```
# boosting.r
RNGkind(sample.kind = 'Rounding')
```

```
## Warning in RNGkind(sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```
library(MASS)      # Boston dataset
library(gbm)       # gbm()
```

```
## Loaded gbm 2.1.5
```

```
dim(Boston)
```

```
## [1] 506  14
```

```
#
# response is medv
# p=13 predictors
n = nrow(Boston)
set.seed(1)
train = sample(1:n,n/2)    # 253 train rows

# boosted regression trees
#==============================================================================================
set.seed(1)
boost1=gbm(medv~.,data=Boston[train,],distribution="gaussian",
           n.trees=5000,interaction.depth=4)
#
boost1
```

```
## gbm(formula = medv ~ ., distribution = "gaussian", data = Boston[train,
##     ], n.trees = 5000, interaction.depth = 4)
## A gradient boosted model with gaussian loss function.
## 5000 iterations were performed.
## There were 13 predictors of which 13 had non-zero influence.
```

```
# for categorical response use distribution="bernoulli"
# depth of each tree limited to 4 splits
# 5000 trees (default is 100 trees)
# interaction.depth is d
#
names(boost1)
```

```
##  [1] "initF"             "fit"               "train.error"
##  [4] "valid.error"       "oobag.improve"     "trees"
##  [7] "c.splits"          "bag.fraction"      "distribution"
## [10] "interaction.depth" "n.minobsinnode"    "num.classes"
## [13] "n.trees"           "nTrain"            "train.fraction"
## [16] "response.name"     "shrinkage"         "var.levels"
## [19] "var.monotone"      "var.names"         "var.type"
## [22] "verbose"           "data"              "Terms"
## [25] "cv.folds"          "call"              "m"
```

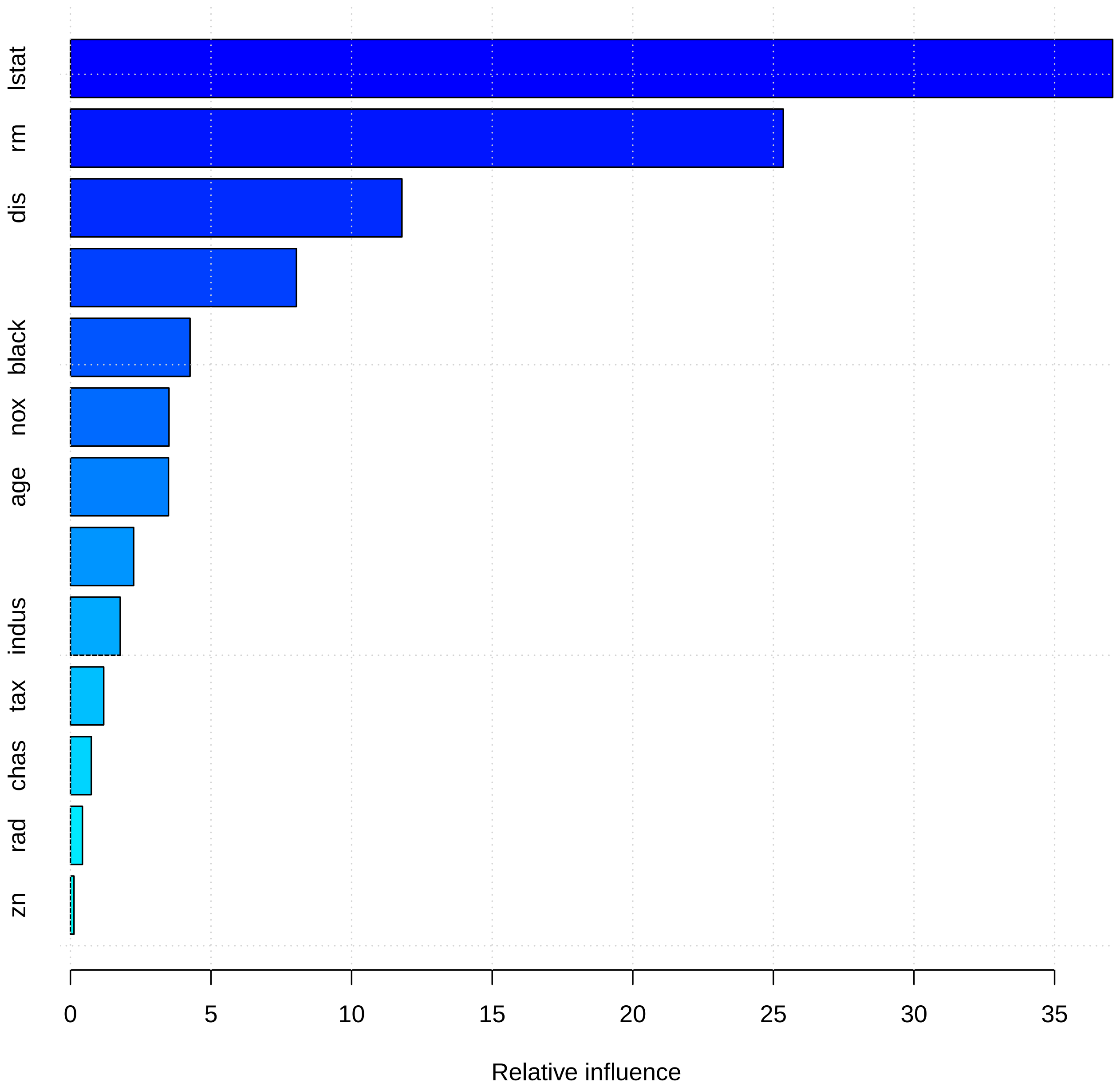
```
#
# lambda -default value
boost1$shrinkage
```

```
## [1] 0.1
```

```r
# importance of predictors
summary(boost1);
```

```
##              var    rel.inf
## lstat     lstat 37.0661275
## rm           rm 25.3533123
## dis         dis 11.7903016
## crim       crim  8.0388750
## black     black  4.2531659
## nox         nox  3.5058570
## age         age  3.4868724
## ptratio ptratio  2.2500385
## indus     indus  1.7725070
## tax         tax  1.1836592
## chas       chas  0.7441319
## rad         rad  0.4274311
## zn           zn  0.1277206
```

```r
grid()
```



```
#
# lstat and rm best predictors

# test MSPE
ytest=Boston[-train,"medv"]        # y values in test set
#
# predict test set -n.trees is required
#
yhat.boost=predict(boost1,newdata=Boston[-train,],n.trees=5000)
mean((yhat.boost-ytest)^2)
```

```
## [1] 10.81479
```

```
#
# little improvement over Random Forest
```

```r
# Try lambda = 0.2   (shrinkage)
#
boost2=gbm(medv~.,data=Boston[train,],distribution="gaussian",
           n.trees=5000,interaction.depth=4,shrinkage=0.2,verbose=F)
yhat.boost=predict(boost2,newdata=Boston[-train,],n.trees=5000)
mean((yhat.boost-ytest)^2)
```

```
## [1] 11.51109
```

```r
#
# not as good as default lambda
```