# Final project (Jin Kweon and Jiyoon Clover Jeong)

*Jin Kweon and Clover Jeong*

*11/22/2017*

## Preprocessing and Exploratory Data Analysis

### a) Missing values

```
train <- read.table("../data/rawdata/adult.data.txt", sep = ",", na.strings = "?",
                    strip.white = T)
test <- read.table("../data/rawdata/adult.test.txt", sep = ",", na.strings = "?",
                   strip.white = T)

dim(train)
```

```
## [1] 32561    15
```

```
dim(test)
```

```
## [1] 16281    15
```

```
colnames(train) <- c("age", "workclass", "fnlwgt", "education", "education-num",
                     "marital-status", "occupation", "relationship", "race", "sex",
                     "capital-gain", "capital-loss", "hours-per-week", "native-country", "income")

colnames(test) <- c("age", "workclass", "fnlwgt", "education", "education-num",
                     "marital-status", "occupation", "relationship", "race", "sex",
                     "capital-gain", "capital-loss", "hours-per-week", "native-country", "income")

#Find missing values and NAs for training set.
for(i in 1:ncol(train)){
  cat("<names of NA rows in", colnames(train)[i], "variable>", "\n")
  cat(rownames(train)[is.na(train[,i])], "\n")
  cat("Number of NA values:  ", length(rownames(train)[is.na(train[,i])]),"\n")
  print("=======================================")
  print("=======================================")

  cat("<names of rows contain missing values in", colnames(train)[i], "variable>", "\n")
  cat(rownames(train[which(train[,i] == ""),]), "\n")
  cat("Number of Missing values :  ", length(rownames(train[which(train[,i] == ""),])), "\n")
  print("=======================================")
  print("=======================================")

  cat("<names of rows contain ? values in", colnames(train)[i], "variable>", "\n")
  cat(rownames(train[which(train[,i] == " ?"),]), "\n")
  cat("Number of ? values :  ", length(rownames(train[which(train[,i] == " ?"),])), "\n")
  print("=======================================")
  print("=======================================")
}
```

```
## <names of NA rows in age variable>
##
```

```
## Number of NA values:   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain missing values in age variable>
##
## Number of Missing values :   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain ? values in age variable>
##
## Number of ? values :   0
## [1] "======================================"
## [1] "======================================"
## <names of NA rows in workclass variable>
## 28 62 70 78 107 129 150 155 161 188 202 222 227 244 267 298 313 327 347 348 355 398 409 431 432 450 4
## Number of NA values:   1836
## [1] "======================================"
## [1] "======================================"
## <names of rows contain missing values in workclass variable>
##
## Number of Missing values :   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain ? values in workclass variable>
##
## Number of ? values :   0
## [1] "======================================"
## [1] "======================================"
## <names of NA rows in fnlwgt variable>
##
## Number of NA values:   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain missing values in fnlwgt variable>
##
## Number of Missing values :   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain ? values in fnlwgt variable>
##
## Number of ? values :   0
## [1] "======================================"
## [1] "======================================"
## <names of NA rows in education variable>
##
## Number of NA values:   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain missing values in education variable>
##
## Number of Missing values :   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain ? values in education variable>
```

```
##
## Number of ? values :   0
## [1] "======================================"
## [1] "======================================"
## <names of NA rows in education-num variable>
##
## Number of NA values:   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain missing values in education-num variable>
##
## Number of Missing values :   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain ? values in education-num variable>
##
## Number of ? values :   0
## [1] "======================================"
## [1] "======================================"
## <names of NA rows in marital-status variable>
##
## Number of NA values:   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain missing values in marital-status variable>
##
## Number of Missing values :   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain ? values in marital-status variable>
##
## Number of ? values :   0
## [1] "======================================"
## [1] "======================================"
## <names of NA rows in occupation variable>
## 28 62 70 78 107 129 150 155 161 188 202 222 227 244 267 298 313 327 347 348 355 398 409 431 432 450 4
## Number of NA values:   1843
## [1] "======================================"
## [1] "======================================"
## <names of rows contain missing values in occupation variable>
##
## Number of Missing values :   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain ? values in occupation variable>
##
## Number of ? values :   0
## [1] "======================================"
## [1] "======================================"
## <names of NA rows in relationship variable>
##
## Number of NA values:   0
## [1] "======================================"
## [1] "======================================"
```

```
## <names of rows contain missing values in relationship variable>
##
## Number of Missing values :   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain ? values in relationship variable>
##
## Number of ? values :   0
## [1] "======================================"
## [1] "======================================"
## <names of NA rows in race variable>
##
## Number of NA values:   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain missing values in race variable>
##
## Number of Missing values :   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain ? values in race variable>
##
## Number of ? values :   0
## [1] "======================================"
## [1] "======================================"
## <names of NA rows in sex variable>
##
## Number of NA values:   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain missing values in sex variable>
##
## Number of Missing values :   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain ? values in sex variable>
##
## Number of ? values :   0
## [1] "======================================"
## [1] "======================================"
## <names of NA rows in capital-gain variable>
##
## Number of NA values:   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain missing values in capital-gain variable>
##
## Number of Missing values :   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain ? values in capital-gain variable>
##
## Number of ? values :   0
## [1] "======================================"
```

```
## [1] "======================================"
## <names of NA rows in capital-loss variable>
##
## Number of NA values:   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain missing values in capital-loss variable>
##
## Number of Missing values :   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain ? values in capital-loss variable>
##
## Number of ? values :   0
## [1] "======================================"
## [1] "======================================"
## <names of NA rows in hours-per-week variable>
##
## Number of NA values:   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain missing values in hours-per-week variable>
##
## Number of Missing values :   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain ? values in hours-per-week variable>
##
## Number of ? values :   0
## [1] "======================================"
## [1] "======================================"
## <names of NA rows in native-country variable>
## 15 39 52 62 94 246 250 298 394 454 558 713 726 730 778 781 888 956 1027 1037 1116 1153 1159 1200 1225
## Number of NA values:   583
## [1] "======================================"
## [1] "======================================"
## <names of rows contain missing values in native-country variable>
##
## Number of Missing values :   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain ? values in native-country variable>
##
## Number of ? values :   0
## [1] "======================================"
## [1] "======================================"
## <names of NA rows in income variable>
##
## Number of NA values:   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain missing values in income variable>
##
## Number of Missing values :   0
```

```
## [1] "======================================"
## [1] "======================================"
## <names of rows contain ? values in income variable>
##
## Number of ? values :   0
## [1] "======================================"
## [1] "======================================"
```

```
# emptytrain <- c()
# for(i in 1:ncol(train)){
#   emptytrain[i] <- sum(train[,i] == "?")
# }
# emptytrain

#Find missing values and NAs for testing set.
for(i in 1:ncol(test)){
  cat("<names of NA rows in", colnames(test)[i], "variable>", "\n")
  cat(rownames(test)[is.na(test[,i])], "\n")
  cat("Number of NA values:  ", length(rownames(test)[is.na(test[,i])]),"\n")
  print("======================================")
  print("======================================")

  cat("<names of rows contain missing values in", colnames(test)[i], "variable>", "\n")
  cat(rownames(test[which(test[,i] == ""),]), "\n")
  cat("Number of Missing values :  ", length(rownames(test[which(test[,i] == ""),])), "\n")
  print("======================================")
  print("======================================")

  cat("<names of rows contain ? values in", colnames(test)[i], "variable>", "\n")
  cat(rownames(test[which(test[,i] == " ?"),]), "\n")
  cat("Number of ? values :  ", length(rownames(test[which(test[,i] == " ?"),])), "\n")
  print("======================================")
  print("======================================")
}
```

```
## <names of NA rows in age variable>
##
## Number of NA values:   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain missing values in age variable>
##
## Number of Missing values :   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain ? values in age variable>
##
## Number of ? values :   0
## [1] "======================================"
## [1] "======================================"
## <names of NA rows in workclass variable>
## 5 7 14 23 36 76 90 101 114 133 183 186 194 229 230 246 267 269 275 317 332 351 379 395 398 414 430 43
## Number of NA values:   963
## [1] "======================================"
## [1] "======================================"
```

```
## <names of rows contain missing values in workclass variable>
##
## Number of Missing values :   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain ? values in workclass variable>
##
## Number of ? values :   0
## [1] "======================================"
## [1] "======================================"
## <names of NA rows in fnlwgt variable>
##
## Number of NA values:   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain missing values in fnlwgt variable>
##
## Number of Missing values :   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain ? values in fnlwgt variable>
##
## Number of ? values :   0
## [1] "======================================"
## [1] "======================================"
## <names of NA rows in education variable>
##
## Number of NA values:   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain missing values in education variable>
##
## Number of Missing values :   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain ? values in education variable>
##
## Number of ? values :   0
## [1] "======================================"
## [1] "======================================"
## <names of NA rows in education-num variable>
##
## Number of NA values:   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain missing values in education-num variable>
##
## Number of Missing values :   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain ? values in education-num variable>
##
## Number of ? values :   0
## [1] "======================================"
```

```
## [1] "======================================"
## <names of NA rows in marital-status variable>
##
## Number of NA values:   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain missing values in marital-status variable>
##
## Number of Missing values :   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain ? values in marital-status variable>
##
## Number of ? values :   0
## [1] "======================================"
## [1] "======================================"
## <names of NA rows in occupation variable>
## 5 7 14 23 36 76 90 101 114 133 183 186 194 229 230 246 267 269 275 317 332 351 379 395 398 414 430 43
## Number of NA values:   966
## [1] "======================================"
## [1] "======================================"
## <names of rows contain missing values in occupation variable>
##
## Number of Missing values :   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain ? values in occupation variable>
##
## Number of ? values :   0
## [1] "======================================"
## [1] "======================================"
## <names of NA rows in relationship variable>
##
## Number of NA values:   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain missing values in relationship variable>
##
## Number of Missing values :   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain ? values in relationship variable>
##
## Number of ? values :   0
## [1] "======================================"
## [1] "======================================"
## <names of NA rows in race variable>
##
## Number of NA values:   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain missing values in race variable>
##
## Number of Missing values :   0
```

```
## [1] "====================================="
## [1] "====================================="
## <names of rows contain ? values in race variable>
##
## Number of ? values :   0
## [1] "====================================="
## [1] "====================================="
## <names of NA rows in sex variable>
##
## Number of NA values:   0
## [1] "====================================="
## [1] "====================================="
## <names of rows contain missing values in sex variable>
##
## Number of Missing values :   0
## [1] "====================================="
## [1] "====================================="
## <names of rows contain ? values in sex variable>
##
## Number of ? values :   0
## [1] "====================================="
## [1] "====================================="
## <names of NA rows in capital-gain variable>
##
## Number of NA values:   0
## [1] "====================================="
## [1] "====================================="
## <names of rows contain missing values in capital-gain variable>
##
## Number of Missing values :   0
## [1] "====================================="
## [1] "====================================="
## <names of rows contain ? values in capital-gain variable>
##
## Number of ? values :   0
## [1] "====================================="
## [1] "====================================="
## <names of NA rows in capital-loss variable>
##
## Number of NA values:   0
## [1] "====================================="
## [1] "====================================="
## <names of rows contain missing values in capital-loss variable>
##
## Number of Missing values :   0
## [1] "====================================="
## [1] "====================================="
## <names of rows contain ? values in capital-loss variable>
##
## Number of ? values :   0
## [1] "====================================="
## [1] "====================================="
## <names of NA rows in hours-per-week variable>
##
```

```
## Number of NA values:   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain missing values in hours-per-week variable>
##
## Number of Missing values :   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain ? values in hours-per-week variable>
##
## Number of ? values :   0
## [1] "======================================"
## [1] "======================================"
## <names of NA rows in native-country variable>
## 20 66 84 189 254 306 330 404 421 472 516 649 666 688 844 1009 1039 1164 1334 1365 1406 1616 1644 180
## Number of NA values:   274
## [1] "======================================"
## [1] "======================================"
## <names of rows contain missing values in native-country variable>
##
## Number of Missing values :   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain ? values in native-country variable>
##
## Number of ? values :   0
## [1] "======================================"
## [1] "======================================"
## <names of NA rows in income variable>
##
## Number of NA values:   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain missing values in income variable>
##
## Number of Missing values :   0
## [1] "======================================"
## [1] "======================================"
## <names of rows contain ? values in income variable>
##
## Number of ? values :   0
## [1] "======================================"
## [1] "======================================"
```

```
# emptytest <- c()
# for(i in 1:ncol(test)){
#   emptytest[i] <- sum(test[,i] == "?")
# }
# emptytest


#Get percentage of missing values
apply(train, 2, function(x) sum(is.na(x))/length(x))*100
```

```
##            age      workclass         fnlwgt      education  education-num
```

```
##       0.000000       5.638647       0.000000       0.000000       0.000000
## marital-status     occupation   relationship           race            sex
##       0.000000       5.660146       0.000000       0.000000       0.000000
##   capital-gain   capital-loss hours-per-week native-country         income
##       0.000000       0.000000       0.000000       1.790486       0.000000
```

```
apply(test, 2, function(x) sum(is.na(x))/length(x))*100
```

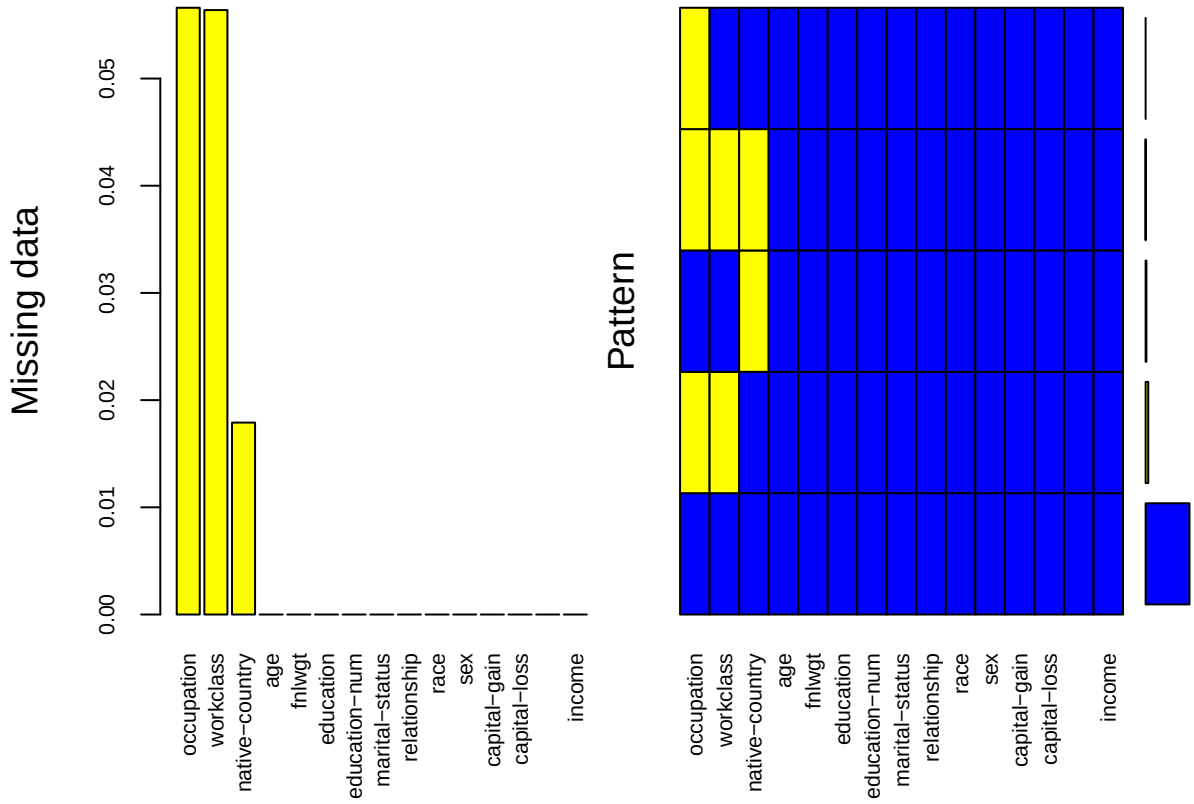
```
##            age      workclass         fnlwgt      education  education-num
##       0.000000       5.914870       0.000000       0.000000       0.000000
## marital-status     occupation   relationship           race            sex
##       0.000000       5.933296       0.000000       0.000000       0.000000
##   capital-gain   capital-loss hours-per-week native-country         income
##       0.000000       0.000000       0.000000       1.682943       0.000000
```

```
#MICE package to see the pattern
md.pattern(train)
```

```
##       age fnlwgt education education-num marital-status relationship race
## 30162   1      1         1             1              1             1    1
##     7   1      1         1             1              1             1    1
##   556   1      1         1             1              1             1    1
##  1809   1      1         1             1              1             1    1
##    27   1      1         1             1              1             1    1
##         0      0         0             0              0             0    0
##       sex capital-gain capital-loss hours-per-week income native-country
## 30162   1            1            1              1      1              1
##     7   1            1            1              1      1              1
##   556   1            1            1              1      1              0
##  1809   1            1            1              1      1              1
##    27   1            1            1              1      1              0
##         0            0            0              0      0            583
##       workclass occupation
## 30162         1          1    0
##     7         1          0    1
##   556         1          1    1
##  1809         0          0    2
##    27         0          0    3
##            1836       1843 4262
```

```
plot <- aggr(train, col = c('blue','yellow'),
                    numbers = TRUE, sortVars = TRUE,
                    labels = names(train), cex.axis=.7,
                    gap = 2, ylab=c("Missing data","Pattern"))
```

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




```
##
##  Variables sorted by number of missings:
##        Variable      Count
##      occupation 0.05660146
##       workclass 0.05638647
##  native-country 0.01790486
##             age 0.00000000
##          fnlwgt 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
```

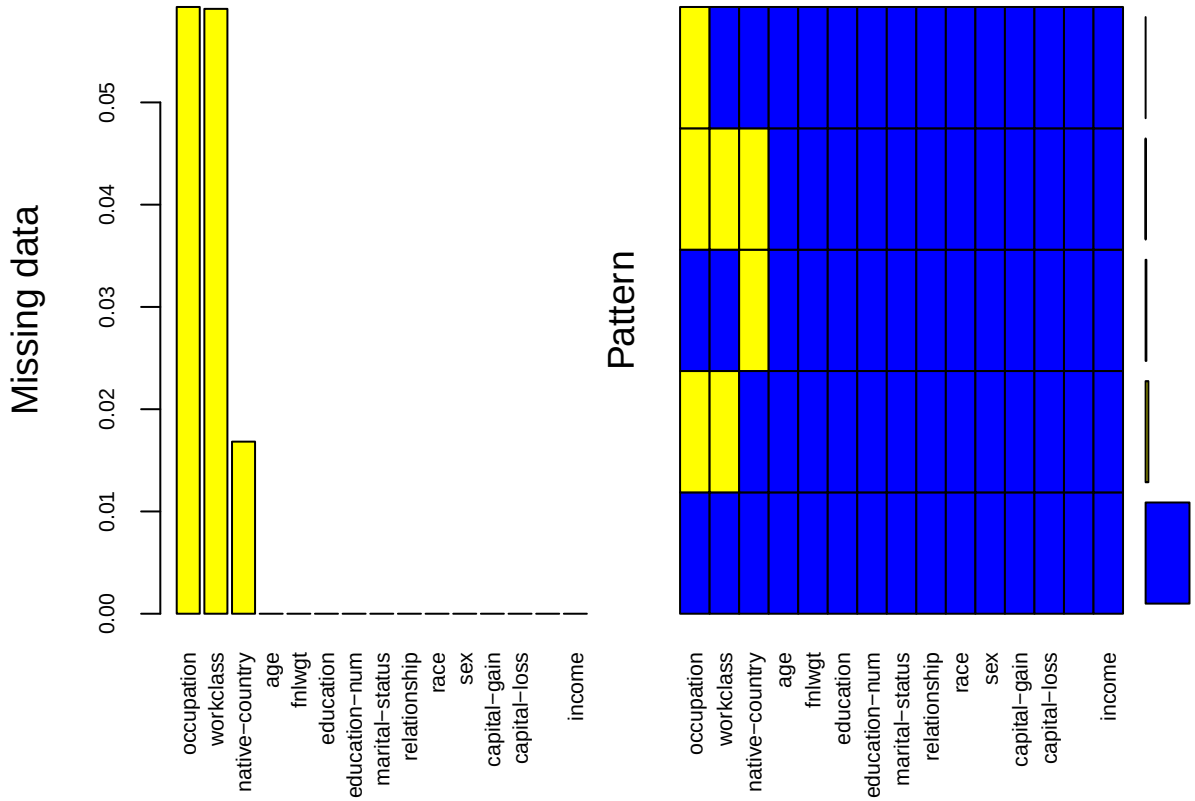
```
md.pattern(test)
```

```
##       age fnlwgt education education-num marital-status relationship race
## 15060   1      1         1             1              1            1    1
##     3   1      1         1             1              1            1    1
##   255   1      1         1             1              1            1    1
##   944   1      1         1             1              1            1    1
##    19   1      1         1             1              1            1    1
##         0      0         0             0              0            0    0
##       sex capital-gain capital-loss hours-per-week income native-country
```

```
## 15060   1            1            1              1       1              1
##     3   1            1            1              1       1              1
##   255   1            1            1              1       1              0
##   944   1            1            1              1       1              1
##    19   1            1            1              1       1              0
##         0            0            0              0       0            274
##       workclass occupation
## 15060         1          1    0
##     3         1          0    1
##   255         1          1    1
##   944         0          0    2
##    19         0          0    3
##             963        966 2203
```

```
plot <- aggr(test, col = c('blue','yellow'),
                   numbers = TRUE, sortVars = TRUE,
                   labels = names(test), cex.axis=.7,
                   gap = 2, ylab=c("Missing data","Pattern"))
```

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




```
##
##  Variables sorted by number of missings:
##        Variable      Count
##      occupation 0.05933296
##       workclass 0.05914870
##  native-country 0.01682943
##             age 0.00000000
```

```
##          fnlwgt 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
```

```
# Hmisc package to impute missing values
# ww <- aregImpute(~ age + workclass + fnlwgt + education + `education-num` + `marital-status` +
#                    occupation + relationship + race + sex + `capital-gain` + `capital-loss` +
#                    `hours-per-week` + income,
#                  data = train, n.impute = 5, group = "income")



#mlr package to impute missing values
# newworkclass <- impute(train[,2], classes = list(factor = imputeMode(), integer = imputeMean()), dumm
#
# newoccupation <- impute(train[,7], classes = list(factor = imputeMode(), integer = imputeMean()), dum
#
# newcountry <- impute(train[,14], classes = list(factor = imputeMode(), integer = imputeMean()), dummy



#missForest package to impute missing values
# foresting <- missForest(train, maxiter = 5, ntree = 100)
# foresting$OOBerror
# newtrain <- foresting$ximp
# write.csv(newtrain, file = "../data/cleandata/newtrain.csv", col.names = T, row.names = F)

newtrain <- read.csv("../data/cleandata/newtrain.csv", header = T)
dim(newtrain)
```

```
## [1] 32561    15
```

```
# foresting2 <- missForest(test, maxiter = 5, ntree = 100)
# foresting2$OOBerror
# newtest <- foresting2$ximp
# write.csv(newtest, file = "../data/cleandata/newtest.csv", col.names = T, row.names = F)
newtest <- read.csv("../data/cleandata/newtest.csv", header = T)
dim(newtest)
```

```
## [1] 16281    15
```

```
#Check whether the data is messed up while imputing missing values
#They should never show 0, as we are supposed to see only missing value has been changed...
#Compare NA with new number in new data set should show NA, not 0.
t <- matrix(0, 1, ncol(train))
for(i in 1:20){
  a <- sample.int(nrow(newtrain), 1)
  t <- rbind(t, (newtrain[a,] == train[a,]))
```

```
}
t <- t[-1,]
t
```

```
##       age workclass fnlwgt education education.num marital.status
## 24789   1         1      1         1             1              1
## 153     1         1      1         1             1              1
## 10914   1         1      1         1             1              1
## 3651    1         1      1         1             1              1
## 3101    1         1      1         1             1              1
## 7576    1         1      1         1             1              1
## 17323   1         1      1         1             1              1
## 28640   1         1      1         1             1              1
## 16940   1         1      1         1             1              1
## 8656    1         1      1         1             1              1
## 14183   1         1      1         1             1              1
## 17703   1         1      1         1             1              1
## 927     1         1      1         1             1              1
## 7286    1         1      1         1             1              1
## 10110   1         1      1         1             1              1
## 18335   1         1      1         1             1              1
## 3424    1         1      1         1             1              1
## 2685    1         1      1         1             1              1
## 19004   1         1      1         1             1              1
## 22712   1         1      1         1             1              1
##       occupation relationship race sex capital.gain capital.loss
## 24789          1            1    1   1            1            1
## 153            1            1    1   1            1            1
## 10914          1            1    1   1            1            1
## 3651           1            1    1   1            1            1
## 3101           1            1    1   1            1            1
## 7576           1            1    1   1            1            1
## 17323          1            1    1   1            1            1
## 28640          1            1    1   1            1            1
## 16940          1            1    1   1            1            1
## 8656           1            1    1   1            1            1
## 14183          1            1    1   1            1            1
## 17703          1            1    1   1            1            1
## 927            1            1    1   1            1            1
## 7286           1            1    1   1            1            1
## 10110          1            1    1   1            1            1
## 18335          1            1    1   1            1            1
## 3424           1            1    1   1            1            1
## 2685           1            1    1   1            1            1
## 19004          1            1    1   1            1            1
## 22712          1            1    1   1            1            1
##       hours.per.week native.country income
## 24789              1              1      1
## 153                1              1      1
## 10914              1              1      1
## 3651               1              1      1
## 3101               1              1      1
## 7576               1              1      1
## 17323              1              1      1
```

```
## 28640              1              1      1
## 16940              1              1      1
## 8656               1              1      1
## 14183              1              1      1
## 17703              1              1      1
## 927                1              1      1
## 7286               1              1      1
## 10110              1              1      1
## 18335              1              1      1
## 3424               1              1      1
## 2685               1              1      1
## 19004              1              1      1
## 22712              1              1      1
```

```
t2 <- matrix(0, 1, ncol(test))
for(i in 1:20){
  a <- sample.int(nrow(newtest), 1)
  t2 <- rbind(t2, (newtest[a,] == test[a,]))
}
t2 <- t2[-1,]
t2
```

```
##       age workclass fnlwgt education education.num marital.status
## 692     1         1      1         1             1              1
## 2070    1         1      1         1             1              1
## 15420   1         1      1         1             1              1
## 930     1         1      1         1             1              1
## 5058    1         1      1         1             1              1
## 11505   1         1      1         1             1              1
## 10287   1         1      1         1             1              1
## 5534    1         1      1         1             1              1
## 14020   1         1      1         1             1              1
## 1024    1         1      1         1             1              1
## 1552    1         1      1         1             1              1
## 11161   1         1      1         1             1              1
## 15224   1         1      1         1             1              1
## 4701    1         1      1         1             1              1
## 11553   1         1      1         1             1              1
## 15209   1         1      1         1             1              1
## 12343   1         1      1         1             1              1
## 3149    1         1      1         1             1              1
## 15205   1         1      1         1             1              1
## 7164    1         1      1         1             1              1
##       occupation relationship race sex capital.gain capital.loss
## 692            1            1    1   1            1            1
## 2070           1            1    1   1            1            1
## 15420          1            1    1   1            1            1
## 930            1            1    1   1            1            1
## 5058           1            1    1   1            1            1
## 11505          1            1    1   1            1            1
## 10287          1            1    1   1            1            1
## 5534           1            1    1   1            1            1
## 14020          1            1    1   1            1            1
## 1024           1            1    1   1            1            1
## 1552           1            1    1   1            1            1
```

```
## 11161          1            1    1   1            1            1
## 15224          1            1    1   1            1            1
## 4701           1            1    1   1            1            1
## 11553          1            1    1   1            1            1
## 15209          1            1    1   1            1            1
## 12343          1            1    1   1            1            1
## 3149           1            1    1   1            1            1
## 15205          1            1    1   1            1            1
## 7164           1            1    1   1            1            1
##       hours.per.week native.country income
## 692                1              1      1
## 2070               1              1      1
## 15420              1              1      1
## 930                1              1      1
## 5058               1              1      1
## 11505              1              1      1
## 10287              1              1      1
## 5534               1              1      1
## 14020              1              1      1
## 1024               1              1      1
## 1552               1              1      1
## 11161              1              1      1
## 15224              1              1      1
## 4701               1              1      1
## 11553              1              1      1
## 15209              1              1      1
## 12343              1              1      1
## 3149               1              1      1
## 15205              1              1      1
## 7164               1              1      1
```

**b) 2 - 5 EDAs**

```
#See structure and summaries before removing outliers
str(newtest)
```

```
## 'data.frame':    16281 obs. of  15 variables:
##  $ age           : int  25 38 28 44 18 34 29 63 24 55 ...
##  $ workclass     : Factor w/ 8 levels "Federal-gov",..: 4 4 2 4 4 4 4 6 4 4 ...
##  $ fnlwgt        : int  226802 89814 336951 160323 103497 198693 227026 104626 369667 104996 ...
##  $ education     : Factor w/ 16 levels "10th","11th",..: 2 12 8 16 16 1 12 15 16 6 ...
##  $ education.num : int  7 9 12 10 10 6 9 15 10 4 ...
##  $ marital.status: Factor w/ 7 levels "Divorced","Married-AF-spouse",..: 5 3 3 3 5 5 5 3 5 3 ...
##  $ occupation    : Factor w/ 14 levels "Adm-clerical",..: 7 5 11 7 12 8 6 10 8 3 ...
##  $ relationship  : Factor w/ 6 levels "Husband","Not-in-family",..: 4 1 1 1 4 2 5 1 5 1 ...
##  $ race          : Factor w/ 5 levels "Amer-Indian-Eskimo",..: 3 5 5 3 5 5 3 5 5 5 ...
##  $ sex           : Factor w/ 2 levels "Female","Male": 2 2 2 2 1 2 2 2 1 2 ...
##  $ capital.gain  : int  0 0 0 7688 0 0 0 3103 0 0 ...
```

```
##  $ capital.loss  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hours.per.week: int  40 50 40 40 30 30 40 32 40 10 ...
##  $ native.country: Factor w/ 40 levels "Cambodia","Canada",..: 38 38 38 38 38 38 38 38 38 38 ...
##  $ income        : Factor w/ 2 levels "<=50K.",">50K.": 1 1 2 2 1 1 1 2 1 1 ...
```

```
summary(newtest)
```

```
##       age                     workclass         fnlwgt
##  Min.   :17.00   Private         :11963   Min.   :  13492
##  1st Qu.:28.00   Self-emp-not-inc: 1433   1st Qu.: 116736
##  Median :37.00   Local-gov       : 1090   Median : 177831
##  Mean   :38.77   State-gov       :  710   Mean   : 189436
##  3rd Qu.:48.00   Self-emp-inc    :  594   3rd Qu.: 238384
##  Max.   :90.00   Federal-gov     :  481   Max.   :1490400
##                  (Other)         :   10
##         education    education.num                  marital.status
##  HS-grad     :5283   Min.   : 1.00   Divorced              :2190
##  Some-college:3587   1st Qu.: 9.00   Married-AF-spouse     :  14
##  Bachelors   :2670   Median :10.00   Married-civ-spouse    :7403
##  Masters     : 934   Mean   :10.07   Married-spouse-absent: 210
##  Assoc-voc   : 679   3rd Qu.:12.00   Never-married         :5434
##  11th        : 637   Max.   :16.00   Separated             : 505
##  (Other)     :2491                   Widowed               : 525
##            occupation           relationship                    race
##  Prof-specialty :2111   Husband       :6523   Amer-Indian-Eskimo:  159
##  Craft-repair   :2040   Not-in-family :4278   Asian-Pac-Islander:  480
##  Exec-managerial:2035   Other-relative: 525   Black              : 1561
##  Adm-clerical   :1967   Own-child     :2513   Other              :  135
##  Sales          :1921   Unmarried     :1679   White              :13946
##  Other-service  :1825   Wife          : 763
##  (Other)        :4382
##      sex          capital.gain    capital.loss    hours.per.week
##  Female: 5421   Min.   :    0   Min.   :   0.0   Min.   : 1.00
##  Male  :10860   1st Qu.:    0   1st Qu.:   0.0   1st Qu.:40.00
##                 Median :    0   Median :   0.0   Median :40.00
##                 Mean   : 1082   Mean   :  87.9   Mean   :40.39
##                 3rd Qu.:    0   3rd Qu.:   0.0   3rd Qu.:45.00
##                 Max.   :99999   Max.   :3770.0   Max.   :99.00
##
##        native.country     income
##  United-States:14892   <=50K.:12435
##  Mexico       :  311   >50K. : 3846
##  Philippines  :  111
##  Puerto-Rico  :   70
##  Germany      :   69
##  Canada       :   61
##  (Other)      :  767
```

```
str(newtrain)
```

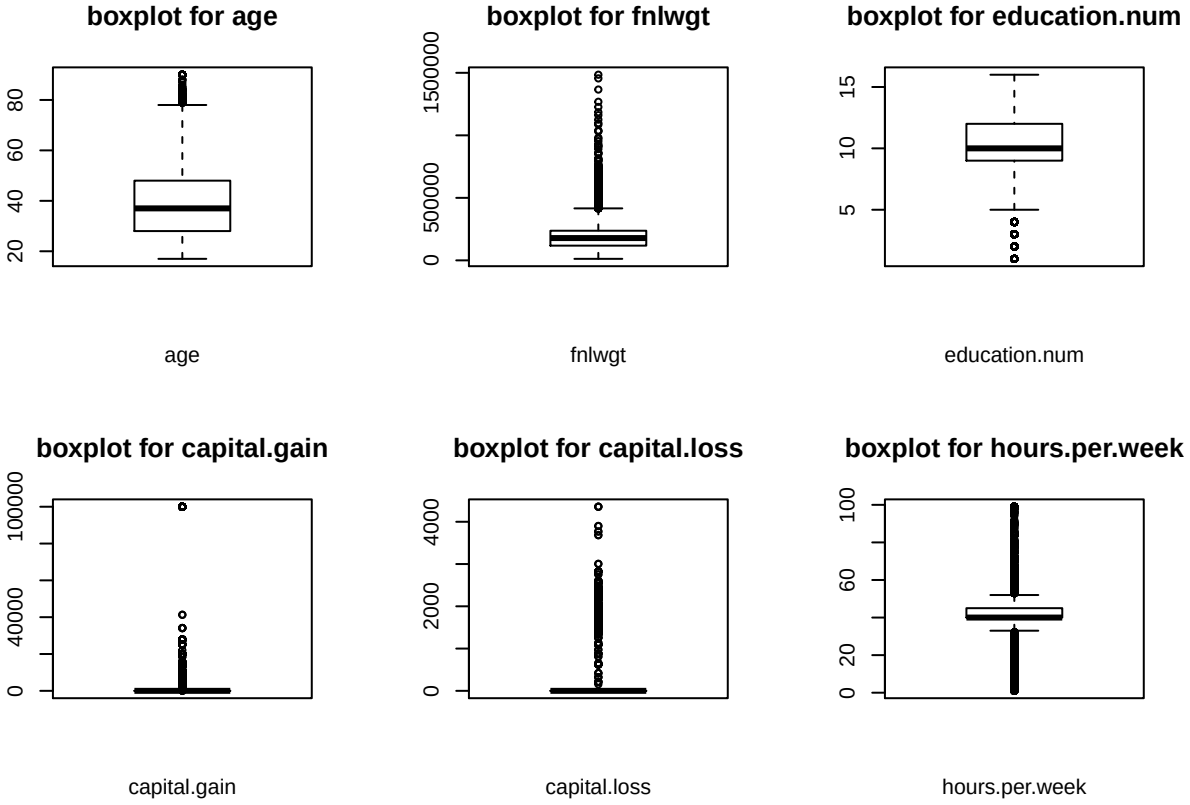
```
## 'data.frame':    32561 obs. of  15 variables:
##  $ age           : int  39 50 38 53 28 37 49 52 31 42 ...
##  $ workclass     : Factor w/ 8 levels "Federal-gov",..: 7 6 4 4 4 4 4 6 4 4 ...
##  $ fnlwgt        : int  77516 83311 215646 234721 338409 284582 160187 209642 45781 159449 ...
##  $ education     : Factor w/ 16 levels "10th","11th",..: 10 10 12 2 10 13 7 12 13 10 ...
```

```
##  $ education.num : int  13 13 9 7 13 14 5 9 14 13 ...
##  $ marital.status: Factor w/ 7 levels "Divorced","Married-AF-spouse",..: 5 3 1 3 3 3 4 3 5 3 ...
##  $ 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 ...
##  $ capital.gain  : int  2174 0 0 0 0 0 0 0 14084 5178 ...
##  $ capital.loss  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hours.per.week: int  40 13 40 40 40 40 16 45 50 40 ...
##  $ native.country: Factor w/ 41 levels "Cambodia","Canada",..: 39 39 39 39 5 39 23 39 39 39 ...
##  $ income        : Factor w/ 2 levels "<=50K",">50K": 1 1 1 1 1 1 1 2 2 2 ...
```

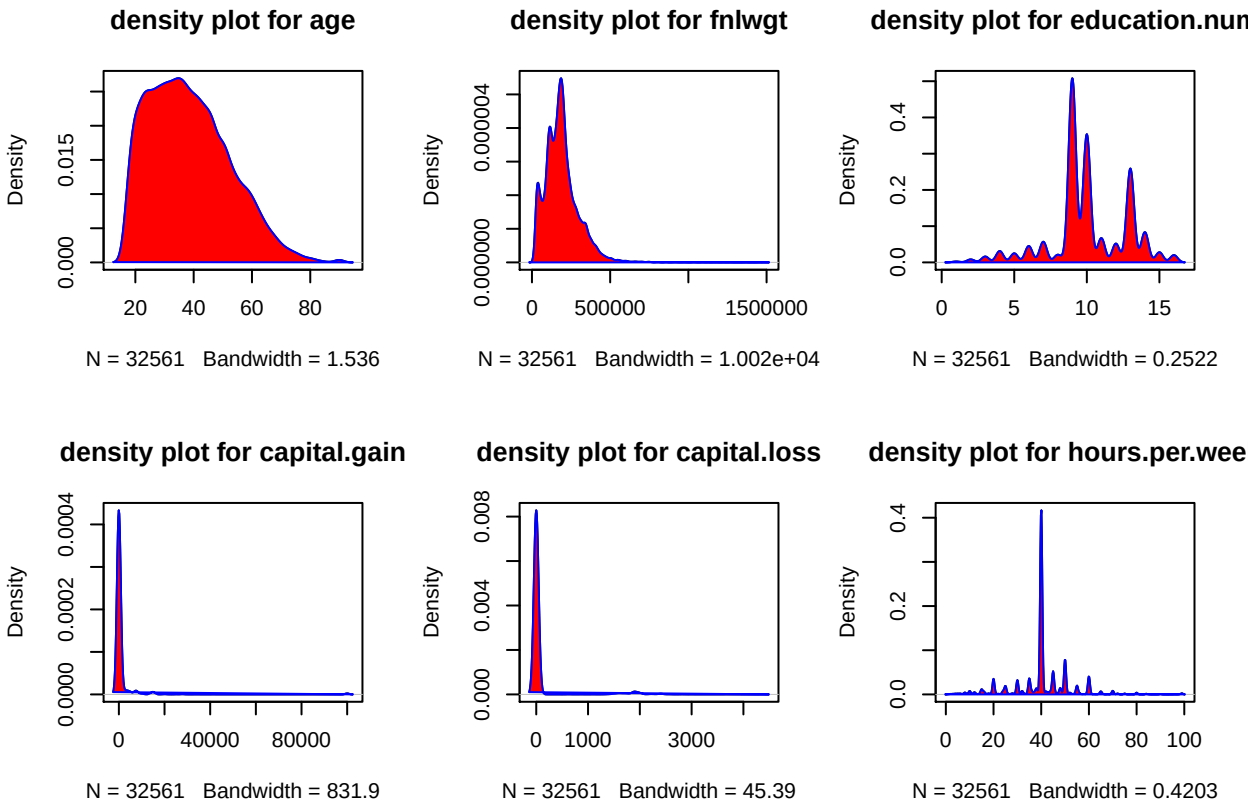
```
summary(newtrain)
```

```
##       age                    workclass         fnlwgt
##  Min.   :17.00   Private         :24068   Min.   :  12285
##  1st Qu.:28.00   Self-emp-not-inc: 2776   1st Qu.: 117827
##  Median :37.00   Local-gov       : 2193   Median : 178356
##  Mean   :38.58   State-gov       : 1352   Mean   : 189778
##  3rd Qu.:48.00   Self-emp-inc    : 1164   3rd Qu.: 237051
##  Max.   :90.00   Federal-gov     :  985   Max.   :1484705
##                  (Other)         :   23
##         education     education.num                 marital.status
##  HS-grad     :10501   Min.   : 1.00   Divorced             : 4443
##  Some-college: 7291   1st Qu.: 9.00   Married-AF-spouse    :   23
##  Bachelors   : 5355   Median :10.00   Married-civ-spouse   :14976
##  Masters     : 1723   Mean   :10.08   Married-spouse-absent:  418
##  Assoc-voc   : 1382   3rd Qu.:12.00   Never-married        :10683
##  11th        : 1175   Max.   :16.00   Separated            : 1025
##  (Other)     : 5134                   Widowed              :  993
##            occupation           relationship                   race
##  Prof-specialty :4295   Husband       :13193   Amer-Indian-Eskimo:  311
##  Craft-repair   :4162   Not-in-family : 8305   Asian-Pac-Islander: 1039
##  Exec-managerial:4129   Other-relative:  981   Black             : 3124
##  Adm-clerical   :3992   Own-child     : 5068   Other             :  271
##  Sales          :3715   Unmarried     : 3446   White             :27816
##  Other-service  :3696   Wife          : 1568
##  (Other)        :8572
##      sex          capital.gain    capital.loss    hours.per.week
##  Female:10771   Min.   :    0   Min.   :   0.0   Min.   : 1.00
##  Male  :21790   1st Qu.:    0   1st Qu.:   0.0   1st Qu.:40.00
##                 Median :    0   Median :   0.0   Median :40.00
##                 Mean   : 1078   Mean   :  87.3   Mean   :40.44
##                 3rd Qu.:    0   3rd Qu.:   0.0   3rd Qu.:45.00
##                 Max.   :99999   Max.   :4356.0   Max.   :99.00
##
##        native.country    income
##  United-States:29675   <=50K:24720
##  Mexico       :  657   >50K : 7841
##  Philippines  :  211
##  Germany      :  137
##  Canada       :  121
##  Puerto-Rico  :  114
##  (Other)      : 1646
```

```r
#Deal with outliers for training sets
continuouscol <- c(1, 3, 5, 11, 12, 13) #subset continous variables

par(mfrow = c(2, 3))
for(i in continuouscol){
  boxplot(newtrain[,i], main = paste("boxplot for", colnames(newtrain[i])),
          xlab = colnames(newtrain[i]))
}
```




```r
for(i in continuouscol){
  den_acc <- density(newtrain[,i], adjust = 1)
  plot(den_acc, main = paste("density plot for", colnames(newtrain[i])))
  polygon(den_acc, col = "red", border = "blue")
}
```




```
outlierstrain <- list()
for(i in continuouscol){
  outliers <- boxplot.stats(newtrain[,i])$out
  numbers <- length(outliers)
  outlierstrain[[i]] <- list(outliers, numbers)
}
head(outlierstrain)
```

```
## [[1]]
## [[1]][[1]]
##   [1] 79 90 80 81 90 88 90 90 80 90 81 82 79 81 80 83 90 90 79 81 90 90 80
##  [24] 90 90 79 79 84 90 80 90 81 83 84 81 79 85 82 79 80 90 90 90 84 80 90
##  [47] 90 79 84 90 79 90 90 90 82 81 90 84 79 81 82 81 80 90 80 84 82 79 90
##  [70] 84 90 83 79 81 80 79 80 79 80 90 90 80 90 90 81 83 82 90 90 81 80 80
##  [93] 90 79 80 82 85 80 79 90 81 79 80 79 81 82 88 90 82 88 84 83 79 86 90
## [116] 90 82 83 81 79 90 80 81 79 84 84 79 90 80 81 81 81 90 87 90 80 80 82
## [139] 90 90 85 82 81
##
## [[1]][[2]]
## [1] 143
##
##
## [[2]]
## NULL
##
## [[3]]
## [[3]][[1]]
##   [1]  544091  507875  446839  432376  494223  428030  483777  633742
```

```
##   [9]  523910  635913  538583  477983  425161  860348  423158  481060
##  [17]  416103  445382 1033222  426017  543162  433665  462440  556660
##  [25]  430828  475028  420537  680390  499233  543028  465507  526968
##  [33]  767403  431192  520586  445824  416745  444304  441454  421132
##  [41]  795830  419721  509350  467108  444554  449257  441620  563883
##  [49]  431745  436006  473040  910398  451940  428350  421871  443040
##  [57]  420895  496743  429507  418324  538319  508336  445382  483201
##  [65]  452205  672412  473547  421065  505119  460046  549430  441591
##  [73]  438696  488720  482082  460835  519627  675421  481987  758700
##  [81]  509364  432565  490332  466224  446219  423460  509364  656036
##  [89]  443508  566117  436253  454508  427686  548510  545483  503012
##  [97]  573583  511361  454941  452405  716416  480861  498785  637222
## [105]  430084  423770  417657  446358  457402  664821  462890  598606
## [113]  457237  465326  503923  572751  580248  519006  617021  437994
## [121]  596776  588905  517995  640383  504725  423863  420917  470663
## [129]  611029  437851  495888  549341  421837  746786  550848  510072
## [137]  449432  430471  416129  511331  446559  452640  456399  469705
## [145]  656036  488720  434710  449354  425627  417136  460835  416338
## [153]  424079  423561  688355  587310  628797  421449  424988  443508
## [161]  632613  499249  445758  416164  473133  450580  506329  445168
## [169]  516337  432376  571853 1184622  913447  476573  632593  595000
## [177]  703067  484475  476391  749105  459465  543922  420282  498325
## [185]  447579  420749  482732  437281  427965  505980  549349  496025
## [193]  562558  642830  435022  443546  523095  436770  436493  704108
## [201]  557082  477106  471452  426001  464536  451996  505980  454614
## [209]  473748  506858  434102  454989  537222  595000  454508  577521
## [217]  424012  431426  604506  564135  427781  469907  503675  444089
## [225]  435835  512103  716066  487486  484298  479765  444743  483596
## [233]  525878  423250  538443  493034  434292  496382  432154  528616
## [241]  515025  433491  421223  428350  446358  455995  659273  435604
## [249]  425092  452924  541737  444822  423024  445940  468706  428584
## [257]  972354  459189  498216  608184  444219  433788  586657 1226583
## [265]  664670  447346  504725  427055  561334  499001  791084  917220
## [273]  430084  508548  511289  416577  512992  431745  427862  637080
## [281]  431861  671292  442612  494638  431307  459007  517000  421446
## [289]  548361  648223  522881  433669  461678  416059  473836  745768
## [297]  523067  508891  486332  418176  417419  464945  454508  476653
## [305]  488706  647882  569761  585203  539563 1038553  567788  732569
## [313]  416165  721161  509629  474136  450924  477697  423711  419658
## [321]  553473  496414  421967  453067  466458  421561  483530  560804
## [329]  447079  528616  485496  425528  502316  467799  469921  444134
## [337]  443179  497300  426431  607848  501172  441700  483822  420973
## [345]  514033  470663  472604  487411  558183  416829  430005  426263
## [353]  439608  456236  420779  541282  518030  459248  548580  526528
## [361]  447739  586657  433375  581071  437727  575442  554986  592930
## [369]  632834  423052  504951  484861  449576  496538  459463  505438
## [377]  479482  467108  467108  849857  426562  558944  420054  691903
## [385]  419691  684015  423605  461678  466498  530099  554317  420054
## [393]  450920  427952  695136  698418  464103  526968  450695  548303
## [401]  529216  526164  506436  439919  734193  737315  544686  468713
## [409]  548361  556652  691830  520775  442429  433669  607799  660870
## [417]  440456  471990  483822  423222  500509  487742  498785  423064
## [425]  532379  426895  493862  424855  469602  432555  424468  428271
## [433]  464502  446140  480717  529104  456110  451744  680390  438711
```

```
## [441]  483450  419053  857532  454063 1484705  424034  421837  425447
## [449]  456956  434467  755858  523484  436861  654141  469864  424034
## [457]  458549  930948  664366  420629  456236  515629  606111  463667
## [465]  431637  509364  634226  458558  483261  420749  446358  428405
## [473]  451996  423297  568490  447882  450246  456236  448626 1268339
## [481]  467579  455995  698363  617860  615893  427382  565313  591711
## [489]  520231  461337  419554  460408  454915  448337  536725  472070
## [497]  430175  446771  485117  500002  462294  443508  418020  435638
## [505]  420277  511517  438139  462255 1366120  495061  420351  431245
## [513]  434894  441210  419394  593246  449432  473133  440138  462838
## [521]  423222  529223  456618  651396  451951  431861  517036  436361
## [529]  497788  529216  441637  526734  543042  428299  427744  501144
## [537]  417668  631947  489085  436798  443855  438427  437890  540712
## [545]  549174  460437  806552  604537  487085  436341  473748  484024
## [553] 1455435  445382  659504  416745  439263  556688  750972  424884
## [561]  607848  454915  419895  548256  493363  463194  450695  422149
## [569]  552354  469056  435503  561489  455361  578377  509500  889965
## [577]  462180  506329  428499  507086  419732  659558  440129  609935
## [585]  521400  608184  425804  415913  513660  424478  422960  445728
## [593]  467108  615367  557236  562336  427474  493443  443546  430554
## [601]  434097  520078  460408  454934  474617  485117  456618  660461
## [609]  423222  442035  533147  497253  617898  449354  419722  440607
## [617]  442045  450544  953588  425622  609789  598995  421633  609789
## [625]  424719  482732  469697  452283  663394  417668  530454  494784
## [633]  436107  543477  452452  481096  420054  495982  556902  421412
## [641]  432052  418405  732102  548256  476334  709445  463072  469454
## [649]  423616  456604  609789  570821  438176  416356  421561  636017
## [657]  703107  544792  434463  434114  423222  418961  595088  438996
## [665]  607848  433705  462832  476334  527162  470875  416415  456572
## [673]  422836  566049  602513  509060  448026  491000  488541  520033
## [681]  554206  429346  455379  443742  520759  421837  694812  578701
## [689]  422013  462869  456618  549413  598802  511289  464103  462294
## [697]  427422  440417  439919  424494  806316  459548  541343  438839
## [705]  439592 1033222  424468  599629  571017  416577  425199  738812
## [713]  497280  447066  477209  431513  618191  544268  557853  535978
## [721]  668319  423024  491421  682947  469572  574271  456460  478829
## [729]  816750  597843  442274  595461  553405  506329  704108  481987
## [737]  460408  515712  551962  572751  745817  422933  473171  481175
## [745]  433170  476558  420986  447488  446512  497486  433330  496856
## [753] 1161363  435836  424591  425049  441542  419691  433330  444607
## [761]  459342  452808  427474  447555  422718  673764  424494  418405
## [769]  446654  434467  479621  472789  454843  456062  588484  809585
## [777]  493689  445382  482927  503454  574271  462820  478994  434268
## [785]  501671  594187  439779  509462  435469  548664  422813  498079
## [793]  431515  447488  466502  558490  456661  509048  419146  468713
## [801]  653574  706026  511068  427965  452640  475324  470203  513416
## [809]  421561  417941  535978  422249  442274  721712  615367  472580
## [817]  549174  437825 1097453  423222  461715  471452  426836  442131
## [825]  477867  461929  478380  479611  419146  472807  515797  475322
## [833]  510072  570562  491000  419134  423024  473133 1085515  500720
## [841]  421633  511668  455361  521665  478457  548361  591711  518530
## [849]  594187  417668  452406  499197  434430  509866  504871  695411
## [857]  420986  442359  462966  761006  484669  423616  467611  440647
## [865]  506830  574005  478205  604045  465974  415913  605502  589809
```

```
## [873]  426467  487347  588003  509629  431426  429897  709798  561334
## [881]  481987  570002  443546 1125613  454915  440706  532845  498328
## [889]  604380  583755  437909  420691  510072  557349  501172  609789
## [897]  476599  424094  557644  706180  425785  606752  417668  673764
## [905]  460214  475324  547886  554206  430035  456236  419740  462832
## [913]  440129  584790  425804  481987  799281  657397  496526  426431
## [921]  440969  487330  444554  512771  466325  440969  512828  422275
## [929]  531055  437666  472166  653574  417605  502837  444304  436798
## [937]  745768  478346  857532  715938  747719  569930  423217  433989
## [945]  475322  585361  452402  425497  502752  492263  543922  766115
## [953]  461337  421561  456922  584259  493034  538822  542265  430283
## [961]  498349  431245  491862  420895  448337  418702  477505  421467
## [969]  469454  749636  433906  437727  668362  449101  981628  470368
## [977]  746432  451059  499935  473625  566537  456367  455553  693066
## [985]  539864  447346  478315  427686  435842  485710  436163  514716
##
## [[3]][[2]]
## [1] 992
##
##
## [[4]]
## NULL
##
## [[5]]
## [[5]][[1]]
##    [1] 4 3 4 4 2 4 3 4 2 1 4 4 3 3 3 4 2 2 2 3 3 2 4 4 4 3 4 4 3 3 4 3 2 1
##   [35] 4 4 4 4 2 2 3 3 4 3 4 3 4 4 3 2 4 4 4 4 4 3 4 4 4 4 4 4 2 4 4 4 4 3 3
##   [69] 4 3 4 4 4 4 4 4 4 4 4 3 4 3 4 4 2 2 3 3 4 3 2 4 4 4 4 3 3 2 2 4 3 4 1 4
##  [103] 1 4 4 4 4 3 3 4 3 4 4 4 4 2 4 3 4 4 3 3 3 1 4 4 4 4 4 4 1 4 4 4 4 3 3 4 4 4 4 4
##  [137] 4 3 4 4 4 3 2 4 4 4 4 1 3 4 4 4 4 4 2 2 4 4 4 4 2 4 4 4 3 4 4 4 4 4 2 4 4 4 4 3 4
##  [171] 3 3 3 4 2 4 4 4 2 4 4 4 4 3 4 4 4 4 3 4 3 4 3 4 3 4 3 4 2 3 3 4 4 4 3 3 4 2 4 3
##  [205] 2 2 4 4 4 2 2 4 4 4 2 2 3 3 3 4 3 4 4 4 4 4 4 1 4 3 4 4 4 4 4 3 4 4 4 4 1 4 4
##  [239] 4 4 4 4 4 4 4 1 3 4 1 4 4 4 2 4 2 4 4 4 4 3 3 3 4 4 4 4 4 4 3 2 2 4 4 4 3 4 4 4 2
##  [273] 4 1 4 4 4 4 4 4 4 4 4 4 3 1 1 1 4 4 4 4 2 4 3 3 3 4 2 4 4 4 4 3 2 4 4 4 4 2 4
##  [307] 1 4 4 4 4 4 3 2 2 4 4 4 4 3 3 3 2 2 4 3 4 3 4 4 4 4 4 3 4 3 4 4 4 3 4 4 4 4 3
##  [341] 4 4 3 3 4 3 4 2 3 2 4 4 3 2 3 4 4 4 4 2 4 4 4 4 4 3 3 4 4 4 2 4 3 1 3 2 4 4 3
##  [375] 3 4 3 3 3 4 4 4 2 4 3 2 3 4 3 4 4 4 3 3 2 4 4 4 4 3 4 3 4 1 4 4 4 2 2 4 3 1 4
##  [409] 3 3 4 3 4 4 4 4 3 3 3 4 4 3 1 4 2 2 4 4 3 3 3 2 4 4 4 4 3 4 4 4 2 3 4 4 4 3 3 4
##  [443] 3 4 4 4 4 4 4 4 4 4 3 2 4 4 3 4 4 4 3 2 4 4 2 4 4 4 4 3 4 4 3 4 4 4 4 2 4 4 4 3 3 4 4 3
##  [477] 1 3 4 2 3 4 2 4 4 4 4 3 4 4 2 2 4 4 2 2 3 4 4 2 3 4 4 3 3 4 4 4 4 3 2 3 3 3 4 4 4 4 4
##  [511] 2 3 4 4 3 2 3 3 3 4 4 3 4 3 4 4 4 4 3 4 4 3 2 4 4 4 3 3 4 4 3 4 3 4 4 3 3 3 3 2 3 3
##  [545] 4 4 1 4 3 4 4 3 2 4 2 4 4 3 3 4 4 3 3 4 2 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 3 2 4
##  [579] 2 4 4 4 3 4 4 4 4 4 4 4 3 3 4 2 4 4 4 3 1 3 4 4 4 1 3 4 4 4 4 4 3 4 2 4 4 4 4 4 2
##  [613] 4 3 4 4 4 4 4 4 3 4 4 4 3 2 3 4 2 4 4 4 4 3 4 4 3 4 4 4 4 4 4 3 4 3 3 4 2 2 3 4 4
##  [647] 3 4 4 4 3 4 4 3 3 4 4 4 4 4 4 4 4 3 3 4 4 3 2 1 4 4 4 3 4 4 3 4 4 3 3 4 4 3 4 2 2 4 4
##  [681] 2 4 4 3 2 4 4 3 4 2 4 4 3 2 4 4 3 4 2 2 4 3 2 4 4 4 4 4 4 4 4 4 4 4 3 4 4 4 3 4 2 4
##  [715] 4 4 4 4 4 4 4 2 4 4 4 4 4 4 3 4 4 4 4 3 1 4 4 4 3 2 4 4 4 3 3 4 4 4 4 3 3 4 4 4 4 3 2 4
##  [749] 4 2 4 3 4 4 4 4 4 4 4 3 4 4 4 4 3 4 1 4 1 4 4 4 4 2 4 4 3 4 4 4 2 4 1 4 3 3 3 4 1 3 4
##  [783] 4 3 4 2 4 2 4 4 4 3 4 4 3 4 4 4 1 4 2 4 3 3 3 2 4 4 3 4 4 4 4 4 2 1 2 4 4 3 4 4 4 4 3
##  [817] 4 3 4 3 1 4 4 3 3 2 4 4 3 3 2 4 4 3 4 4 3 4 4 4 4 4 4 3 4 4 4 4 4 4 4 4 3 2 4 4 2 4 3 3 3
##  [851] 4 4 4 4 4 4 3 3 4 4 4 4 4 3 3 2 4 4 4 4 4 1 4 4 2 4 4 4 4 4 4 3 4 4 4 4 2 4 4 4 4 4 1 4
##  [885] 1 4 4 4 4 4 4 4 2 4 1 4 4 1 4 4 4 4 4 4 3 4 4 1 4 4 4 4 4 4 3 4 4 3 3 3 4 4 3 3 2 3 4 4
##  [919] 4 1 4 4 2 4 4 4 4 4 4 3 4 4 3 4 4 4 4 3 1 4 4 4 4 3 4 4 2 4 4 4 3 4 4 3 4 4 4 3 2 4 4 4 4 1
##  [953] 4 4 1 4 4 4 4 4 4 4 3 2 3 4 4 3 3 2 3 3 4 4 4 4 2 4 4 4 2 4 4 3 1 4 4 4 2 4 1 4 4
```

```
##  [987] 3 3 3 3 3 4 3 4 3 3 2 4 3 4 4 4 4 4 4 3 4 3 3 4 3 4 3 2 4 4 4 3 4 3
## [1021] 4 3 2 2 4 2 4 4 4 4 2 4 2 3 3 2 3 4 1 4 3 3 3 4 3 4 2 4 4 3 3 4 2 3
## [1055] 3 4 3 4 3 3 4 2 3 4 4 3 4 3 4 4 4 4 4 4 4 3 4 4 4 4 3 3 4 2 3 4 3 3
## [1089] 2 2 2 2 4 4 3 2 4 4 4 3 2 2 3 4 3 2 4 2 4 4 3 4 4 4 3 4 4 4 3 3 4 3
## [1123] 3 3 4 3 3 4 2 3 4 4 2 4 2 2 2 4 3 4 4 3 3 2 2 4 2 4 3 3 2 4 3 2 4 3
## [1157] 3 4 4 4 4 4 4 2 1 4 2 2 4 4 2 4 4 1 2 4 4 4 3 3 3 1 4 2 3 4 1 4 4 2
## [1191] 3 2 4 4 1 4 4 4
##
## [[5]][[2]]
## [1] 1198
##
##
## [[6]]
## NULL
```

```
fnlwgttrainout <- tail(order(rank(newtrain[,3])), 15)
fnlout <- c()
for(i in 1:length(fnlwgttrainout)){
  fnlout[i] <- newtrain[fnlwgttrainout[i], 3]
}

#head(order(rank(newtrain[,5])))

table(newtrain[,11])
```

```
##
##     0   114   401   594   914   991  1055  1086  1111  1151  1173  1409
## 29849     6     2    34     8     5    25     4     1     8     3     7
##  1424  1455  1471  1506  1639  1797  1831  1848  2009  2036  2050  2062
##     3     1     7    15     1     7     7     6     3     4     5     2
##  2105  2174  2176  2202  2228  2290  2329  2346  2354  2387  2407  2414
##     9    48    23    16     5     5     6     6    11     1    19     8
##  2463  2538  2580  2597  2635  2653  2829  2885  2907  2936  2961  2964
##    11     1    12    20    11     5    31    24    11     3     3     9
##  2977  2993  3103  3137  3273  3325  3411  3418  3432  3456  3464  3471
##     8     2    97    37     6    53    24     5     4     2    23     8
##  3674  3781  3818  3887  3908  3942  4064  4101  4386  4416  4508  4650
##    14    12     7     6    32    14    42    20    70    12    12    41
##  4687  4787  4865  4931  4934  5013  5060  5178  5455  5556  5721  6097
##     3    23    17     1     7    69     1    97    11     5     3     1
##  6360  6418  6497  6514  6723  6767  6849  7298  7430  7443  7688  7896
##     3     9    11     5     2     5    27   246     9     5   284     3
##  7978  8614  9386  9562 10520 10566 10605 11678 13550 14084 14344 15020
##     1    55    22     4    43     6    12     2    27    41    26     5
## 15024 15831 18481 20051 22040 25124 25236 27828 34095 41310 99999
##   347     6     2    37     1     4    11    34     5     2   159
```

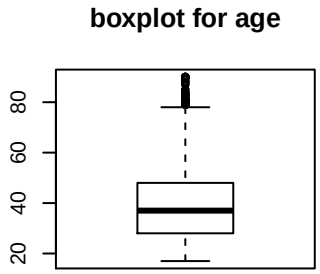
```
gainout <- tail(order(rank(newtrain[,11])), 159)

#Outliers removing for training sets.
dim(newtrain)
```

```
## [1] 32561    15
```

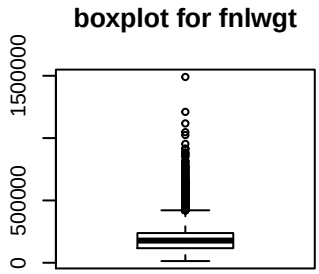
```
newtrain <- newtrain[-gainout, ]
dim(newtrain)
```

```
## [1] 32402    15
```

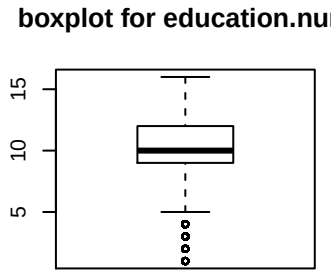
```
#Deal with outliers for testing sets
for(i in continuouscol){
  boxplot(newtest[,i], main = paste("boxplot for", colnames(newtest[i])),
          xlab = colnames(newtest[i]))
}
```



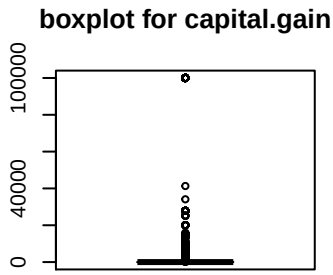

age



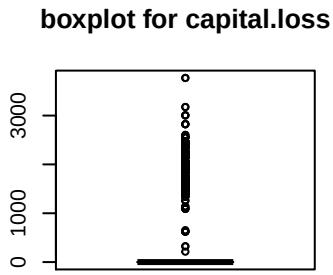

fnlwgt



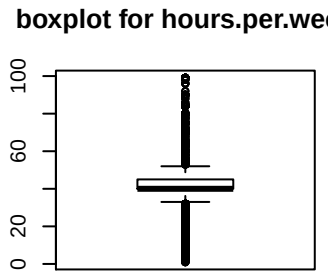

education.num




capital.gain

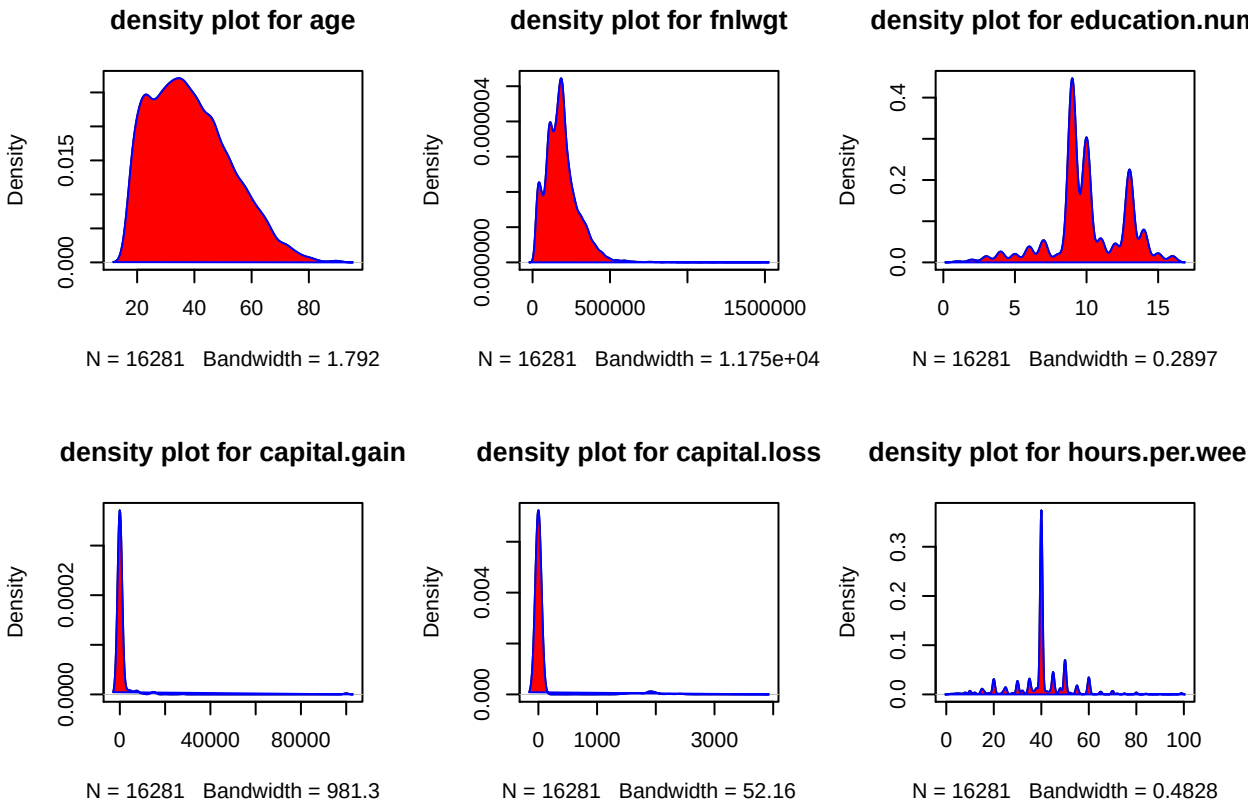



capital.loss




hours.per.week

```
for(i in continuouscol){
  den_acc <- density(newtest[,i], adjust = 1)
  plot(den_acc, main = paste("density plot for", colnames(newtest[i])))
  polygon(den_acc, col = "red", border = "blue")
}
```




```
outlierstest <- list()
for(i in continuouscol){
  outliers <- boxplot.stats(newtest[,i])$out
  numbers <- length(outliers)
  outlierstest[[i]] <- list(outliers, numbers)
}
head(outlierstest)
```

```
## [[1]]
## [[1]][[1]]
##  [1] 79 80 90 79 80 81 82 83 81 85 80 90 81 84 81 89 81 83 81 82 80 90 81
## [24] 83 80 90 90 84 80 80 80 81 90 85 90 81 81 80 80 79 81 80 88 87 90 79
## [47] 83 79 80 90 79 79 81 81 90 82 90 87 81 88 80 81 80 81 90 88 89 84 80
## [70] 80 83 79 81
##
## [[1]][[2]]
## [1] 73
##
##
## [[2]]
## NULL
##
## [[3]]
## [[3]][[1]]
##   [1]  444554  432824  465326  445382  479296  428420  456736  537222
##   [9]  513100  447488  512864  500068  446894  599057  479179  471990
##  [17]  457162  455379  542610  479600  448026  437200  652784  573446
##  [25]  453233  662460  426589  629900  499971  450770  481987  478373
```

```
##  [33]  486194  509364  632733  504725  560313  651702  644278  535852
##  [41]  445758  452353  475775  455469  522241  427744  473206  427541
##  [49]  581128  444725  608881  490871  430151  431245  451019  430336
##  [57]  433602  437994  436431  914061  624006  510072  484475  505365
##  [65]  593246  714597  816750  491214  446724  552529  454717  425622
##  [73]  575172  475322  622192  566066  493732  427437  427320  614113
##  [81]  445365  472517  459556  548568  565769  429832  424988  426350
##  [89]  789600  424340  447144  864960  497414  471876  723746  427422
##  [97]  421837  692831  535869  433624  638116  467936  698039  427812
## [105]  472861  449101  677398  464621  547931  497039  451742  460322
## [113]  666014  474568  452640  765214  445480  761800  460356 1047822
## [121]  436651  544319  617917  450695  429696  443377  522881  437161
## [129]  421010  479296  459189  469005  457070  750972  505365  458609
## [137]  520231  589155  538193  428251  454321  455399  477345  470486
## [145]  437318  588739  449578  486436  588484  449101  528618  806552
## [153]  478354  467936  505168  858091  451327  482082  663291  447554
## [161]  451603  455995  460408  581025  453983  656488  421633  478457
## [169]  422836  557349  421350  498267  442478  421228  655066  426431
## [177]  494371  737315  541755  436198  594521  442656  491000  455995
## [185]  430672  496856  589838  479296  605504  490332  423453  445382
## [193]  558752  448862  429281  772919  884434  495288  488720  444554
## [201]  604045  437940  697806  632271  497788  464484  587310  467759
## [209]  472344  438587  427055  538243  441227  459465  454950  439777
## [217] 1490400  768659  764638  437458  517995  718736  433682  477083
## [225]  442478  547108  474229  498833  882849  453663  443508  498411
## [233]  504423  746660  488459  423883  457357  501671  786418  565313
## [241]  483201  466458  424934  450200  465334  482096  451603  465725
## [249]  502633  473133  477867  435356  478457  653215  437825  576645
## [257]  510643  538099  425502  432480  482211  539019  496743  455379
## [265]  421132  452402  531055  454076  434081  452402  434710  446947
## [273]  472411  594187  685955  442116  435835  430278  548361  606111
## [281]  459192  592029  426263  513977  647591  566066  553588  433325
## [289]  491607  624572  488706  535740  607118  482677  420973  426431
## [297]  580591  449172  438427  557853  446390  487751  469263  478972
## [305]  441949  430930  635913  485944  557805  626493  444134  433580
## [313]  493034  914061  456736  557349  443336  953588  473547  457710
## [321]  471768  558344  421871  430710  481258  590204  679853  421474
## [329]  443809  516701  443546  535762  438321  814850  427812  874728
## [337]  497525  434102  450141  441949  438429  506830  478277  594194
## [345]  445480  452963  498267  538583  602513  589809  421474  507492
## [353]  546118  446647  530099  453686  443377 1117718  427248  461725
## [361]  460259  849067  590941  572285  608441  720428  423311  436361
## [369]  463601  557359  454024  431515  590522  443546  433592  479406
## [377]  430195  421633  428299  484911  478836  513440  744929  534775
## [385]  511231  598995  456592  525848  442359  458168  457453  913447
## [393]  584259  694105  441227  448841  606347  437566  495366 1024535
## [401]  427474  811615  431551  461929  533660  445382  427475 1210504
## [409]  426263  425830  421837  427770  447210  455995  435836  425816
## [417]  490645  513977  553405  497414  742903  431745  553405  504941
## [425]  450141  456665  449376  487770  448026  443858  473449  440934
## [433]  456430  421200  426589  484879  438696  435638  535027  464552
## [441]  443701  438427  513719  439263  425444  454585  428251  618130
## [449]  542762  771836  473133  464552  435266  437161  462964  423605
## [457]  618808  573446  432204  461484  455379  504871  532969  455665
```

```
## [465]  425127  449925  427515  607658  422933  430340  440129
##
## [[3]][[2]]
## [1] 471
##
##
## [[4]]
## NULL
##
## [[5]]
## [[5]][[1]]
##   [1] 4 4 3 4 4 4 4 4 4 3 2 3 4 4 2 4 4 3 3 2 4 3 3 4 3 3 4 4 4 1 1 4 3 2 4
##  [36] 4 2 3 4 4 1 4 1 4 4 4 3 4 4 3 4 3 4 2 4 2 4 4 4 3 4 2 4 4 3 3 1 1 4 3
##  [71] 4 2 3 4 3 3 3 4 4 4 4 4 3 3 3 2 2 4 4 4 4 3 3 4 3 3 3 3 1 2 3 3 3 1 4
## [106] 4 4 4 4 4 4 4 2 3 4 4 3 4 4 4 3 3 3 4 4 1 4 4 4 3 4 2 4 2 4 4 4 4 3 3
## [141] 4 4 1 4 3 4 4 4 3 4 4 4 3 3 3 4 2 2 4 2 4 4 4 4 4 4 4 4 4 2 4 4 3 4 1
## [176] 2 3 4 3 2 4 1 4 2 3 3 4 4 4 1 2 2 4 3 4 4 4 4 3 2 4 4 4 4 3 3 3 4 3 4
## [211] 2 4 4 4 3 4 3 2 4 4 3 4 2 2 4 1 2 3 4 2 4 4 4 4 4 2 4 4 4 3 4 3 4 3 4
## [246] 3 4 3 4 3 4 4 4 4 3 3 3 2 3 4 3 4 4 4 3 1 2 2 2 2 3 1 2 3 4 4 4 1 1 2
## [281] 4 4 4 4 2 4 3 4 3 1 3 3 1 3 4 4 4 4 4 4 3 3 3 3 3 3 4 4 4 4 3 4 4 3 2
## [316] 4 4 2 4 4 3 4 3 4 4 4 4 4 4 2 3 4 4 3 2 4 2 4 4 4 4 2 3 4 4 3 3 4 3 2 3
## [351] 4 2 3 4 4 3 4 4 2 4 4 3 2 4 4 4 2 4 4 4 3 4 3 3 4 2 4 2 3 3 3 4 3 4 3
## [386] 4 1 4 3 4 4 3 4 2 4 2 3 3 4 3 2 1 1 2 3 3 4 3 1 3 3 2 4 3 4 3 3 3 4 3
## [421] 4 4 2 3 3 3 3 1 3 3 2 4 3 4 1 2 3 4 4 4 4 4 4 3 3 2 3 4 4 3 4 2 4 4 4
## [456] 4 4 2 4 2 4 2 4 4 3 4 3 2 4 3 4 4 3 4 4 4 4 4 3 4 4 3 4 3 4 4 3 2 4 2
## [491] 2 4 2 4 3 4 4 3 4 3 4 3 4 1 1 4 3 2 4 4 4 4 3 3 4 4 2 4 4 4 3 4 3 1 4
## [526] 3 3 4 3 4 4 4 4 4 4 4 4 4 3 2 3 4 3 4 4 4 4 4 3 4 4 3 4 3 4 2 2 3 2 3
## [561] 3 3 4 4 4 1 3 3 3 4 4 1 3 4 2 3 3 3 2 3 3 4 4 4 3 4 4 1 4 4 4 4 4 4 4
## [596] 4
##
## [[5]][[2]]
## [1] 596
##
##
## [[6]]
## NULL
```

```
table(newtest[,11])
```

```
##
##     0   114   401   594   914   991  1055  1086  1151  1173  1264  1409
## 14958     2     3    18     2     1    12     4     5     2     2     3
##  1424  1455  1471  1506  1731  1797  1831  1848  2036  2062  2105  2174
##     1     3     2     9     1     3     2     3     1     1     6    26
##  2176  2202  2290  2329  2346  2354  2407  2414  2463  2538  2580  2597
##     8    12     5     1     2    10     6     2     4     4     8    11
##  2635  2653  2829  2885  2907  2936  2961  2964  2977  2993  3103  3137
##     3     6    11     6     7     1     1     5     3     1    55    14
##  3273  3325  3411  3418  3456  3464  3471  3674  3781  3818  3887  3908
##     1    28    10     3     4    10     3     8     4     4     2    10
##  3942  4064  4101  4386  4416  4508  4650  4687  4787  4865  4931  4934
##     4    12     9    38    12    11    22     1    12     8     3     3
##  5013  5060  5178  5455  5556  5721  6097  6418  6497  6514  6612  6723
##    48     1    49     7     1     4     1     7     4     5     1     3
##  6767  6849  7262  7298  7430  7443  7688  7896  7978  8614  9386  9562
```

```
##     1    15     1   118     6     2   126     1     1    27     9     1
## 10520 10566 10605 11678 13550 14084 14344 15020 15024 15831 20051 25124
##    21     2     7     2    15     8     8     5   166     2    12     2
## 25236 27828 34095 41310 99999
##     3    24     1     1    85
```

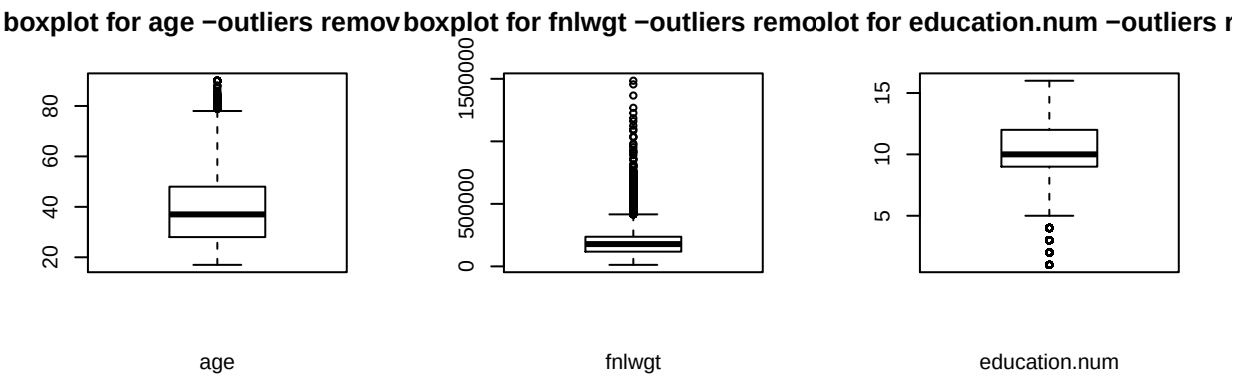
```
gainout <- tail(order(rank(newtest[,11])), 85)

#Outliers removing for training sets.
dim(newtest)
```

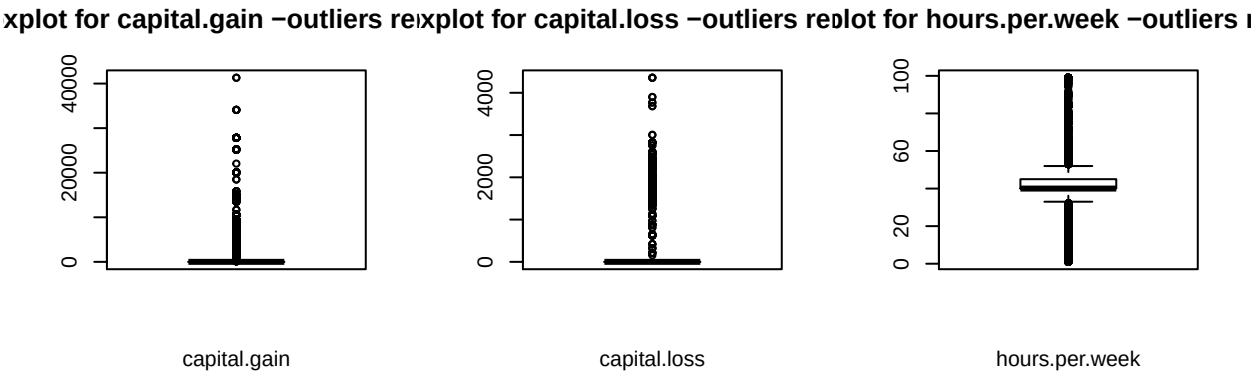
```
## [1] 16281    15
```

```
newtest <- newtest[-gainout, ]
dim(newtest)
```

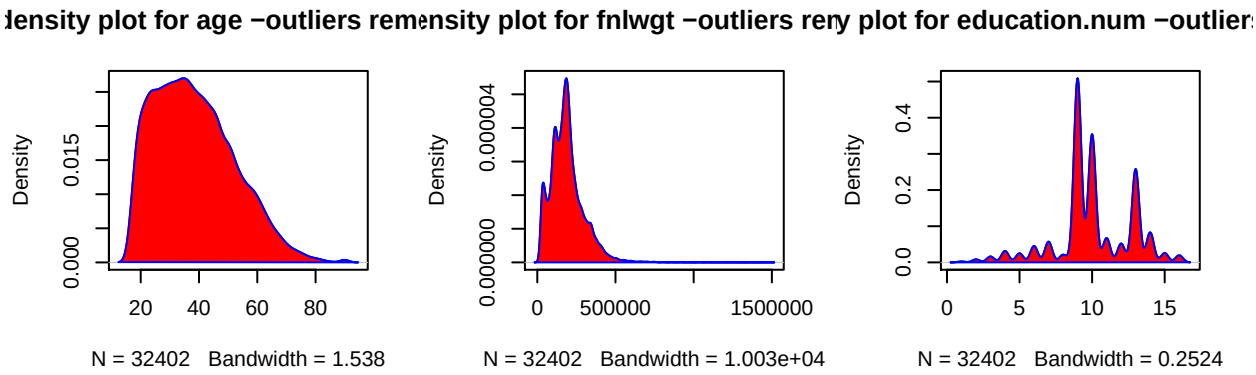
```
## [1] 16196    15
```

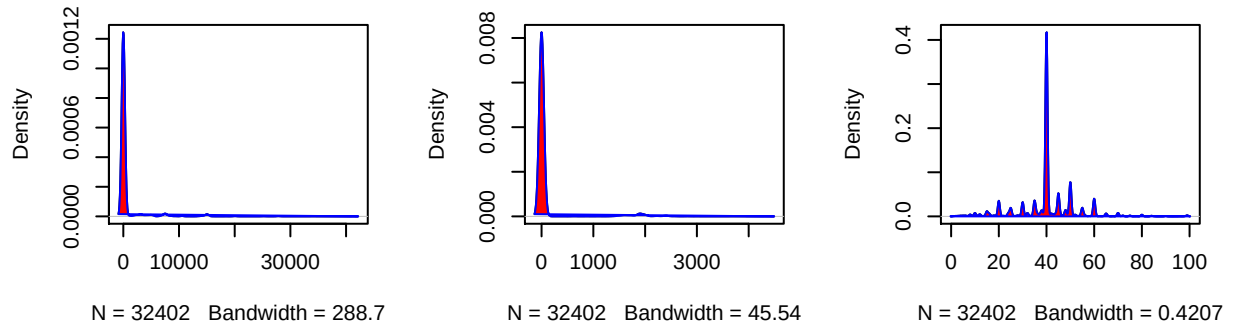
```
#Plots after removing outliers training
for(i in continuouscol){
  boxplot(newtrain[,i], main = paste("boxplot for", colnames(newtrain[i]), "-outliers removed"),
          xlab = colnames(newtrain[i]))
}
```



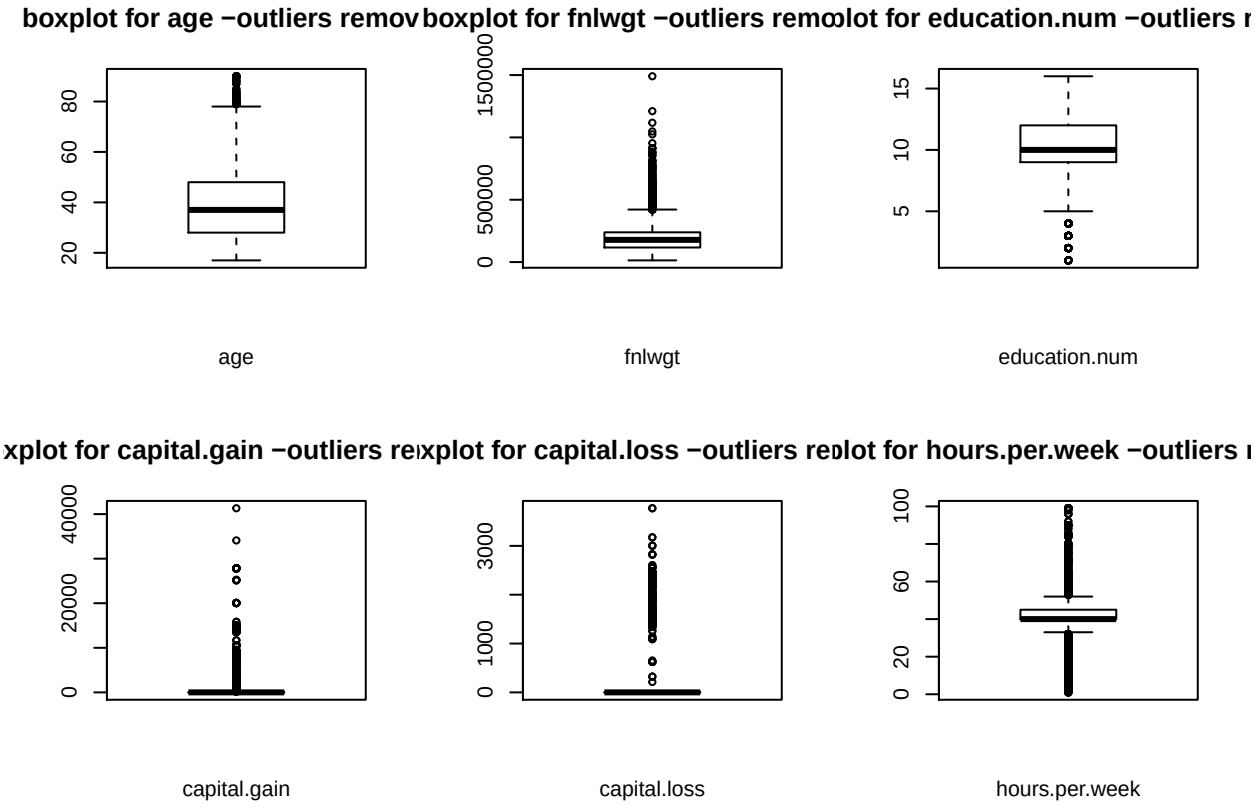




```
for(i in continuouscol){
  den_acc <- density(newtrain[,i], adjust = 1)
  plot(den_acc, main = paste("density plot for", colnames(newtrain[i]), "-outliers removed"))
  polygon(den_acc, col = "red", border = "blue")
}
```

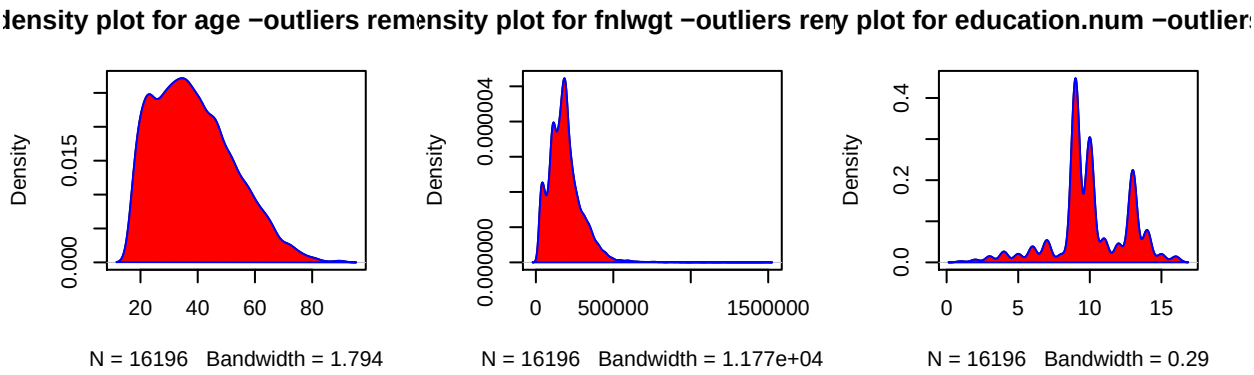


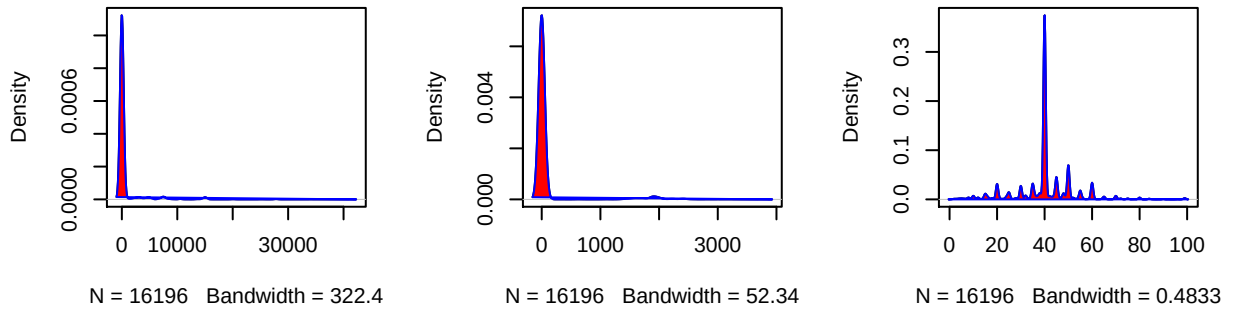




```
#Plots after removing outliers testing
for(i in continuouscol){
  boxplot(newtest[,i], main = paste("boxplot for", colnames(newtest[i]), "-outliers removed"),
          xlab = colnames(newtest[i]))
}
```




```
for(i in continuouscol){
  den_acc <- density(newtest[,i], adjust = 1)
  plot(den_acc, main = paste("density plot for", colnames(newtest[i]), "-outliers removed"))
  polygon(den_acc, col = "red", border = "blue")
}
```




### c) 6 - 8 EDAs

```
#See structure and summaries after removing outliers
str(newtest)
```

```
## 'data.frame':    16196 obs. of  15 variables:
##  $ age           : int  25 38 28 44 18 34 29 63 24 55 ...
##  $ workclass     : Factor w/ 8 levels "Federal-gov",..: 4 4 2 4 4 4 4 6 4 4 ...
##  $ fnlwgt        : int  226802 89814 336951 160323 103497 198693 227026 104626 369667 104996 ...
##  $ education     : Factor w/ 16 levels "10th","11th",..: 2 12 8 16 16 1 12 15 16 6 ...
##  $ education.num : int  7 9 12 10 10 6 9 15 10 4 ...
##  $ marital.status: Factor w/ 7 levels "Divorced","Married-AF-spouse",..: 5 3 3 3 5 5 5 3 5 3 ...
##  $ occupation    : Factor w/ 14 levels "Adm-clerical",..: 7 5 11 7 12 8 6 10 8 3 ...
##  $ relationship  : Factor w/ 6 levels "Husband","Not-in-family",..: 4 1 1 1 4 2 5 1 5 1 ...
##  $ race          : Factor w/ 5 levels "Amer-Indian-Eskimo",..: 3 5 5 3 5 5 3 5 5 5 ...
##  $ sex           : Factor w/ 2 levels "Female","Male": 2 2 2 2 1 2 2 2 1 2 ...
##  $ capital.gain  : int  0 0 0 7688 0 0 0 3103 0 0 ...
##  $ capital.loss  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hours.per.week: int  40 50 40 40 30 30 40 32 40 10 ...
##  $ native.country: Factor w/ 40 levels "Cambodia","Canada",..: 38 38 38 38 38 38 38 38 38 38 ...
##  $ income        : Factor w/ 2 levels "<=50K.",">50K.": 1 1 2 2 1 1 1 2 1 1 ...
```

```
summary(newtest)
```

```
##       age                     workclass         fnlwgt
##  Min.   :17.00   Private         :11919   Min.   :  13492
##  1st Qu.:28.00   Self-emp-not-inc: 1421   1st Qu.: 116808
##  Median :37.00   Local-gov       : 1089   Median : 177856
##  Mean   :38.72   State-gov       :  707   Mean   : 189529
##  3rd Qu.:48.00   Self-emp-inc    :  570   3rd Qu.: 238567
##  Max.   :90.00   Federal-gov     :  480   Max.   :1490400
##                  (Other)         :   10
##         education    education.num                marital.status
##  HS-grad     :5272   Min.   : 1.00   Divorced             :2181
##  Some-college:3583   1st Qu.: 9.00   Married-AF-spouse    :  13
##  Bachelors   :2648   Median :10.00   Married-civ-spouse   :7340
##  Masters     : 922   Mean   :10.06   Married-spouse-absent: 210
##  Assoc-voc   : 677   3rd Qu.:12.00   Never-married        :5425
##  11th        : 637   Max.   :16.00   Separated            : 503
##  (Other)     :2457                   Widowed              : 524
##            occupation           relationship                    race
##  Prof-specialty :2077   Husband       :6465   Amer-Indian-Eskimo:  159
##  Craft-repair   :2032   Not-in-family :4262   Asian-Pac-Islander:  475
##  Exec-managerial:2009   Other-relative: 525   Black              : 1558
##  Adm-clerical   :1965   Own-child     :2511   Other              :  134
##  Sales          :1912   Unmarried     :1676   White              :13870
##  Other-service  :1824   Wife          : 757
##  (Other)        :4377
##      sex          capital.gain      capital.loss     hours.per.week
##  Female: 5407   Min.   :    0.0   Min.   :   0.00   Min.   : 1.00
##  Male  :10789   1st Qu.:    0.0   1st Qu.:   0.00   1st Qu.:40.00
##                 Median :    0.0   Median :   0.00   Median :40.00
##                 Mean   :  562.8   Mean   :  88.36   Mean   :40.33
##                 3rd Qu.:    0.0   3rd Qu.:   0.00   3rd Qu.:45.00
##                 Max.   :41310.0   Max.   :3770.00   Max.   :99.00
##
##        native.country     income
##  United-States:14813   <=50K.:12435
##  Mexico       :  310   >50K. : 3761
##  Philippines  :  109
##  Puerto-Rico  :   70
##  Germany      :   69
##  Canada       :   61
##  (Other)      :  764
```

```
str(newtrain)
```

```
## 'data.frame':    32402 obs. of  15 variables:
##  $ age           : int  39 50 38 53 28 37 49 52 31 42 ...
##  $ workclass     : Factor w/ 8 levels "Federal-gov",..: 7 6 4 4 4 4 4 6 4 4 ...
##  $ fnlwgt        : 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 ...
##  $ education.num : int  13 13 9 7 13 14 5 9 14 13 ...
##  $ marital.status: Factor w/ 7 levels "Divorced","Married-AF-spouse",..: 5 3 1 3 3 3 4 3 5 3 ...
##  $ 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 ...
##  $ capital.gain  : int  2174 0 0 0 0 0 0 0 14084 5178 ...
##  $ capital.loss  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hours.per.week: int  40 13 40 40 40 40 16 45 50 40 ...
##  $ native.country: Factor w/ 41 levels "Cambodia","Canada",..: 39 39 39 39 5 39 23 39 39 39 ...
##  $ income        : Factor w/ 2 levels "<=50K",">50K": 1 1 1 1 1 1 1 2 2 2 ...
```

```
summary(newtrain)
```

```
##       age                    workclass         fnlwgt
##  Min.   :17.00   Private         :23984   Min.   :  12285
##  1st Qu.:28.00   Self-emp-not-inc: 2747   1st Qu.: 117793
##  Median :37.00   Local-gov       : 2187   Median : 178383
##  Mean   :38.54   State-gov       : 1351   Mean   : 189763
##  3rd Qu.:48.00   Self-emp-inc    : 1127   3rd Qu.: 237049
##  Max.   :90.00   Federal-gov     :  983   Max.   :1484705
##                  (Other)         :   23
##         education     education.num                  marital.status
##  HS-grad     :10478   Min.   : 1.00   Divorced              : 4432
##  Some-college: 7277   1st Qu.: 9.00   Married-AF-spouse     :   23
##  Bachelors   : 5314   Median :10.00   Married-civ-spouse    :14844
##  Masters     : 1705   Mean   :10.07   Married-spouse-absent:  417
##  Assoc-voc   : 1381   3rd Qu.:12.00   Never-married         :10671
##  11th        : 1175   Max.   :16.00   Separated             : 1023
##  (Other)     : 5072                   Widowed               :  992
##            occupation           relationship                    race
##  Prof-specialty :4228   Husband       :13072   Amer-Indian-Eskimo:  311
##  Craft-repair   :4154   Not-in-family : 8284   Asian-Pac-Islander: 1029
##  Exec-managerial:4085   Other-relative:  981   Black             : 3117
##  Adm-clerical   :3986   Own-child     : 5066   Other             :  269
##  Other-service  :3694   Unmarried     : 3442   White             :27676
##  Sales          :3690   Wife          : 1557
##  (Other)        :8565
##      sex          capital.gain      capital.loss     hours.per.week
##  Female:10749   Min.   :    0.0   Min.   :   0.00   Min.   : 1.00
##  Male  :21653   1st Qu.:    0.0   1st Qu.:   0.00   1st Qu.:40.00
##                 Median :    0.0   Median :   0.00   Median :40.00
##                 Mean   :  592.2   Mean   :  87.73   Mean   :40.39
##                 3rd Qu.:    0.0   3rd Qu.:   0.00   3rd Qu.:45.00
##                 Max.   :41310.0   Max.   :4356.00   Max.   :99.00
##
##        native.country    income
##  United-States:29528   <=50K:24720
##  Mexico       :  656   >50K : 7682
##  Philippines  :  210
##  Germany      :  137
##  Canada       :  120
##  Puerto-Rico  :  114
##  (Other)      : 1637
```

```
#Analyzing/checking before discretizing
table(newtrain[,14])
```

```
##
```

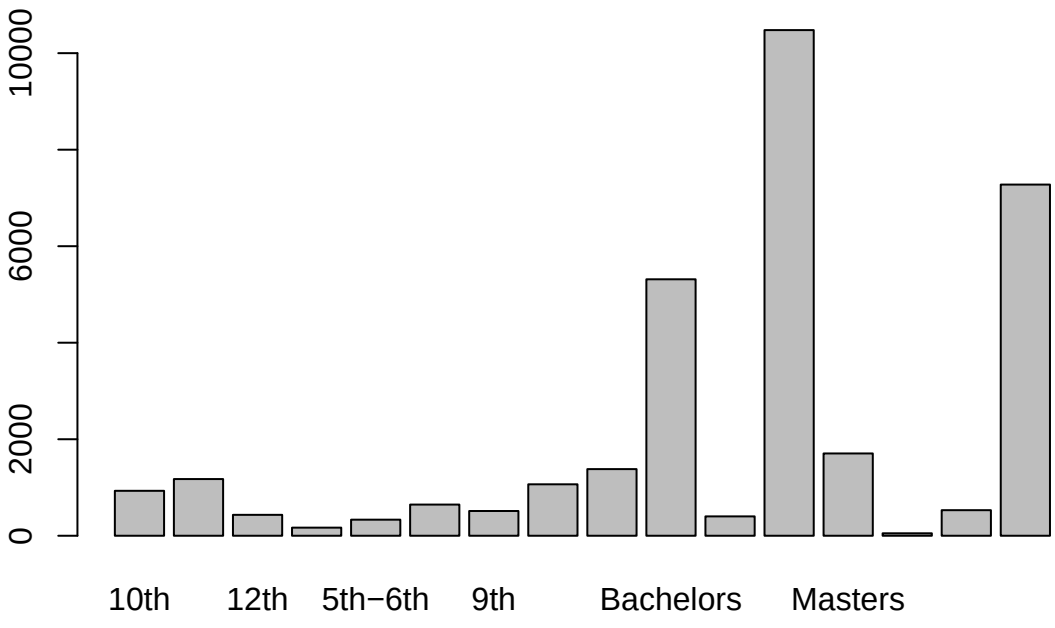
```
##                    Cambodia                     Canada
##                          20                        120
##                       China                   Columbia
##                          79                         59
##                        Cuba         Dominican-Republic
##                          95                         70
##                     Ecuador                El-Salvador
##                          28                        106
##                     England                     France
##                          90                         29
##                     Germany                     Greece
##                         137                         29
##                   Guatemala                      Haiti
##                          64                         44
##          Holand-Netherlands                    Honduras
##                           1                         13
##                        Hong                     Hungary
##                          23                         13
##                       India                        Iran
##                         104                         43
##                     Ireland                     Italy
##                          24                         74
##                     Jamaica                     Japan
##                          81                         66
##                        Laos                    Mexico
##                          22                        656
##                   Nicaragua Outlying-US(Guam-USVI-etc)
##                          34                         14
##                        Peru                Philippines
##                          31                        210
##                      Poland                   Portugal
##                          60                         37
##                 Puerto-Rico                   Scotland
##                         114                         12
##                       South                     Taiwan
##                          89                         56
##                    Thailand            Trinadad&Tobago
##                          19                         19
##               United-States                    Vietnam
##                       29528                         73
##                  Yugoslavia
##                          16
```

```
table(newtest[,14])
```

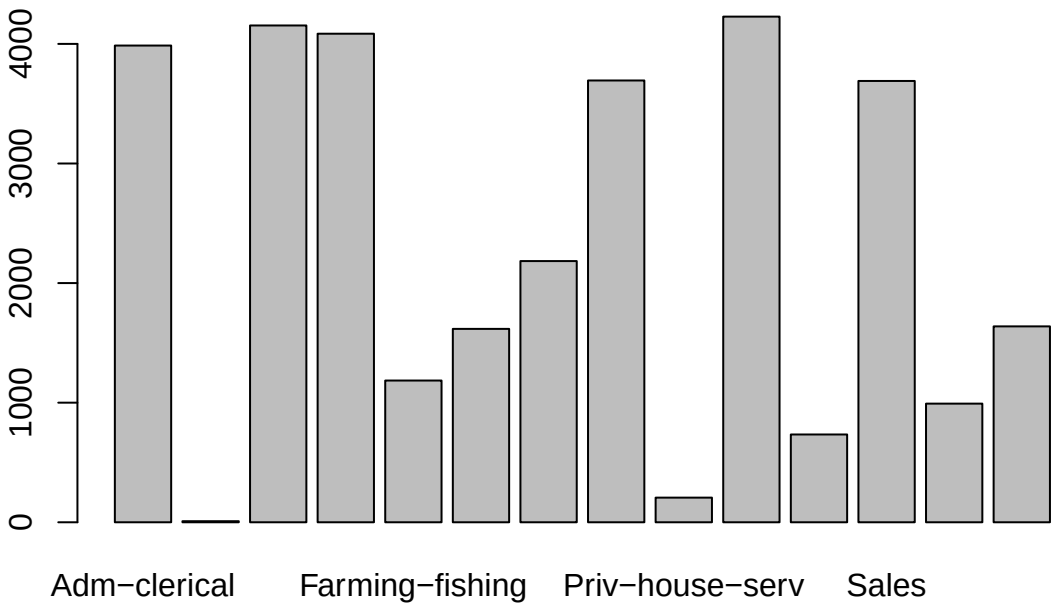
```
##
##                    Cambodia                     Canada
##                          12                         61
##                       China                   Columbia
##                          50                         26
##                        Cuba         Dominican-Republic
##                          43                         34
##                     Ecuador                El-Salvador
##                          17                         49
##                     England                     France
```

```
##                         38                          9
##                    Germany                     Greece
##                         69                         20
##                  Guatemala                      Haiti
##                         24                         31
##                   Honduras                       Hong
##                          7                         10
##                    Hungary                      India
##                          6                         56
##                       Iran                    Ireland
##                         16                         13
##                      Italy                    Jamaica
##                         32                         25
##                      Japan                      Laos
##                         32                          5
##                     Mexico                  Nicaragua
##                        310                         15
## Outlying-US(Guam-USVI-etc)                       Peru
##                          9                         15
##                Philippines                     Poland
##                        109                         27
##                   Portugal                  Puerto-Rico
##                         30                         70
##                   Scotland                      South
##                          9                         37
##                     Taiwan                   Thailand
##                         17                         13
##             Trinadad&Tobago              United-States
##                          8                      14813
##                    Vietnam                 Yugoslavia
##                         22                          7
```

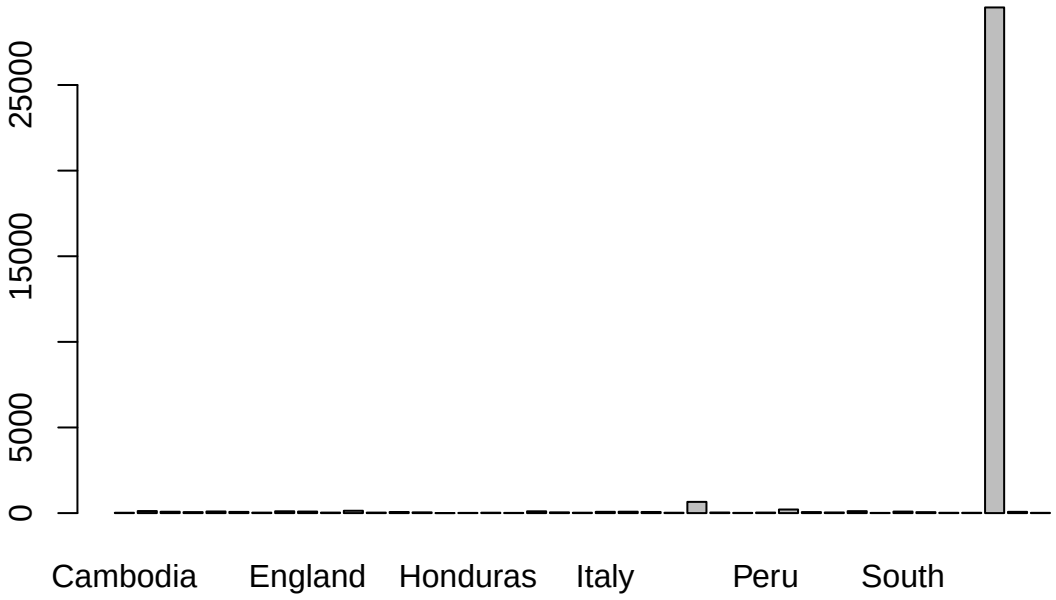
```
plot(newtrain$education)
```



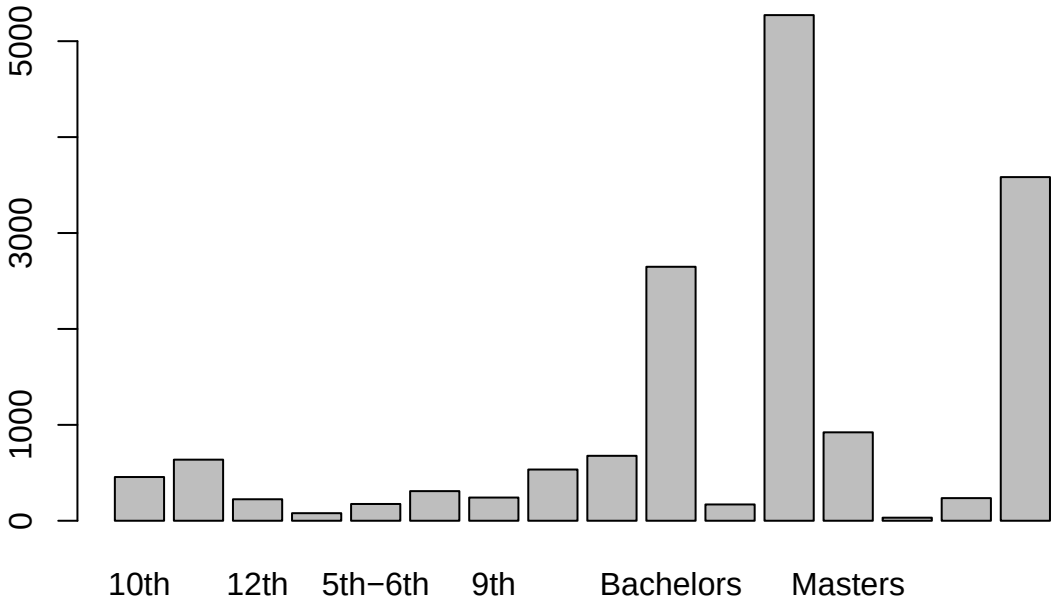
```
plot(newtrain$occupation)
```



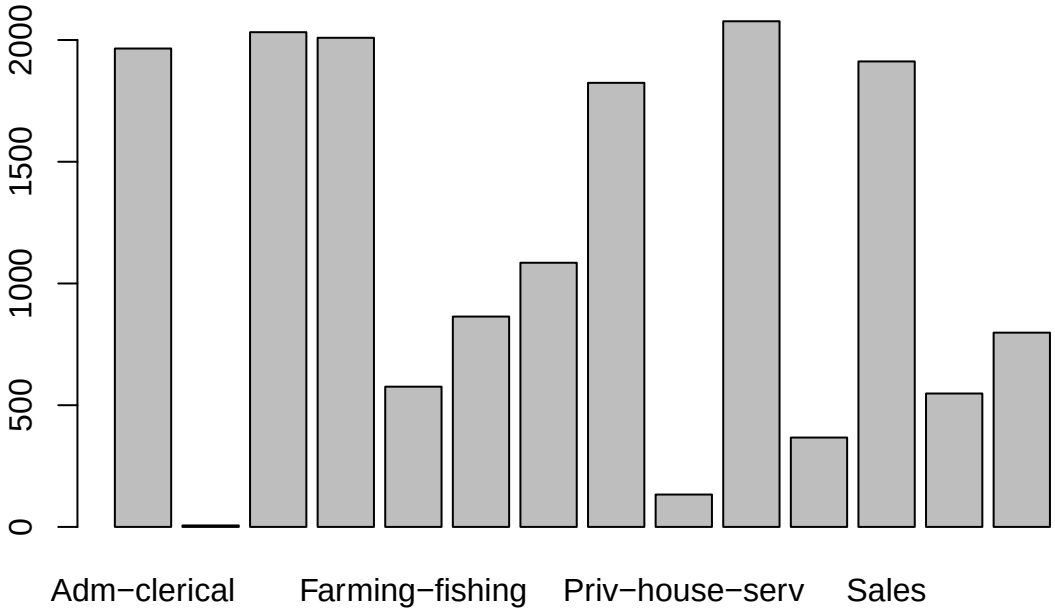

```
plot(newtrain$native.country)
```



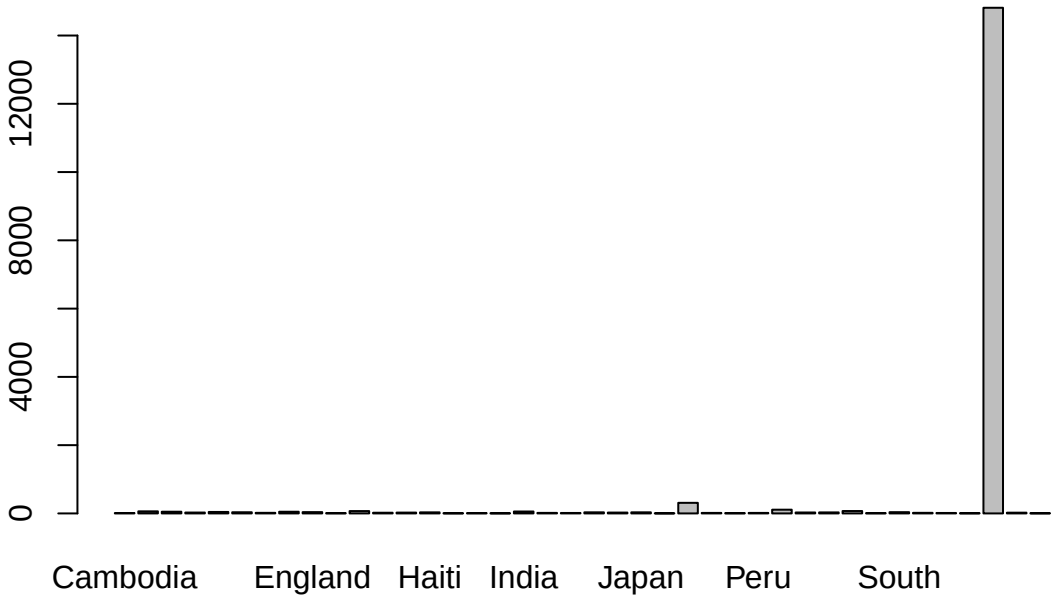

```
plot(newtest$education)
```




```
plot(newtest$occupation)
```




```
plot(newtest$native.country)
```




```
#Discretize training set
# discretetrainage <- discretize(newtrain$age, method = "interval", categories = 10)
# discretetrainfnlwgt <- discretize(newtrain$fnlwgt, method = "interval", categories = 10)
# discretetrainedunum <- discretize(newtrain$education.num, method = "interval", categories = 10)
# discretetraingain <- discretize(newtrain$capital.gain, method = "interval", categories = 10)
# discretetrainloss <- discretize(newtrain$capital.loss, method = "interval", categories = 10)
# discretetrainhours <- discretize(newtrain$hours.per.week, method = "interval", categories = 10)

countrydis <- function(vector){
  len <- length(vector)
  for(i in 1:len){
      if(vector[i] == "United-States"){
        vector[i] <- vector[i]
      }else if(vector[i] == "Mexico"){
        vector[i] <- vector[i]
      }else if(vector[i] == "Philippines"){
        vector[i] <- vector[i]
      }else{
        vector[i] <- "other_countries"
      }
  }
  return(vector)
}

#discretetraincountry <- as.factor(countrydis(as.character(newtrain$native.country)))



#Discretize testing set
# discretetestage <- discretize(newtest$age, method = "interval", categories = 10)
# discretetestfnlwgt <- discretize(newtest$fnlwgt, method = "interval", categories = 10)
# discretetestedunum <- discretize(newtest$education.num, method = "interval", categories = 10)
# discretetestgain <- discretize(newtest$capital.gain, method = "interval", categories = 10)
# discretetestloss <- discretize(newtest$capital.loss, method = "interval", categories = 10)
# discretetesthours <- discretize(newtest$hours.per.week, method = "interval", categories = 10)
```

```
#discretetestcountry <- as.factor(countrydis(as.character(newtest$native.country)))

#Combine training and testing to make the same intervals for discretizing
newtrain$type <- "train"
newtest$type <- "test"

combined <- rbind(newtrain, newtest)

discreteage <- discretize(combined$age, method = "interval", categories = 10)
discretefnlwgt <- discretize(combined$fnlwgt, method = "interval", categories = 10)
discreteedunum <- discretize(combined$education.num, method = "interval", categories = 10)
discretegain <- discretize(combined$capital.gain, method = "interval", categories = 7) #not enough data
discreteloss <- discretize(combined$capital.loss, method = "interval", categories = 7) #not enough data
discretehours <- discretize(combined$hours.per.week, method = "interval", categories = 10)
discretecountry <- as.factor(countrydis(as.character(combined$native.country)))

combined$age <- discreteage
combined$fnlwgt <- discretefnlwgt
combined$education.num <- discreteedunum
combined$capital.gain <- discretegain
combined$capital.loss <- discreteloss
combined$hours.per.week <- discretehours
combined$native.country <- discretecountry

dim(combined)
```

```
## [1] 48598    16
```

```
newtrain2 <- combined[1:sum(combined$type == "train"), -16]
newtest2 <- combined[(sum(combined$type == "train")+1):nrow(combined), -16]
dim(newtrain2)
```

```
## [1] 32402    15
```

```
dim(newtest2)
```

```
## [1] 16196    15
```

```
#Assignining discretized variables
# newtrain2 <- newtrain
# newtest2 <- newtest
# dim(newtrain2)
# dim(newtest2)
#
# newtrain2$age <- discretetrainage
# newtrain2$fnlwgt <- discretetrainfnlwgt
# newtrain2$education.num <- discretetrainedunum
# newtrain2$capital.gain <- discretetraingain
# newtrain2$capital.loss <- discretetrainloss
# newtrain2$hours.per.week <- discretetrainhours
# newtrain2$native.country <- discretetraincountry
#
# newtest2$age <- discretetestage
# newtest2$fnlwgt <- discretetestfnlwgt
# newtest2$education.num <- discretetestedunum
# newtest2$capital.gain <- discretetestgain
```

```
# newtest2$capital.loss <- discretetestloss
# newtest2$hours.per.week <- discretetesthours
# newtest2$native.country <- discretetestcountry



#Dummify training set
dumtrainwork <- dummy(newtrain$workclass)
dumtrainedu <- dummy(newtrain$education)
dumtrainmarry <- dummy(newtrain$marital.status)
dumtrainoccu <- dummy(newtrain$occupation)
dumtrainrelation <- dummy(newtrain$relationship)
dumtrainrace <- dummy(newtrain$race)
dumtrainsex <- dummy(newtrain$sex)



#Dummify testing set
dumtestwork <- dummy(newtest$workclass)
dumtestedu <- dummy(newtest$education)
dumtestmarry <- dummy(newtest$marital.status)
dumtestoccu <- dummy(newtest$occupation)
dumtestrelation <- dummy(newtest$relationship)
dumtestrace <- dummy(newtest$race)
dumtestsex <- dummy(newtest$sex)



#Take out columns
newtrain2 <- newtrain2[,-c(2, 4, 6, 7, 8, 9, 10)]
newtest2 <- newtest2[,-c(2, 4, 6, 7, 8, 9, 10)]



#Assigning dummified variables
newtrain2 <- cbind(newtrain2, dumtrainwork, dumtrainedu, dumtrainmarry, dumtrainoccu,
                   dumtrainrelation, dumtrainrace, dumtrainsex)
newtrain2[, 60] <- newtrain2$income
newtrain2 <- newtrain2[,-8]
names(newtrain2)[59]<- "income"
dim(newtrain2)
```

```
## [1] 32402    59
```

```
newtest2 <- cbind(newtest2, dumtestwork, dumtestedu, dumtestmarry, dumtestoccu,
                   dumtestrelation, dumtestrace, dumtestsex)
newtest2[, 60] <- newtest2$income
newtest2 <- newtest2[,-8]
names(newtest2)[59]<- "income"
dim(newtest2)
```

```
## [1] 16196    59
```

```
#fixing...
newtrain2$income <- droplevels(newtrain2$income, c("<=50K.", ">50K."))
```

```
newtest2$income <- droplevels(newtest2$income, c("<=50K", ">50K"))

newtest2$income <- as.character(newtest2$income)
newtest2$income <- substr(newtest2$income, 1, nchar(newtest2$income)-1)
newtest2$income <- as.factor(newtest2$income)

dim(newtrain2)
```

```
## [1] 32402    59
```

```
dim(newtest2)
```

```
## [1] 16196    59
```

```
str(newtrain2)
```

```
## 'data.frame':    32402 obs. of  59 variables:
##  $ age                   : Factor w/ 10 levels "[17.0,24.3)",..: 4 5 3 5 2 3 5 5 2 4 ...
##  $ fnlwgt                : Factor w/ 10 levels "[  12285, 160096)",..: 1 1 2 2 3 2 2 2 1 1 ...
##  $ education.num         : Factor w/ 10 levels "[ 1.0, 2.5)",..: 9 9 6 5 9 9 3 6 9 9 ...
##  $ capital.gain          : Factor w/ 7 levels "[    0, 5901)",..: 1 1 1 1 1 1 1 1 3 1 ...
##  $ capital.loss          : Factor w/ 7 levels "[   0, 622)",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ hours.per.week        : Factor w/ 10 levels "[ 1.0,10.8)",..: 4 2 4 4 4 4 2 5 6 4 ...
##  $ native.country        : Factor w/ 4 levels "Mexico","other_countries",..: 4 4 4 4 2 4 2 4 4 4 ...
##  $ Local-gov             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Never-worked          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Private               : num  0 0 1 1 1 1 1 0 1 1 ...
##  $ Self-emp-inc          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Self-emp-not-inc      : num  0 1 0 0 0 0 0 1 0 0 ...
##  $ State-gov             : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ Without-pay           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ 11th                  : num  0 0 0 1 0 0 0 0 0 0 ...
##  $ 12th                  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ 1st-4th               : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ 5th-6th               : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ 7th-8th               : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ 9th                   : num  0 0 0 0 0 0 1 0 0 0 ...
##  $ Assoc-acdm            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Assoc-voc             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Bachelors             : num  1 1 0 0 1 0 0 0 0 1 ...
##  $ Doctorate             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ HS-grad               : num  0 0 1 0 0 0 0 1 0 0 ...
##  $ Masters               : num  0 0 0 0 0 1 0 0 1 0 ...
##  $ Preschool             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Prof-school           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Some-college          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Married-AF-spouse     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Married-civ-spouse    : num  0 1 0 1 1 1 0 1 0 1 ...
##  $ Married-spouse-absent: num  0 0 0 0 0 0 1 0 0 0 ...
##  $ Never-married         : num  1 0 0 0 0 0 0 0 1 0 ...
##  $ Separated             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Widowed               : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Armed-Forces          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Craft-repair          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Exec-managerial       : num  0 1 0 0 0 1 0 1 0 1 ...
```

```
##  $ Farming-fishing      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Handlers-cleaners    : num  0 0 1 1 0 0 0 0 0 0 ...
##  $ Machine-op-inspct    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Other-service        : num  0 0 0 0 0 0 1 0 0 0 ...
##  $ Priv-house-serv      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Prof-specialty       : num  0 0 0 0 1 0 0 0 1 0 ...
##  $ Protective-serv      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Sales                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Tech-support         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Transport-moving     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Not-in-family        : num  1 0 1 0 0 0 1 0 1 0 ...
##  $ Other-relative       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Own-child            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Unmarried            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Wife                 : num  0 0 0 0 1 1 0 0 0 0 ...
##  $ Asian-Pac-Islander   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Black                : num  0 0 0 1 1 0 1 0 0 0 ...
##  $ Other                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ White                : num  1 1 1 0 0 1 0 1 1 1 ...
##  $ Male                 : num  1 1 1 1 0 0 0 1 0 1 ...
##  $ income               : Factor w/ 2 levels "<=50K",">50K": 1 1 1 1 1 1 1 2 2 2 ...
```

```
str(newtest2)
```

```
## 'data.frame':    16196 obs. of  59 variables:
##  $ age                  : Factor w/ 10 levels "[17.0,24.3)",..: 2 3 2 4 1 3 2 7 1 6 ...
##  $ fnlwgt               : Factor w/ 10 levels "[  12285, 160096)",..: 2 1 3 2 1 2 2 1 3 1 ...
##  $ education.num        : Factor w/ 10 levels "[ 1.0, 2.5)",..: 5 6 8 7 7 4 6 10 7 3 ...
##  $ capital.gain         : Factor w/ 7 levels "[    0, 5901)",..: 1 1 1 2 1 1 1 1 1 1 ...
##  $ capital.loss         : Factor w/ 7 levels "[   0, 622)",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ hours.per.week       : Factor w/ 10 levels "[ 1.0,10.8)",..: 4 6 4 4 3 3 4 4 4 1 ...
##  $ native.country       : Factor w/ 4 levels "Mexico","other_countries",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ Local-gov            : num  0 0 1 0 0 0 0 0 0 0 ...
##  $ Never-worked         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Private              : num  1 1 0 1 1 1 1 0 1 1 ...
##  $ Self-emp-inc         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Self-emp-not-inc     : num  0 0 0 0 0 0 0 1 0 0 ...
##  $ State-gov            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Without-pay          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ 11th                 : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ 12th                 : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ 1st-4th              : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ 5th-6th              : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ 7th-8th              : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ 9th                  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Assoc-acdm           : num  0 0 1 0 0 0 0 0 0 0 ...
##  $ Assoc-voc            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Bachelors            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Doctorate            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ HS-grad              : num  0 1 0 0 0 0 1 0 0 0 ...
##  $ Masters              : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Preschool            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Prof-school          : num  0 0 0 0 0 0 0 1 0 0 ...
##  $ Some-college         : num  0 0 0 1 1 0 0 0 1 0 ...
##  $ Married-AF-spouse    : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
##  $ Married-civ-spouse   : num  0 1 1 1 0 0 0 1 0 1 ...
##  $ Married-spouse-absent: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Never-married        : num  1 0 0 0 1 1 1 0 1 0 ...
##  $ Separated            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Widowed              : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Armed-Forces         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Craft-repair         : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ Exec-managerial      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Farming-fishing      : num  0 1 0 0 0 0 0 0 0 0 ...
##  $ Handlers-cleaners    : num  0 0 0 0 0 0 1 0 0 0 ...
##  $ Machine-op-inspct    : num  1 0 0 1 0 0 0 0 0 0 ...
##  $ Other-service        : num  0 0 0 0 0 1 0 0 1 0 ...
##  $ Priv-house-serv      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Prof-specialty       : num  0 0 0 0 0 0 0 1 0 0 ...
##  $ Protective-serv      : num  0 0 1 0 0 0 0 0 0 0 ...
##  $ Sales                : num  0 0 0 0 1 0 0 0 0 0 ...
##  $ Tech-support         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Transport-moving     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Not-in-family        : num  0 0 0 0 0 1 0 0 0 0 ...
##  $ Other-relative       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Own-child            : num  1 0 0 0 1 0 0 0 0 0 ...
##  $ Unmarried            : num  0 0 0 0 0 0 1 0 1 0 ...
##  $ Wife                 : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Asian-Pac-Islander   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Black                : num  1 0 0 1 0 0 1 0 0 0 ...
##  $ Other                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ White                : num  0 1 1 0 1 1 0 1 1 1 ...
##  $ Male                 : num  1 1 1 1 0 1 1 1 0 1 ...
##  $ income               : Factor w/ 2 levels "<=50K",">50K": 1 1 2 2 1 1 1 2 1 1 ...
```

```
newtrain2 <- read.csv("../data/cleandata/newtrain2.csv", header = T)
newtest2 <- read.csv("../data/cleandata/newtest2.csv", header = T)
str(newtrain2)
```

```
## 'data.frame':    32402 obs. of  59 variables:
##  $ age                  : Factor w/ 10 levels "[17.0,24.3)",..: 4 5 3 5 2 3 5 5 2 4 ...
##  $ fnlwgt               : Factor w/ 10 levels "[  12285, 160096)",..: 1 1 2 2 3 2 2 2 1 1 ...
##  $ education.num        : Factor w/ 10 levels "[ 1.0, 2.5)",..: 9 9 6 5 9 9 3 6 9 9 ...
##  $ capital.gain         : Factor w/ 7 levels "[    0, 5901)",..: 1 1 1 1 1 1 1 1 3 1 ...
##  $ capital.loss         : Factor w/ 7 levels "[   0, 622)",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ hours.per.week       : Factor w/ 10 levels "[ 1.0,10.8)",..: 4 2 4 4 4 4 2 5 6 4 ...
##  $ native.country       : Factor w/ 4 levels "Mexico","other_countries",..: 4 4 4 4 2 4 2 4 4 4 ...
##  $ Local.gov            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Never.worked         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Private              : int  0 0 1 1 1 1 1 0 1 1 ...
##  $ Self.emp.inc         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Self.emp.not.inc     : int  0 1 0 0 0 0 0 1 0 0 ...
##  $ State.gov            : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ Without.pay          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X11th                : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ X12th                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X1st.4th             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X5th.6th             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X7th.8th             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X9th                 : int  0 0 0 0 0 0 1 0 0 0 ...
```

```
##  $ Assoc.acdm           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Assoc.voc            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Bachelors            : int  1 1 0 0 1 0 0 0 0 1 ...
##  $ Doctorate            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ HS.grad              : int  0 0 1 0 0 0 0 1 0 0 ...
##  $ Masters              : int  0 0 0 0 0 1 0 0 1 0 ...
##  $ Preschool            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Prof.school          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Some.college         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Married.AF.spouse    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Married.civ.spouse   : int  0 1 0 1 1 1 0 1 0 1 ...
##  $ Married.spouse.absent: int  0 0 0 0 0 0 1 0 0 0 ...
##  $ Never.married        : int  1 0 0 0 0 0 0 0 1 0 ...
##  $ Separated            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Widowed              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Armed.Forces         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Craft.repair         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Exec.managerial      : int  0 1 0 0 0 1 0 1 0 1 ...
##  $ Farming.fishing      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Handlers.cleaners    : int  0 0 1 1 0 0 0 0 0 0 ...
##  $ Machine.op.inspct    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Other.service        : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ Priv.house.serv      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Prof.specialty       : int  0 0 0 0 1 0 0 0 1 0 ...
##  $ Protective.serv      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Sales                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Tech.support         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Transport.moving     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Not.in.family        : int  1 0 1 0 0 0 1 0 1 0 ...
##  $ Other.relative       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Own.child            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Unmarried            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Wife                 : int  0 0 0 0 1 1 0 0 0 0 ...
##  $ Asian.Pac.Islander   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Black                : int  0 0 0 1 1 0 1 0 0 0 ...
##  $ Other                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ White                : int  1 1 1 0 0 1 0 1 1 1 ...
##  $ Male                 : int  1 1 1 1 0 0 0 1 0 1 ...
##  $ income               : Factor w/ 2 levels "<=50K",">50K": 1 1 1 1 1 1 1 2 2 2 ...
```

```
str(newtest2)
```

```
## 'data.frame':    16196 obs. of  59 variables:
##  $ age                  : Factor w/ 10 levels "[17.0,24.3)",..: 2 3 2 4 1 3 2 7 1 6 ...
##  $ fnlwgt               : Factor w/ 10 levels "[  12285, 160096)",..: 2 1 3 2 1 2 2 1 3 1 ...
##  $ education.num        : Factor w/ 10 levels "[ 1.0, 2.5)",..: 5 6 8 7 7 4 6 10 7 3 ...
##  $ capital.gain         : Factor w/ 7 levels "[    0, 5901)",..: 1 1 1 2 1 1 1 1 1 1 ...
##  $ capital.loss         : Factor w/ 7 levels "[   0, 622)",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ hours.per.week       : Factor w/ 10 levels "[ 1.0,10.8)",..: 4 6 4 4 3 3 4 4 4 1 ...
##  $ native.country       : Factor w/ 4 levels "Mexico","other_countries",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ Local.gov            : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ Never.worked         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Private              : int  1 1 0 1 1 1 1 0 1 1 ...
##  $ Self.emp.inc         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Self.emp.not.inc     : int  0 0 0 0 0 0 0 1 0 0 ...
```

```
##  $ State.gov            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Without.pay          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X11th                : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ X12th                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X1st.4th             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X5th.6th             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X7th.8th             : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ X9th                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Assoc.acdm           : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ Assoc.voc            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Bachelors            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Doctorate            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ HS.grad              : int  0 1 0 0 0 0 1 0 0 0 ...
##  $ Masters              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Preschool            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Prof.school          : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Some.college         : int  0 0 0 1 1 0 0 0 1 0 ...
##  $ Married.AF.spouse    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Married.civ.spouse   : int  0 1 1 1 0 0 0 1 0 1 ...
##  $ Married.spouse.absent: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Never.married        : int  1 0 0 0 1 1 1 0 1 0 ...
##  $ Separated            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Widowed              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Armed.Forces         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Craft.repair         : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ Exec.managerial      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Farming.fishing      : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ Handlers.cleaners    : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ Machine.op.inspct    : int  1 0 0 1 0 0 0 0 0 0 ...
##  $ Other.service        : int  0 0 0 0 0 1 0 0 1 0 ...
##  $ Priv.house.serv      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Prof.specialty       : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Protective.serv      : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ Sales                : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ Tech.support         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Transport.moving     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Not.in.family        : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ Other.relative       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Own.child            : int  1 0 0 0 1 0 0 0 0 0 ...
##  $ Unmarried            : int  0 0 0 0 0 0 1 0 1 0 ...
##  $ Wife                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Asian.Pac.Islander   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Black                : int  1 0 0 1 0 0 1 0 0 0 ...
##  $ Other                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ White                : int  0 1 1 0 1 1 0 1 1 1 ...
##  $ Male                 : int  1 1 1 1 0 1 1 1 0 1 ...
##  $ income               : Factor w/ 2 levels "<=50K",">50K": 1 1 2 2 1 1 1 2 1 1 ...
```

```
#Check if train and test datasets have different factor level
for(i in 1:7){
  cat(names(newtest2)[i], "\n")
  print(levels(newtest2[,i]))
  cat("\n")
  print(levels(newtrain2[,i]))
```

```
  cat("\n")
}
```

```
## age
##  [1] "[17.0,24.3)" "[24.3,31.6)" "[31.6,38.9)" "[38.9,46.2)" "[46.2,53.5)"
##  [6] "[53.5,60.8)" "[60.8,68.1)" "[68.1,75.4)" "[75.4,82.7)" "[82.7,90.0]"
##
##  [1] "[17.0,24.3)" "[24.3,31.6)" "[31.6,38.9)" "[38.9,46.2)" "[46.2,53.5)"
##  [6] "[53.5,60.8)" "[60.8,68.1)" "[68.1,75.4)" "[75.4,82.7)" "[82.7,90.0]"
##
## fnlwgt
##  [1] "[  12285, 160096)" "[ 160096, 307908)" "[ 307908, 455720)"
##  [4] "[ 455720, 603531)" "[ 603531, 751342)" "[ 751342, 899154)"
##  [7] "[ 899154,1046966)" "[1046966,1194777)" "[1194777,1342588)"
## [10] "[1342588,1490400]"
##
##  [1] "[  12285, 160096)" "[ 160096, 307908)" "[ 307908, 455720)"
##  [4] "[ 455720, 603531)" "[ 603531, 751342)" "[ 751342, 899154)"
##  [7] "[ 899154,1046966)" "[1046966,1194777)" "[1194777,1342588)"
## [10] "[1342588,1490400]"
##
## education.num
##  [1] "[ 1.0, 2.5)" "[ 2.5, 4.0)" "[ 4.0, 5.5)" "[ 5.5, 7.0)" "[ 7.0, 8.5)"
##  [6] "[ 8.5,10.0)" "[10.0,11.5)" "[11.5,13.0)" "[13.0,14.5)" "[14.5,16.0]"
##
##  [1] "[ 1.0, 2.5)" "[ 2.5, 4.0)" "[ 4.0, 5.5)" "[ 5.5, 7.0)" "[ 7.0, 8.5)"
##  [6] "[ 8.5,10.0)" "[10.0,11.5)" "[11.5,13.0)" "[13.0,14.5)" "[14.5,16.0]"
##
## capital.gain
## [1] "[    0, 5901)" "[ 5901,11803)" "[11803,17704)" "[17704,23606)"
## [5] "[23606,29507)" "[29507,35409)" "[35409,41310]"
##
## [1] "[    0, 5901)" "[ 5901,11803)" "[11803,17704)" "[17704,23606)"
## [5] "[23606,29507)" "[29507,35409)" "[35409,41310]"
##
## capital.loss
## [1] "[   0, 622)" "[ 622,1245)" "[1245,1867)" "[1867,2489)" "[2489,3111)"
## [6] "[3111,3734)" "[3734,4356]"
##
## [1] "[   0, 622)" "[ 622,1245)" "[1245,1867)" "[1867,2489)" "[2489,3111)"
## [6] "[3111,3734)" "[3734,4356]"
##
## hours.per.week
##  [1] "[ 1.0,10.8)" "[10.8,20.6)" "[20.6,30.4)" "[30.4,40.2)" "[40.2,50.0)"
##  [6] "[50.0,59.8)" "[59.8,69.6)" "[69.6,79.4)" "[79.4,89.2)" "[89.2,99.0]"
##
##  [1] "[ 1.0,10.8)" "[10.8,20.6)" "[20.6,30.4)" "[30.4,40.2)" "[40.2,50.0)"
##  [6] "[50.0,59.8)" "[59.8,69.6)" "[69.6,79.4)" "[79.4,89.2)" "[89.2,99.0]"
##
## native.country
## [1] "Mexico"          "other_countries" "Philippines"     "United-States"
##
## [1] "Mexico"          "other_countries" "Philippines"     "United-States"
```

```r
#Remove white space in factor variables to visualize factor correctly in the future plots
newtrain3 <- newtrain2
newtest3 <- newtest2
for(i in 1:7){
  newtrain3[,i] <- as.factor(gsub(" ", "", newtrain2[,i], fixed = TRUE))
  newtest3[,i] <- as.factor(gsub(" ", "", newtest2[,i], fixed = TRUE))
}
```

# Classification Tree

## Normal way

```r
# Fit the tree
tree1 <- tree(income ~., newtrain3)

# brief summary of tree1 object
tree1
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 32402 35490.00 <=50K ( 0.762916 0.237084 )
##    2) Married.civ.spouse < 0.5 17558  8343.00 <=50K ( 0.936098 0.063902 )
##      4) capital.gain: [0,5901),[29507,35409) 17235  6762.00 <=50K ( 0.950798 0.049202 )
##        8) education.num: [1.0,2.5),[10.0,11.5),[11.5,13.0),[2.5,4.0),[4.0,5.5),[5.5,7.0),[7.0,8.5),[8
##        9) education.num: [13.0,14.5),[14.5,16.0] 3392  2861.00 <=50K ( 0.850531 0.149469 ) *
##      5) capital.gain: [11803,17704),[17704,23606),[23606,29507),[5901,11803) 323   275.00 >50K ( 0.15
##    3) Married.civ.spouse > 0.5 14844 20380.00 <=50K ( 0.558071 0.441929 )
##      6) education.num: [1.0,2.5),[10.0,11.5),[11.5,13.0),[2.5,4.0),[4.0,5.5),[5.5,7.0),[7.0,8.5),[8.5
##       12) capital.gain: [0,5901),[35409,41310] 10049 12300.00 <=50K ( 0.698477 0.301523 )
##         24) education.num: [1.0,2.5),[2.5,4.0),[4.0,5.5),[5.5,7.0),[7.0,8.5) 1665  1119.00 <=50K ( 0
##         25) education.num: [10.0,11.5),[11.5,13.0),[8.5,10.0) 8384 10750.00 <=50K ( 0.659470 0.340530
##       13) capital.gain: [11803,17704),[17704,23606),[5901,11803) 426    94.80 >50K ( 0.023474 0.976526
##      7) education.num: [13.0,14.5),[14.5,16.0] 4369  5240.00 >50K ( 0.287251 0.712749 )
##       14) capital.gain: [0,5901),[35409,41310] 3822  4836.00 >50K ( 0.327839 0.672161 ) *
##       15) capital.gain: [11803,17704),[17704,23606),[5901,11803) 547    26.44 >50K ( 0.003656 0.996344
```

```r
# summary of tree1
tree1.summary <- summary(tree1)
tree1.summary
```

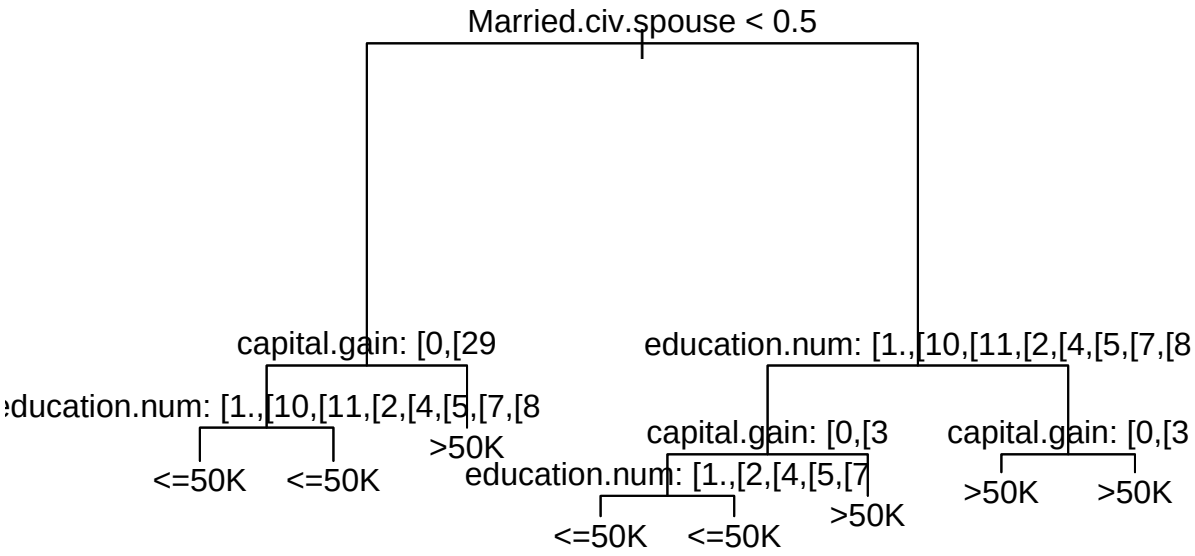
```
##
```

```
## Classification tree:
## tree(formula = income ~ ., data = newtrain3)
## Variables actually used in tree construction:
## [1] "Married.civ.spouse" "capital.gain"       "education.num"
## Number of terminal nodes:  8
## Residual mean deviance:  0.7152 = 23170 / 32390
## Misclassification error rate: 0.1602 = 5192 / 32402
```

```
# training accuracy rate
1 - (tree1.summary$misclass[1] / tree1.summary$misclass[2])
```

```
## [1] 0.839763
```

```
# Make plot of the tree
plot(tree1)
text(tree1, pretty= T)
```




```
set.seed (100)

income <- newtest3$income

treepred <- predict (tree1, newtest3, type = "class")
table <- table(treepred ,income)
table
```

```
##          income
## treepred <=50K  >50K
##    <=50K 11783  1927
##    >50K    652  1834
```

```
# Misclassification Rate for test dataset
( table[1, 2] + table[2, 1] ) / sum(table)
```

```
## [1] 0.1592368
```

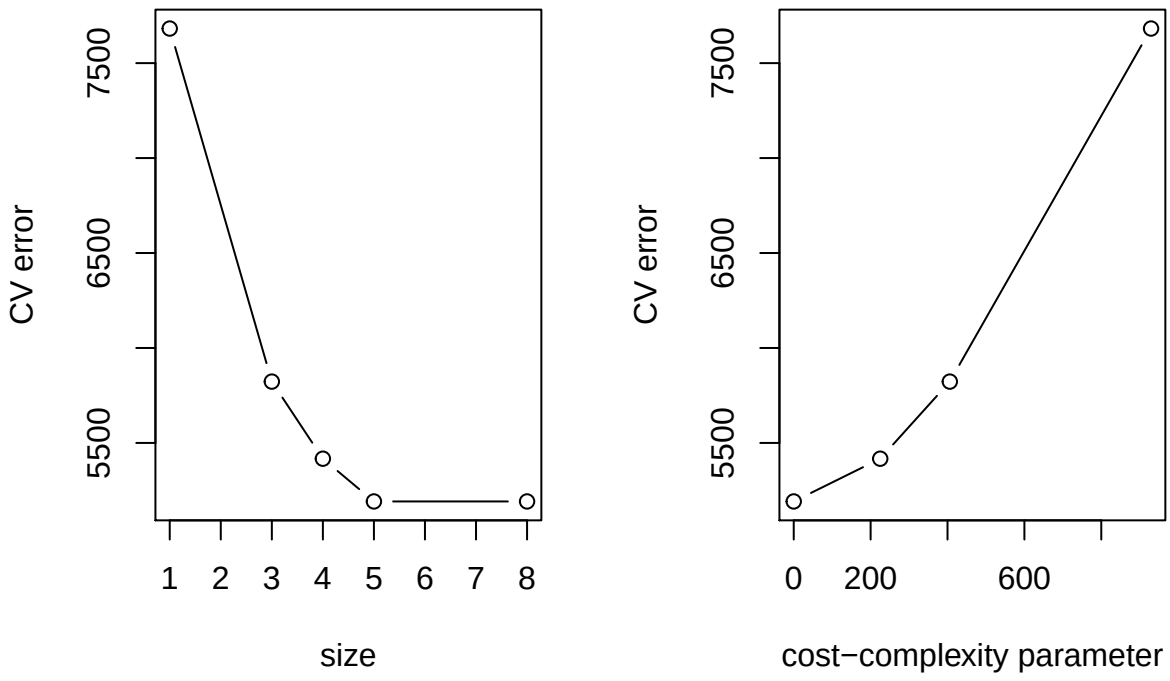
```
# Accuracy Rate for test dataset
( table[1, 1] + table[2, 2] ) / sum(table)
```

```
## [1] 0.8407632
```

```r
set.seed(100)
cv.tree1 <- cv.tree(tree1, FUN=prune.misclass)
cv.tree1
```

```
## $size
## [1] 8 5 4 3 1
##
## $dev
## [1] 5192 5192 5417 5823 7682
##
## $k
## [1]  -Inf   0.0 225.0 406.0 929.5
##
## $method
## [1] "misclass"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```r
# Plot the error rate as a function of both size and cost complexity parameter.

par(mfrow = c(1, 2))
plot(cv.tree1$size, cv.tree1$dev, type="b", xlab = "size", ylab = "CV error")
plot(cv.tree1$k, cv.tree1$dev, type="b",  xlab = "cost-complexity parameter", ylab = "CV error")
```




```r
# Get the best number of node of tree
best <- cv.tree1$size[which(cv.tree1$dev == min(cv.tree1$dev))[1]]
best
```

```
## [1] 8
```

```r
# prune the tree
prune.tree1 <- prune.misclass(tree1, best = best)
plot(prune.tree1)
```

```
text(prune.tree1, pretty =T)


# Performance of pruned tree on the test dataset
treepred <- predict(prune.tree1 , newtest3 ,type = "class")
table <- table(treepred, income)
table
```

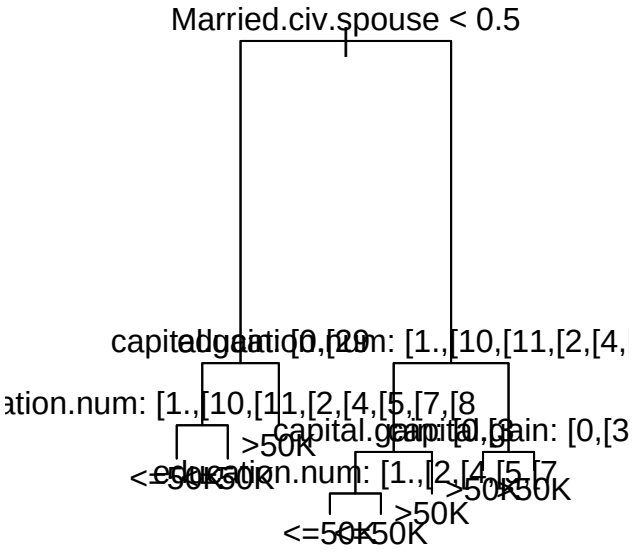
```
##          income
## treepred <=50K  >50K
##    <=50K 11783  1927
##    >50K    652  1834
```

```
# Misclassification Rate of prunned tree
( table[1, 2] + table[2, 1] ) / sum(table)
```

```
## [1] 0.1592368
```

```
# Accuracy Rate of prunned tree
( table[1, 1] + table[2, 2] ) / sum(table)
```

```
## [1] 0.8407632
```




### other way to tune

```
trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)
set.seed(100)


# Training the Decision Tree classifier with criterion as information (cross Entropy)
dtree_fit <- caret::train(income ~., data = newtrain3,
                          method = "rpart",
                    parms = list(split = "information"),
                    trControl = trctrl,
                    tuneLength = 10)
```

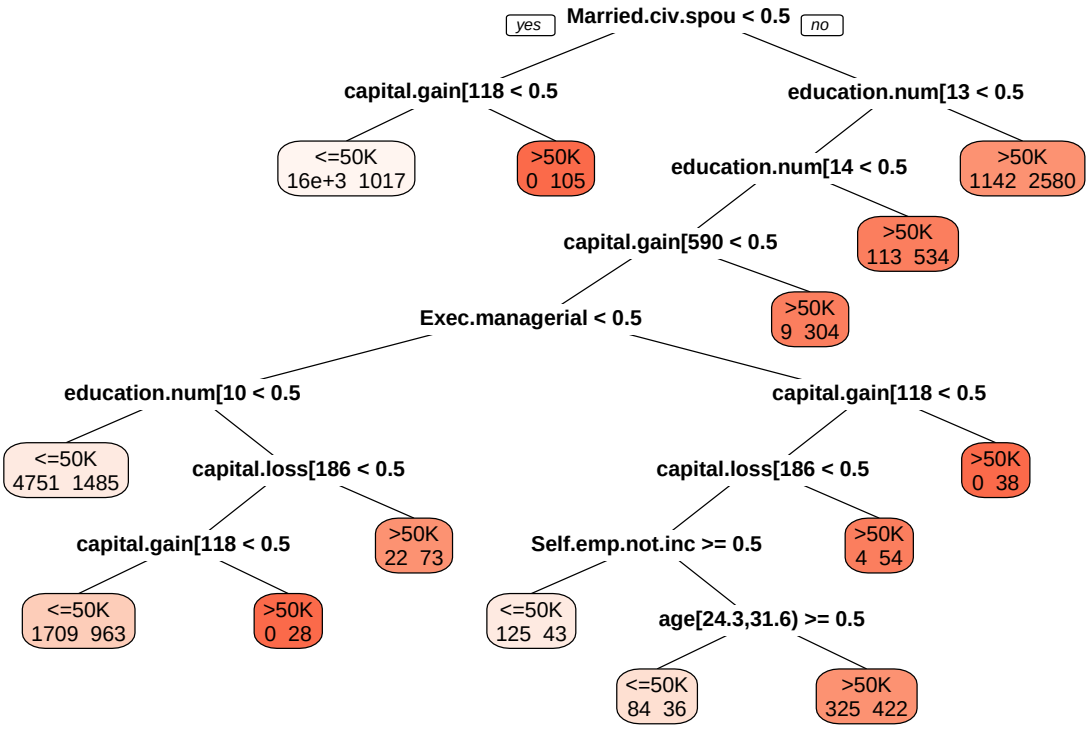
```
dtree_fit
```

```
## CART
##
## 32402 samples
##    58 predictor
##     2 classes: '<=50K', '>50K'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 29162, 29161, 29162, 29162, 29162, 29162, ...
## Resampling results across tuning parameters:
##
##   cp           Accuracy   Kappa
##   0.002928925  0.8404109  0.5178431
##   0.003124186  0.8398452  0.5156259
##   0.005272065  0.8349484  0.5019023
##   0.006378547  0.8333332  0.4915460
##   0.008982036  0.8325102  0.4758966
##   0.009177298  0.8325102  0.4758966
##   0.013668316  0.8304938  0.4594283
##   0.038401458  0.8245066  0.4343096
##   0.054803437  0.8121516  0.3771283
##   0.093595418  0.7819376  0.1583614
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was cp = 0.002928925.
```

```
# Tuning parameter - cp
dtree_fit$bestTune
```

```
##           cp
## 1 0.002928925
```

```
# plot classification tree  - part of the factor names are missing ******
prp(dtree_fit$finalModel, box.palette = "Reds", tweak = 0.8,
    fallen.leaves = FALSE, faclen = 0, extra = 1)
```



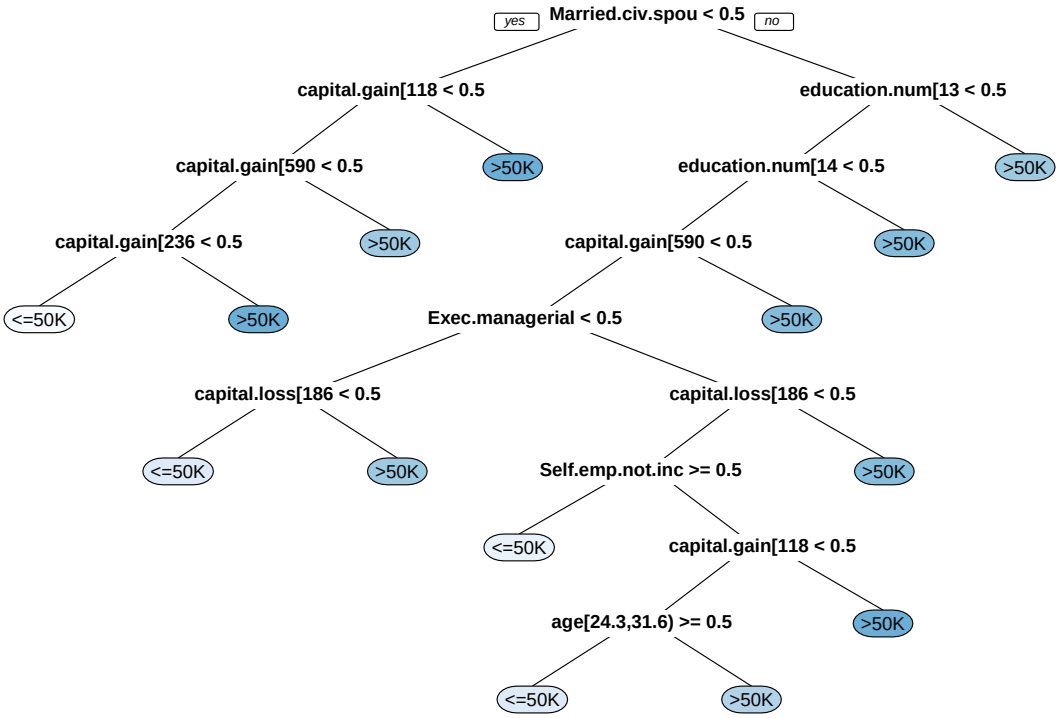

```
# prediction
testpred <- predict(dtree_fit, newdata = newtest3)
confusionMatrix(testpred, newtest3$income)  #check accuracy
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction <=50K  >50K
##      <=50K 11632  1760
##      >50K    803  2001
##
##                Accuracy : 0.8418
##                  95% CI : (0.836, 0.8473)
##     No Information Rate : 0.7678
##     P-Value [Acc > NIR] : < 0.00000000000000022
##
##                   Kappa : 0.513
##  Mcnemar's Test P-Value : < 0.00000000000000022
##
##             Sensitivity : 0.9354
##             Specificity : 0.5320
##          Pos Pred Value : 0.8686
##          Neg Pred Value : 0.7136
##              Prevalence : 0.7678
##          Detection Rate : 0.7182
##    Detection Prevalence : 0.8269
##       Balanced Accuracy : 0.7337
##
##        'Positive' Class : <=50K
##
```

```r
#Training the Decision Tree classifier with criterion as gini index
set.seed(100)
dtree_fit_gini <- caret::train(income ~., data = newtrain3, method = "rpart",
                   parms = list(split = "gini"),
                   trControl=trctrl,
                   tuneLength = 10)
dtree_fit_gini
```

```
## CART
##
## 32402 samples
##    58 predictor
##     2 classes: '<=50K', '>50K'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 29162, 29161, 29162, 29162, 29162, 29162, ...
## Resampling results across tuning parameters:
##
##   cp           Accuracy   Kappa
##   0.002928925  0.8441865  0.5373763
##   0.003124186  0.8437133  0.5355654
##   0.005272065  0.8395159  0.5215652
##   0.006378547  0.8378081  0.5148709
##   0.008982036  0.8343104  0.4935476
##   0.009177298  0.8338372  0.4905254
##   0.013668316  0.8306687  0.4605568
##   0.038401458  0.8245066  0.4343096
##   0.054803437  0.8125837  0.3791621
##   0.093595418  0.7819376  0.1583614
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was cp = 0.002928925.
```

```r
#Plot decision tree from gini index criterion
prp(dtree_fit_gini$finalModel, box.palette = "Blues", tweak = 0.6)
```




```
#Tuning parameter - cp
dtree_fit_gini$bestTune
```

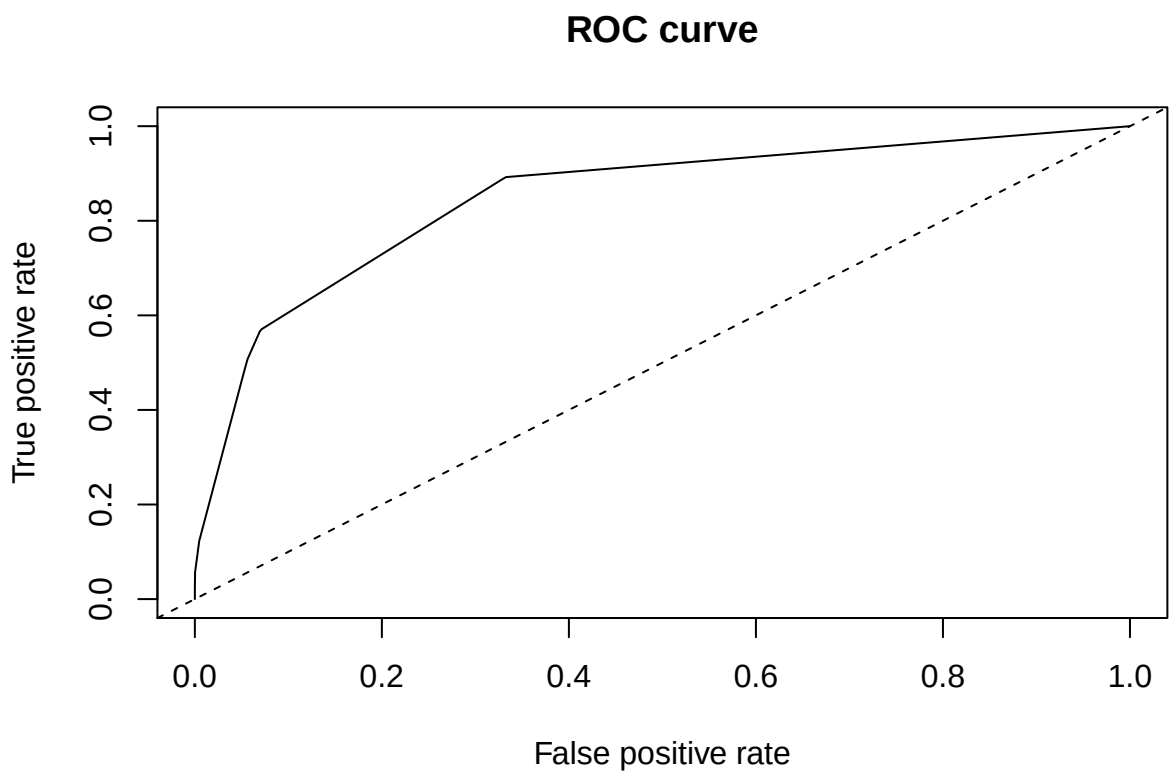
```
##           cp
## 1 0.002928925
```

```
#Accuracy and confusion matrix from Gini index criterion
test_pred_gini <- predict(dtree_fit_gini, newdata = newtest3)
confusionMatrix(test_pred_gini, newtest3$income )  #check accuracy
```

```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction <=50K  >50K
##      <=50K 11572  1633
##      >50K    863  2128
##
##                Accuracy : 0.8459
##                  95% CI : (0.8402, 0.8514)
##     No Information Rate : 0.7678
##     P-Value [Acc > NIR] : < 0.00000000000000022
##
##                   Kappa : 0.5346
##  Mcnemar's Test P-Value : < 0.00000000000000022
##
##             Sensitivity : 0.9306
##             Specificity : 0.5658
##          Pos Pred Value : 0.8763
##          Neg Pred Value : 0.7115
##              Prevalence : 0.7678
##          Detection Rate : 0.7145
##    Detection Prevalence : 0.8153
```

```
##       Balanced Accuracy : 0.7482
##
##        'Positive' Class : <=50K
##
```

```
#ROC Curve  : https://stackoverflow.com/questions/30818188/roc-curve-in-r-using-rpart-package


#Getting predicted >50K of income probabilities
gini_prob <- predict(dtree_fit_gini, newdata = newtest3, type = "prob")[,2]
gini_prediction <- prediction(gini_prob, newtest3$income)
gini_performance <- performance(gini_prediction, measure = "tpr", x.measure = "fpr")

#Plot ROC curve
plot(gini_performance, main="ROC curve")
abline(a=0, b=1, lty=2)
```




```
#Calculate AUC
performance(gini_prediction, measure="auc")@y.values[[1]]
```

```
## [1] 0.8474155
```

```
#Pick the best threshold
str(gini_performance)
```

```
## Formal class 'performance' [package "ROCR"] with 6 slots
##   ..@ x.name      : chr "False positive rate"
##   ..@ y.name      : chr "True positive rate"
##   ..@ alpha.name  : chr "Cutoff"
##   ..@ x.values    :List of 1
##   .. ..$ : num [1:13] 0 0 0.000241 0.000643 0.004664 ...
```

```
##   ..@ y.values    :List of 1
##   .. ..$ : num [1:13] 0 0.0207 0.0569 0.0625 0.1226 ...
##   ..@ alpha.values:List of 1
##   .. ..$ : num [1:13] Inf 1 0.971 0.931 0.825 ...
```

```
cutoffs <- data.frame(cut = gini_performance@alpha.values[[1]],
                      fpr = gini_performance@x.values[[1]],
                      tpr = gini_performance@y.values[[1]])
head(cutoffs)
```

```
##         cut          fpr        tpr
## 1       Inf 0.0000000000 0.00000000
## 2 1.0000000 0.0000000000 0.02073917
## 3 0.9712460 0.0002412545 0.05689976
## 4 0.9310345 0.0006433454 0.06248338
## 5 0.8253478 0.0046642541 0.12257378
## 6 0.7065868 0.0067551267 0.13852699
```

```
roc <- pROC::roc(newtest3$income, gini_prob)
threshold <- coords(roc, "best", ret = "threshold")
cat("The best threshold is :  " , threshold, "\n")
```

```
## The best threshold is :   0.1548338
```

```
#Get accuracy rate of testset data using the optimal threshold  ****
confusionMatrix(table(gini_prob > threshold, newtest3$income == ">50K"))
```

```
## Confusion Matrix and Statistics
##
##
##         FALSE TRUE
##   FALSE  8298  405
##   TRUE   4137 3356
##
##                Accuracy : 0.7196
##                  95% CI : (0.7126, 0.7265)
##     No Information Rate : 0.7678
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.4157
##  Mcnemar's Test P-Value : <0.0000000000000002
##
##             Sensitivity : 0.6673
##             Specificity : 0.8923
##          Pos Pred Value : 0.9535
##          Neg Pred Value : 0.4479
##              Prevalence : 0.7678
##          Detection Rate : 0.5123
##    Detection Prevalence : 0.5374
##       Balanced Accuracy : 0.7798
##
##        'Positive' Class : FALSE
##
```

```
confusionMatrix(gini_prob > threshold, newtest3$income == ">50K")
```

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

```
##
##           Reference
## Prediction FALSE TRUE
##      FALSE  8298  405
##      TRUE   4137 3356
##
##                Accuracy : 0.7196
##                  95% CI : (0.7126, 0.7265)
##     No Information Rate : 0.7678
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.4157
##  Mcnemar's Test P-Value : <0.0000000000000002
##
##             Sensitivity : 0.6673
##             Specificity : 0.8923
##          Pos Pred Value : 0.9535
##          Neg Pred Value : 0.4479
##              Prevalence : 0.7678
##          Detection Rate : 0.5123
##    Detection Prevalence : 0.5374
##       Balanced Accuracy : 0.7798
##
##        'Positive' Class : FALSE
##
```

### Using R part

https://stackoverflow.com/questions/46042966/set-threshold-for-the-probability-result-from-decision-tree

```
set.seed(100)

# Classification tree using cross entropy criterion
tree <- rpart(income ~., data = newtrain3,
              control = rpart.control(cp = 0.004), method = "class",
              parms = list(split = 'information') )
  # minsplit = 2, minbucket = 1


#Pick the optimal tuning parameter
cp <- tree$cptable[which.min(tree$cptable[,"xerror"]),"CP"]


#Prune the tree using the optimal cp
treepruned <- prune(tree, cp = cp)

#Treepruned object
treepruned
```

```
## n= 32402
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
```

```
##   1) root 32402 7682 <=50K (0.76291587 0.23708413)
##     2) Married.civ.spouse< 0.5 17558 1122 <=50K (0.93609751 0.06390249)
##       4) capital.gain=[0,5901),[29507,35409) 17235  848 <=50K (0.95079780 0.04920220) *
##       5) capital.gain=[11803,17704),[17704,23606),[23606,29507),[5901,11803) 323   49 >50K (0.151702
##     3) Married.civ.spouse>=0.5 14844 6560 <=50K (0.55807060 0.44192940)
##       6) education.num=[1.0,2.5),[10.0,11.5),[11.5,13.0),[2.5,4.0),[4.0,5.5),[5.5,7.0),[7.0,8.5),[8.5
##        12) capital.gain=[0,5901),[35409,41310] 10049 3030 <=50K (0.69847746 0.30152254)
##          24) education.num=[1.0,2.5),[2.5,4.0),[4.0,5.5),[5.5,7.0),[7.0,8.5) 1665  175 <=50K (0.8948
##          25) education.num=[10.0,11.5),[11.5,13.0),[8.5,10.0) 8384 2855 <=50K (0.65947042 0.34052958
##            50) age=[17.0,24.3),[24.3,31.6),[60.8,68.1),[68.1,75.4),[75.4,82.7),[82.7,90.0] 2393  483
##            51) age=[31.6,38.9),[38.9,46.2),[46.2,53.5),[53.5,60.8) 5991 2372 <=50K (0.60407278 0.395
##             102) capital.loss=[0,622),[1245,1867),[2489,3111) 5756 2186 <=50K (0.62022238 0.37977762
##               204) education.num=[8.5,10.0) 3169 1019 <=50K (0.67844746 0.32155254) *
##               205) education.num=[10.0,11.5),[11.5,13.0) 2587 1167 <=50K (0.54889834 0.45110166)
##                 410) Exec.managerial< 0.5 2160  904 <=50K (0.58148148 0.41851852) *
##                 411) Exec.managerial>=0.5 427  164 >50K (0.38407494 0.61592506) *
##             103) capital.loss=[1867,2489) 235   49 >50K (0.20851064 0.79148936) *
##        13) capital.gain=[11803,17704),[17704,23606),[5901,11803) 426   10 >50K (0.02347418 0.9765258
##       7) education.num=[13.0,14.5),[14.5,16.0] 4369 1255 >50K (0.28725109 0.71274891) *
```

```
printcp(treepruned)
```

```
##
## Classification tree:
## rpart(formula = income ~ ., data = newtrain3, method = "class",
##     parms = list(split = "information"), control = rpart.control(cp = 0.004))
##
## Variables actually used in tree construction:
## [1] age                capital.gain       capital.loss
## [4] education.num      Exec.managerial    Married.civ.spouse
##
## Root node error: 7682/32402 = 0.23708
##
## n= 32402
##
##          CP nsplit rel error  xerror      xstd
## 1 0.1209971      0   1.00000 1.00000 0.0099655
## 2 0.0528508      2   0.75801 0.75801 0.0089967
## 3 0.0292892      3   0.70515 0.70515 0.0087434
## 4 0.0059446      4   0.67587 0.67587 0.0085955
## 5 0.0040000      9   0.64514 0.64619 0.0084398
```

```
#summary information
summary(treepruned, digits = 3)
```

```
## Call:
## rpart(formula = income ~ ., data = newtrain3, method = "class",
##     parms = list(split = "information"), control = rpart.control(cp = 0.004))
##   n= 32402
##
##        CP nsplit rel error xerror    xstd
## 1 0.12100      0     1.000  1.000 0.00997
## 2 0.05285      2     0.758  0.758 0.00900
## 3 0.02929      3     0.705  0.705 0.00874
## 4 0.00594      4     0.676  0.676 0.00860
```

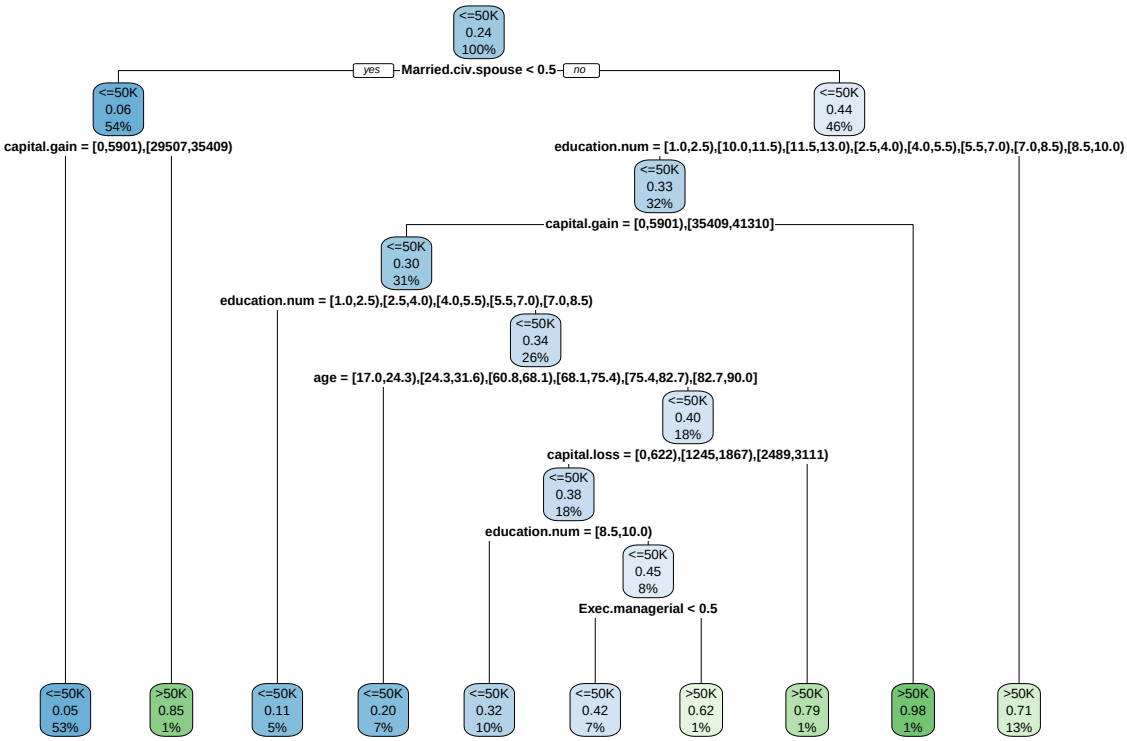
```
## 5 0.00400      9     0.645  0.646 0.00844
##
## Variable importance
## Married.civ.spouse      Never.married      Not.in.family
##                 26                 14                 10
##      education.num       capital.gain               Male
##                  9                  8                  8
##                age          Bachelors          Own.child
##                  7                  5                  4
##     Prof.specialty            Masters       capital.loss
##                  2                  2                  1
##        Prof.school
##                  1
##
## Node number 1: 32402 observations,    complexity param=0.121
##   predicted class=<=50K  expected loss=0.237  P(node) =1
##     class counts: 24720  7682
##    probabilities: 0.763 0.237
##   left son=2 (17558 obs) right son=3 (14844 obs)
##   Primary splits:
##       Married.civ.spouse < 0.5 to the left,     improve=3390, (0 missing)
##       Never.married      < 0.5 to the right,    improve=1990, (0 missing)
##       capital.gain       splits as  LRRRLLR,    improve=1630, (0 missing)
##       education.num      splits as  LLLRRLLLLL, improve=1530, (0 missing)
##       age                splits as  LLRRRRRLLL, improve=1490, (0 missing)
##   Surrogate splits:
##       Never.married < 0.5 to the right,    agree=0.787, adj=0.536, (0 split)
##       Not.in.family < 0.5 to the right,    agree=0.713, adj=0.373, (0 split)
##       Male          < 0.5 to the left,     agree=0.688, adj=0.320, (0 split)
##       age           splits as  LLRRRRRRRL, agree=0.648, adj=0.231, (0 split)
##       Own.child     < 0.5 to the right,    agree=0.609, adj=0.146, (0 split)
##
## Node number 2: 17558 observations,    complexity param=0.0293
##   predicted class=<=50K  expected loss=0.0639  P(node) =0.542
##     class counts: 16436  1122
##    probabilities: 0.936 0.064
##   left son=4 (17235 obs) right son=5 (323 obs)
##   Primary splits:
##       capital.gain   splits as  LRRRL-R,    improve=653, (0 missing)
##       education.num  splits as  LLLRRLLLLL, improve=465, (0 missing)
##       hours.per.week splits as  LLLLRRRRRR, improve=351, (0 missing)
##       age            splits as  LLRRRRRRRL, improve=308, (0 missing)
##       Own.child      < 0.5 to the right,    improve=231, (0 missing)
##
## Node number 3: 14844 observations,    complexity param=0.121
##   predicted class=<=50K  expected loss=0.442  P(node) =0.458
##     class counts:  8284  6560
##    probabilities: 0.558 0.442
##   left son=6 (10475 obs) right son=7 (4369 obs)
##   Primary splits:
##       education.num   splits as  LLLRRLLLLL, improve=933, (0 missing)
##       capital.gain    splits as  LRR--LR,    improve=769, (0 missing)
##       Bachelors       < 0.5 to the left,     improve=342, (0 missing)
##       Exec.managerial < 0.5 to the left,     improve=336, (0 missing)
```

```
##       Prof.specialty  < 0.5 to the left,     improve=324, (0 missing)
##   Surrogate splits:
##       Bachelors      < 0.5 to the left,  agree=0.890, adj=0.626, (0 split)
##       Prof.specialty < 0.5 to the left,  agree=0.791, adj=0.289, (0 split)
##       Masters        < 0.5 to the left,  agree=0.772, adj=0.226, (0 split)
##       Prof.school    < 0.5 to the left,  agree=0.731, adj=0.085, (0 split)
##       Doctorate      < 0.5 to the left,  agree=0.724, adj=0.063, (0 split)
##
## Node number 4: 17235 observations
##   predicted class=<=50K  expected loss=0.0492  P(node) =0.532
##     class counts: 16387   848
##    probabilities: 0.951 0.049
##
## Node number 5: 323 observations
##   predicted class=>50K   expected loss=0.152  P(node) =0.00997
##     class counts:    49   274
##    probabilities: 0.152 0.848
##
## Node number 6: 10475 observations,    complexity param=0.0529
##   predicted class=<=50K  expected loss=0.329  P(node) =0.323
##     class counts:  7029  3446
##    probabilities: 0.671 0.329
##   left son=12 (10049 obs) right son=13 (426 obs)
##   Primary splits:
##       capital.gain    splits as  LRR--LR,    improve=436, (0 missing)
##       education.num   splits as  LRR--LLLLR, improve=232, (0 missing)
##       age             splits as  LLRRRRLLLL, improve=187, (0 missing)
##       Exec.managerial < 0.5 to the left,     improve=134, (0 missing)
##       capital.loss    splits as  LLRL---,    improve=101, (0 missing)
##
## Node number 7: 4369 observations
##   predicted class=>50K   expected loss=0.287  P(node) =0.135
##     class counts:  1255  3114
##    probabilities: 0.287 0.713
##
## Node number 12: 10049 observations,    complexity param=0.00594
##   predicted class=<=50K  expected loss=0.302  P(node) =0.31
##     class counts:  7019  3030
##    probabilities: 0.698 0.302
##   left son=24 (1665 obs) right son=25 (8384 obs)
##   Primary splits:
##       education.num   splits as  LRR--LLLLR, improve=214.0, (0 missing)
##       age             splits as  LLRRRRLLLL, improve=177.0, (0 missing)
##       capital.loss    splits as  LLRL---,    improve=119.0, (0 missing)
##       Exec.managerial < 0.5 to the left,     improve=110.0, (0 missing)
##       hours.per.week  splits as  LLLRRRRRRR, improve= 77.8, (0 missing)
##   Surrogate splits:
##       X7th.8th < 0.5 to the right, agree=0.869, adj=0.212, (0 split)
##       X11th    < 0.5 to the right, agree=0.869, adj=0.208, (0 split)
##       X9th     < 0.5 to the right, agree=0.857, adj=0.138, (0 split)
##       X5th.6th < 0.5 to the right, agree=0.851, adj=0.102, (0 split)
##       X12th    < 0.5 to the right, agree=0.847, adj=0.074, (0 split)
##
## Node number 13: 426 observations
```

```
##   predicted class=>50K   expected loss=0.0235  P(node) =0.0131
##     class counts:    10   416
##    probabilities: 0.023 0.977
##
## Node number 24: 1665 observations
##   predicted class=<=50K  expected loss=0.105  P(node) =0.0514
##     class counts:  1490   175
##    probabilities: 0.895 0.105
##
## Node number 25: 8384 observations,    complexity param=0.00594
##   predicted class=<=50K  expected loss=0.341  P(node) =0.259
##     class counts:  5529  2855
##    probabilities: 0.659 0.341
##   left son=50 (2393 obs) right son=51 (5991 obs)
##   Primary splits:
##       age             splits as  LLRRRRLLLL, improve=152.0, (0 missing)
##       capital.loss    splits as  LLRL---,    improve=101.0, (0 missing)
##       Exec.managerial < 0.5 to the left,     improve= 79.8, (0 missing)
##       hours.per.week  splits as  LLLRRRRRRR, improve= 69.8, (0 missing)
##       education.num   splits as  -RR------L, improve= 68.5, (0 missing)
##   Surrogate splits:
##       hours.per.week splits as  LLRRRRRRRR, agree=0.721, adj=0.021, (0 split)
##       Own.child      < 0.5 to the right,    agree=0.718, adj=0.011, (0 split)
##       Not.in.family  < 0.5 to the right,    agree=0.715, adj=0.003, (0 split)
##       Without.pay    < 0.5 to the right,    agree=0.715, adj=0.002, (0 split)
##       fnlwgt         splits as  L-R-RRRRRR, agree=0.715, adj=0.000, (0 split)
##
## Node number 50: 2393 observations
##   predicted class=<=50K  expected loss=0.202  P(node) =0.0739
##     class counts:  1910   483
##    probabilities: 0.798 0.202
##
## Node number 51: 5991 observations,    complexity param=0.00594
##   predicted class=<=50K  expected loss=0.396  P(node) =0.185
##     class counts:  3619  2372
##    probabilities: 0.604 0.396
##   left son=102 (5756 obs) right son=103 (235 obs)
##   Primary splits:
##       capital.loss    splits as  LLRL---,    improve=79.9, (0 missing)
##       education.num   splits as  -RR------L, improve=54.7, (0 missing)
##       HS.grad         < 0.5 to the right,    improve=54.7, (0 missing)
##       Exec.managerial < 0.5 to the left,     improve=52.9, (0 missing)
##       Other.service   < 0.5 to the right,    improve=42.4, (0 missing)
##
## Node number 102: 5756 observations,    complexity param=0.00594
##   predicted class=<=50K  expected loss=0.38  P(node) =0.178
##     class counts:  3570  2186
##    probabilities: 0.620 0.380
##   left son=204 (3169 obs) right son=205 (2587 obs)
##   Primary splits:
##       education.num    splits as  -RR------L, improve=50.7, (0 missing)
##       HS.grad          < 0.5 to the right,    improve=50.7, (0 missing)
##       Exec.managerial  < 0.5 to the left,     improve=46.9, (0 missing)
##       Other.service    < 0.5 to the right,    improve=39.6, (0 missing)
```

```
##       Self.emp.not.inc < 0.5 to the right,    improve=35.3, (0 missing)
##   Surrogate splits:
##       HS.grad        < 0.5 to the right, agree=1.000, adj=1.000, (0 split)
##       Some.college   < 0.5 to the left,  agree=0.868, adj=0.706, (0 split)
##       Assoc.voc      < 0.5 to the left,  agree=0.629, adj=0.175, (0 split)
##       Assoc.acdm     < 0.5 to the left,  agree=0.604, adj=0.119, (0 split)
##       Prof.specialty < 0.5 to the left,  agree=0.574, adj=0.053, (0 split)
##
## Node number 103: 235 observations
##   predicted class=>50K   expected loss=0.209  P(node) =0.00725
##     class counts:    49   186
##    probabilities: 0.209 0.791
##
## Node number 204: 3169 observations
##   predicted class=<=50K  expected loss=0.322  P(node) =0.0978
##     class counts:  2150  1019
##    probabilities: 0.678 0.322
##
## Node number 205: 2587 observations,    complexity param=0.00594
##   predicted class=<=50K  expected loss=0.451  P(node) =0.0798
##     class counts:  1420  1167
##    probabilities: 0.549 0.451
##   left son=410 (2160 obs) right son=411 (427 obs)
##   Primary splits:
##       Exec.managerial  < 0.5 to the left,     improve=28.0, (0 missing)
##       Self.emp.not.inc < 0.5 to the right,    improve=21.5, (0 missing)
##       Farming.fishing  < 0.5 to the right,    improve=20.3, (0 missing)
##       Other.service    < 0.5 to the right,    improve=19.1, (0 missing)
##       fnlwgt           splits as  --L-RRLRRL, improve=13.2, (0 missing)
##   Surrogate splits:
##       fnlwgt splits as  --L-LLLRLL, agree=0.835, adj=0.002, (0 split)
##
## Node number 410: 2160 observations
##   predicted class=<=50K  expected loss=0.419  P(node) =0.0667
##     class counts:  1256   904
##    probabilities: 0.581 0.419
##
## Node number 411: 427 observations
##   predicted class=>50K   expected loss=0.384  P(node) =0.0132
##     class counts:   164   263
##    probabilities: 0.384 0.616
```

```
#rpart tree
rpart.plot(treepruned)
```




```
#Confusion matrix - train data
confusionMatrix(newtrain3$income, predict(treepruned, newdata = newtrain3,
                                          type="class"))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction <=50K  >50K
##      <=50K 23193  1527
##      >50K   3429  4253
##
##                Accuracy : 0.847
##                  95% CI : (0.8431, 0.8509)
##     No Information Rate : 0.8216
##     P-Value [Acc > NIR] : < 0.00000000000000022
##
##                   Kappa : 0.5377
##  Mcnemar's Test P-Value : < 0.00000000000000022
##
##             Sensitivity : 0.8712
##             Specificity : 0.7358
##          Pos Pred Value : 0.9382
##          Neg Pred Value : 0.5536
##              Prevalence : 0.8216
##          Detection Rate : 0.7158
##    Detection Prevalence : 0.7629
##       Balanced Accuracy : 0.8035
##
##        'Positive' Class : <=50K
##
```

```r
#Confusion matrix - test data
confusionMatrix(newtest3$income,  predict(treepruned, newdata = newtest3,
                                          type="class"))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction <=50K  >50K
##      <=50K 11680   755
##      >50K   1708  2053
##
##                Accuracy : 0.8479
##                  95% CI : (0.8423, 0.8534)
##     No Information Rate : 0.8266
##     P-Value [Acc > NIR] : 0.0000000000001788
##
##                   Kappa : 0.5322
##  Mcnemar's Test P-Value : < 0.00000000000000022
##
##             Sensitivity : 0.8724
##             Specificity : 0.7311
##          Pos Pred Value : 0.9393
##          Neg Pred Value : 0.5459
##              Prevalence : 0.8266
##          Detection Rate : 0.7212
##    Detection Prevalence : 0.7678
##       Balanced Accuracy : 0.8018
##
##        'Positive' Class : <=50K
##
```

### Fancy way

# Bagged Tree