In [3]:
```
library(rJava)
library(RWeka)
```

In [4]:
```
?Weka_classifier_trees
```

Weka_classifier_trees {RWeka} R Documentation

R/Weka Classifier Trees

Description

R interfaces to Weka regression and classification tree learners.

Usage

J48(formula, data, subset, na.action,control = Weka_control(), options = NULL)

LMT(formula, data, subset, na.action,control = Weka_control(), options = NULL)

M5P(formula, data, subset, na.action,control = Weka_control(), options = NULL)

DecisionStump(formula, data, subset, na.action, control = Weka_control(), options = NULL)

Arguments

formula :a symbolic description of the model to be fit.

data :an optional data frame containing the variables in the model.

subset :an optional vector specifying a subset of observations to be used in the fitting process.

na.action :a function which indicates what should happen when the data contain NAs. See model.frame for details.

control :an object of class Weka_control giving options to be passed to the Weka learner. Available options can be obtained on-line using the Weka Option Wizard WOW, or the Weka documentation.

options :a named list of further options, or NULL (default). See Details.

(a) formula - If we want to predict X based on attributes Y and Z, the formula takes the form of X ~ Y + Z. If we want to predict X based on all the attributes, you can use the shorthand X ~ .

(b) data - This is the training data.

(c) control - This is used to set various parameters for the decision tree. You can view the available controls with,

In [5]:
```
WOW(J48)
```

```
-U      Use unpruned tree.
-O      Do not collapse tree.
-C <pruning confidence>
        Set confidence threshold for pruning.  (default 0.25)
        Number of arguments: 1.
-M <minimum number of instances>
        Set minimum number of instances per leaf.  (default 2)
        Number of arguments: 1.
-R      Use reduced error pruning.
-N <number of folds>
        Set number of folds for reduced error pruning. One fold is used
        as pruning set.  (default 3)
        Number of arguments: 1.
-B      Use binary splits only.
-S      Do not perform subtree raising.
-L      Do not clean up after the tree has been built.
-A      Laplace smoothing for predicted probabilities.
-J      Do not use MDL correction for info gain on numeric attributes.
-Q <seed>
        Seed for random data shuffling (default 1).
        Number of arguments: 1.
-doNotMakeSplitPointActualValue
        Do not make split point actual value.
-output-debug-info
        If set, classifier is run in debug mode and may output
        additional info to the console
-do-not-check-capabilities
        If set, classifier capabilities are not checked before
        classifier is built (use with caution).
-num-decimal-places
        The number of decimal places for the output of numbers in the
        model (default 2).
        Number of arguments: 1.
-batch-size
        The desired batch size for batch prediction (default 100).
        Number of arguments: 1.
```

In [8]:
```
adult.train <- read.csv("adult_train.csv")
head(adult.train)
```

age	workclass	education	marital.status	occupation	relationship	race	sex	capital.gain	capital.loss	hours.per.week	native
59	Private	Some-college	Married-civ-spouse	Adm-clerical	Other-relative	White	Female	0	0	16	United·
21	Private	Some-college	Never-married	Adm-clerical	Own-child	Black	Male	0	0	50	United·
38	Private	Bachelors	Divorced	Prof-specialty	Unmarried	Black	Female	15020	0	45	United·
33	Private	Some-college	Married-civ-spouse	Handlers-cleaners	Husband	White	Male	0	0	50	United·
52	Self-emp-not-inc	HS-grad	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	98	United·
42	Federal-gov	Bachelors	Married-civ-spouse	Exec-managerial	Husband	White	Male	7298	0	50	United·

In [12]:
```
mdl <- J48(income ~ ., data = adult.train, control = Weka_control(M = 10, C = 0.25, B = F))
```

In [15]:
```
adult.test <- read.csv("adult_test.csv")
evaluate_Weka_classifier(mdl, newdata = adult.test, class = T)
```

```
=== Summary ===

Correctly Classified Instances         769               76.9    %
Incorrectly Classified Instances       231               23.1    %
Kappa statistic                          0.5366
Mean absolute error                      0.3016
Root mean squared error                  0.4139
Relative absolute error                 60.3336 %
Root relative squared error             82.7864 %
Total Number of Instances             1000

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.663    0.128    0.834      0.663    0.739      0.548    0.811     0.784      <=50K
                 0.872    0.337    0.727      0.872    0.793      0.548    0.811     0.762      >50K
Weighted Avg.    0.769    0.234    0.780      0.769    0.766      0.548    0.811     0.773

=== Confusion Matrix ===

   a   b   <-- classified as
 327 166 |   a =  <=50K
  65 442 |   b =  >50K
```

In [17]:
```
#Consider two models mdl.1 and mdl.2.
mdl.1 <- J48(income ~ ., data = adult.train, control = Weka_control(M = 2, U = T, B = T))
mdl.2 <- J48(income ~ ., data = adult.train, control = Weka_control(M = 2, U = F, B = F))
#compare these two models with k-fold cross, k=10
evaluate_Weka_classifier(mdl.1, class = T, numFolds = 10)
print("@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@")
evaluate_Weka_classifier(mdl.2, class = T, numFolds = 10)
```

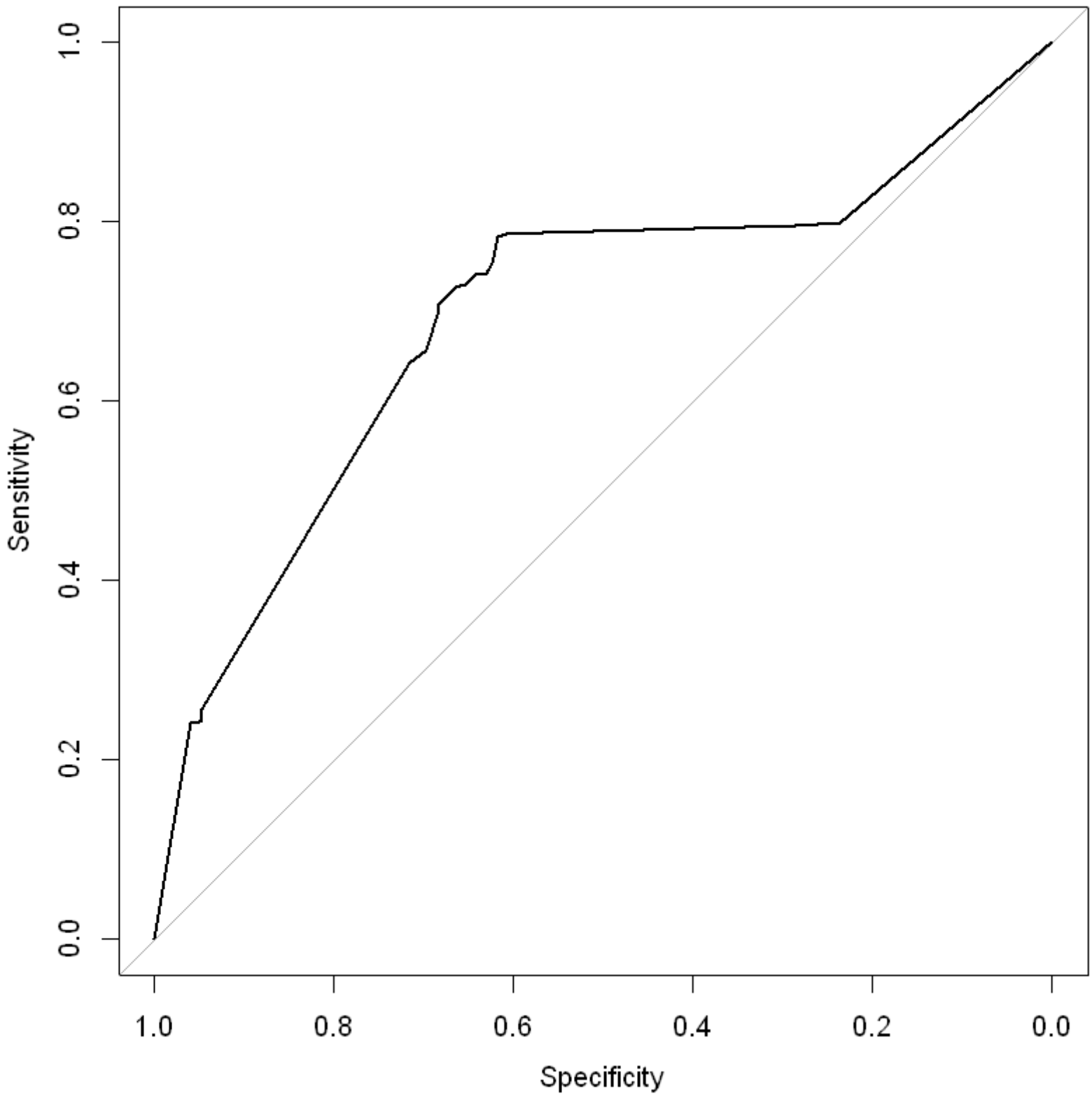
```
=== 10 Fold Cross Validation ===

=== Summary ===

Correctly Classified Instances        3114               77.85   %
Incorrectly Classified Instances       886               22.15   %
Kappa statistic                          0.5571
Mean absolute error                      0.2398
Root mean squared error                  0.4206
Relative absolute error                 47.9556 %
Root relative squared error             84.1195 %
Total Number of Instances             4000

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.761    0.204    0.790      0.761    0.775      0.557    0.817     0.776      <=50K
                 0.796    0.239    0.768      0.796    0.782      0.557    0.817     0.763      >50K
Weighted Avg.    0.779    0.221    0.779      0.779    0.778      0.557    0.817     0.769

=== Confusion Matrix ===

    a    b   <-- classified as
 1528  479 |    a =  <=50K
  407 1586 |    b =  >50K

[1] "@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@"

=== 10 Fold Cross Validation ===

=== Summary ===

Correctly Classified Instances        3203               80.075  %
Incorrectly Classified Instances       797               19.925  %
Kappa statistic                          0.6015
Mean absolute error                      0.2696
Root mean squared error                  0.3827
Relative absolute error                 53.9175 %
Root relative squared error             76.5332 %
Total Number of Instances             4000

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.791    0.189    0.808      0.791    0.799      0.602    0.862     0.824      <=50K
                 0.811    0.209    0.794      0.811    0.802      0.602    0.862     0.847      >50K
Weighted Avg.    0.801    0.199    0.801      0.801    0.801      0.602    0.862     0.835

=== Confusion Matrix ===

    a    b   <-- classified as
 1587  420 |    a =  <=50K
  377 1616 |    b =  >50K
```

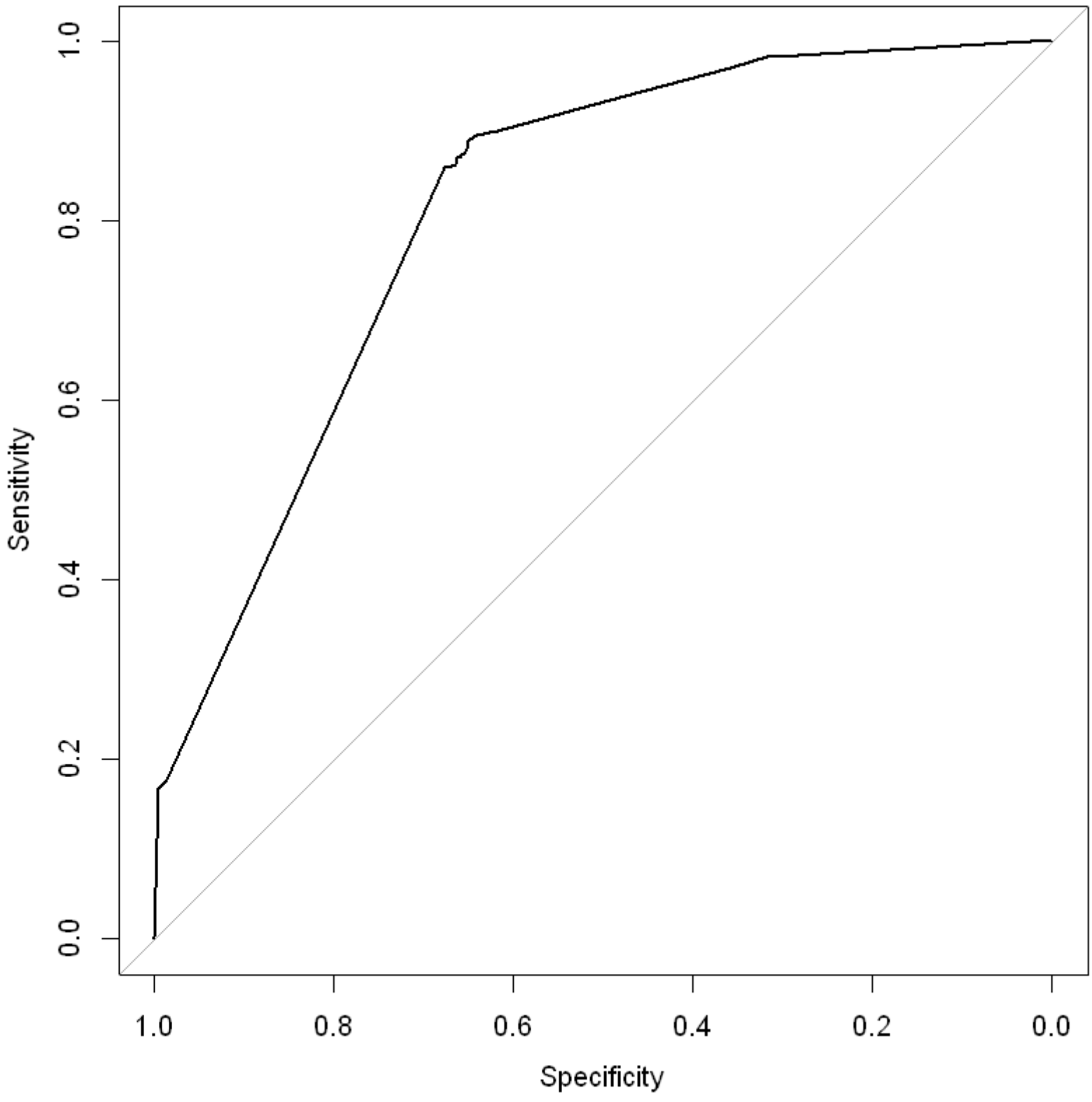
In [18]:
```
##compare these two models with test data
evaluate_Weka_classifier(mdl.1, class = T, newdata = adult.test)
evaluate_Weka_classifier(mdl.2, class = T, newdata = adult.test)
```

```
=== Summary ===

Correctly Classified Instances         697               69.7    %
Incorrectly Classified Instances       303               30.3    %
Kappa statistic                          0.3934
Mean absolute error                      0.3267
Root mean squared error                  0.5092
Relative absolute error                 65.3599 %
Root relative squared error            101.8514 %
Total Number of Instances             1000

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.665    0.272    0.704      0.665    0.684      0.394    0.697     0.618      <=50K
                 0.728    0.335    0.691      0.728    0.709      0.394    0.697     0.698      >50K
Weighted Avg.    0.697    0.304    0.697      0.697    0.697      0.394    0.697     0.658

=== Confusion Matrix ===

   a   b   <-- classified as
 328 165 |   a =  <=50K
 138 369 |   b =  >50K

=== Summary ===

Correctly Classified Instances         767               76.7    %
Incorrectly Classified Instances       233               23.3    %
Kappa statistic                          0.5325
Mean absolute error                      0.3
Root mean squared error                  0.414
Relative absolute error                 60.0032 %
Root relative squared error             82.8064 %
Total Number of Instances             1000

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.657    0.126    0.835      0.657    0.736      0.545    0.813     0.790      <=50K
                 0.874    0.343    0.724      0.874    0.792      0.545    0.813     0.762      >50K
Weighted Avg.    0.767    0.236    0.779      0.767    0.764      0.545    0.813     0.776

=== Confusion Matrix ===

   a   b   <-- classified as
 324 169 |   a =  <=50K
  64 443 |   b =  >50K
```

In [19]:
```
#Performance Evaluation with ROC
library(pROC)

#Get the predicted class probabilities of adult.test using mdl.1 and mdl.2
p.1 <- predict(mdl.1, newdata = adult.test, type = c("prob"))
p.2 <- predict(mdl.2, newdata = adult.test, type = c("prob"))

#Calculate the ROC.
roc.1 <- roc(adult.test$income, p.1[,1])
roc.2 <- roc(adult.test$income, p.2[,1])
#plot ROC
plot(roc.1)
plot(roc.2)
```

```
Type 'citation("pROC")' for a citation.

Attaching package: 'pROC'

The following objects are masked from 'package:stats':

    cov, smooth, var
```





ROC

In statistics, a receiver operating characteristic curve, or ROC curve, is a graphical plot that illustrates the performance of a binary classifier system as its discrimination threshold is varied. The curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The true-positive rate is also known as sensitivity, recall or probability of detection[1] in machine learning. The false-positive rate is also known as the fall-out or probability of false alarm and can be calculated as (1 − specificity).

In [ ]: