2.445 0.014496 *
## education 12th                    0.58063    0.27804   2.088 0.036771 *
## education 1st-4th                -0.41068    0.46847  -0.877 0.380683
## education 5th-6th                -0.90073    0.37000  -2.434 0.014915 *
## education 7th-8th                -0.36613    0.24122  -1.518 0.129057
## education 9th                    -0.30841    0.28210  -1.093 0.274276
## education Assoc-acdm              1.56385    0.18352   8.522  < 2e-16 ***
## education Assoc-voc               1.32485    0.17895   7.403 1.33e-13 ***
## education Bachelors               1.99343    0.16620  11.994  < 2e-16 ***
## education Doctorate               3.34235    0.23996  13.929  < 2e-16 ***
## education HS-grad                 0.88138    0.16174   5.449 5.06e-08 ***
## education Masters                 2.50298    0.17606  14.216  < 2e-16 ***
## education Preschool             -17.42398 1089.16906  -0.016 0.987236
## education Prof-school             2.95594    0.22181  13.327  < 2e-16 ***
## education Some-college            1.17005    0.16431   7.121 1.07e-12 ***
## marital_status_group Never-married  -0.58392    0.07022  -8.315  < 2e-16 ***
## marital_status_group Separated    -1.13155    0.15510  -7.296 2.97e-13 ***
## marital_status_group Widowed      -0.65966    0.13694  -4.817 1.46e-06 ***
## marital_status_groupMarried        0.41489    0.06994   5.932 2.99e-09 ***
## occupation Armed-Forces          -16.98508 2556.10288  -0.007 0.994698
## occupation Craft-repair            0.50834    0.08358   6.082 1.18e-09 ***
## occupation Exec-managerial         1.20427    0.07967  15.115  < 2e-16 ***
## occupation Farming-fishing        -0.84362    0.14542  -5.801 6.58e-09 ***
## occupation Handlers-cleaners      -0.47724    0.15155  -3.149 0.001638 **
## occupation Machine-op-inspct       0.14003    0.10323   1.356 0.174947
## occupation Other-service           0.08672    0.09828   0.882 0.377555
## occupation Priv-house-serv       -13.43644  123.43560  -0.109 0.913318
## occupation Prof-specialty          0.93845    0.08211  11.429  < 2e-16 ***
## occupation Protective-serv         0.90731    0.13013   6.972 3.11e-12 ***
## occupation Sales                   0.63520    0.08604   7.383 1.55e-13 ***
## occupation Tech-support            1.38643    0.11157  12.427  < 2e-16 ***
## occupation Transport-moving        0.27822    0.10663   2.609 0.009078 **
## relationship Not-in-family        -0.81646    0.06869 -11.886  < 2e-16 ***
## relationship Other-relative       -1.51463    0.17277  -8.767  < 2e-16 ***
## relationship Own-child            -2.20122    0.13786 -15.967  < 2e-16 ***
## relationship Unmarried            -0.74760    0.08048  -9.290  < 2e-16 ***
## relationship Wife                  0.92459    0.09022  10.249  < 2e-16 ***
## race Asian-Pac-Islander            0.93842    0.20817   4.508 6.55e-06 ***
## race Black                         0.60824    0.19683   3.090 0.002000 **
## race Other                         0.18219    0.29347   0.621 0.534727
## race White                        -0.10825    0.18861  -0.574 0.565996
## sex Male                           0.05503    0.05254   1.047 0.294909
## cap_gainLow                      -27.15991  176.92092  -0.154 0.877993
```

```
## cap_gainMedium                      -24.24531  176.92088  -0.137 0.890999
## cap_gainZero                        -26.53021  176.92087  -0.150 0.880800
## cap_lossLow                          -1.40354    0.19837  -7.075 1.49e-12 ***
## cap_lossMedium                       -0.54391    0.18287  -2.974 0.002936 **
## cap_lossZero                         -1.75086    0.13893 -12.602  < 2e-16 ***
## weekly_hours45 to 50 hrs              0.69506    0.05740  12.108  < 2e-16 ***
## weekly_hours50 to 60 hrs              0.74273    0.06247  11.889  < 2e-16 ***
## weekly_hours60 to 70 hrs              1.08106    0.11777   9.179  < 2e-16 ***
## weekly_hours70 to 80 hours            0.64544    0.18700   3.452 0.000557 ***
## weekly_hoursgreater than 80 hrs       0.51969    0.21924   2.370 0.017768 *
## weekly_hoursless than 40 hrs         -0.65293    0.05922 -11.026  < 2e-16 ***
## native_region Central-Asia            1.36352    0.25067   5.440 5.34e-08 ***
## native_region East-Asia               0.92786    0.17013   5.454 4.93e-08 ***
## native_region Europe-East            -0.17319    0.35058  -0.494 0.621300
## native_region Europe-West             0.27973    0.18756   1.491 0.135848
## native_region South-America          -2.07036    0.55667  -3.719 0.000200 ***
## native_regionNorth America           -0.39016    0.13452  -2.900 0.003728 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 33299  on 24019  degrees of freedom
## Residual deviance: 17734  on 23952  degrees of freedom
## AIC: 17870
##
## Number of Fisher Scoring iterations: 17
```

From the model summary, if we use an alpha level of 0.05, we would say that all the variables used are significant so we will keep all variables used in the current model.

```
#running the anova function to analyze the analysis of variance:
anova(model_glm, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: income
##
## Terms added sequentially (first to last)
##
##
##                       Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                                  24019      33299
## age_stand              1   2207.6     24018      31091  < 2e-16 ***
## workclass              6    965.8     24012      30125  < 2e-16 ***
## education             15   4022.7     23997      26103  < 2e-16 ***
## marital_status_group   4   2478.3     23993      23624  < 2e-16 ***
## occupation            13   1087.8     23980      22537  < 2e-16 ***
## relationship           5    764.5     23975      21772  < 2e-16 ***
## race                   4    659.3     23971      21113  < 2e-16 ***
## sex                    1      4.3     23970      21109  0.03744 *
## cap_gain               3   2175.5     23967      18933  < 2e-16 ***
## cap_loss               3    339.1     23964      18594  < 2e-16 ***
```

```
## weekly_hours          6    590.0     23958      18004  < 2e-16 ***
## native_region         6    270.3     23952      17734  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

All variables used are significant with probability (Pr) values < 0.05. Education, marital_status, occupation, cap_loss, cap_gain have the highest deviance values.

#Evaluating the Model

Now we will test the logistic regression model on the test data.

```
predicted_results_glm <- predict(model_glm, X_test, type='response')
head(predicted_results_glm)
```

```
##           3          13          18          20          34          45
## 0.048248052 0.010549387 0.044427596 0.660348898 0.066823850 0.009682766
```

The predicted results are the probabilities for income to be equal to 0 (<=50K) for each instance in our test dataset. we need to determine an ideal cutoff value to interpret the results in terms of 0(<=50K) or 1(>50K).

```
library(InformationValue)
```

```
set.seed(123)
ideal_cutoff_glm <-
  optimalCutoff(
  actuals = X_test$income,
  predictedScores = predicted_results_glm,
  optimiseFor = "Both")
```

check ideal cutoff

```
## [1] 0.39
```

So the ideal cutoff value for oour prediction is 0.4599963. Now we can use the cutoff value to recode predictions.

```
##  3 13 18 20 34 45
##  0  0  0  1  0  0
```

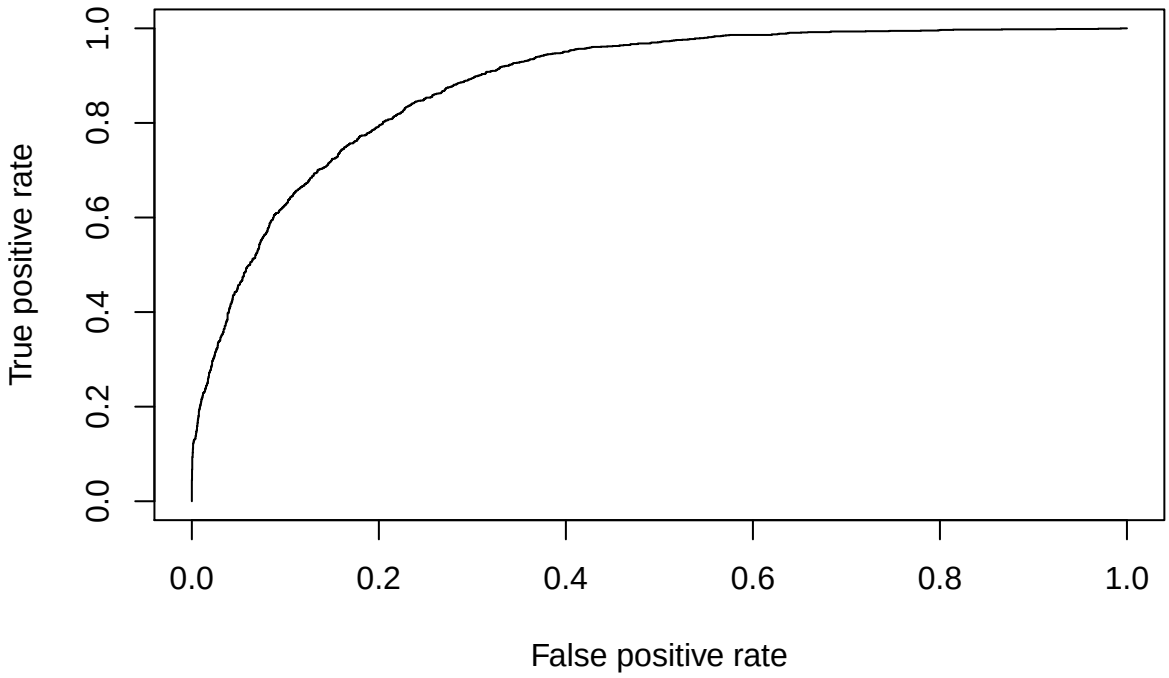
Evaluate the model against the test data by creating a confusion matrix and use it to derive the model's accuracy/

```
## [1] 0.7809856
```

The accuracy is 78%.

ROC curve plots the True Positive Rate with the False Positive Rate, at different threshold settings.

AUC is the area under the curve, and the better a model is the closer the AUC is to 1, rather than to 0.5. Values above 0.80 indicate that the model does a good job in discriminating between the two categories which comprise our target variable.



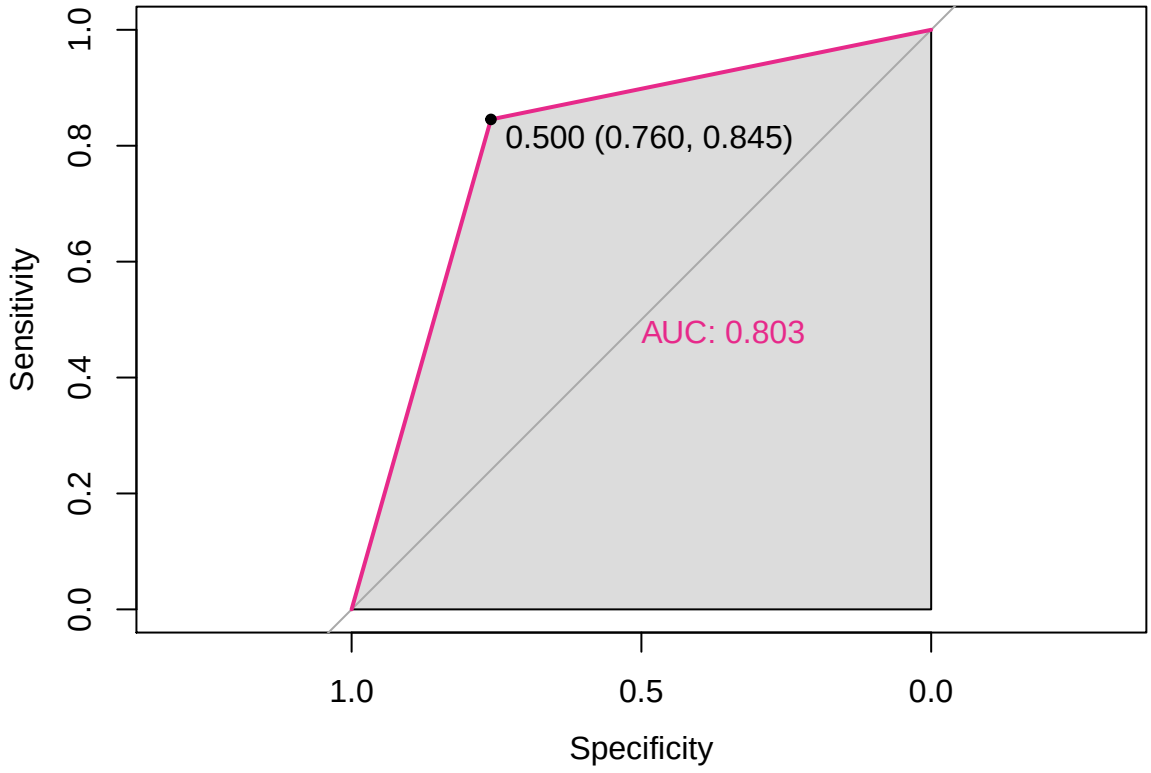

```
## AUC is 0.8843838 .
```

```
#ROC & AUC
roc.glm = roc(X_test$income,as.vector(ifelse(p_glm >ideal_cutoff_glm, 1,0)))
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
auc.glm = auc(roc.glm)
```



We can also look at the importance of each predictor individually using a filter approach.

To examine the absolute value of the t-statistic for each predictor:

```
varImp(model_glm)
```

```
##                                Overall
## age_stand                   16.657899380
## workclass Local-gov          5.010273174
## workclass Private            8.592660004
## workclass Self-emp-inc       1.178105542
## workclass Self-emp-not-inc   5.748216912
## workclass State-gov          5.969828984
## workclass Without-pay        0.010057488
## education 11th               2.444738336
## education 12th               2.088298859
## education 1st-4th            0.876638456
## education 5th-6th            2.434437489
## education 7th-8th            1.517831602
## education 9th                1.093268250
## education Assoc-acdm         8.521518872
## education Assoc-voc          7.403404102
## education Bachelors         11.994025713
## education Doctorate         13.928595342
## education HS-grad            5.449286030
## education Masters           14.216370974
## education Preschool          0.015997495
```

```
## education Prof-school              13.326598158
## education Some-college              7.120793610
## marital_status_group Never-married  8.315263416
## marital_status_group Separated      7.295792736
## marital_status_group Widowed        4.817276866
## marital_status_groupMarried         5.932115368
## occupation Armed-Forces             0.006644913
## occupation Craft-repair             6.082350505
## occupation Exec-managerial         15.114964425
## occupation Farming-fishing          5.801413485
## occupation Handlers-cleaners        3.149046518
## occupation Machine-op-inspct        1.356478787
## occupation Other-service            0.882410297
## occupation Priv-house-serv          0.108853864
## occupation Prof-specialty          11.429073033
## occupation Protective-serv          6.972472686
## occupation Sales                    7.382838326
## occupation Tech-support            12.427054643
## occupation Transport-moving         2.609108108
## relationship Not-in-family         11.886466142
## relationship Other-relative         8.766562333
## relationship Own-child             15.967292210
## relationship Unmarried              9.289612743
## relationship Wife                  10.248714016
## race Asian-Pac-Islander             4.507861767
## race Black                          3.090251043
## race Other                          0.620805783
## race White                          0.573957975
## sex Male                            1.047414269
## cap_gainLow                         0.153514399
## cap_gainMedium                      0.137040414
## cap_gainZero                        0.149955205
## cap_lossLow                         7.075363944
## cap_lossMedium                      2.974342755
## cap_lossZero                       12.602428522
## weekly_hours45 to 50 hrs           12.108429640
## weekly_hours50 to 60 hrs           11.888711261
## weekly_hours60 to 70 hrs            9.179098117
## weekly_hours70 to 80 hours          3.451521218
## weekly_hoursgreater than 80 hrs     2.370419701
## weekly_hoursless than 40 hrs       11.025821265
## native_region Central-Asia          5.439550343
## native_region East-Asia             5.453775198
## native_region Europe-East           0.494008280
## native_region Europe-West           1.491431159
## native_region South-America         3.719166290
## native_regionNorth America          2.900300938
```

According to the results of the varImp function, age, education - Bachelors, education - Doctorate, education - Masters, education- Prof - school, occupation Exec-managerial, occupation Prof-specialty, occupation Tech-support, relationship Not-in-family, relationship Own-child, capital loss of zero, hours per week (45 - 50, 50 -60, and less than 40) are the most important predictors for income status.

Deployment

The underlying code for this markdown can be found on github Assignment 1 The prediction model is deployed in ShinyApp.The application accepts inputs from user and classify income as above or below 50K.The url for Income Classifier

Conclusion and Discussion.

For the census adult income data set, since the model was built on the data collected in 1994, some values could be outdated and it's likely that the model doesn't not predict well nowadays. If other potential predictors could be collected, such as the city people reside in, the number of people in the household, the age group of children in the household, the highest education their parents received, the model could be further improved.

There are missing value and outliers in this imbalanced dataset, we removed missing values for the sake of limited time and used different ways to clean the data. We tried differnt supervised learning algorithms on different sets of features.

We then use accuracy and AUC to evaluate the models. For our training and testing sets, the different algorithms give relatively close results. More specifically, the random forest model gives a 80% overall accuracy, 81% balanced accuracy, and AUC 0.81; KNN with the default Euclidean distance metric gives an accuracy of 77%, balanced accuracy of 80% and AUC 0.79. The accuracy of logistic regression on transformed variables is 80% and the balanced accuracy is 81%, with AUC 0.81. When we removed the final weight variable and/or train models using the original age values, the accuracy of the models almost did not change at all, but the computational speed becomes much slower.

We chose the logistic regression model trained on the original age values for deployment, because of it's relatively high accuracy and ease of interpretation.

Bibliography

https://www.youtube.com/watch?v=mf1A_8K84rw&ab_channel=DavidDalpiaz

https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm

https://rpubs.com/vassitar/us_census_preprocessing

Intefrate.AI Inc. (2019). Respnsible AI Consumer Enterprise.

RPubs by Rstudio, https://rpubs.com/Cher/403319

Hadley Wickham & Garret Grolemund , R for Data Science, O'Reilly in January 2017