PQHS471_Midterm

Ruipeng Wei

2/26/2019

library package

```
set.seed(471)
library(caret)

## Warning: package 'caret' was built under R version 3.4.4

## Loading required package: lattice

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.4.4

library(simputation)
library(knitr)
library(ggthemes)

## Warning: package 'ggthemes' was built under R version 3.4.4

library(gridExtra)
library(scales)
library(dplyr)

## Warning: package 'dplyr' was built under R version 3.4.4

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:gridExtra':
##
##     combine

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(ggplot2)
library(psych)

## Warning: package 'psych' was built under R version 3.4.4
```

```
##
## Attaching package: 'psych'

## The following objects are masked from 'package:scales':
##
##     alpha, rescale

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha

library(randomForest)

## Warning: package 'randomForest' was built under R version 3.4.4

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:psych':
##
##     outlier

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:gridExtra':
##
##     combine

## The following object is masked from 'package:simputation':
##
##     na.roughfix

## The following object is masked from 'package:ggplot2':
##
##     margin
```

input data

```
test <- read.csv("census_test.csv")
train <- read.csv("census_train.csv")

#duplicates in training set
ifelse(length(unique(train[,1])) == nrow(train),"No duplicates","Duplicates d
etected!")

## [1] "Duplicates detected!"
```

```r
#missing data in training set
s <- vector()
train <- as.matrix(train)
train[train==" ?"] <- NA
train <- as.data.frame(train)
sum(!complete.cases(train))
```

```
## [1] 1829
```

```r
for(i in 1:ncol(train)){
    s[i] <- sum(is.na(train[,i]))
}
s <- cbind(colnames(train),s)
s
```

```
##                          s
##  [1,] "age"              "0"
##  [2,] "workclass"        "1404"
##  [3,] "fnlwgt"           "0"
##  [4,] "education"        "0"
##  [5,] "education.num"    "0"
##  [6,] "marital.status"   "0"
##  [7,] "occupation"       "1411"
##  [8,] "relationship"     "0"
##  [9,] "race"             "0"
## [10,] "sex"              "0"
## [11,] "capital.gain"     "0"
## [12,] "capital.loss"     "0"
## [13,] "hours.per.week"   "0"
## [14,] "native.country"   "437"
## [15,] "income"           "0"
```

We could see that there are 1404 missing data in workclass, 1411 missing data in occupation and 437 missing data in native country.

```r
#missing data in training set
table(rowSums(is.na(train)))
```

```
##
##     0     1     2     3
## 23171   425  1385    19
```

According to this result, there are 23171 observations without any missing data, 425 observations have one missing data, 1385 observations have two missing data while 19 observations have 3 missing data.

Training data has 25000 observations, observations have missing data are 1829 which less than 10% of all training data. If I remove all observations with missing data, it will not arouse serious problem. Thus, I will remove all observations with missing data.

```
train <- train[complete.cases(train),]
str(train)
```

```
## 'data.frame':    23171 obs. of  15 variables:
##  $ age           : Factor w/ 73 levels "17","18","19",..: 29 47 10 3 10 28
17 38 32 22 ...
##  $ workclass     : Factor w/ 8 levels " Federal-gov",..: 4 6 4 4 4 4 4 4 4
4 ...
##  $ fnlwgt        : Factor w/ 17865 levels "  12285","  13769",..: 9448 103
29 4830 4769 16592 17828 14983 8275 3071 1871 ...
##  $ education     : Factor w/ 16 levels " 10th"," 11th",..: 16 10 13 12 12
16 10 7 10 9 ...
##  $ education.num : Factor w/ 16 levels " 1"," 2"," 3",..: 10 13 14 9 9 10
13 5 13 11 ...
##  $ marital.status: Factor w/ 7 levels " Divorced"," Married-AF-spouse",..:
3 3 5 5 1 6 5 3 3 3 ...
##  $ occupation    : Factor w/ 14 levels " Adm-clerical",..: 3 12 10 8 7 7 1
0 4 10 7 ...
##  $ relationship  : Factor w/ 6 levels " Husband"," Not-in-family",..: 1 1
2 4 2 2 2 1 1 1 ...
##  $ race          : Factor w/ 5 levels " Amer-Indian-Eskimo",..: 5 5 5 3 5
3 3 5 5 5 ...
##  $ sex           : Factor w/ 2 levels " Female"," Male": 2 2 2 2 2 2 2 2 2
2 ...
##  $ capital.gain  : Factor w/ 118 levels "    0","  114",..: 1 1 98 1 1 1 1
1 1 1 ...
##  $ capital.loss  : Factor w/ 88 levels "   0"," 155",..: 1 1 1 1 1 1 1 1 1
1 ...
##  $ hours.per.week: Factor w/ 91 levels " 1"," 2"," 3",..: 40 15 40 40 40 4
0 50 45 45 40 ...
##  $ native.country: Factor w/ 41 levels " Cambodia"," Canada",..: 39 39 39
39 39 39 39 39 22 39 ...
##  $ income        : Factor w/ 2 levels " <=50K"," >50K": 2 1 2 1 1 1 2 2 2
2 ...
```

```
train$age <- as.numeric(as.character(train$age))
train$fnlwgt <- as.numeric(as.character(train$fnlwgt))
train$education.num <- as.integer(as.character(train$education.num))
train$capital.gain <- as.numeric(as.character(train$capital.gain))
train$capital.loss <- as.numeric(as.character(train$capital.loss))
train$hours.per.week <- as.numeric(as.character(train$hours.per.week))
train$workclass <- factor(train$workclass, levels(train$workclass), ordered=T
)
train$education <- factor(train$education, levels=c(" Preschool"," 1st-4th","
5th-6th"," 7th-8th"," 9th"," 10th"," 11th"," 12th"," HS-grad"," Some-college"
," Assoc-voc"," Assoc-acdm"," Bachelors"," Masters"," Prof-school"," Doctorat
e"),ordered = T)
train$marital.status <- factor(train$marital.status,levels(train$marital.stat
us),ordered = T)
train$occupation <- factor(train$occupation,levels(train$occupation),ordered
```

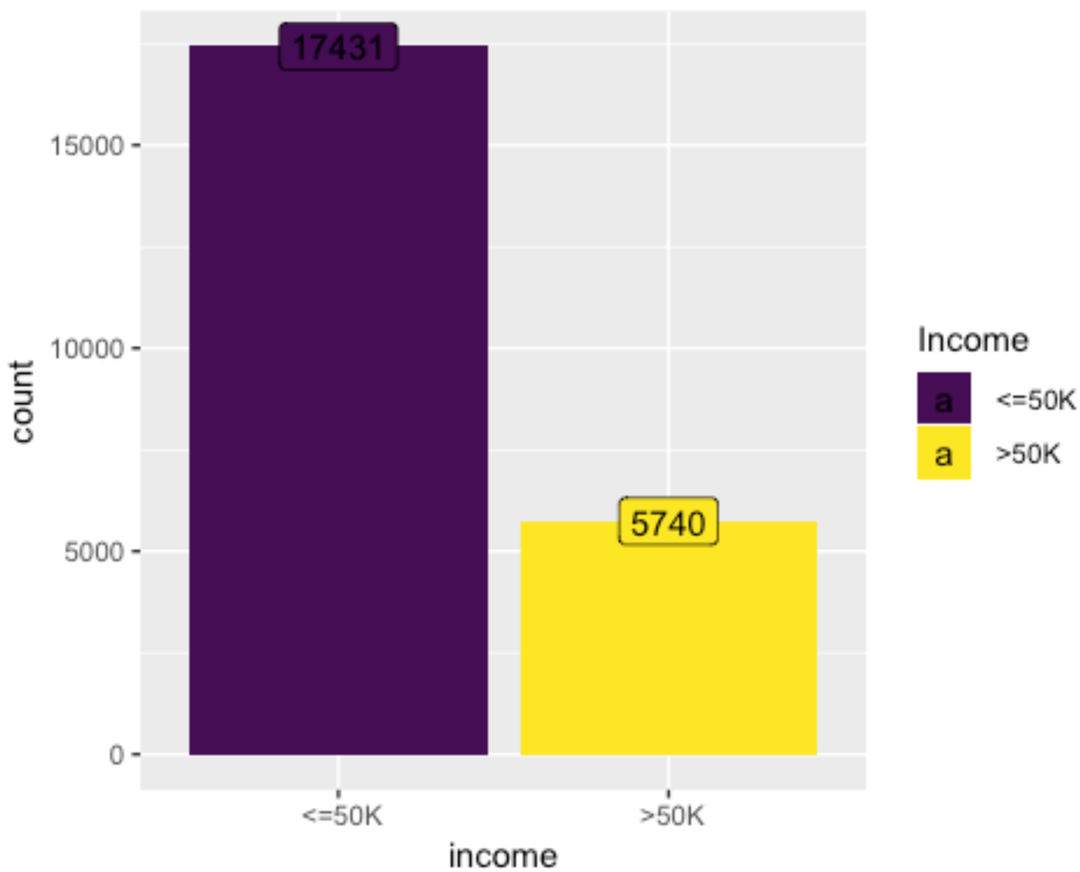
```
= T)
train$relationship <- factor(train$relationship,levels(train$relationship),or
dered = T)
train$race <- factor(train$race,levels(train$race),ordered = T)
train$sex <- factor(train$sex,levels(train$sex),ordered = T)
train$native.country <- factor(train$native.country,levels(train$native.count
ry),ordered = T)
train$income <- factor(train$income,levels(train$income),ordered = T)
summary(train)
```

```
##       age                    workclass          fnlwgt
##  Min.   :17.00    Private         :17136   Min.   :  13769
##  1st Qu.:28.00    Self-emp-not-inc: 1916   1st Qu.: 117674
##  Median :37.00    Local-gov       : 1582   Median : 178344
##  Mean   :38.47    State-gov       :  977   Mean   : 189752
##  3rd Qu.:47.00    Self-emp-inc    :  827   3rd Qu.: 237528
##  Max.   :90.00    Federal-gov     :  723   Max.   :1484705
##                  (Other)          :   10
##          education     education.num                  marital.status
##   HS-grad      :7555   Min.   : 1.00    Divorced             : 3254
##   Some-college:5125   1st Qu.: 9.00    Married-AF-spouse    :   17
##   Bachelors    :3869   Median :10.00    Married-civ-spouse   :10805
##   Masters      :1231   Mean   :10.11    Married-spouse-absent:  288
##   Assoc-voc    : 998   3rd Qu.:13.00    Never-married        : 7458
##   11th         : 802   Max.   :16.00    Separated            :  721
##  (Other)       :3591                    Widowed              :  628
##            occupation            relationship
##   Prof-specialty :3111    Husband        :9568
##   Craft-repair   :3110    Not-in-family  :5934
##   Exec-managerial:3088    Other-relative: 701
##   Adm-clerical   :2850    Own-child      :3425
##   Sales          :2730    Unmarried      :2446
##   Other-service  :2478    Wife           :1097
##  (Other)         :5804
##                    race           sex          capital.gain
##   Amer-Indian-Eskimo:  219    Female: 7466   Min.   :    0
##   Asian-Pac-Islander:  696    Male  :15705   1st Qu.:    0
##   Black             : 2131                   Median :    0
##   Other             :  186                   Mean   : 1078
##   White             :19939                   3rd Qu.:    0
##                                              Max.   :99999
##
##   capital.loss     hours.per.week        native.country      income
##  Min.   :   0.00   Min.   : 1.00    United-States:21122    <=50K:17431
##  1st Qu.:   0.00   1st Qu.:40.00    Mexico       :  480    >50K : 5740
##  Median :   0.00   Median :40.00    Philippines  :  145
##  Mean   :  89.11   Mean   :40.93    Germany      :  103
##  3rd Qu.:   0.00   3rd Qu.:45.00    Puerto-Rico  :   85
##  Max.   :3900.00   Max.   :99.00    Canada       :   82
##                                    (Other)       : 1154
```

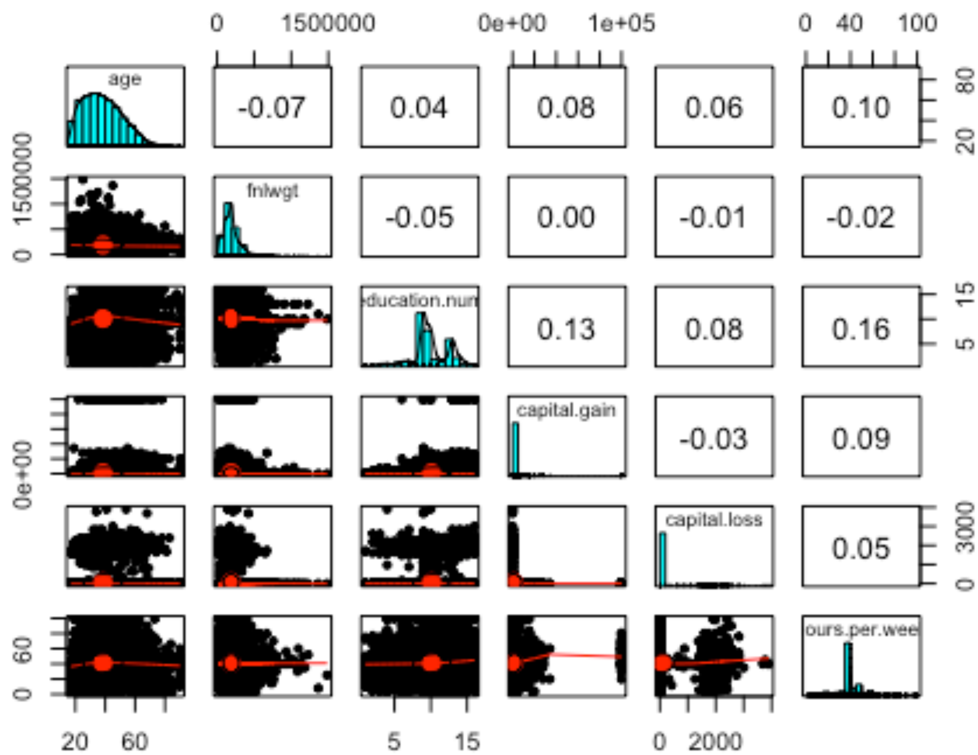
After removing missing data, there are 23717 observations in the training dataset. And the outcome - income - is a binary outcome which contains two levels “>50k” and “<=50k”. Thus, logistic regression will be the first trial. Before doing analysis, the distribution of capital gain in a little weird.Thus I will do some data exploratory before build model.

explortatory of training data

```
train %>%
ggplot(aes(x=income, fill = income))+
        geom_histogram(stat = "count")+
        geom_label(stat='count',aes(label=..count..))+
        labs(fill = "Income")

## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

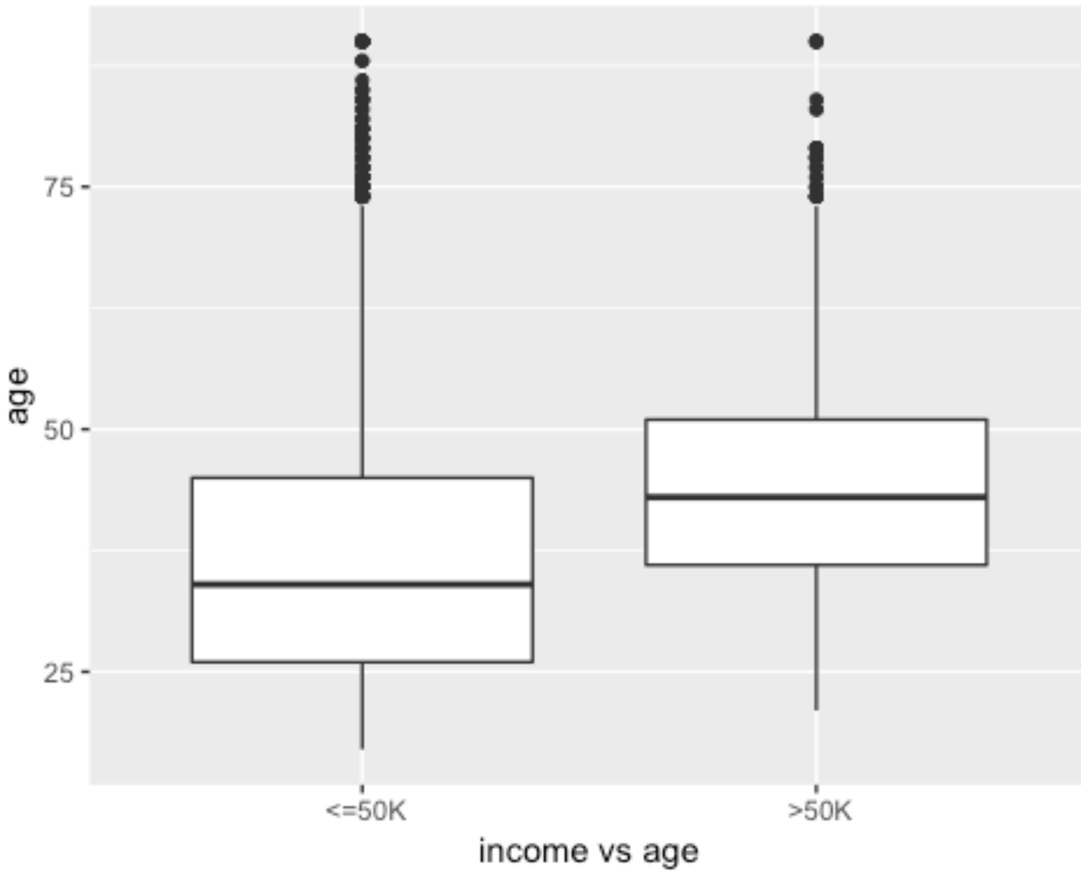



```
pairs.panels(train[c(1,3,5,11,12,13)])  # select columns 1-4
```



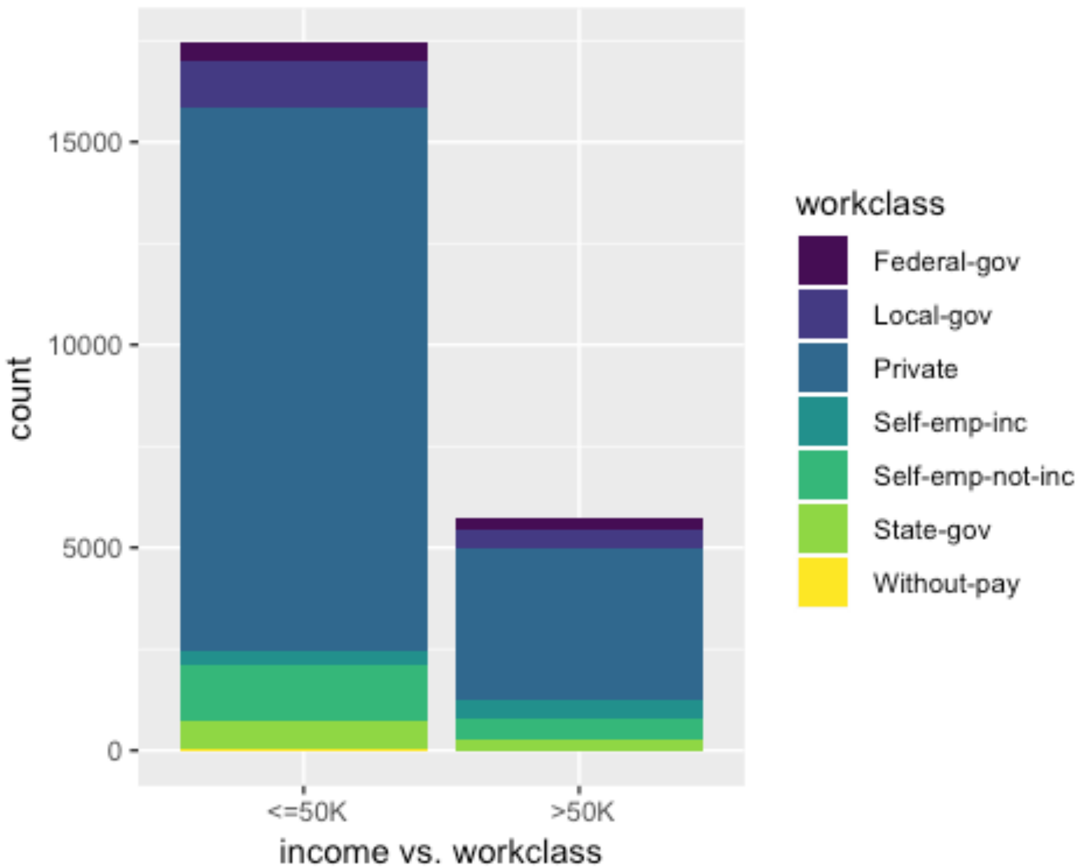
The numeric variables do not correlate with each other tightly.

```
ggplot(train,aes(x= income, y = age))+
  geom_boxplot()+
        labs(x = "income vs age")
```



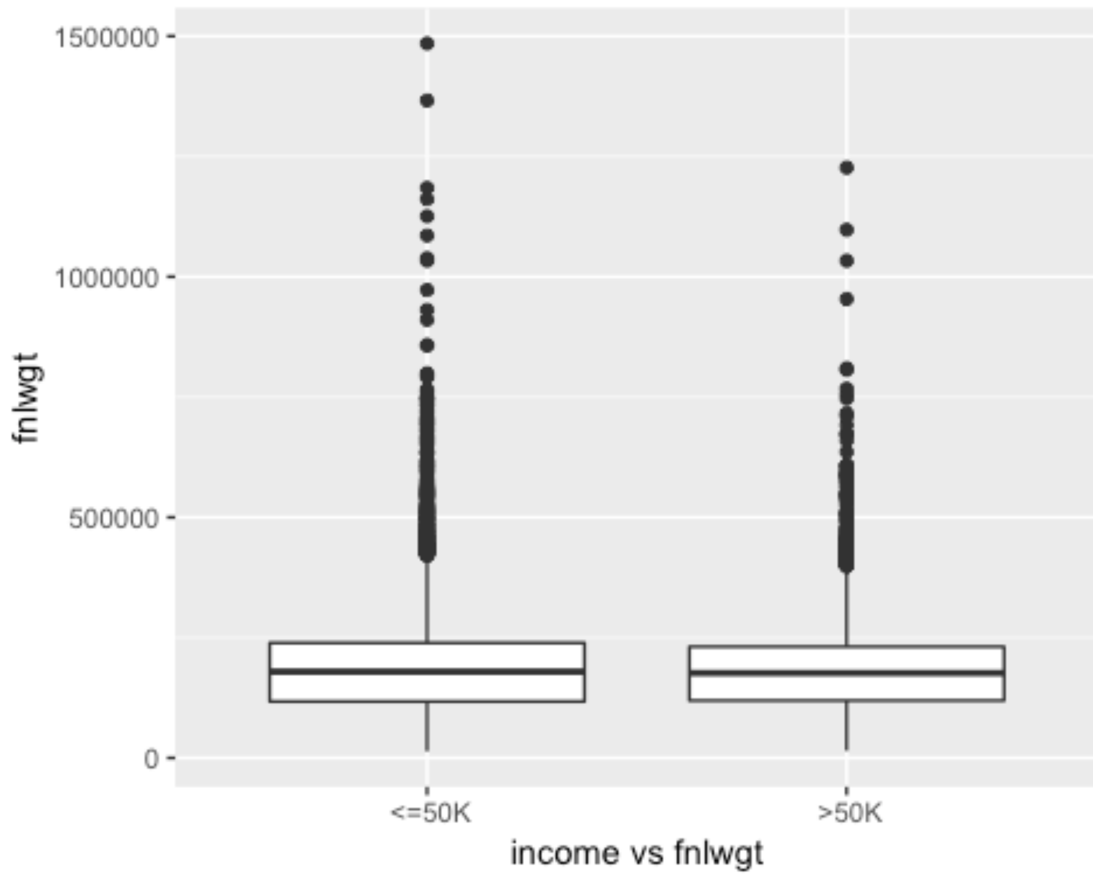
It is obvious that people whose income >50k have higher average/median age.

```
ggplot(train,aes(x=income, fill=workclass))+
  geom_histogram(stat = "count")+
        labs(x = "income vs. workclass")

## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



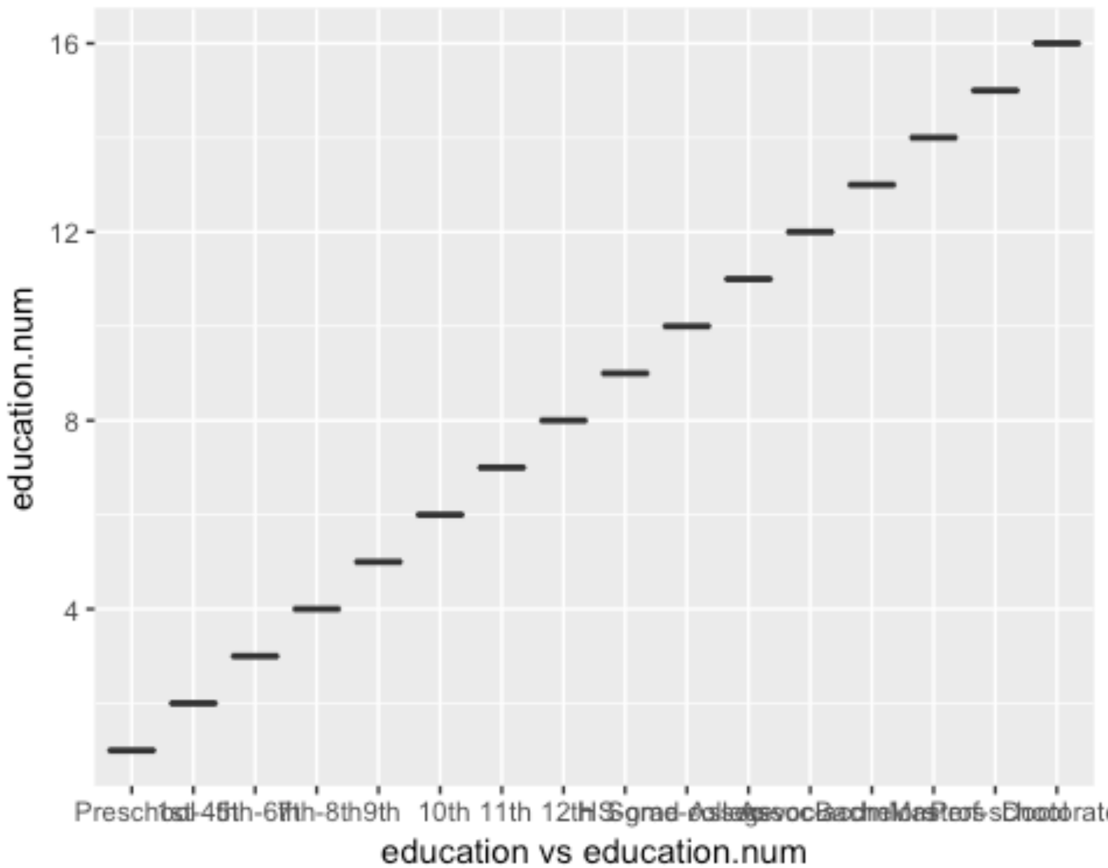
In both kind of income, people have workclass in private are the most.

```
ggplot(train,aes(x= income, y = fnlwgt))+
  geom_boxplot()+
        labs(x = "income vs fnlwgt")
```



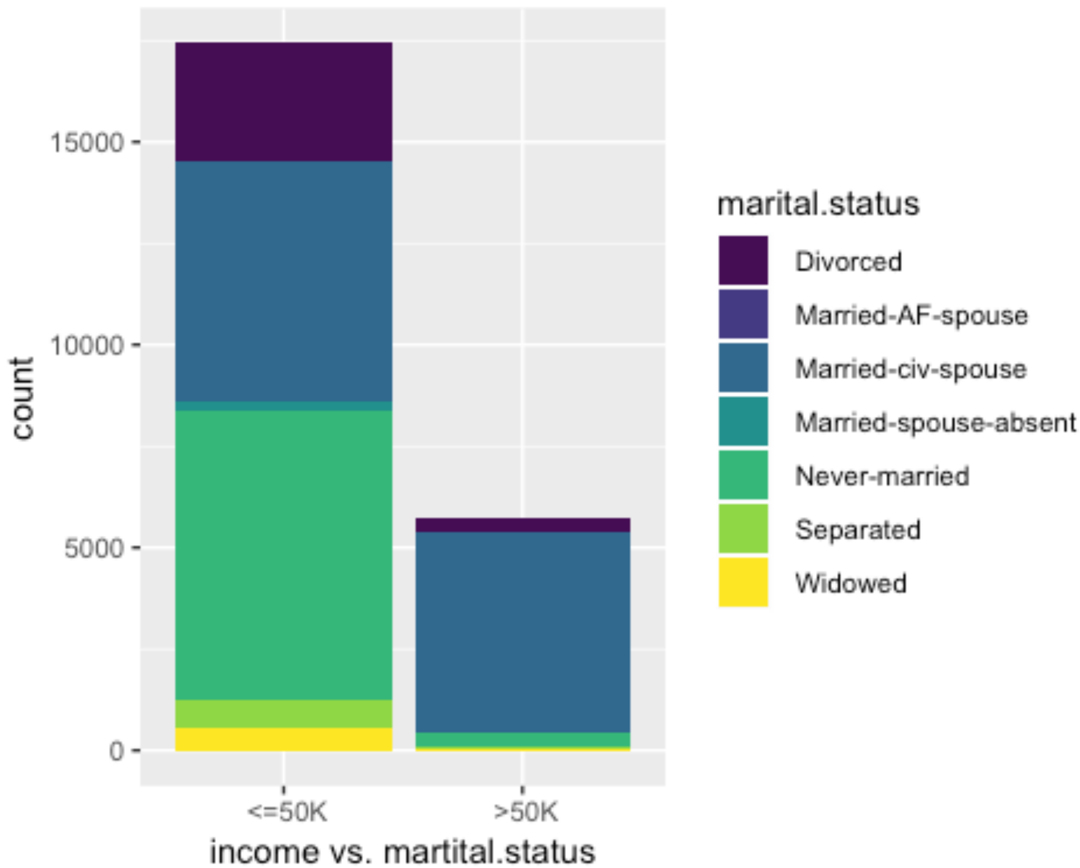
The distribution of the the final sampling weight of the two groups are very similar from the box plot.

```
ggplot(train,aes(x= education, y = education.num))+
  geom_boxplot()+
        labs(x = "education vs education.num")
```



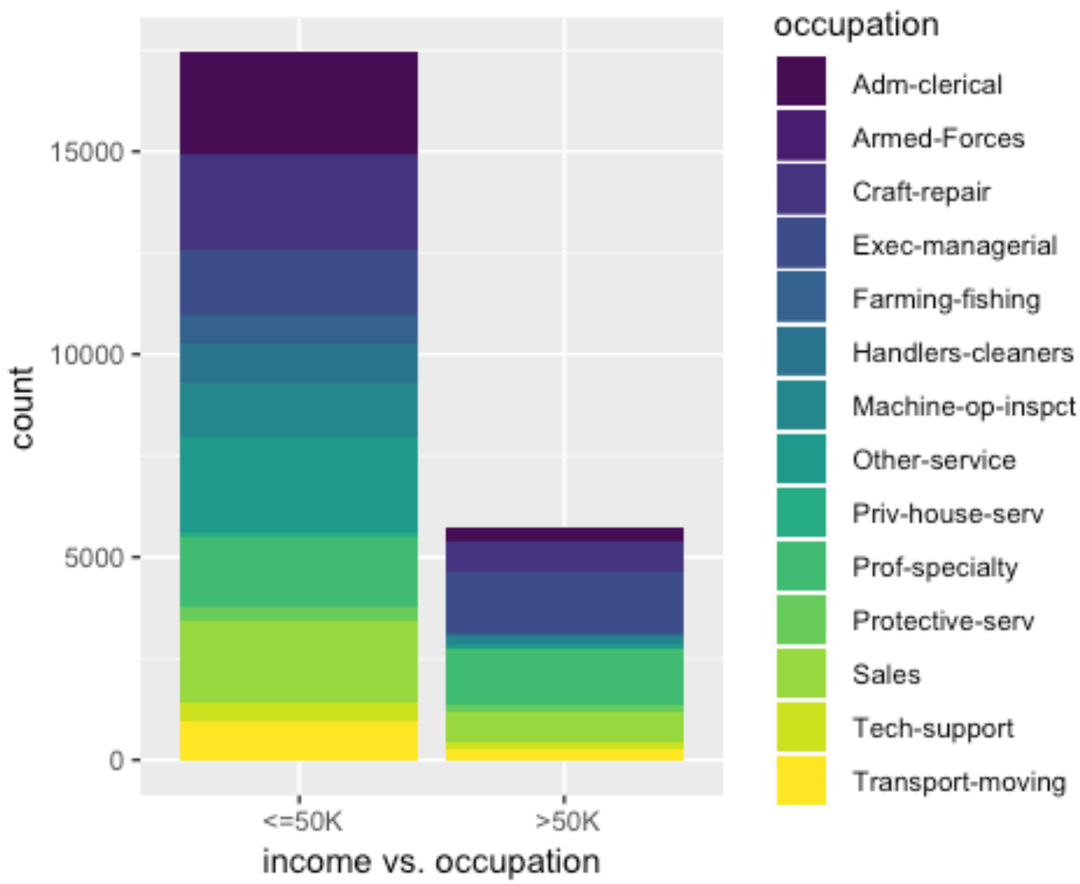
The education and education.num are highly correlated.When the education.num increase from 1 to 16, the education are also change from Preschool to Doctorate. Thus, may be only one of the two will be enough to be considered in the prediction model.

```
ggplot(train,aes(x=income, fill=marital.status))+
  geom_histogram(stat = "count")+
        labs(x = "income vs. martital.status")

## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



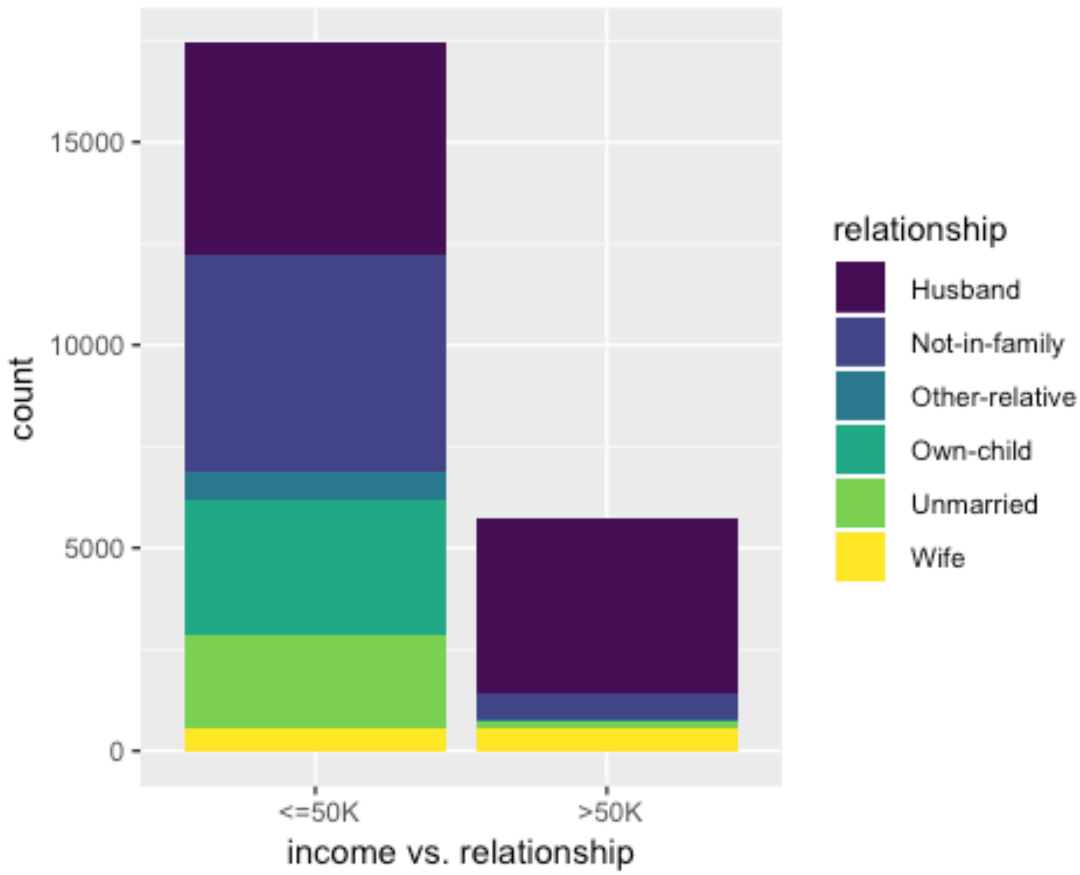
The distribution of the marital status in two group of income are very different. Most people whose income is less than 50K are either divorced, married-civ spouse or never-married. But most people whose income >50K are Married-civ-spouse.

```
ggplot(train,aes(x=income, fill=occupation))+
  geom_histogram(stat = "count")+
        labs(x = "income vs. occupation")

## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



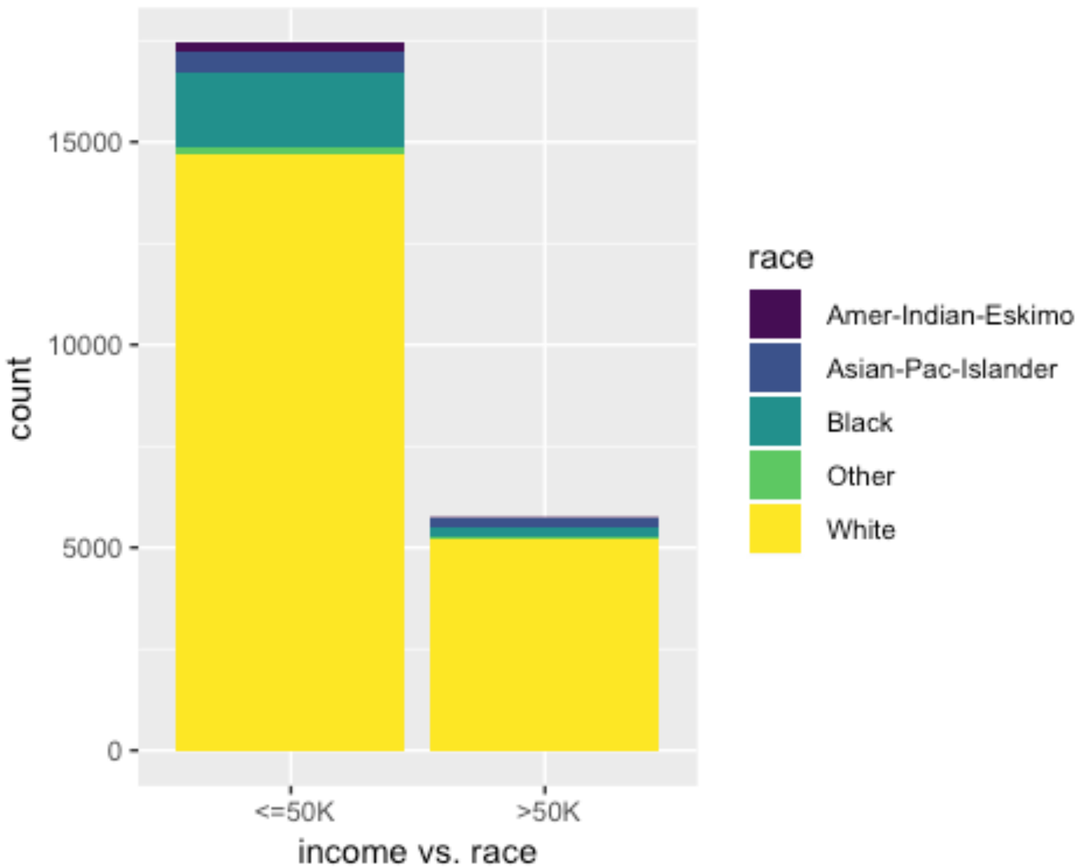
The occupations distribution of the two income groups are very different. People whose income are >50K are unlikely to be Handlers-cleaners, Machine-op-inspect and Other-services.

```
ggplot(train,aes(x=income, fill=relationship))+
  geom_histogram(stat = "count")+
        labs(x = "income vs. relationship")

## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



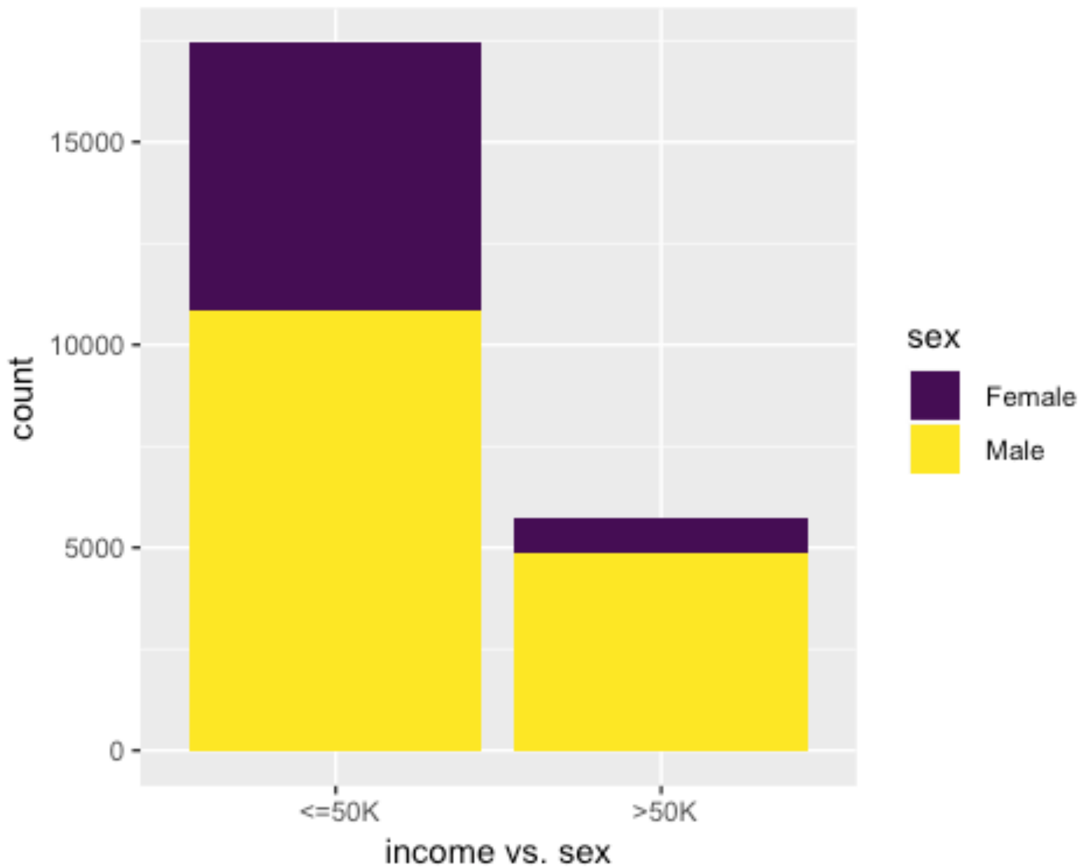
People whose income are > 50K have high possibility be husband in relationship. But people whose income are <=50K are more likely to be in various relationship.

```
ggplot(train,aes(x=income, fill=race))+
  geom_histogram(stat = "count")+
        labs(x = "income vs. race")

## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



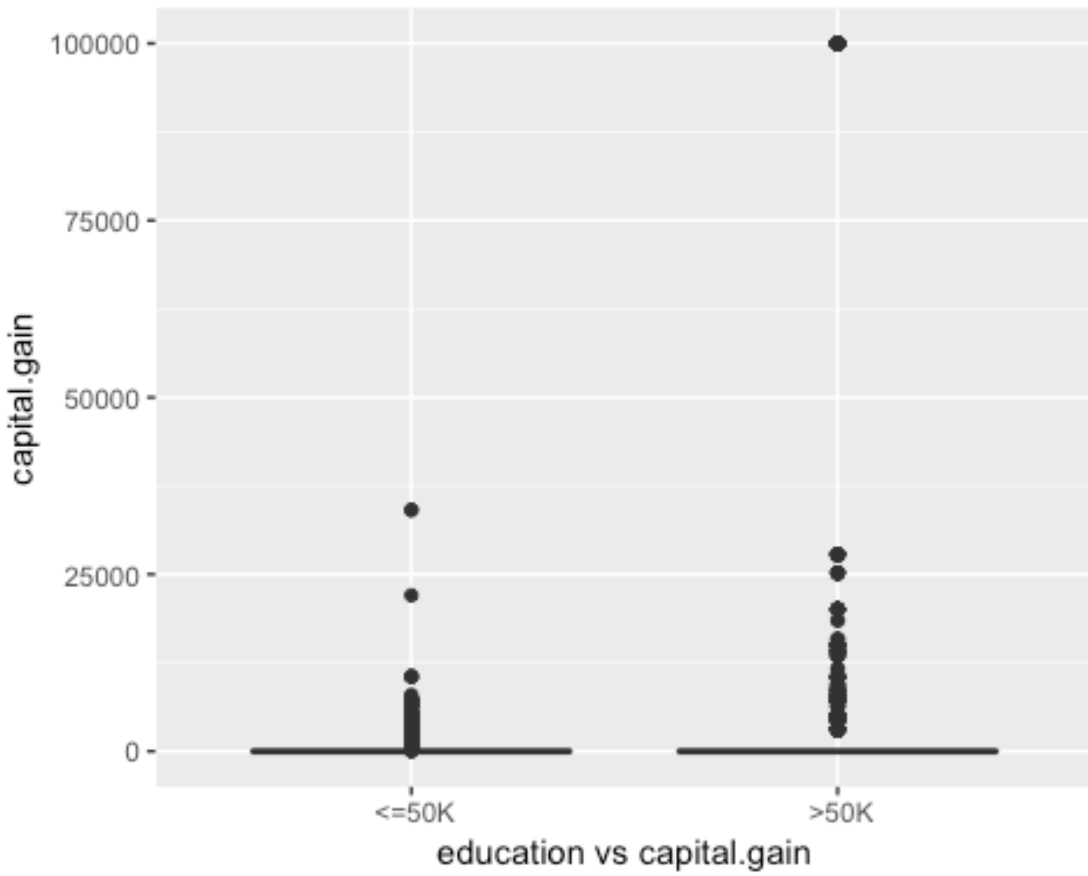
Race in Black has more proportion in the income <= 50K group.

```
ggplot(train,aes(x=income, fill=sex))+
  geom_histogram(stat = "count")+
        labs(x = "income vs. sex")

## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



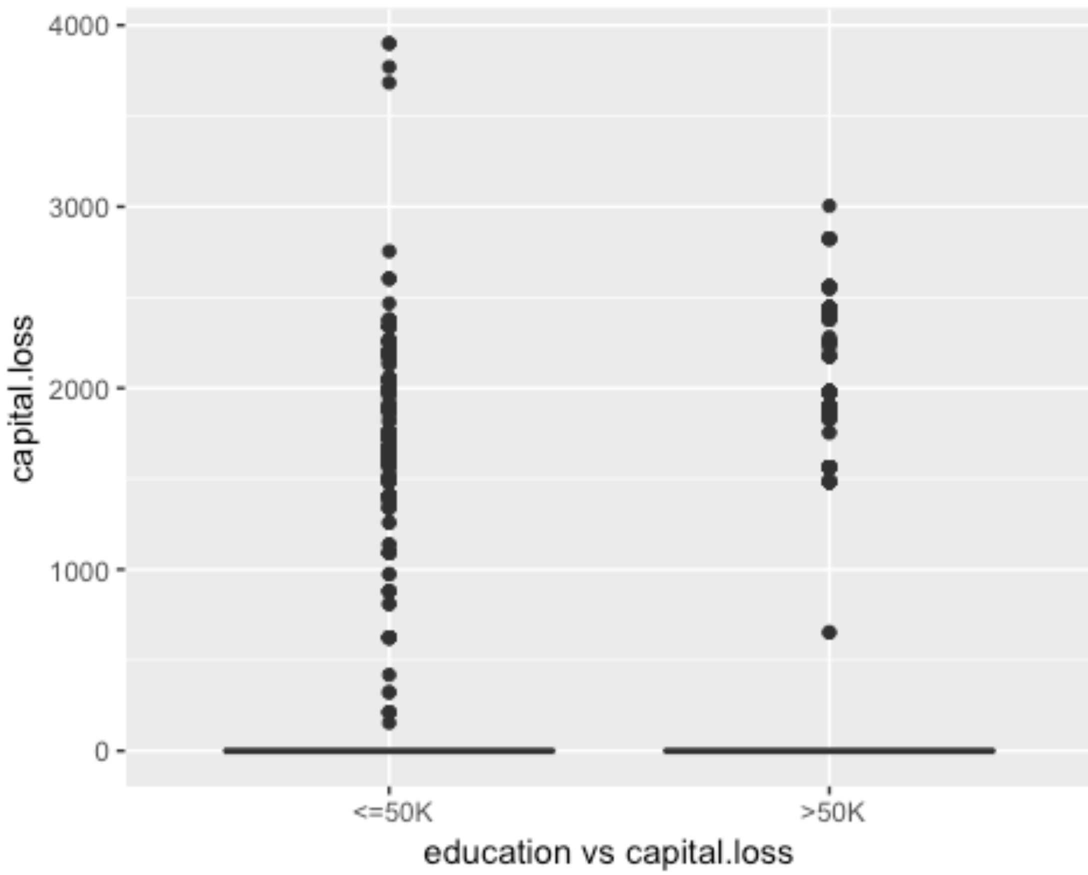
Female are less in the income group >50K.

```
ggplot(train,aes(x= income, y = capital.gain))+
  geom_boxplot()+
        labs(x = "education vs capital.gain")
```



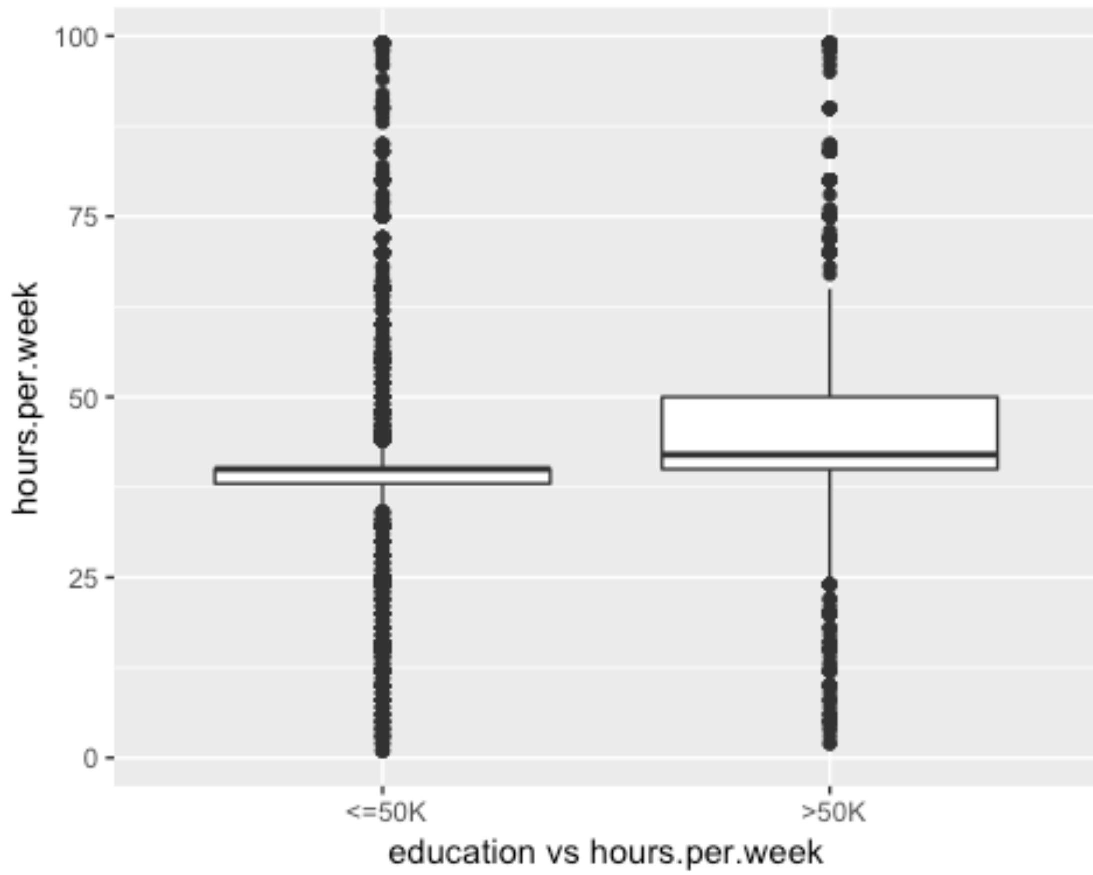
The distribution of capital.gain in the two income groups are right skewed. And the median of the two are both 0. But people whose income >50K has some outliers fall on about 100000.

```
ggplot(train,aes(x= income, y = capital.loss))+
  geom_boxplot()+
        labs(x = "education vs capital.loss")
```



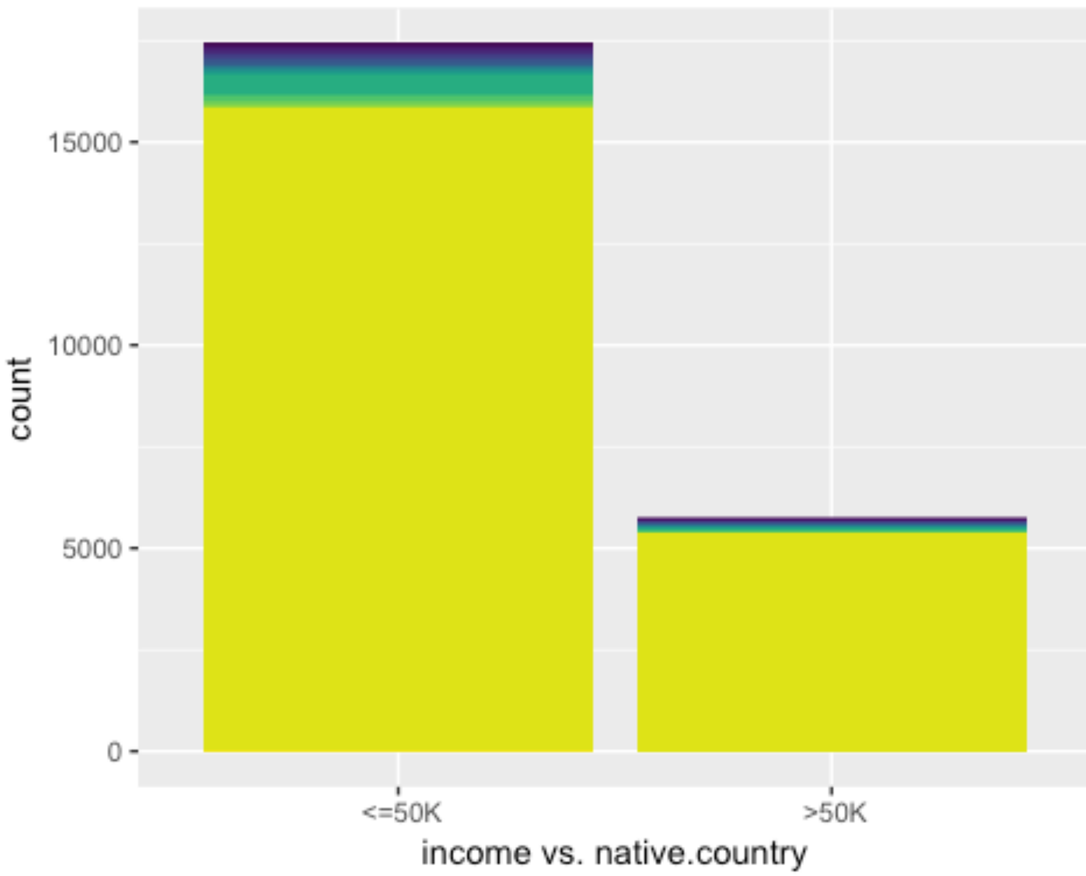
The distribution of capital.loss of the two income groups are also right skewed. But the group <=50K has much more outliters.

```
ggplot(train,aes(x= income, y = hours.per.week))+
  geom_boxplot()+
        labs(x = "education vs hours.per.week")
```



The range of the hours.per.week in the two groups are in the same range, but the median and mean of the >50K are larger.

```
ggplot(train,aes(x=income, fill=native.country))+
  geom_histogram(stat = "count")+
        labs(x = "income vs. native.country")+
  theme(legend.position="none")

## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



It is hard to tell which country contribute to which group more except for United-States.

logistic regression

```
log.m1 <- glm(income ~ age + workclass + fnlwgt + education.num + marital.sta
tus + occupation + relationship + race + sex + capital.gain + capital.loss +
hours.per.week + native.country, data = train, family="binomial")

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

summary(log.m1)

##
## Call:
## glm(formula = income ~ age + workclass + fnlwgt + education.num +
##     marital.status + occupation + relationship + race + sex +
##     capital.gain + capital.loss + hours.per.week + native.country,
##     family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.4217  -0.5096  -0.1808  -0.0079   4.0650
##
```

```
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)    
## (Intercept)         -1.171e+01  4.175e+01  -0.281 0.779032    
## age                  2.747e-02  1.945e-03  14.124  < 2e-16 ***
## workclass.L         -7.832e+00  1.364e+02  -0.057 0.954216    
## workclass.Q         -6.788e+00  1.313e+02  -0.052 0.958755    
## workclass.C         -5.318e+00  9.823e+01  -0.054 0.956826    
## workclass^4         -2.675e+00  5.817e+01  -0.046 0.963321    
## workclass^5         -1.719e+00  2.625e+01  -0.065 0.947790    
## workclass^6         -6.360e-01  7.916e+00  -0.080 0.935958    
## fnlwgt               8.050e-07  2.017e-07   3.990 6.59e-05 ***
## education.num        2.823e-01  1.113e-02  25.360  < 2e-16 ***
## marital.status.L    -1.654e+00  3.027e-01  -5.463 4.67e-08 ***
## marital.status.Q    -4.987e-01  1.912e-01  -2.609 0.009091 ** 
## marital.status.C     2.418e+00  3.533e-01   6.843 7.75e-12 ***
## marital.status^4    -1.606e+00  3.936e-01  -4.080 4.50e-05 ***
## marital.status^5    -1.177e-01  2.683e-01  -0.439 0.660917    
## marital.status^6     2.143e-01  2.363e-01   0.907 0.364506    
## occupation.L         4.522e-01  6.557e-01   0.690 0.490438    
## occupation.Q         1.374e+00  6.987e-01   1.966 0.049268 *  
## occupation.C         3.254e-01  4.571e-01   0.712 0.476631    
## occupation^4        -1.547e+00  5.065e-01  -3.055 0.002254 ** 
## occupation^5        -1.242e+00  9.021e-01  -1.377 0.168431    
## occupation^6         9.332e-01  7.750e-01   1.204 0.228518    
## occupation^7         3.635e-01  9.836e-01   0.370 0.711695    
## occupation^8         5.987e-01  6.109e-01   0.980 0.327116    
## occupation^9        -1.488e+00  6.759e-01  -2.202 0.027687 *  
## occupation^10       -1.609e+00  6.876e-01  -2.340 0.019288 *  
## occupation^11        1.157e+00  2.354e-01   4.917 8.80e-07 ***
## occupation^12        1.284e+00  9.559e-01   1.343 0.179172    
## occupation^13        1.275e+00  8.301e-01   1.536 0.124632    
## relationship.L       7.559e-01  9.755e-02   7.748 9.32e-15 ***
## relationship.Q       1.040e+00  2.780e-01   3.740 0.000184 ***
## relationship.C       7.303e-01  1.164e-01   6.272 3.55e-10 ***
## relationship^4      -5.975e-01  2.329e-01  -2.565 0.010313 *  
## relationship^5      -1.195e-01  1.999e-01  -0.598 0.549853    
## race.L               1.373e-01  2.170e-01   0.633 0.526735    
## race.Q              -1.105e-01  1.904e-01  -0.580 0.561749    
## race.C               7.300e-01  2.710e-01   2.694 0.007063 ** 
## race^4               1.131e-01  2.111e-01   0.536 0.592127    
## sex.L                6.415e-01  6.681e-02   9.602  < 2e-16 ***
## capital.gain         3.311e-04  1.254e-05  26.412  < 2e-16 ***
## capital.loss         6.452e-04  4.420e-05  14.598  < 2e-16 ***
## hours.per.week       3.162e-02  1.959e-03  16.145  < 2e-16 ***
## native.country.L    -2.380e+00  9.144e+01  -0.026 0.979237    
## native.country.Q     1.875e+00  1.188e+02   0.016 0.987406    
## native.country.C     1.220e+00  1.501e+02   0.008 0.993515    
## native.country^4     3.066e+00  7.222e+01   0.042 0.966133    
## native.country^5     5.311e-01  1.696e+02   0.003 0.997501    
## native.country^6    -1.740e+00  6.724e+01  -0.026 0.979359    
```

```
## native.country^7  -3.895e+00  1.393e+02  -0.028 0.977690
## native.country^8   2.803e+00  1.532e+02   0.018 0.985403
## native.country^9   5.561e-01  5.134e+01   0.011 0.991358
## native.country^10 -3.090e+00  1.701e+02  -0.018 0.985507
## native.country^11  2.242e+00  9.088e+01   0.025 0.980318
## native.country^12  5.558e+00  1.418e+02   0.039 0.968746
## native.country^13  1.981e+00  1.418e+02   0.014 0.988850
## native.country^14 -1.366e+00  7.122e+01  -0.019 0.984696
## native.country^15 -1.959e+00  1.820e+02  -0.011 0.991416
## native.country^16 -4.682e+00  8.530e+01  -0.055 0.956226
## native.country^17 -2.493e-01  1.324e+02  -0.002 0.998498
## native.country^18  5.801e+00  1.582e+02   0.037 0.970747
## native.country^19  4.212e+00  8.926e+01   0.047 0.962369
## native.country^20  3.261e-02  1.793e+02   0.000 0.999855
## native.country^21 -1.932e+00  9.137e+01  -0.021 0.983132
## native.country^22 -1.468e+00  1.333e+02  -0.011 0.991208
## native.country^23 -3.701e+00  1.830e+02  -0.020 0.983869
## native.country^24 -1.048e-01  8.618e+01  -0.001 0.999030
## native.country^25  4.420e-01  1.729e+02   0.003 0.997960
## native.country^26 -1.752e+00  1.280e+02  -0.014 0.989077
## native.country^27 -1.693e+00  1.239e+02  -0.014 0.989098
## native.country^28  5.906e+00  2.078e+02   0.028 0.977326
## native.country^29 -5.306e-01  1.367e+02  -0.004 0.996903
## native.country^30 -3.091e+00  1.596e+02  -0.019 0.984547
## native.country^31  2.511e+00  5.976e+01   0.042 0.966480
## native.country^32 -7.144e+00  1.994e+02  -0.036 0.971425
## native.country^33 -1.156e-01  1.471e+02  -0.001 0.999373
## native.country^34  1.150e+00  2.389e+02   0.005 0.996159
## native.country^35 -1.662e+00  1.587e+02  -0.010 0.991645
## native.country^36 -3.394e+00  2.896e+02  -0.012 0.990651
## native.country^37  3.298e+00  1.538e+02   0.021 0.982888
## native.country^38  1.486e-01  1.051e+02   0.001 0.998872
## native.country^39 -6.170e+00  3.016e+02  -0.020 0.983680
## native.country^40 -3.435e+00  1.167e+02  -0.029 0.976516
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 25943  on 23170  degrees of freedom
## Residual deviance: 14792  on 23089  degrees of freedom
## AIC: 14956
##
## Number of Fisher Scoring iterations: 13
```

```
log.m1$rule.5 <- ifelse(log.m1$fitted.values >= 0.5,"predicted >50K", "predic
ted <=50K")
table(log.m1$rule.5,train$income)
```

```
##
##                     <=50K  >50K
##   predicted <=50K  16189  2197
##   predicted >50K    1242  3543

pre_test_log.m1 <- table(log.m1$rule.5,train$income)

#accuracy in train data
sum(diag(pre_test_log.m1))/sum(pre_test_log.m1)

## [1] 0.8515817
```

This is the result when removing education variable and using all the other variables. The accuracy of this model on test data set is about 85.2%.

According to the significance of the variables in the model, workclass and native country do not have any significance, occupation has significance only in several levels. Thus, it is reasonable to remove these three variables.

```
log.m2 <- glm(income ~ age + fnlwgt + education.num + marital.status + relati
onship + race + sex + capital.gain + capital.loss + hours.per.week, data = tr
ain, family="binomial")

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

log.m2$rule.5 <- ifelse(log.m2$fitted.values >= 0.5,"predicted >50K", "predic
ted <=50K")
table(log.m2$rule.5,train$income)

##
##                     <=50K  >50K
##   predicted <=50K  16170  2402
##   predicted >50K    1261  3338

pre_test_log.m2 <- table(log.m2$rule.5,train$income)
summary(log.m2)

##
## Call:
## glm(formula = income ~ age + fnlwgt + education.num + marital.status +
##     relationship + race + sex + capital.gain + capital.loss +
##     hours.per.week, family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.3316  -0.5513  -0.2059  -0.0275   3.7073
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -9.049e+00  2.170e-01 -41.707  < 2e-16 ***
## age                2.700e-02  1.842e-03  14.656  < 2e-16 ***
## fnlwgt             8.201e-07  1.945e-07   4.217 2.48e-05 ***
```

```
## education.num     3.628e-01  9.253e-03  39.214  < 2e-16 ***
## marital.status.L -1.613e+00  2.940e-01  -5.486 4.11e-08 ***
## marital.status.Q -4.410e-01  1.869e-01  -2.360  0.01828 *
## marital.status.C  2.305e+00  3.429e-01   6.721 1.81e-11 ***
## marital.status^4 -1.564e+00  3.815e-01  -4.101 4.12e-05 ***
## marital.status^5 -1.521e-01  2.601e-01  -0.585  0.55875
## marital.status^6  2.563e-01  2.306e-01   1.112  0.26625
## relationship.L    6.951e-01  9.493e-02   7.322 2.45e-13 ***
## relationship.Q    1.118e+00  2.701e-01   4.138 3.50e-05 ***
## relationship.C    7.266e-01  1.138e-01   6.386 1.70e-10 ***
## relationship^4   -6.322e-01  2.271e-01  -2.783  0.00538 **
## relationship^5   -1.286e-01  1.950e-01  -0.660  0.50952
## race.L            9.472e-02  2.065e-01   0.459  0.64652
## race.Q            8.438e-02  1.804e-01   0.468  0.63996
## race.C            7.761e-01  2.441e-01   3.179  0.00148 **
## race^4            3.669e-01  1.861e-01   1.971  0.04871 *
## sex.L             6.074e-01  6.504e-02   9.338  < 2e-16 ***
## capital.gain      3.280e-04  1.215e-05  26.998  < 2e-16 ***
## capital.loss      6.624e-04  4.306e-05  15.383  < 2e-16 ***
## hours.per.week    3.081e-02  1.832e-03  16.820  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 25943  on 23170  degrees of freedom
## Residual deviance: 15465  on 23148  degrees of freedom
## AIC: 15511
##
## Number of Fisher Scoring iterations: 7
```

```
#accuracy in train data
sum(diag(pre_test_log.m2))/sum(pre_test_log.m2)
```

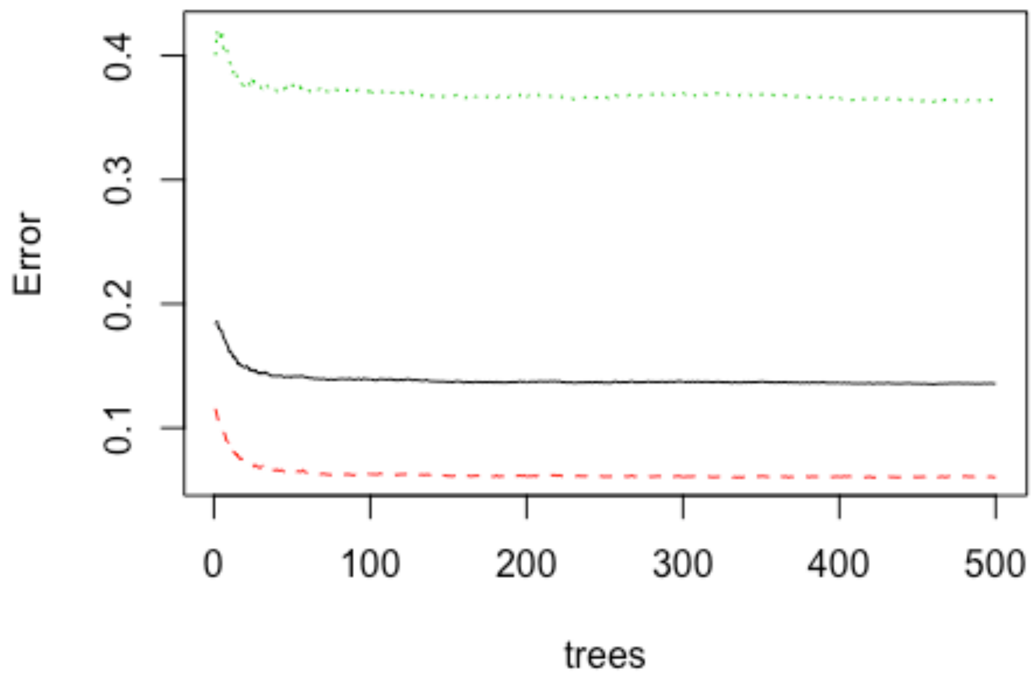
```
## [1] 0.8419145
```

The accuracy decrease to 84.2% which is acceptable.

All above are logistic regression models, which are very little flexible. Thus, a flexible method, random forest is worth to try.

randomForest

```
rf1 <- randomForest(income ~ age + workclass + fnlwgt + education.num + marit
al.status + occupation + relationship + race + sex + capital.gain + capital.l
oss + hours.per.week + native.country, data = train)

matplot(rf1$err.rate, type='l', xlab='trees', ylab='Error')
```



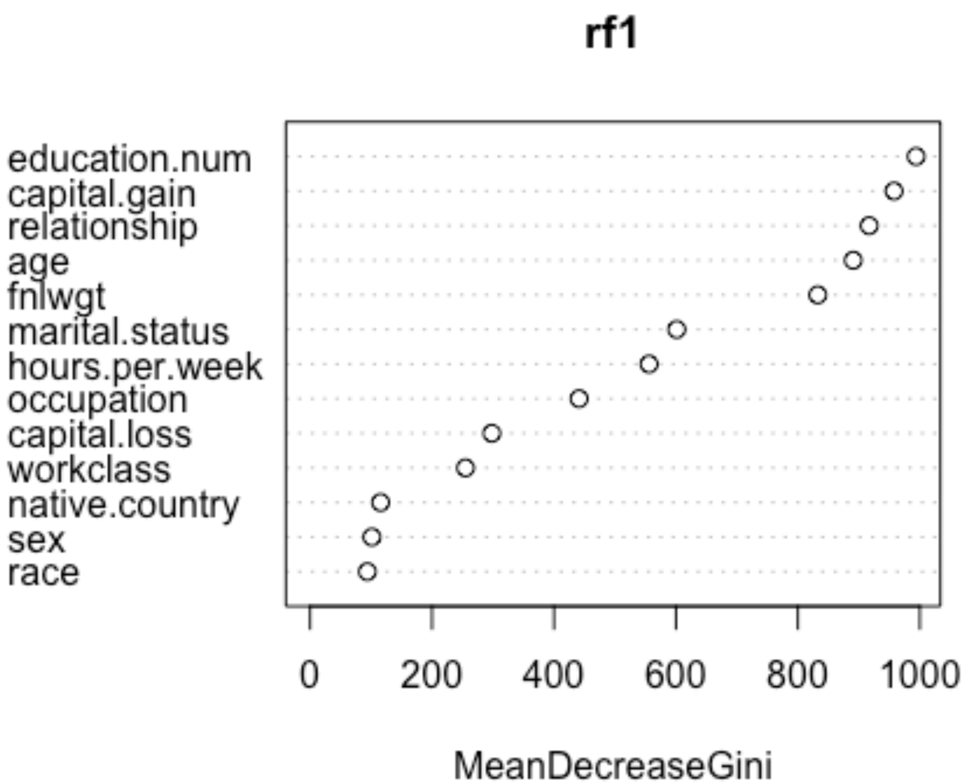

```
table(train$income,predict(rf1))

##
##          <=50K  >50K
##    <=50K 16377  1054
##    >50K   2093  3647

pre_test_rf1 <- table(train$income,predict(rf1))

#accuracy in train data
sum(diag(pre_test_rf1))/sum(pre_test_rf1)

## [1] 0.8641837
```

The accuracy of random forest considering all variables are 86.3%, a little higher than logistic regression model. When the number of tree larger than 100, the error become stable. I will use ntree = 500 in the following analysis.

```
importance(rf1)

##                 MeanDecreaseGini
## age                    890.76026
## workclass              254.94332
## fnlwgt                 832.59857
```

```
## education.num          993.91802
## marital.status         601.25847
## occupation             441.20232
## relationship           916.92215
## race                    93.98745
## sex                    101.41945
## capital.gain           957.66295
## capital.loss           298.02893
## hours.per.week         555.98222
## native.country         115.61689
```

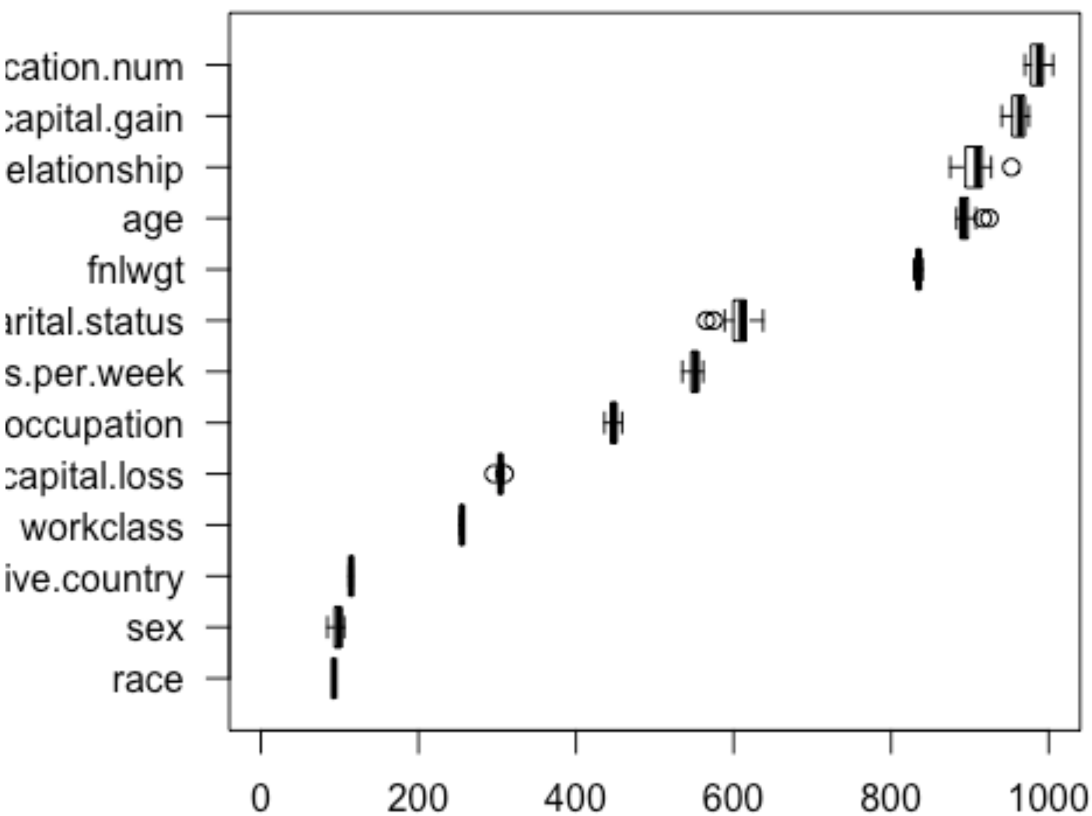
```
varImpPlot(rf1)
```



From the importance shows above, race, sex and native.country have the least importance. Since the randomness of this method, I would like to try several times and take a look at the importance.

```
importance.multirun = matrix(,20,13)
for(i in 1:20)
importance.multirun[i,] = randomForest(income ~ age + workclass + fnlwgt + ed
ucation.num + marital.status + occupation + relationship + race + sex + capit
al.gain + capital.loss + hours.per.week + native.country, data = train, ntree
= 500)$importance
```

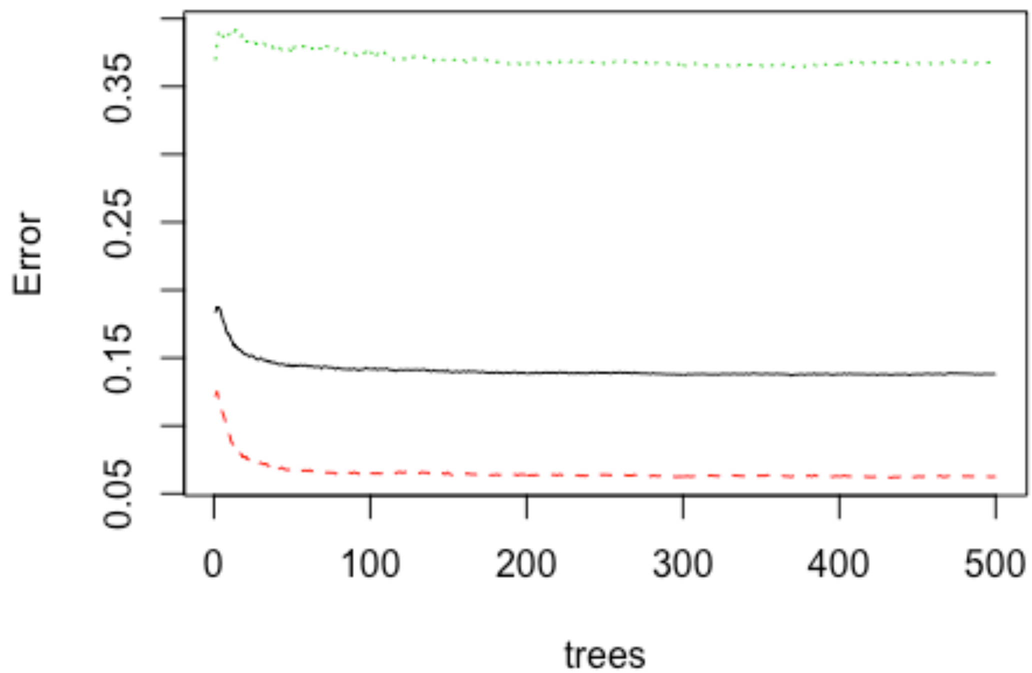
```
colnames(importance.multirun) = rownames(rf1$importance)
par(mar=c(3,5,1,1))
idx = order(apply(importance.multirun, 2, median))
boxplot(importance.multirun[, idx], horizontal=T, las=1, ylim=c(0,1000))
```



According to the value of each importance, the ranges of the value are relative small. Race, sex and native country still have the smallest importance. I will try another two model here. I will remove race, sex and native.country for the first one due to the importance. And for the other one, I will remove native.country, workclass and occupation due to the missing value in the both train and test sets.

reduce variable with small importance

```
rf2 <- randomForest(income ~ age + workclass + fnlwgt + education.num + marit
al.status + occupation + relationship + capital.gain + capital.loss + hours.p
er.week, data = train, ntree=500)

matplot(rf2$err.rate, type='l', xlab='trees', ylab='Error')
```

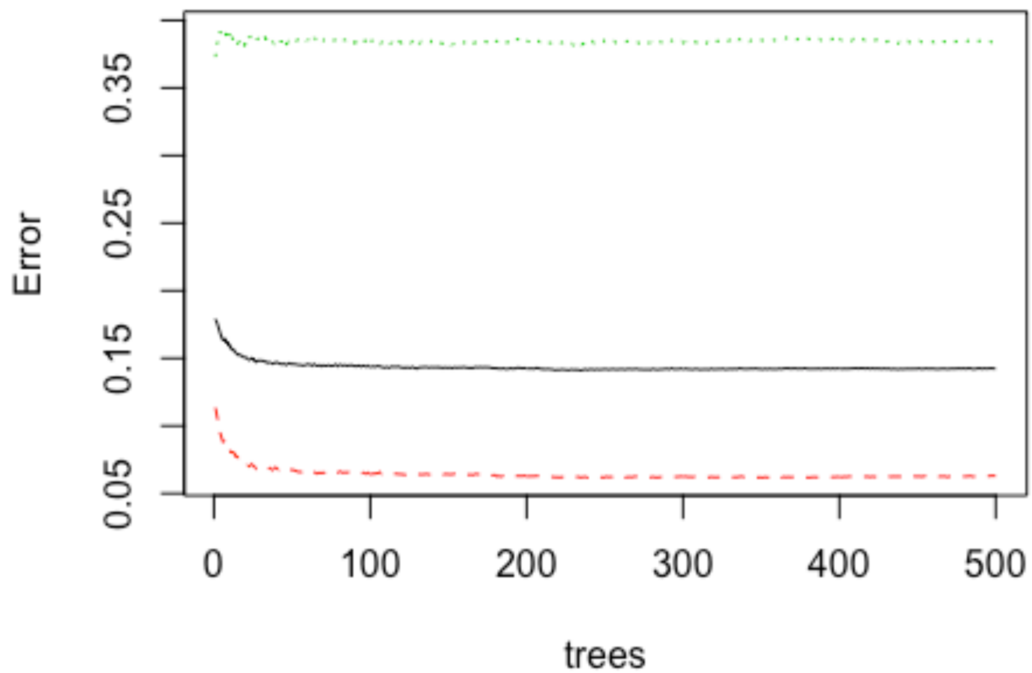



```
table(train$income,predict(rf2))

##
##           <=50K  >50K
##    <=50K  16343  1088
##    >50K    2108  3632

pre_test_rf2 <- table(train$income,predict(rf2))

#accuracy in train data
sum(diag(pre_test_rf2))/sum(pre_test_rf2)

## [1] 0.862069
```

The accuracy of the model removing the least important three variables does not change much, it decrease a little from 86.2% to 86.1%.

reducing the variable with missing value

```
rf3 <- randomForest(income ~ age + fnlwgt + education.num + marital.status +
relationship + race + sex + capital.gain + capital.loss + hours.per.week, dat
a = train, ntree = 500)

matplot(rf3$err.rate, type='l', xlab='trees', ylab='Error')
```



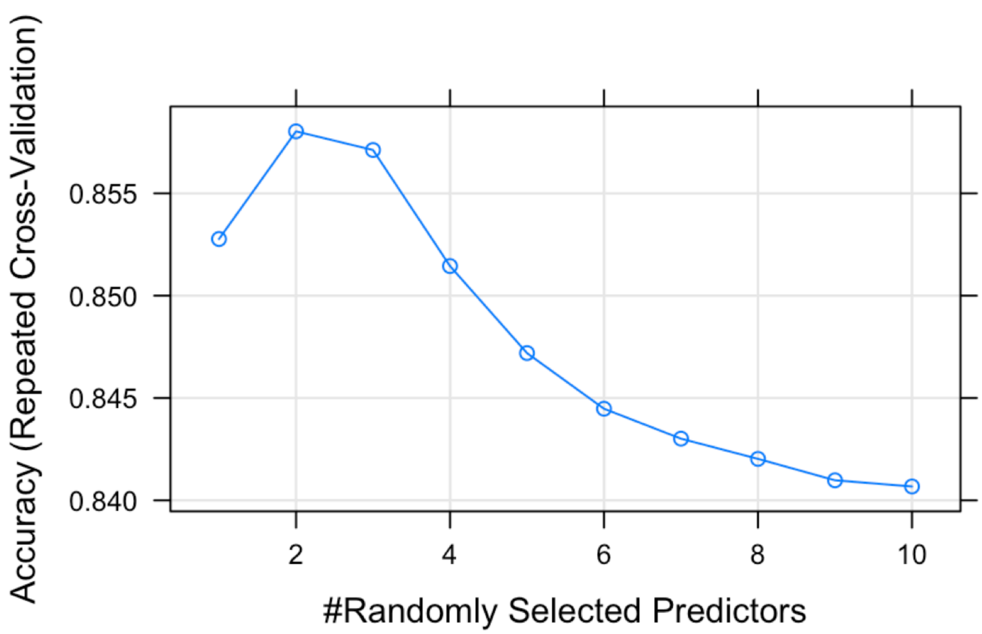

```
table(train$income,predict(rf3))

##
##          <=50K  >50K
##    <=50K 16334  1097
##    >50K   2200  3540

pre_test_rf3 <- table(train$income,predict(rf3))

#accuracy in train data
sum(diag(pre_test_rf3))/sum(pre_test_rf3)

## [1] 0.8577101
```

It also decrease a little from 86.2% to 85.8% when removing the three variables contain missing values. I think it is doable to kick of these three variable from the model. Also, remove these variables will provide enough degree of freedom to do cross validation.

In order to improve the performance of the model, I would like to use cross validation to choose a proper value of mtry.

randomForest with cv

```
train_rf <- train
train_rf <- apply(train,2,function(x)gsub('\\s+', '',x))
train_rf <- as.data.frame(train_rf)
train_rf$income <- as.character(train_rf$income)
train_rf$native.country <- as.character(train_rf$native.country)
train_rf[which(train_rf[,15]==">50K"),][,15] <- "larger_than_50K"
train_rf[which(train_rf[,15]=="<=50K"),][,15] <- "less_than_50K"
train_rf$age <- as.numeric(as.character(train_rf$age))
train_rf$fnlwgt <- as.numeric(as.character(train_rf$fnlwgt))
train_rf$education.num <- as.numeric(as.character(train_rf$education.num))
train_rf$capital.gain <- as.numeric(as.character(train_rf$capital.gain))
train_rf$capital.loss <- as.numeric(as.character(train_rf$capital.loss))
train_rf$hours.per.week <- as.numeric(as.character(train_rf$hours.per.week))


cv <- trainControl(method="repeatedcv", number=10, repeats=8, classProbs=TRUE
)


rf5 <- train(x=train_rf[,c(1,3,5,6,8:13)], y=train_rf$income, trControl=cv,tu
neGrid=data.frame(mtry=1:10), method="rf", ntree=500)


plot(rf5)
```



According to the plot, when mtry = 1, randomForest has the highest accuracy value.

```r
rf5
rf5$finalModel
```

```
Random Forest

23171 samples
   10 predictor
    2 classes: 'larger_than_50K', 'less_than_50K'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 8 times)
Summary of sample sizes: 20854, 20854, 20853, 20854, 20854, 20854,
...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
   1    0.8527684  0.5470879
   2    0.8580284  0.5869132
   3    0.8571166  0.5935203
   4    0.8514469  0.5811443
   5    0.8471959  0.5712481
   6    0.8444770  0.5646670
   7    0.8430151  0.5608605
   8    0.8420278  0.5579188
   9    0.8409813  0.5552177
  10    0.8406792  0.5542001
```

```
Accuracy was used to select the optimal model using the
 largest value.
The final value used for the model was mtry = 2.

Call:
 randomForest(x = x, y = y, ntree = 500, mtry = param$mtry)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 14.17%
Confusion matrix:
                larger_than_50K less_than_50K class.error
larger_than_50K            3442          2298  0.40034843
less_than_50K               985         16446  0.05650852
```

After cross validation, the accuracy is 85.8%, does not have any improvement.

Prediction in test set

Since the accuracy of these models do not have significant difference in training set, I will use the logistic regression and randomForest to predict the test set.

test dataset

```
test$marital.status <- factor(test$marital.status,levels(test$marital.status)
,ordered = T)
test$occupation <- factor(test$occupation,levels(test$occupation),ordered = T
)
test$relationship <- factor(test$relationship,levels(test$relationship),order
ed = T)
test$race <- factor(test$race,levels(test$race),ordered = T)
test$sex <- factor(test$sex,levels(test$sex),ordered = T)
test$native.country <- factor(test$native.country,levels(test$native.country)
,ordered = T)
test$income <- factor(test$income,levels(test$income),ordered = T)
```

using logistic regression

```
pre_test_rf2 <- predict(log.m2,newdata = test[,c(1,3,5,6,8:13)])
pre_test_rf2 <- as.numeric(pre_test_rf2)
pre_test_rf2 <- exp(pre_test_rf2)/(1+exp(pre_test_rf2))
pre_test_rf2 <- as.data.frame(pre_test_rf2)
pre_test_rf2 <- cbind.data.frame(pre_test_rf2,test$income)
colnames(pre_test_rf2) <- c("predict","real")
pre_test_rf2[which(pre_test_rf2$predict<0.5),]$predict <- "<=50K"
pre_test_rf2[which(pre_test_rf2$predict!="<=50K"),]$predict <- ">50K"
pre_test_rf2_table <- table(pre_test_rf2$predict,pre_test_rf2$real)

#accuracy in train data
sum(diag(pre_test_rf2_table))/sum(pre_test_rf2_table)

## [1] 0.836265
```

The accuracy of the logistic regression model in test set is about 83.6%.

using randomForest

```
test_rf <- test
test_rf <- apply(test,2,function(x)gsub('\\s+', '',x))
test_rf <- as.data.frame(test_rf)
test_rf$income <- as.character(test_rf$income)
test_rf$native.country <- as.character(test_rf$native.country)
test_rf$age <- as.numeric(as.character(test_rf$age))
test_rf$fnlwgt <- as.numeric(as.character(test_rf$fnlwgt))
test_rf$education.num <- as.numeric(as.character(test_rf$education.num))
test_rf$capital.gain <- as.numeric(as.character(test_rf$capital.gain))
```

```r
test_rf$capital.loss <- as.numeric(as.character(test_rf$capital.loss))
test_rf$hours.per.week <- as.numeric(as.character(test_rf$hours.per.week))

table(predict(rf5$finalModel, test_rf[,c(1,3,5,6,8:13)]),test$income)
```

```
                 <=50K  >50K
larger_than_50K    323  1067
less_than_50K     5395   776
```

Thus, the accuracy of randomForest is about 85.5%, a little higher than the logistic regression model.