Stats 315B: Homework 1

Joe Higgins, Austin Wang, Jessica Wetstone

Due 5/19/2018

Question 1

Random forests predict with an ensemble of bagged trees each trained on a bootstrap sample randomly drawn from the original training data. Additional random variation among the trees is induced by choosing the variable for each split from a small randomly chosen subset of all of the predictor variables when building each tree. What are the advantages and disadvantages of this random variable selection strategy? How can one introduce additional tree variation in the forest without randomly selecting subsets of variables?

Question 2

Why is it necessary to use regularization in linear regression when the number of predictor variables is greater than the number of observations in the training sample? Explain how regularization helps in this case. Are there other situations where regularization might help? What is the potential disadvantage of introducing regularization? Why is sparsity a reasonable assumption in the boosting context. Is it always? If not, why not?

Question 3

Show that the convex members of the power family of penalties, except for the lasso, have the property that solutions to $\hat{a}(\lambda) = argmin_a \hat{R}(a) + \lambda P_\gamma(a)$ have nonzero values for all coefficients at each path point indexed by λ. By contrast the convex members of the elastic net (except ridge) can produce solutions with many zero valued coefficients at various path points.

Question 4

Show that the variable x_{j*} that has the maximum absolute correlation with $j^* = argmax_{1\leq j\leq J}|E(yx_j)|$ is the same as the one that best predicts y using squared—error loss. This shows that the base learner most correlated with the generalized residual is the one that best predicts it with squared—error loss.

Question 5

Binary classification: Spam Email. The data set for this problem is spam_stats315B.csv, with documentation files spam_stats315B_info.txt and spam_stats315B_names,txt. The data set is a collection of 4601 emails of which 1813 were considered spam, i.e. unsolicited commercial email. The data set consists of 58 attributes of which 57 are continuous predictors and one is a class label that indicates whether the email was considered spam (1) or not (0). Among the 57 predictor attributes are: percentage of the word "free" in the email, percentage of exclamation marks in the email, etc. See file spam_stats315B_names.txt for the full list of attributes. The goal is, of course, to predict whether or not an email is "spam". This data set is used for illustration in the tutorial Boosting with R Programming. The data set spam_stats315B_train.csv represents a subsample of these emails randomly selected from spam_stats315B.csv to be used for training. The file spam_stats315B_test.csv contains the remaining emails to be used for evaluating results.

(a) Based on the training data, fit a gbm model for predicting whether or not an email is "spam", following the example in the tutorial. What is your estimate of the misclassification rate? Of all the spam emails of the test set what percentage was misclassified, and of all the non-spam emails in the test set what percentage was misclassified?

```r
rm(list = ls())
data_path <- paste(getwd(),'/data',sep='')
setwd(data_path)

#data labels
rflabs<-c("make", "address", "all", "3d", "our", "over", "remove",
  "internet","order", "mail", "receive", "will",
  "people", "report", "addresses","free", "business",
  "email", "you", "credit", "your", "font","000","money",
  "hp", "hpl", "george", "650", "lab", "labs",
  "telnet", "857", "data", "415", "85", "technology", "1999",
  "parts","pm", "direct", "cs", "meeting", "original", "project",
  "re","edu", "table", "conference", ";", "(", "[", "!", "$", "#",
  "CAPAVE", "CAPMAX", "CAPTOT","type")

#load the data
train <- read.csv(file="spam_stats315B_train.csv", header=FALSE, sep=",")
test <- read.csv(file="spam_stats315B_test.csv", header=FALSE, sep=",")
colnames(train)<-rflabs
colnames(test)<-rflabs

#randomize order of data
set.seed(131)
x <- train[sample(nrow(train)),]

#fit model on training data
set.seed(444) # random for bag.fraction
gbm0 <- gbm(type~., data=x, train.fraction=1,
            interaction.depth=4, shrinkage=.05,
            n.trees=2500, bag.fraction=0.5, cv.folds=5,
            distribution="bernoulli", verbose=T)
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.2867             nan     0.0500    0.0272
##      2        1.2377             nan     0.0500    0.0232
##      3        1.1911             nan     0.0500    0.0227
##      4        1.1499             nan     0.0500    0.0202
##      5        1.1058             nan     0.0500    0.0224
##      6        1.0713             nan     0.0500    0.0170
##      7        1.0331             nan     0.0500    0.0195
##      8        0.9969             nan     0.0500    0.0177
##      9        0.9639             nan     0.0500    0.0163
##     10        0.9349             nan     0.0500    0.0140
##     20        0.7237             nan     0.0500    0.0073
##     40        0.5177             nan     0.0500    0.0050
##     60        0.4070             nan     0.0500    0.0016
##     80        0.3477             nan     0.0500    0.0007
##    100        0.3113             nan     0.0500    0.0002
##    120        0.2848             nan     0.0500    0.0005
##    140        0.2655             nan     0.0500    0.0005
##    160        0.2497             nan     0.0500    0.0001
##    180        0.2368             nan     0.0500    0.0001
##    200        0.2237             nan     0.0500    0.0001
```

```
##    220        0.2127            nan     0.0500   -0.0001
##    240        0.2026            nan     0.0500   -0.0001
##    260        0.1932            nan     0.0500   -0.0000
##    280        0.1846            nan     0.0500    0.0000
##    300        0.1757            nan     0.0500   -0.0001
##    320        0.1692            nan     0.0500    0.0000
##    340        0.1626            nan     0.0500   -0.0000
##    360        0.1570            nan     0.0500   -0.0000
##    380        0.1504            nan     0.0500    0.0000
##    400        0.1438            nan     0.0500   -0.0001
##    420        0.1383            nan     0.0500   -0.0001
##    440        0.1337            nan     0.0500   -0.0000
##    460        0.1286            nan     0.0500   -0.0001
##    480        0.1241            nan     0.0500   -0.0001
##    500        0.1199            nan     0.0500   -0.0000
##    520        0.1161            nan     0.0500   -0.0000
##    540        0.1128            nan     0.0500   -0.0001
##    560        0.1090            nan     0.0500   -0.0000
##    580        0.1057            nan     0.0500   -0.0001
##    600        0.1026            nan     0.0500   -0.0001
##    620        0.0993            nan     0.0500   -0.0000
##    640        0.0961            nan     0.0500   -0.0001
##    660        0.0930            nan     0.0500   -0.0000
##    680        0.0900            nan     0.0500   -0.0000
##    700        0.0873            nan     0.0500   -0.0001
##    720        0.0849            nan     0.0500   -0.0001
##    740        0.0820            nan     0.0500   -0.0000
##    760        0.0797            nan     0.0500    0.0000
##    780        0.0778            nan     0.0500   -0.0000
##    800        0.0759            nan     0.0500   -0.0000
##    820        0.0733            nan     0.0500    0.0000
##    840        0.0712            nan     0.0500   -0.0000
##    860        0.0693            nan     0.0500    0.0000
##    880        0.0668            nan     0.0500   -0.0001
##    900        0.0648            nan     0.0500   -0.0001
##    920        0.0632            nan     0.0500   -0.0000
##    940        0.0616            nan     0.0500   -0.0001
##    960        0.0599            nan     0.0500   -0.0001
##    980        0.0582            nan     0.0500   -0.0000
##   1000        0.0569            nan     0.0500   -0.0000
##   1020        0.0554            nan     0.0500   -0.0000
##   1040        0.0541            nan     0.0500   -0.0001
##   1060        0.0528            nan     0.0500   -0.0001
##   1080        0.0515            nan     0.0500   -0.0000
##   1100        0.0502            nan     0.0500   -0.0001
##   1120        0.0490            nan     0.0500   -0.0001
##   1140        0.0479            nan     0.0500   -0.0000
##   1160        0.0465            nan     0.0500   -0.0001
##   1180        0.0454            nan     0.0500   -0.0000
##   1200        0.0444            nan     0.0500   -0.0000
##   1220        0.0433            nan     0.0500   -0.0000
##   1240        0.0423            nan     0.0500   -0.0000
##   1260        0.0413            nan     0.0500   -0.0000
##   1280        0.0404            nan     0.0500   -0.0000
```

```
##   1300        0.0394            nan     0.0500   -0.0000
##   1320        0.0385            nan     0.0500   -0.0000
##   1340        0.0375            nan     0.0500   -0.0000
##   1360        0.0366            nan     0.0500   -0.0000
##   1380        0.0357            nan     0.0500   -0.0000
##   1400        0.0349            nan     0.0500   -0.0000
##   1420        0.0341            nan     0.0500   -0.0000
##   1440        0.0333            nan     0.0500   -0.0001
##   1460        0.0326            nan     0.0500   -0.0000
##   1480        0.0319            nan     0.0500   -0.0000
##   1500        0.0312            nan     0.0500   -0.0000
##   1520        0.0305            nan     0.0500   -0.0000
##   1540        0.0296            nan     0.0500   -0.0000
##   1560        0.0290            nan     0.0500   -0.0000
##   1580        0.0284            nan     0.0500   -0.0000
##   1600        0.0277            nan     0.0500   -0.0000
##   1620        0.0270            nan     0.0500   -0.0000
##   1640        0.0264            nan     0.0500   -0.0000
##   1660        0.0258            nan     0.0500   -0.0000
##   1680        0.0254            nan     0.0500   -0.0000
##   1700        0.0248            nan     0.0500   -0.0000
##   1720        0.0243            nan     0.0500   -0.0000
##   1740        0.0237            nan     0.0500   -0.0000
##   1760        0.0232            nan     0.0500   -0.0000
##   1780        0.0226            nan     0.0500   -0.0000
##   1800        0.0222            nan     0.0500   -0.0000
##   1820        0.0217            nan     0.0500   -0.0000
##   1840        0.0212            nan     0.0500   -0.0000
##   1860        0.0208            nan     0.0500   -0.0000
##   1880        0.0202            nan     0.0500   -0.0000
##   1900        0.0198            nan     0.0500   -0.0000
##   1920        0.0194            nan     0.0500   -0.0000
##   1940        0.0190            nan     0.0500   -0.0000
##   1960        0.0186            nan     0.0500   -0.0000
##   1980        0.0182            nan     0.0500   -0.0000
##   2000        0.0178            nan     0.0500   -0.0000
##   2020        0.0174            nan     0.0500   -0.0000
##   2040        0.0171            nan     0.0500   -0.0000
##   2060        0.0167            nan     0.0500   -0.0000
##   2080        0.0163            nan     0.0500   -0.0000
##   2100        0.0160            nan     0.0500   -0.0000
##   2120        0.0157            nan     0.0500   -0.0000
##   2140        0.0153            nan     0.0500   -0.0000
##   2160        0.0150            nan     0.0500   -0.0000
##   2180        0.0147            nan     0.0500   -0.0000
##   2200        0.0144            nan     0.0500    0.0000
##   2220        0.0141            nan     0.0500   -0.0000
##   2240        0.0138            nan     0.0500   -0.0000
##   2260        0.0134            nan     0.0500   -0.0000
##   2280        0.0132            nan     0.0500   -0.0000
##   2300        0.0129            nan     0.0500   -0.0000
##   2320        0.0127            nan     0.0500   -0.0000
##   2340        0.0124            nan     0.0500   -0.0000
##   2360        0.0121            nan     0.0500   -0.0000
```

```
##   2380        0.0119             nan     0.0500   -0.0000
##   2400        0.0116             nan     0.0500   -0.0000
##   2420        0.0114             nan     0.0500   -0.0000
##   2440        0.0112             nan     0.0500   -0.0000
##   2460        0.0110             nan     0.0500   -0.0000
##   2480        0.0107             nan     0.0500   -0.0000
##   2500        0.0104             nan     0.0500   -0.0000
```

```
#estimate of misclassification rate
y_hat_train <- predict(gbm0, x, type="response", n.trees=300)
y_hat_train[y_hat_train > 0.5] <- 1
y_hat_train[y_hat_train <= 0.5] <- 0
y <- x$type
correct <- y == y_hat_train
pct_correct <- sum(correct)/length(correct)

#What is your estimate of the misclassification rate?
1-pct_correct
```

```
## [1] 0.02575807
```

```
#misclassification rate test
y_hat_test <- predict(gbm0, test, type="response", n.trees=300)
y_hat_test[y_hat_test > 0.5] <- 1
y_hat_test[y_hat_test <= 0.5] <- 0
y <- test$type
correct <- y == y_hat_test

#Of all the spam emails of the test set what percentage was misclassified?
spam_correct <- correct[y==1]
pct_spam_correct <- sum(spam_correct)/length(spam_correct)
1-pct_spam_correct
```

```
## [1] 0.05501618
```

```
#Of all the non-spam emails in the test set what percentage was misclassified?
nonspam_correct <- correct[y==0]
pct_nonspam_correct <- sum(nonspam_correct)/length(nonspam_correct)
1-pct_nonspam_correct
```

```
## [1] 0.02729258
```

(b) Your classifier in part (a) can be used as a spam filter. One of the possible disadvantages of such a spam filter is that it might filter out too many good (non-spam) emails. Therefore, a better spam filter might be the one that penalizes misclassifying non-spam emails more heavily than the spam ones. Suppose that you want to build a spam filter that "throws out" no more that 0.3% of the good (non-spam) emails. You have to find and use a cost matrix that penalizes misclassifying "good" emails as"spam" more than misclassifying "spam" emails as "good" by the method of trial and error. Once you have constructed your final spam filter with the property described above, answer the following questions:

```
w <- x$type
w[w == 1] <- 9
w[w == 0] <- 1
```

```r
gbm1 <- gbm(type~., data=x , train.fraction=1,
            interaction.depth=4, shrinkage=.05,
            n.trees=2500, bag.fraction=0.5, cv.folds=5,
            distribution="bernoulli", verbose=T,
            weights = w)
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1         0.7772             nan     0.0500    0.0248
##      2         0.7389             nan     0.0500    0.0188
##      3         0.7066             nan     0.0500    0.0159
##      4         0.6784             nan     0.0500    0.0138
##      5         0.6515             nan     0.0500    0.0122
##      6         0.6261             nan     0.0500    0.0116
##      7         0.6051             nan     0.0500    0.0102
##      8         0.5849             nan     0.0500    0.0094
##      9         0.5691             nan     0.0500    0.0078
##     10         0.5547             nan     0.0500    0.0068
##     20         0.4437             nan     0.0500    0.0038
##     40         0.3289             nan     0.0500    0.0020
##     60         0.2703             nan     0.0500    0.0016
##     80         0.2369             nan     0.0500    0.0002
##    100         0.2120             nan     0.0500    0.0008
##    120         0.1942             nan     0.0500    0.0002
##    140         0.1803             nan     0.0500   -0.0000
##    160         0.1704             nan     0.0500   -0.0001
##    180         0.1613             nan     0.0500   -0.0001
##    200         0.1533             nan     0.0500    0.0001
##    220         0.1467             nan     0.0500    0.0001
##    240         0.1408             nan     0.0500   -0.0000
##    260         0.1359             nan     0.0500   -0.0000
##    280         0.1310             nan     0.0500   -0.0000
##    300         0.1254             nan     0.0500    0.0000
##    320         0.1200             nan     0.0500    0.0000
##    340         0.1160             nan     0.0500   -0.0001
##    360         0.1123             nan     0.0500   -0.0000
##    380         0.1091             nan     0.0500   -0.0001
##    400         0.1056             nan     0.0500   -0.0000
##    420         0.1022             nan     0.0500   -0.0001
##    440         0.0986             nan     0.0500   -0.0000
##    460         0.0953             nan     0.0500   -0.0001
##    480         0.0921             nan     0.0500   -0.0000
##    500         0.0891             nan     0.0500    0.0000
##    520         0.0872             nan     0.0500   -0.0000
##    540         0.0844             nan     0.0500   -0.0001
##    560         0.0825             nan     0.0500   -0.0000
##    580         0.0804             nan     0.0500   -0.0000
##    600         0.0776             nan     0.0500    0.0000
##    620         0.0756             nan     0.0500   -0.0000
##    640         0.0739             nan     0.0500   -0.0000
##    660         0.0712             nan     0.0500   -0.0000
##    680         0.0694             nan     0.0500   -0.0000
##    700         0.0672             nan     0.0500   -0.0001
##    720         0.0651             nan     0.0500   -0.0000
##    740         0.0632             nan     0.0500   -0.0000
```

```
##    760        0.0608            nan     0.0500   -0.0000
##    780        0.0590            nan     0.0500   -0.0000
##    800        0.0574            nan     0.0500   -0.0000
##    820        0.0559            nan     0.0500   -0.0000
##    840        0.0547            nan     0.0500   -0.0000
##    860        0.0529            nan     0.0500   -0.0000
##    880        0.0516            nan     0.0500   -0.0000
##    900        0.0499            nan     0.0500   -0.0000
##    920        0.0482            nan     0.0500    0.0000
##    940        0.0469            nan     0.0500   -0.0001
##    960        0.0458            nan     0.0500   -0.0000
##    980        0.0449            nan     0.0500   -0.0000
##   1000        0.0436            nan     0.0500   -0.0000
##   1020        0.0426            nan     0.0500    0.0000
##   1040        0.0417            nan     0.0500   -0.0001
##   1060        0.0406            nan     0.0500   -0.0000
##   1080        0.0397            nan     0.0500   -0.0000
##   1100        0.0388            nan     0.0500   -0.0000
##   1120        0.0378            nan     0.0500   -0.0000
##   1140        0.0372            nan     0.0500   -0.0000
##   1160        0.0365            nan     0.0500   -0.0001
##   1180        0.0357            nan     0.0500   -0.0000
##   1200        0.0346            nan     0.0500   -0.0000
##   1220        0.0336            nan     0.0500   -0.0000
##   1240        0.0326            nan     0.0500   -0.0000
##   1260        0.0317            nan     0.0500   -0.0000
##   1280        0.0309            nan     0.0500   -0.0000
##   1300        0.0301            nan     0.0500   -0.0000
##   1320        0.0293            nan     0.0500   -0.0000
##   1340        0.0285            nan     0.0500   -0.0000
##   1360        0.0278            nan     0.0500    0.0000
##   1380        0.0271            nan     0.0500   -0.0000
##   1400        0.0266            nan     0.0500   -0.0000
##   1420        0.0259            nan     0.0500   -0.0000
##   1440        0.0252            nan     0.0500   -0.0000
##   1460        0.0246            nan     0.0500   -0.0000
##   1480        0.0239            nan     0.0500   -0.0000
##   1500        0.0231            nan     0.0500   -0.0000
##   1520        0.0227            nan     0.0500   -0.0000
##   1540        0.0220            nan     0.0500   -0.0000
##   1560        0.0214            nan     0.0500    0.0000
##   1580        0.0209            nan     0.0500   -0.0000
##   1600        0.0204            nan     0.0500   -0.0000
##   1620        0.0201            nan     0.0500   -0.0000
##   1640        0.0196            nan     0.0500    0.0000
##   1660        0.0192            nan     0.0500   -0.0000
##   1680        0.0188            nan     0.0500   -0.0000
##   1700        0.0185            nan     0.0500   -0.0000
##   1720        0.0181            nan     0.0500   -0.0000
##   1740        0.0177            nan     0.0500   -0.0000
##   1760        0.0173            nan     0.0500   -0.0000
##   1780        0.0169            nan     0.0500   -0.0000
##   1800        0.0165            nan     0.0500   -0.0000
##   1820        0.0162            nan     0.0500   -0.0000
```

```
##   1840        0.0157             nan     0.0500   -0.0000
##   1860        0.0153             nan     0.0500   -0.0000
##   1880        0.0150             nan     0.0500   -0.0000
##   1900        0.0146             nan     0.0500   -0.0000
##   1920        0.0142             nan     0.0500   -0.0000
##   1940        0.0139             nan     0.0500   -0.0000
##   1960        0.0136             nan     0.0500   -0.0000
##   1980        0.0132             nan     0.0500   -0.0000
##   2000        0.0128             nan     0.0500   -0.0000
##   2020        0.0126             nan     0.0500   -0.0000
##   2040        0.0123             nan     0.0500   -0.0000
##   2060        0.0120             nan     0.0500   -0.0000
##   2080        0.0117             nan     0.0500   -0.0000
##   2100        0.0115             nan     0.0500   -0.0000
##   2120        0.0113             nan     0.0500   -0.0000
##   2140        0.0110             nan     0.0500   -0.0000
##   2160        0.0108             nan     0.0500   -0.0000
##   2180        0.0106             nan     0.0500   -0.0000
##   2200        0.0104             nan     0.0500    0.0000
##   2220        0.0102             nan     0.0500   -0.0000
##   2240        0.0099             nan     0.0500   -0.0000
##   2260        0.0097             nan     0.0500   -0.0000
##   2280        0.0095             nan     0.0500   -0.0000
##   2300        0.0094             nan     0.0500   -0.0000
##   2320        0.0092             nan     0.0500   -0.0000
##   2340        0.0090             nan     0.0500    0.0000
##   2360        0.0088             nan     0.0500   -0.0000
##   2380        0.0086             nan     0.0500   -0.0000
##   2400        0.0084             nan     0.0500    0.0000
##   2420        0.0082             nan     0.0500   -0.0000
##   2440        0.0081             nan     0.0500   -0.0000
##   2460        0.0079             nan     0.0500   -0.0000
##   2480        0.0077             nan     0.0500   -0.0000
##   2500        0.0076             nan     0.0500   -0.0000
```

```
#misclassification rate on training set
y_hat_train <- predict(gbm1, x, type="response", n.trees=300)
y_hat_train[y_hat_train > 0.5] <- 1
y_hat_train[y_hat_train <= 0.5] <- 0
y <- x$type
correct_train <- y == y_hat_train

#Of all the spam emails of the test set what percentage was misclassified?
spam_correct_train <- correct_train[y==1]
pct_spam_correct_train <- sum(spam_correct_train)/length(spam_correct_train)
1-pct_spam_correct_train
```

```
## [1] 0.002463054
```

```
#Of all the non-spam emails in the test set what percentage was misclassified?
nonspam_correct_train <- correct_train[y==0]
pct_nonspam_correct_train <- sum(nonspam_correct_train)/length(nonspam_correct_train)
1-pct_nonspam_correct_train
```

```
## [1] 0.09897242
```

(i) What is the overall misclassification error of your final filter and what is the percentage of good emails and spam emails that were misclassified respectively?

```
#misclassification rate on training set
y_hat_test <- predict(gbm1, test, type="response", n.trees=300)
y_hat_test[y_hat_test > 0.5] <- 1
y_hat_test[y_hat_test <= 0.5] <- 0
y <- test$type
correct_test <- y == y_hat_test
spam_correct_test <- correct_test[y==1]
nonspam_correct_test <- correct_test[y==0]

#Overall misclassification:
pct_correct_train <- sum(correct_train)/length(correct_train)
pct_correct_test <- sum(correct_test)/length(correct_test)
1-pct_correct_train
```

```
## [1] 0.06064558
```

```
1-pct_correct_test
```

```
## [1] 0.07301173
```

```
#Spam misclassification:
pct_spam_correct_test <- sum(spam_correct_test)/length(spam_correct_test)
1-pct_spam_correct_train
```

```
## [1] 0.002463054
```

```
1-pct_spam_correct_test
```

```
## [1] 0.009708738
```

```
#Nonspam misclassification:
pct_nonspam_correct_test <- sum(nonspam_correct_test)/length(nonspam_correct_test)
1-pct_nonspam_correct_train
```

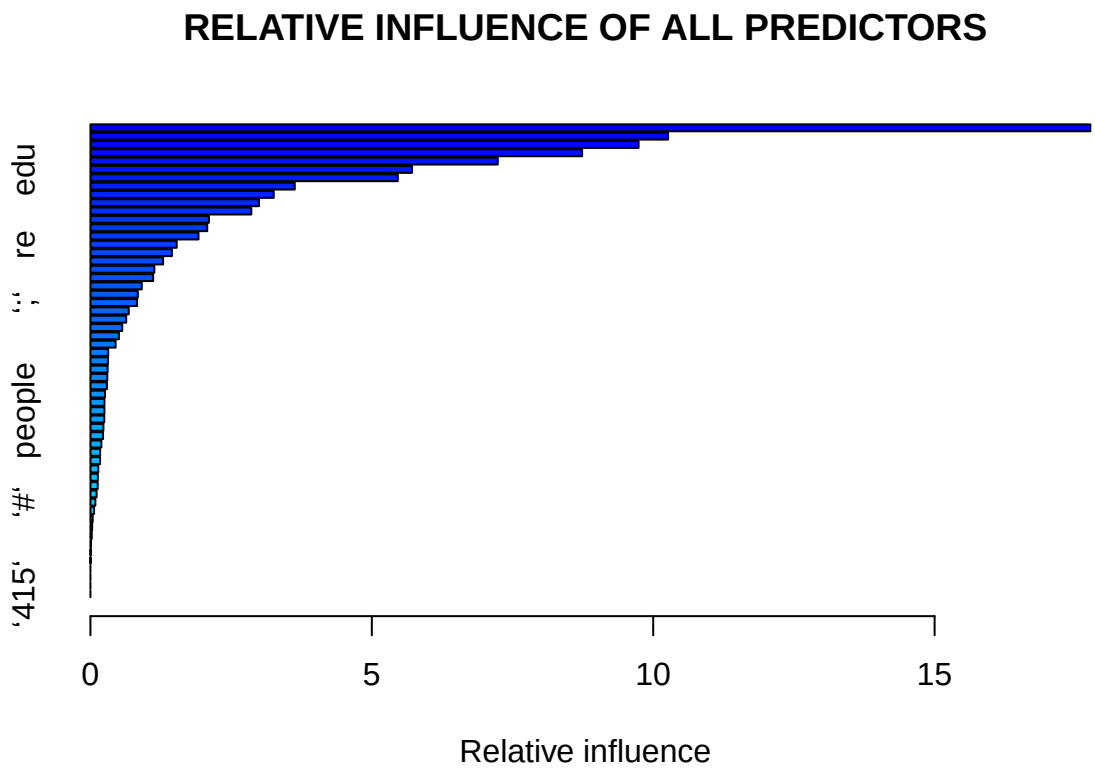
```
## [1] 0.09897242
```

```
1-pct_nonspam_correct_test
```

```
## [1] 0.1157205
```

(ii) What are the important variables in discriminating good emails from spam for your spam filter?

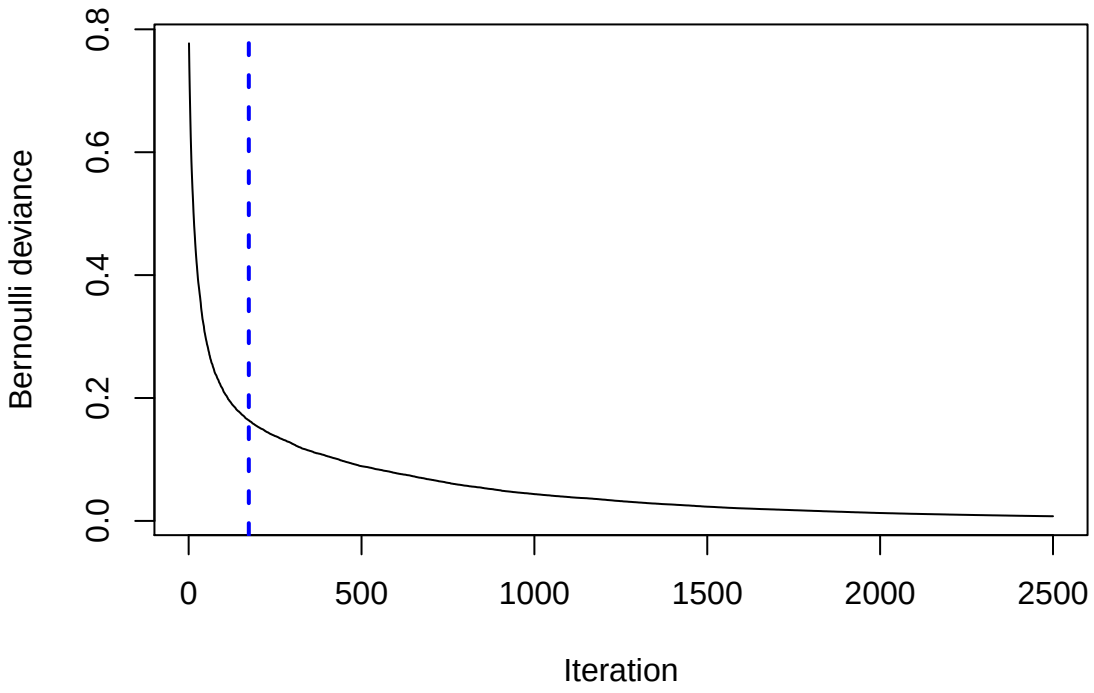
```
top5 <-summary(gbm1,main="RELATIVE INFLUENCE OF ALL PREDICTORS")$var[1:5]
```



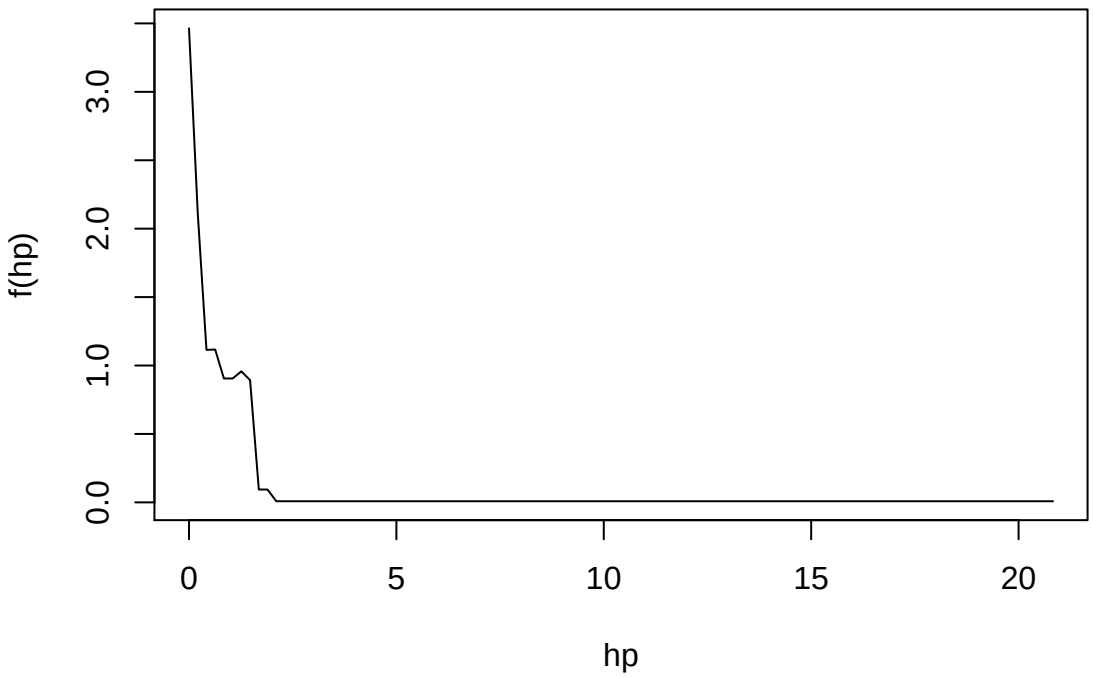
###(iii) Using the interpreting tools provided by gbm, describe the dependence of the response on the most important attributes.

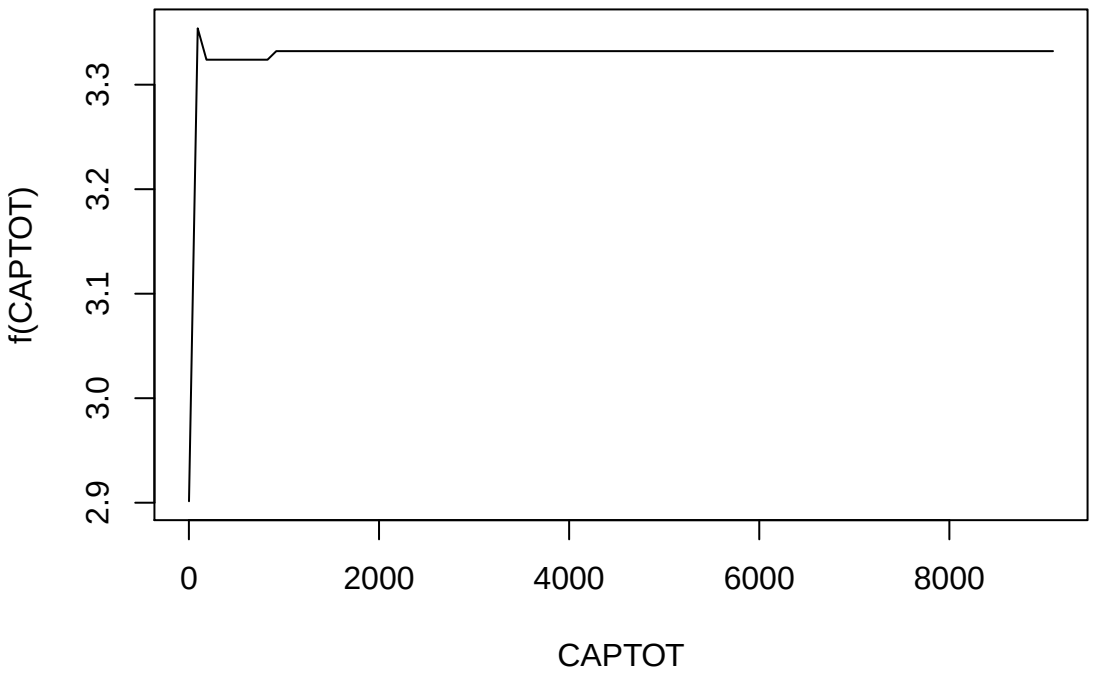
```
best.iter <- gbm.perf(gbm1,method="OOB")#Best iteration by OOB
```

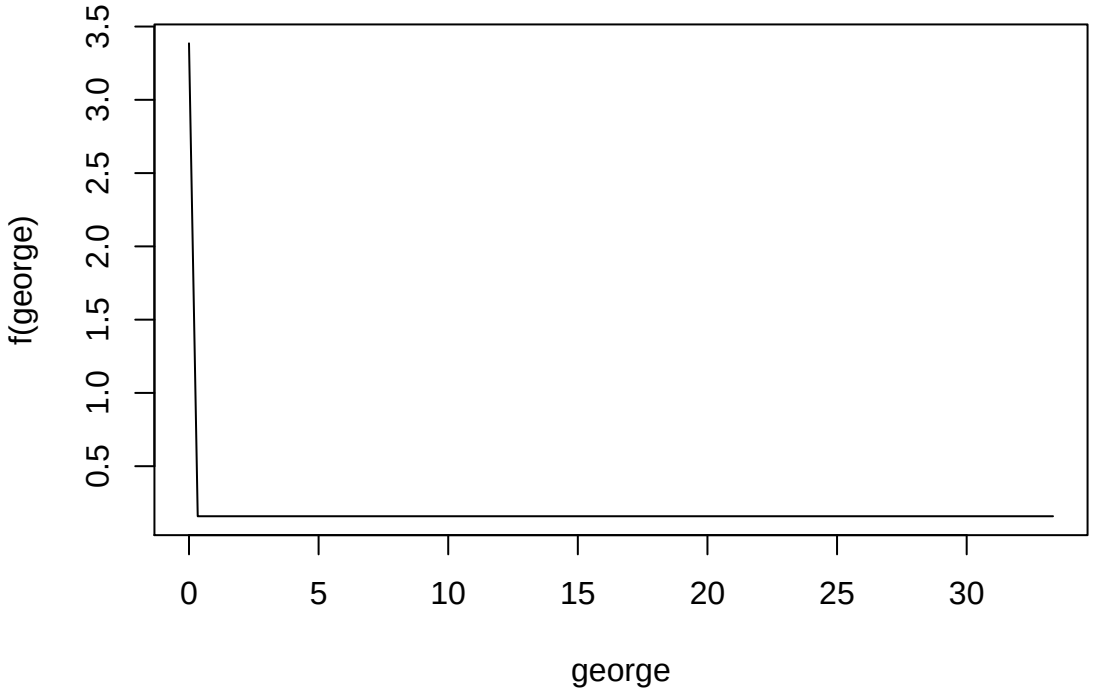
```
## Warning in gbm.perf(gbm1, method = "OOB"): OOB generally underestimates the
## optimal number of iterations although predictive performance is reasonably
## competitive. Using cv.folds>0 when calling gbm usually results in improved
## predictive performance.
```

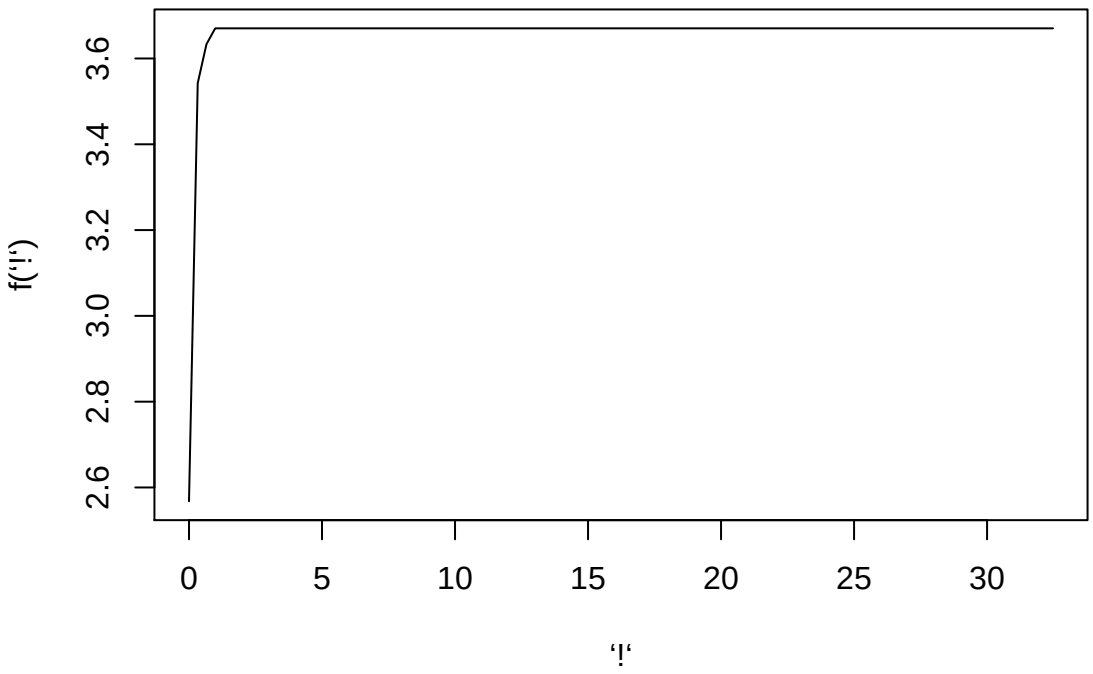


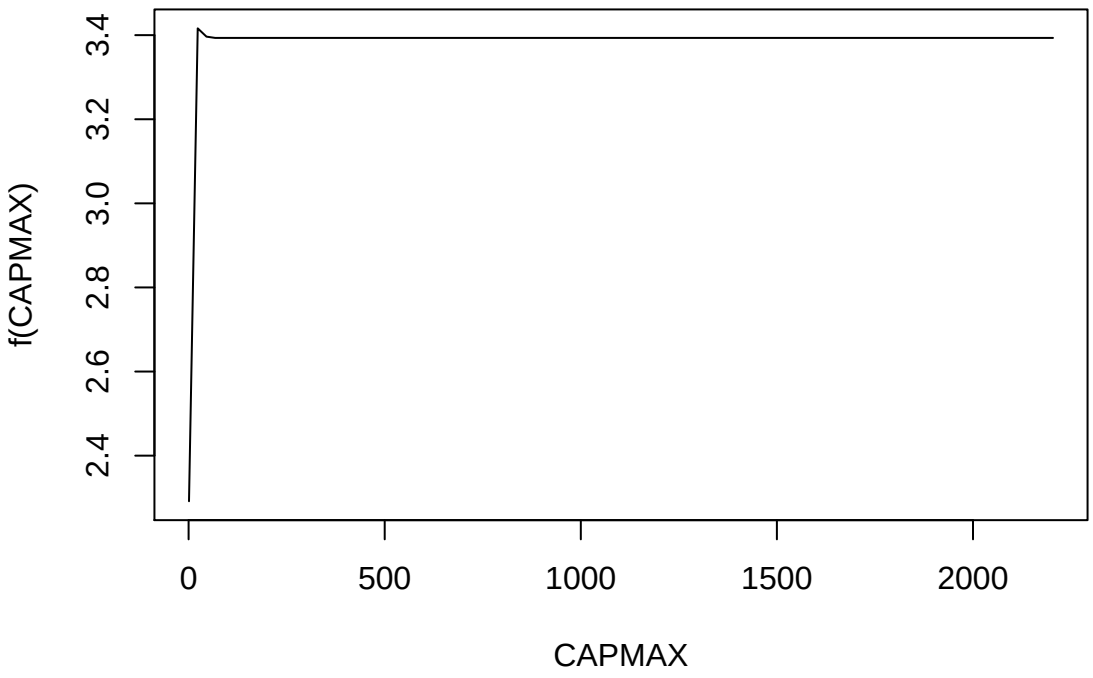

```
top5 <- gsub("`","",top5)
top5_indexes <- match(top5, rflabs)
for(i in top5_indexes){
  plot(gbm1,i,best.iter)
}
```


f(hp)
0.0
1.0
2.0
3.0
0
5
10
15
20
hp


f(CAPTOT)
2.9
3.0
3.1
3.2
3.3
0
2000
4000
6000
8000
CAPTOT


f(george)
0.5
1.0
1.5
2.0
2.5
3.0
3.5
0
5
10
15
20
25
30
george


f(‘i‘)
2.6
2.8
3.0
3.2
3.4
3.6
0
5
10
15
20
25
30
‘i‘



Question 7

Regression: California Housing. The data set calif_stats315B.csv consists of aggregated data from 20,640 California census blocks (from the 1990 census). The goal is to predict the median house value in each neighborhood from the others described in calif_stats315B.txt. Fit a gbm model to the data and write a short report that should include at least

(a) The prediction accuracy of gbm on the data set.

```
rm(list = ls())
data_path <- paste(getwd(),'/data',sep='')
setwd(data_path)

#data labls
labels <-c(
  "house_value",
  "median_income",
  "housing_median_age",
  "average_no_rooms",
  "average_no_bedrooms",
  "population",
  "average_occupancy",
  "latitude",
```

```
  "longitude"
)

#load the data
data <- read.csv(file="calif_stats315B.csv", header=FALSE, sep=",")
colnames(data) <- labels

#fit model on training data
set.seed(444) # random for bag.fraction
model <- gbm(house_value~., data=data, train.fraction=1,
            interaction.depth=4, shrinkage=.05,
            n.trees=2500, bag.fraction=0.5, cv.folds=5,
            distribution="gaussian", verbose=T)
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.2675             nan     0.0500    0.0635
##      2        1.2098             nan     0.0500    0.0574
##      3        1.1579             nan     0.0500    0.0526
##      4        1.1096             nan     0.0500    0.0489
##      5        1.0654             nan     0.0500    0.0448
##      6        1.0256             nan     0.0500    0.0395
##      7        0.9880             nan     0.0500    0.0369
##      8        0.9548             nan     0.0500    0.0336
##      9        0.9229             nan     0.0500    0.0315
##     10        0.8938             nan     0.0500    0.0286
##     20        0.6982             nan     0.0500    0.0138
##     40        0.5131             nan     0.0500    0.0091
##     60        0.4108             nan     0.0500    0.0049
##     80        0.3644             nan     0.0500    0.0014
##    100        0.3368             nan     0.0500    0.0011
##    120        0.3178             nan     0.0500    0.0006
##    140        0.3046             nan     0.0500    0.0011
##    160        0.2950             nan     0.0500    0.0004
##    180        0.2870             nan     0.0500   -0.0000
##    200        0.2816             nan     0.0500    0.0005
##    220        0.2748             nan     0.0500    0.0000
##    240        0.2697             nan     0.0500   -0.0000
##    260        0.2654             nan     0.0500    0.0003
##    280        0.2616             nan     0.0500   -0.0000
##    300        0.2585             nan     0.0500    0.0001
##    320        0.2547             nan     0.0500   -0.0001
##    340        0.2512             nan     0.0500    0.0001
##    360        0.2485             nan     0.0500    0.0001
##    380        0.2454             nan     0.0500   -0.0000
##    400        0.2428             nan     0.0500   -0.0000
##    420        0.2403             nan     0.0500    0.0000
##    440        0.2379             nan     0.0500   -0.0000
##    460        0.2355             nan     0.0500   -0.0000
##    480        0.2336             nan     0.0500    0.0001
##    500        0.2316             nan     0.0500    0.0000
##    520        0.2297             nan     0.0500    0.0000
##    540        0.2276             nan     0.0500    0.0001
##    560        0.2261             nan     0.0500    0.0000
##    580        0.2240             nan     0.0500    0.0002
```

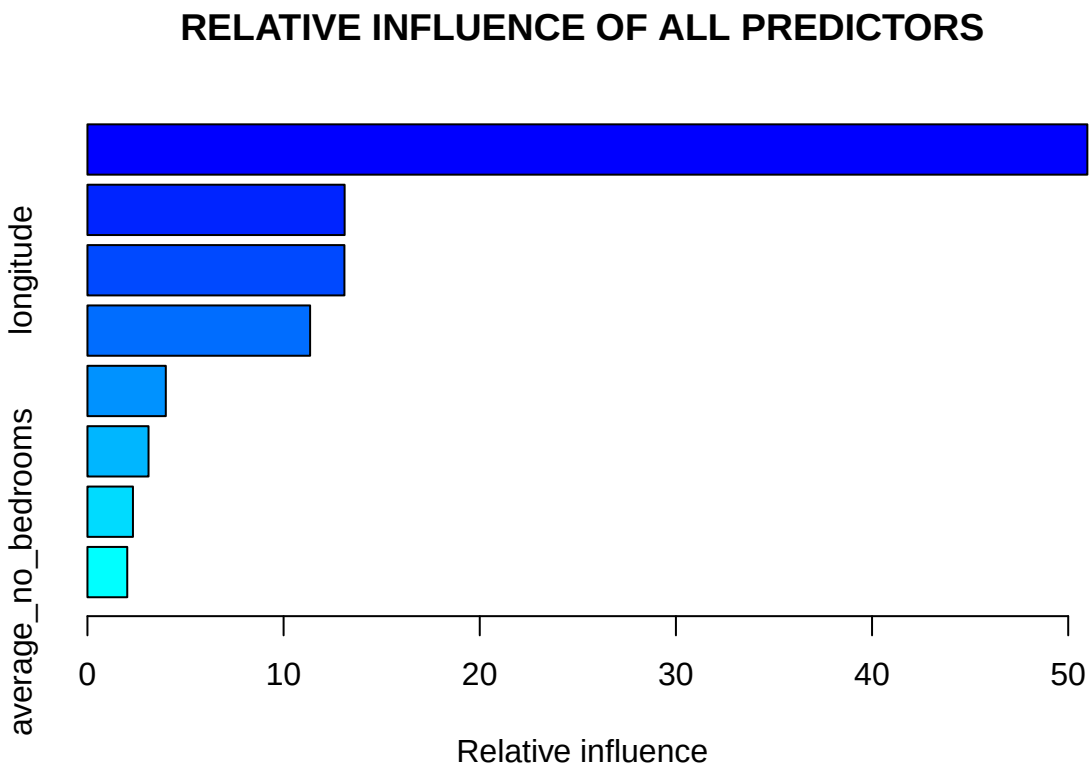
```
##    600        0.2223            nan     0.0500   -0.0000
##    620        0.2209            nan     0.0500    0.0000
##    640        0.2195            nan     0.0500   -0.0000
##    660        0.2181            nan     0.0500   -0.0000
##    680        0.2168            nan     0.0500   -0.0000
##    700        0.2154            nan     0.0500    0.0001
##    720        0.2142            nan     0.0500   -0.0000
##    740        0.2129            nan     0.0500   -0.0000
##    760        0.2116            nan     0.0500    0.0001
##    780        0.2099            nan     0.0500    0.0000
##    800        0.2088            nan     0.0500   -0.0000
##    820        0.2074            nan     0.0500    0.0000
##    840        0.2061            nan     0.0500   -0.0001
##    860        0.2049            nan     0.0500   -0.0000
##    880        0.2038            nan     0.0500   -0.0000
##    900        0.2027            nan     0.0500   -0.0001
##    920        0.2018            nan     0.0500   -0.0000
##    940        0.2009            nan     0.0500   -0.0000
##    960        0.2002            nan     0.0500   -0.0000
##    980        0.1993            nan     0.0500   -0.0000
##   1000        0.1984            nan     0.0500   -0.0000
##   1020        0.1975            nan     0.0500   -0.0001
##   1040        0.1965            nan     0.0500   -0.0000
##   1060        0.1958            nan     0.0500   -0.0000
##   1080        0.1951            nan     0.0500   -0.0000
##   1100        0.1940            nan     0.0500   -0.0000
##   1120        0.1931            nan     0.0500   -0.0000
##   1140        0.1923            nan     0.0500   -0.0000
##   1160        0.1917            nan     0.0500   -0.0001
##   1180        0.1909            nan     0.0500   -0.0001
##   1200        0.1900            nan     0.0500   -0.0000
##   1220        0.1890            nan     0.0500   -0.0000
##   1240        0.1880            nan     0.0500   -0.0000
##   1260        0.1874            nan     0.0500   -0.0000
##   1280        0.1866            nan     0.0500   -0.0000
##   1300        0.1859            nan     0.0500    0.0000
##   1320        0.1852            nan     0.0500   -0.0000
##   1340        0.1847            nan     0.0500   -0.0000
##   1360        0.1840            nan     0.0500   -0.0000
##   1380        0.1830            nan     0.0500   -0.0000
##   1400        0.1825            nan     0.0500   -0.0000
##   1420        0.1819            nan     0.0500   -0.0000
##   1440        0.1813            nan     0.0500   -0.0000
##   1460        0.1808            nan     0.0500   -0.0000
##   1480        0.1802            nan     0.0500    0.0000
##   1500        0.1796            nan     0.0500   -0.0000
##   1520        0.1788            nan     0.0500    0.0000
##   1540        0.1778            nan     0.0500    0.0000
##   1560        0.1773            nan     0.0500   -0.0000
##   1580        0.1768            nan     0.0500   -0.0000
##   1600        0.1763            nan     0.0500   -0.0000
##   1620        0.1756            nan     0.0500   -0.0000
##   1640        0.1749            nan     0.0500   -0.0000
##   1660        0.1743            nan     0.0500   -0.0001
```

```
##   1680        0.1738             nan     0.0500   -0.0000
##   1700        0.1733             nan     0.0500   -0.0001
##   1720        0.1728             nan     0.0500   -0.0000
##   1740        0.1724             nan     0.0500   -0.0000
##   1760        0.1717             nan     0.0500   -0.0000
##   1780        0.1713             nan     0.0500   -0.0001
##   1800        0.1708             nan     0.0500   -0.0000
##   1820        0.1702             nan     0.0500   -0.0000
##   1840        0.1696             nan     0.0500   -0.0001
##   1860        0.1691             nan     0.0500    0.0000
##   1880        0.1686             nan     0.0500    0.0000
##   1900        0.1680             nan     0.0500   -0.0000
##   1920        0.1675             nan     0.0500   -0.0000
##   1940        0.1671             nan     0.0500   -0.0000
##   1960        0.1667             nan     0.0500   -0.0000
##   1980        0.1662             nan     0.0500    0.0000
##   2000        0.1655             nan     0.0500   -0.0000
##   2020        0.1649             nan     0.0500   -0.0000
##   2040        0.1643             nan     0.0500   -0.0000
##   2060        0.1639             nan     0.0500    0.0000
##   2080        0.1634             nan     0.0500    0.0000
##   2100        0.1628             nan     0.0500   -0.0000
##   2120        0.1625             nan     0.0500   -0.0000
##   2140        0.1619             nan     0.0500   -0.0000
##   2160        0.1615             nan     0.0500   -0.0000
##   2180        0.1612             nan     0.0500   -0.0000
##   2200        0.1608             nan     0.0500   -0.0000
##   2220        0.1604             nan     0.0500   -0.0000
##   2240        0.1600             nan     0.0500   -0.0000
##   2260        0.1595             nan     0.0500   -0.0000
##   2280        0.1592             nan     0.0500   -0.0000
##   2300        0.1587             nan     0.0500   -0.0000
##   2320        0.1582             nan     0.0500   -0.0000
##   2340        0.1577             nan     0.0500   -0.0000
##   2360        0.1573             nan     0.0500   -0.0000
##   2380        0.1568             nan     0.0500   -0.0000
##   2400        0.1564             nan     0.0500   -0.0000
##   2420        0.1560             nan     0.0500   -0.0000
##   2440        0.1556             nan     0.0500   -0.0000
##   2460        0.1551             nan     0.0500    0.0000
##   2480        0.1548             nan     0.0500   -0.0000
##   2500        0.1543             nan     0.0500   -0.0000
```

```r
y <- data$house_value
y_hat <- predict(model, data, type="response", n.trees=300)
MSE <- sum((y_hat - y)^2)/length(y)
```

(b) Identification of the most important variables.

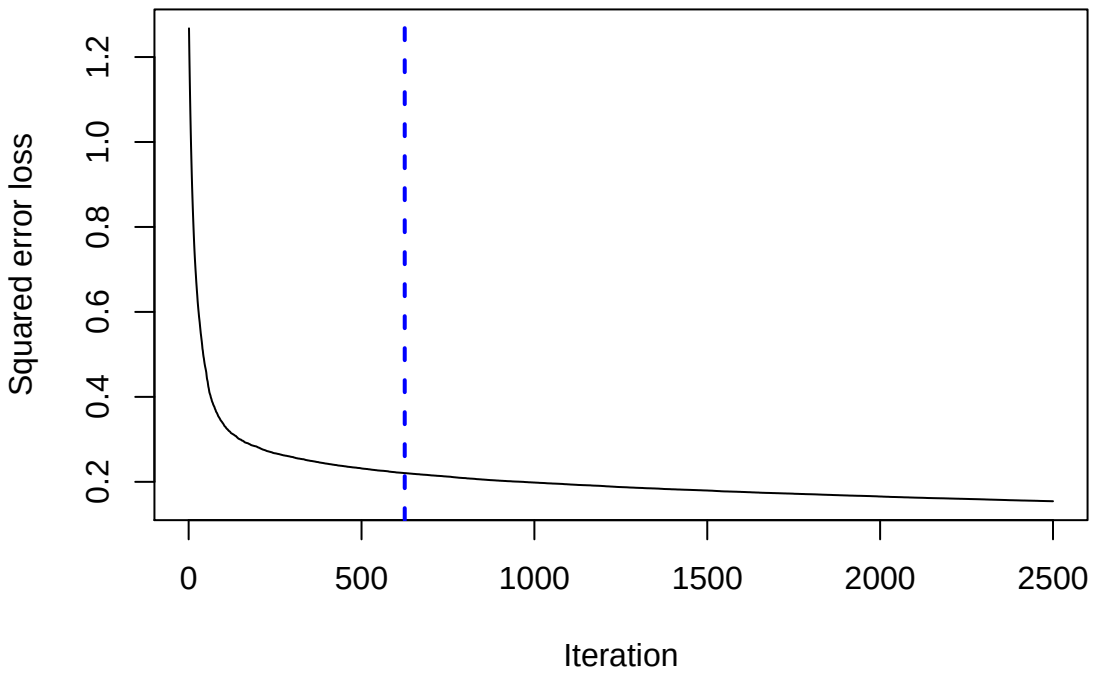
```r
top4 <-summary(model,main="RELATIVE INFLUENCE OF ALL PREDICTORS")$var[1:4]
```



(c) Comments on the dependence of the response on the most important variables (you may want to consider partial dependence plots (plot) on single and pairs of variables, etc.).

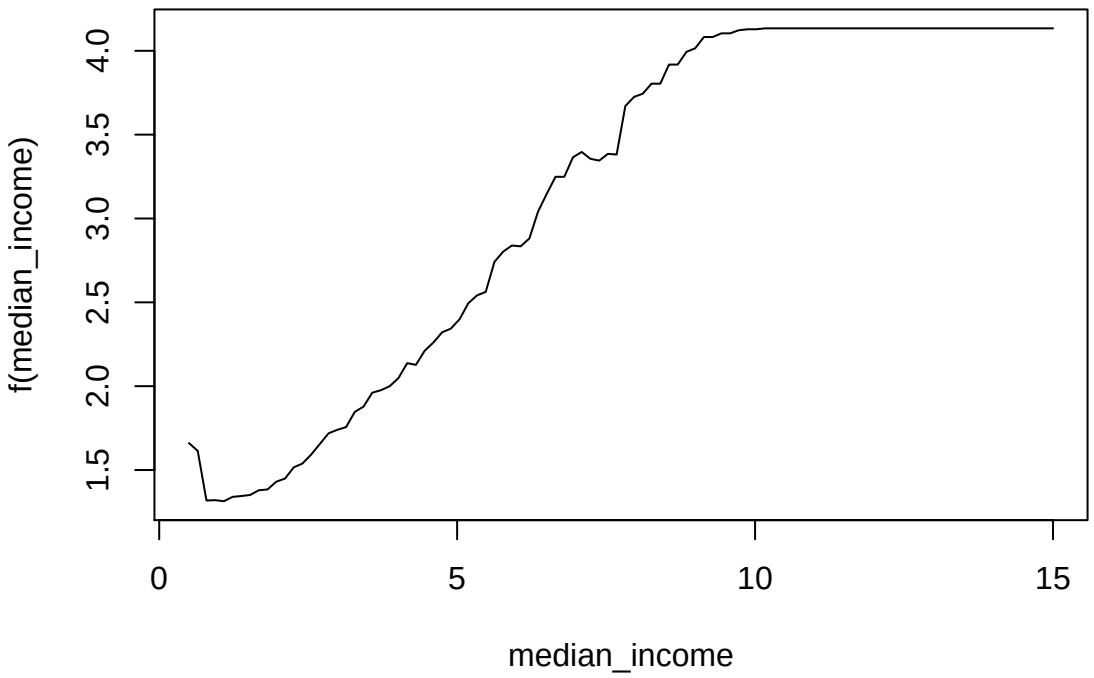
```
best.iter <- gbm.perf(model,method="OOB")#Best iteration by OOB
```

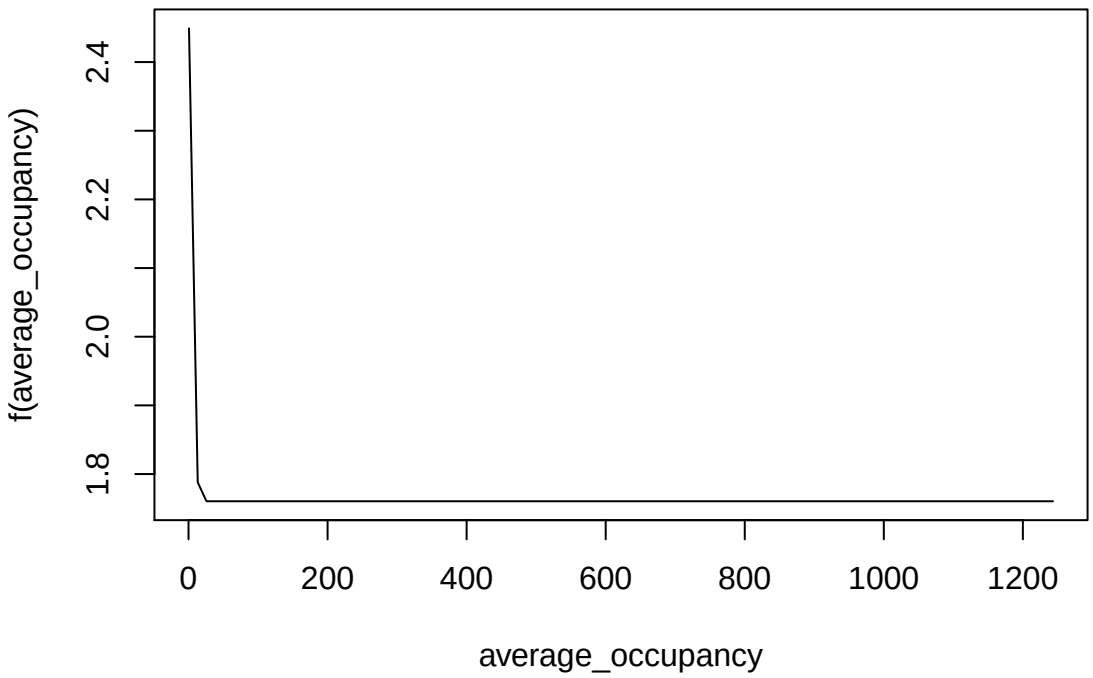
```
## Warning in gbm.perf(model, method = "OOB"): OOB generally underestimates
## the optimal number of iterations although predictive performance is
## reasonably competitive. Using cv.folds>0 when calling gbm usually results
## in improved predictive performance.
```

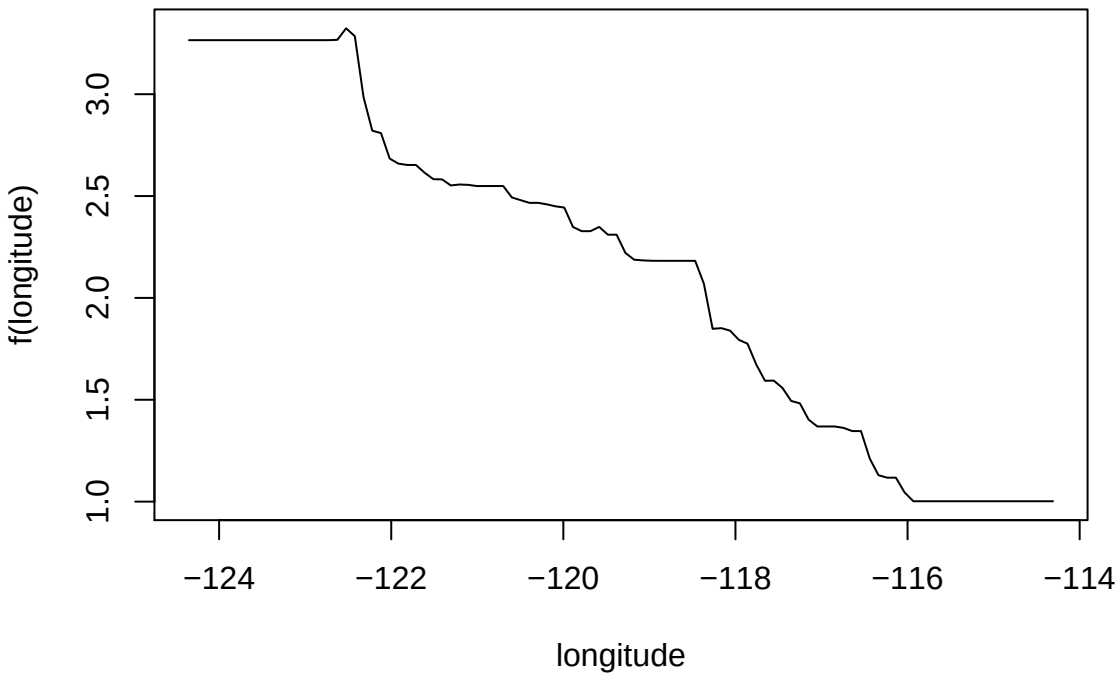


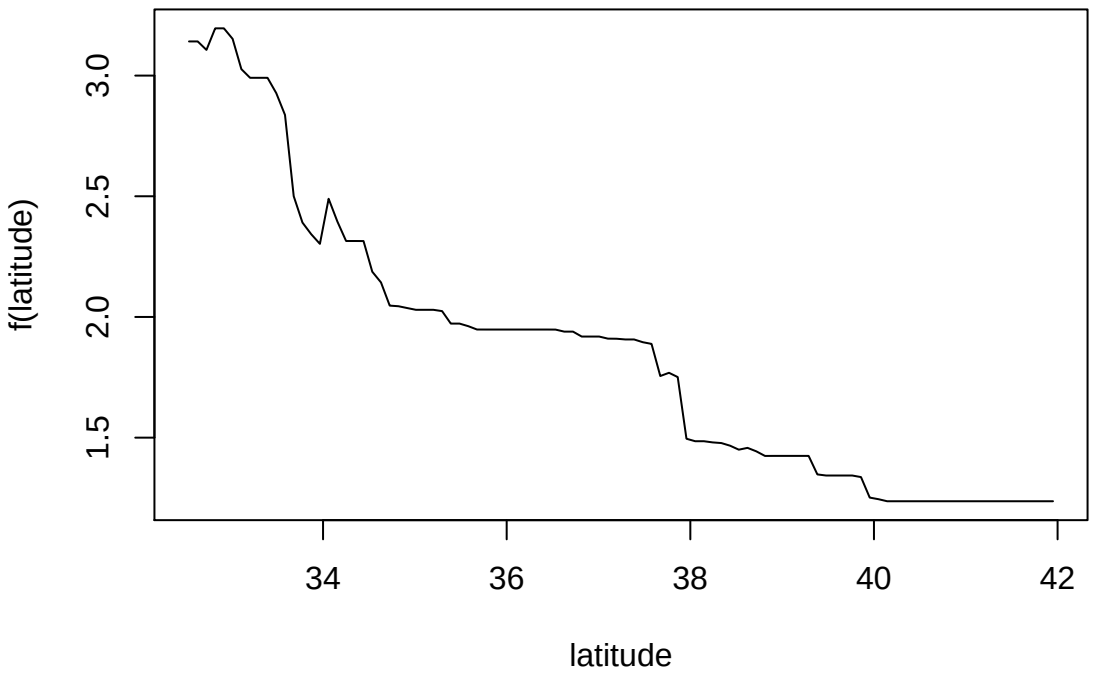

```
top4 <- gsub("`","",top4)
top4_indexes <- match(top4, labels) - 1 #-1 because first label is response var

#main effects
for(i in top4_indexes){
  plot(model,i,best.iter)
}
```

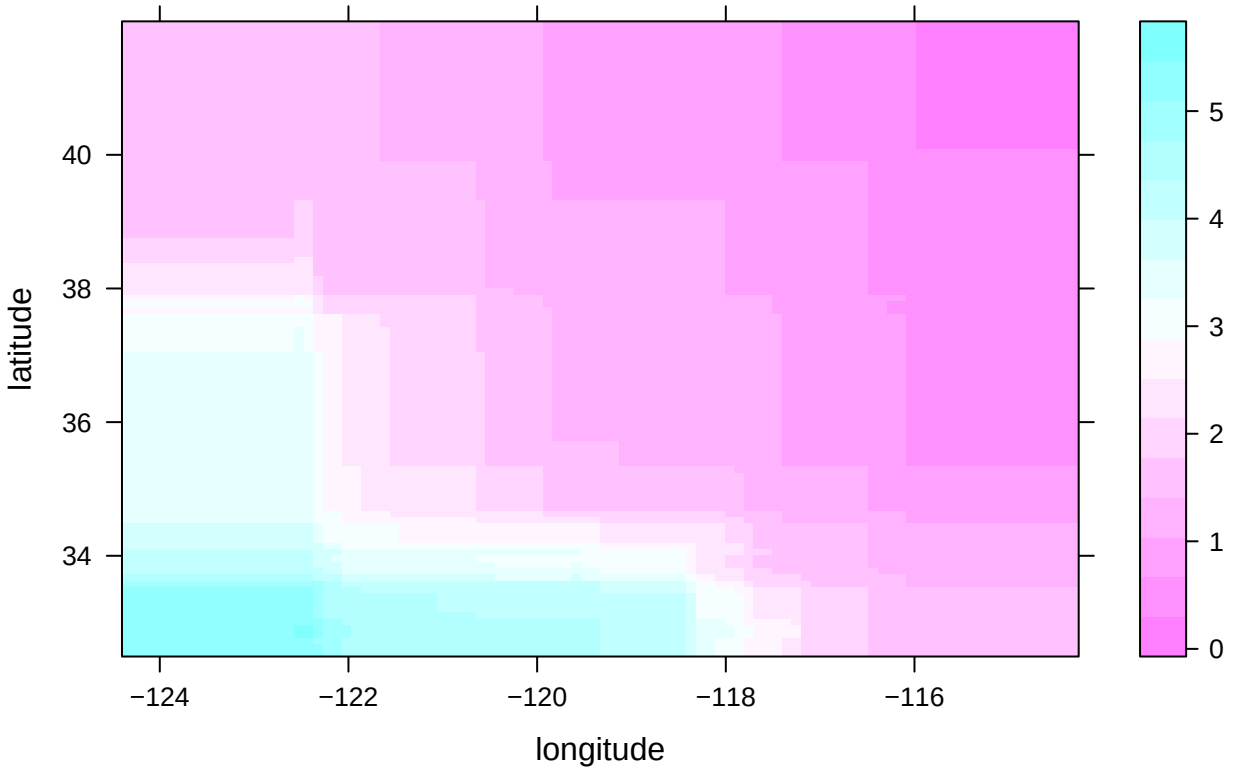

f(median_income)
1.5
2.0
2.5
3.0
3.5
4.0
0
5
10
15
median_income


f(average_occupancy)
1.8
2.0
2.2
2.4
0
200
400
600
800
1000
1200
average_occupancy


f(longitude)
1.0
1.5
2.0
2.5
3.0
−124
−122
−120
−118
−116
−114
longitude




```
#longitude, latitude interaction effects
plot(model,c(8,7),best.iter)
```



Question 8

Regression: Marketing data. The data set age_stats315B.csv was already used in Homework 1. Review age_stats315B.txt for the information about order of attributes etc.

(a) Fit a gbm model for predicting age form the other demographic attributes and compare the accuracy with the accuracy of your best single tree from Homework 1.

```
rm(list = ls())
data_path <- paste(getwd(),'/data',sep='')
setwd(data_path)

#Read and type data
age_data <- read.csv('age_stats315B.csv')
factor_columns <- c(
  'Occup',
  'TypeHome',
  'sex',
  'MarStat',
  'DualInc',
  'HouseStat',
  'Ethnic',
  'Lang'
```

```
)
age_data[factor_columns] <- lapply(age_data[factor_columns], as.factor)

#fit a GBM model
set.seed(444) # random for bag.fraction
model_gbm <- gbm(age~., data=age_data, train.fraction=1,
            interaction.depth=4, shrinkage=.05,
            n.trees=2500, bag.fraction=0.5, cv.folds=5,
            distribution="gaussian", verbose=T)
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        2.5440             nan     0.0500    0.1643
##      2        2.3946             nan     0.0500    0.1454
##      3        2.2620             nan     0.0500    0.1329
##      4        2.1393             nan     0.0500    0.1189
##      5        2.0304             nan     0.0500    0.1083
##      6        1.9291             nan     0.0500    0.1016
##      7        1.8394             nan     0.0500    0.0913
##      8        1.7575             nan     0.0500    0.0830
##      9        1.6825             nan     0.0500    0.0728
##     10        1.6141             nan     0.0500    0.0676
##     20        1.1819             nan     0.0500    0.0303
##     40        0.8691             nan     0.0500    0.0058
##     60        0.7719             nan     0.0500    0.0021
##     80        0.7304             nan     0.0500    0.0010
##    100        0.7104             nan     0.0500    0.0006
##    120        0.6990             nan     0.0500    0.0002
##    140        0.6888             nan     0.0500    0.0004
##    160        0.6804             nan     0.0500    0.0001
##    180        0.6732             nan     0.0500    0.0000
##    200        0.6673             nan     0.0500    0.0000
##    220        0.6622             nan     0.0500   -0.0000
##    240        0.6558             nan     0.0500   -0.0001
##    260        0.6517             nan     0.0500   -0.0001
##    280        0.6473             nan     0.0500   -0.0002
##    300        0.6434             nan     0.0500    0.0000
##    320        0.6401             nan     0.0500   -0.0000
##    340        0.6357             nan     0.0500    0.0001
##    360        0.6315             nan     0.0500    0.0000
##    380        0.6284             nan     0.0500   -0.0002
##    400        0.6246             nan     0.0500   -0.0000
##    420        0.6220             nan     0.0500   -0.0001
##    440        0.6189             nan     0.0500   -0.0000
##    460        0.6161             nan     0.0500   -0.0001
##    480        0.6138             nan     0.0500   -0.0001
##    500        0.6108             nan     0.0500   -0.0000
##    520        0.6080             nan     0.0500   -0.0001
##    540        0.6055             nan     0.0500   -0.0001
##    560        0.6027             nan     0.0500   -0.0002
##    580        0.6006             nan     0.0500   -0.0002
##    600        0.5983             nan     0.0500   -0.0001
##    620        0.5961             nan     0.0500   -0.0002
##    640        0.5941             nan     0.0500   -0.0001
##    660        0.5922             nan     0.0500   -0.0001
```

```
##    680        0.5906           nan     0.0500   -0.0001
##    700        0.5890           nan     0.0500   -0.0001
##    720        0.5871           nan     0.0500   -0.0001
##    740        0.5852           nan     0.0500   -0.0002
##    760        0.5834           nan     0.0500   -0.0002
##    780        0.5817           nan     0.0500   -0.0003
##    800        0.5799           nan     0.0500   -0.0001
##    820        0.5780           nan     0.0500   -0.0001
##    840        0.5763           nan     0.0500   -0.0001
##    860        0.5745           nan     0.0500   -0.0002
##    880        0.5730           nan     0.0500   -0.0003
##    900        0.5717           nan     0.0500   -0.0001
##    920        0.5701           nan     0.0500   -0.0002
##    940        0.5687           nan     0.0500   -0.0001
##    960        0.5672           nan     0.0500   -0.0001
##    980        0.5655           nan     0.0500   -0.0002
##   1000        0.5640           nan     0.0500   -0.0000
##   1020        0.5624           nan     0.0500   -0.0000
##   1040        0.5607           nan     0.0500   -0.0002
##   1060        0.5589           nan     0.0500   -0.0002
##   1080        0.5569           nan     0.0500   -0.0001
##   1100        0.5556           nan     0.0500   -0.0001
##   1120        0.5543           nan     0.0500   -0.0001
##   1140        0.5533           nan     0.0500   -0.0001
##   1160        0.5518           nan     0.0500   -0.0001
##   1180        0.5505           nan     0.0500   -0.0001
##   1200        0.5489           nan     0.0500   -0.0001
##   1220        0.5478           nan     0.0500   -0.0001
##   1240        0.5465           nan     0.0500   -0.0002
##   1260        0.5455           nan     0.0500   -0.0001
##   1280        0.5443           nan     0.0500   -0.0002
##   1300        0.5431           nan     0.0500   -0.0003
##   1320        0.5421           nan     0.0500   -0.0001
##   1340        0.5411           nan     0.0500   -0.0002
##   1360        0.5398           nan     0.0500   -0.0001
##   1380        0.5389           nan     0.0500   -0.0002
##   1400        0.5377           nan     0.0500   -0.0002
##   1420        0.5369           nan     0.0500   -0.0001
##   1440        0.5354           nan     0.0500   -0.0002
##   1460        0.5344           nan     0.0500   -0.0003
##   1480        0.5333           nan     0.0500   -0.0001
##   1500        0.5323           nan     0.0500   -0.0001
##   1520        0.5313           nan     0.0500   -0.0001
##   1540        0.5303           nan     0.0500   -0.0001
##   1560        0.5293           nan     0.0500   -0.0001
##   1580        0.5283           nan     0.0500   -0.0001
##   1600        0.5274           nan     0.0500   -0.0001
##   1620        0.5263           nan     0.0500   -0.0002
##   1640        0.5254           nan     0.0500   -0.0001
##   1660        0.5247           nan     0.0500   -0.0001
##   1680        0.5240           nan     0.0500   -0.0002
##   1700        0.5230           nan     0.0500   -0.0001
##   1720        0.5220           nan     0.0500   -0.0001
##   1740        0.5212           nan     0.0500   -0.0002
```

```
##   1760        0.5202             nan     0.0500   -0.0002
##   1780        0.5192             nan     0.0500   -0.0001
##   1800        0.5183             nan     0.0500   -0.0001
##   1820        0.5175             nan     0.0500   -0.0002
##   1840        0.5167             nan     0.0500   -0.0002
##   1860        0.5160             nan     0.0500   -0.0001
##   1880        0.5151             nan     0.0500   -0.0001
##   1900        0.5142             nan     0.0500   -0.0001
##   1920        0.5131             nan     0.0500    0.0001
##   1940        0.5122             nan     0.0500   -0.0000
##   1960        0.5115             nan     0.0500   -0.0001
##   1980        0.5109             nan     0.0500   -0.0001
##   2000        0.5101             nan     0.0500   -0.0002
##   2020        0.5093             nan     0.0500   -0.0002
##   2040        0.5085             nan     0.0500   -0.0002
##   2060        0.5078             nan     0.0500   -0.0001
##   2080        0.5070             nan     0.0500   -0.0002
##   2100        0.5062             nan     0.0500   -0.0001
##   2120        0.5054             nan     0.0500   -0.0001
##   2140        0.5046             nan     0.0500   -0.0001
##   2160        0.5037             nan     0.0500   -0.0002
##   2180        0.5029             nan     0.0500   -0.0002
##   2200        0.5024             nan     0.0500   -0.0002
##   2220        0.5015             nan     0.0500   -0.0001
##   2240        0.5008             nan     0.0500   -0.0001
##   2260        0.5002             nan     0.0500   -0.0002
##   2280        0.4996             nan     0.0500   -0.0002
##   2300        0.4988             nan     0.0500   -0.0001
##   2320        0.4981             nan     0.0500   -0.0001
##   2340        0.4973             nan     0.0500   -0.0001
##   2360        0.4968             nan     0.0500   -0.0001
##   2380        0.4961             nan     0.0500   -0.0001
##   2400        0.4954             nan     0.0500   -0.0001
##   2420        0.4947             nan     0.0500   -0.0002
##   2440        0.4942             nan     0.0500   -0.0002
##   2460        0.4936             nan     0.0500   -0.0001
##   2480        0.4929             nan     0.0500   -0.0002
##   2500        0.4925             nan     0.0500   -0.0003
```

```r
#fit a tree
model_tree <- rpart(age ~ ., data = age_data, method = "anova",
                    control=rpart.control(minbucket = 10,
                                          xval = 10,
                                          maxsurrogate = 5,
                                          usesurrogate = 2,
                                          cp=0.0001))

# Find the minimum cross-validation error + one SD
min_error_window <- min(model_tree$cptable[,"xerror"] + model_tree$cptable[,"xstd"])

# Find the simplest model with xerror within the min_error_window
best_cp <- first(model_tree$cptable[which(model_tree$cptable[,"xerror"] < min_error_window),"CP"])
best_single_tree <- prune(model_tree, cp = best_cp)
```

```r
#Make predictions
y_hat_gbm <- predict(model_gbm, age_data, type="response", n.trees=300)
y_hat_tree <- predict(best_single_tree, age_data)

#Compare errors
y <- age_data$age
MSE_gbm <- sum((y_hat_gbm - y)^2)/length(y)
MSE_tree <- sum((y_hat_tree - y)^2)/length(y)

MSE_gbm
```

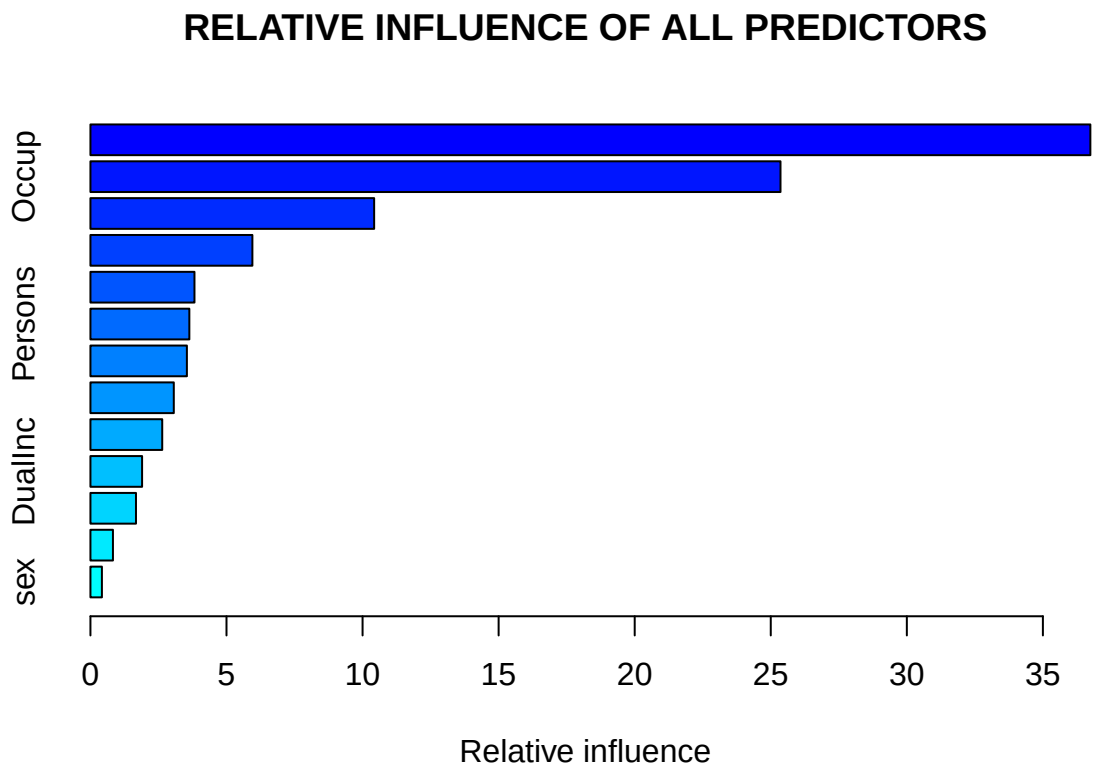
```
## [1] 0.643421
```

```r
MSE_tree
```

```
## [1] 0.6992926
```

(b) Identify the most important variables.

```r
top4 <-summary(model_gbm,main="RELATIVE INFLUENCE OF ALL PREDICTORS")$var[1:4]
```




```r
top4
```

```
## [1] MarStat   Occup     HouseStat Edu
## 13 Levels: DualInc Edu Ethnic HouseStat Income Lang LiveBA ... Under18
```

Question 9

Multiclass classification: marketing data. The data set occup_stats315B.csv comes from the same marketing database used in Homework 1. The description of the attributes can be found in occup_stats315B.txt. The goal in this problem is to fit a gbm model to predict the type of occupation from the 13 other demographic variables.

(a) Report the test set misclassification error for gbm on the data set, and also the misclassification error for each class.

```
rm(list = ls())
data_path <- paste(getwd(),'/data',sep='')
setwd(data_path)

#Read and type data
labels <-c(
  "Occup",
  "TypeHome",
  "sex",
  "MarStat",
  "age",
  "Edu",
  "Income",
  "LiveBA",
  "DualInc",
  "Persons",
  "Under18",
  "HouseStat",
  "Ethnic",
  "Lang"
)
factor_columns <- c(
  'TypeHome',
  'sex',
  'MarStat',
  'Occup',
  'LiveBA',
  'DualInc',
  'HouseStat',
  'Ethnic',
  'Lang'
)

occup_labels <- c(
  "Professional/Managerial",
  "Sales Worker",
  "Factory Worker/Laborer/Driver",
  "Clerical/Service Worker",
  "Homemaker",
  "Student, HS or College",
  "Military",
  "Retired",
  "Unemployed"
```

```
)

#load the data
house_data <- read.csv('occup_stats315B.csv', header=FALSE, sep=",")
colnames(house_data) <- labels
house_data[factor_columns] <- lapply(house_data[factor_columns], as.factor)

#fit GBM model for occupation
set.seed(444) # random for bag.fraction
model <- gbm(Occup ~ ., data=house_data, train.fraction=1,
            interaction.depth=4, shrinkage=.05,
            n.trees=2500, bag.fraction=0.5, cv.folds=5,
            distribution="multinomial", verbose=T)
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        2.1972             nan     0.0500    0.2420
##      2        2.0687             nan     0.0500    0.1803
##      3        1.9725             nan     0.0500    0.1451
##      4        1.8935             nan     0.0500    0.1191
##      5        1.8290             nan     0.0500    0.1030
##      6        1.7743             nan     0.0500    0.0867
##      7        1.7265             nan     0.0500    0.0774
##      8        1.6839             nan     0.0500    0.0674
##      9        1.6473             nan     0.0500    0.0610
##     10        1.6141             nan     0.0500    0.0533
##     20        1.4182             nan     0.0500    0.0187
##     40        1.2869             nan     0.0500    0.0030
##     60        1.2357             nan     0.0500    0.0001
##     80        1.2060             nan     0.0500    0.0001
##    100        1.1844             nan     0.0500   -0.0007
##    120        1.1672             nan     0.0500   -0.0013
##    140        1.1519             nan     0.0500   -0.0009
##    160        1.1362             nan     0.0500   -0.0010
##    180        1.1238             nan     0.0500   -0.0013
##    200        1.1121             nan     0.0500   -0.0009
##    220        1.1006             nan     0.0500   -0.0010
##    240        1.0901             nan     0.0500   -0.0016
##    260        1.0804             nan     0.0500   -0.0011
##    280        1.0706             nan     0.0500   -0.0013
##    300        1.0613             nan     0.0500   -0.0011
##    320        1.0530             nan     0.0500   -0.0012
##    340        1.0449             nan     0.0500   -0.0013
##    360        1.0370             nan     0.0500   -0.0012
##    380        1.0299             nan     0.0500   -0.0013
##    400        1.0228             nan     0.0500   -0.0011
##    420        1.0162             nan     0.0500   -0.0013
##    440        1.0095             nan     0.0500   -0.0013
##    460        1.0026             nan     0.0500   -0.0013
##    480        0.9960             nan     0.0500   -0.0014
##    500        0.9894             nan     0.0500   -0.0014
##    520        0.9835             nan     0.0500   -0.0014
##    540        0.9777             nan     0.0500   -0.0013
##    560        0.9721             nan     0.0500   -0.0013
##    580        0.9663             nan     0.0500   -0.0015
```

```
##    600        0.9610           nan     0.0500   -0.0014
##    620        0.9556           nan     0.0500   -0.0011
##    640        0.9503           nan     0.0500   -0.0015
##    660        0.9449           nan     0.0500   -0.0016
##    680        0.9398           nan     0.0500   -0.0018
##    700        0.9347           nan     0.0500   -0.0011
##    720        0.9297           nan     0.0500   -0.0012
##    740        0.9253           nan     0.0500   -0.0016
##    760        0.9204           nan     0.0500   -0.0012
##    780        0.9157           nan     0.0500   -0.0010
##    800        0.9111           nan     0.0500   -0.0013
##    820        0.9071           nan     0.0500   -0.0015
##    840        0.9029           nan     0.0500   -0.0014
##    860        0.8986           nan     0.0500   -0.0016
##    880        0.8947           nan     0.0500   -0.0014
##    900        0.8908           nan     0.0500   -0.0014
##    920        0.8866           nan     0.0500   -0.0011
##    940        0.8830           nan     0.0500   -0.0012
##    960        0.8789           nan     0.0500   -0.0015
##    980        0.8749           nan     0.0500   -0.0014
##   1000        0.8710           nan     0.0500   -0.0015
##   1020        0.8669           nan     0.0500   -0.0009
##   1040        0.8631           nan     0.0500   -0.0015
##   1060        0.8591           nan     0.0500   -0.0011
##   1080        0.8555           nan     0.0500   -0.0015
##   1100        0.8521           nan     0.0500   -0.0012
##   1120        0.8489           nan     0.0500   -0.0012
##   1140        0.8455           nan     0.0500   -0.0014
##   1160        0.8421           nan     0.0500   -0.0013
##   1180        0.8387           nan     0.0500   -0.0012
##   1200        0.8354           nan     0.0500   -0.0016
##   1220        0.8317           nan     0.0500   -0.0015
##   1240        0.8284           nan     0.0500   -0.0014
##   1260        0.8250           nan     0.0500   -0.0012
##   1280        0.8217           nan     0.0500   -0.0016
##   1300        0.8185           nan     0.0500   -0.0017
##   1320        0.8155           nan     0.0500   -0.0014
##   1340        0.8120           nan     0.0500   -0.0012
##   1360        0.8092           nan     0.0500   -0.0016
##   1380        0.8063           nan     0.0500   -0.0014
##   1400        0.8032           nan     0.0500   -0.0012
##   1420        0.8001           nan     0.0500   -0.0012
##   1440        0.7971           nan     0.0500   -0.0012
##   1460        0.7940           nan     0.0500   -0.0011
##   1480        0.7910           nan     0.0500   -0.0015
##   1500        0.7881           nan     0.0500   -0.0013
##   1520        0.7854           nan     0.0500   -0.0014
##   1540        0.7827           nan     0.0500   -0.0015
##   1560        0.7801           nan     0.0500   -0.0013
##   1580        0.7773           nan     0.0500   -0.0015
##   1600        0.7748           nan     0.0500   -0.0013
##   1620        0.7722           nan     0.0500   -0.0010
##   1640        0.7694           nan     0.0500   -0.0013
##   1660        0.7667           nan     0.0500   -0.0012
```

```
##   1680        0.7642             nan     0.0500   -0.0013
##   1700        0.7616             nan     0.0500   -0.0016
##   1720        0.7588             nan     0.0500   -0.0012
##   1740        0.7561             nan     0.0500   -0.0015
##   1760        0.7537             nan     0.0500   -0.0015
##   1780        0.7513             nan     0.0500   -0.0015
##   1800        0.7489             nan     0.0500   -0.0013
##   1820        0.7464             nan     0.0500   -0.0010
##   1840        0.7439             nan     0.0500   -0.0013
##   1860        0.7414             nan     0.0500   -0.0012
##   1880        0.7387             nan     0.0500   -0.0014
##   1900        0.7363             nan     0.0500   -0.0012
##   1920        0.7341             nan     0.0500   -0.0013
##   1940        0.7320             nan     0.0500   -0.0013
##   1960        0.7295             nan     0.0500   -0.0012
##   1980        0.7272             nan     0.0500   -0.0013
##   2000        0.7249             nan     0.0500   -0.0013
##   2020        0.7224             nan     0.0500   -0.0014
##   2040        0.7202             nan     0.0500   -0.0015
##   2060        0.7179             nan     0.0500   -0.0016
##   2080        0.7156             nan     0.0500   -0.0012
##   2100        0.7135             nan     0.0500   -0.0013
##   2120        0.7114             nan     0.0500   -0.0014
##   2140        0.7093             nan     0.0500   -0.0015
##   2160        0.7075             nan     0.0500   -0.0011
##   2180        0.7053             nan     0.0500   -0.0015
##   2200        0.7031             nan     0.0500   -0.0012
##   2220        0.7012             nan     0.0500   -0.0014
##   2240        0.6989             nan     0.0500   -0.0010
##   2260        0.6968             nan     0.0500   -0.0015
##   2280        0.6948             nan     0.0500   -0.0013
##   2300        0.6927             nan     0.0500   -0.0012
##   2320        0.6906             nan     0.0500   -0.0014
##   2340        0.6887             nan     0.0500   -0.0015
##   2360        0.6865             nan     0.0500   -0.0012
##   2380        0.6845             nan     0.0500   -0.0013
##   2400        0.6826             nan     0.0500   -0.0011
##   2420        0.6804             nan     0.0500   -0.0014
##   2440        0.6787             nan     0.0500   -0.0011
##   2460        0.6769             nan     0.0500   -0.0013
##   2480        0.6751             nan     0.0500   -0.0013
##   2500        0.6731             nan     0.0500   -0.0012
```

```r
#determine misclassification rate
y_hat <- predict(model, house_data, type="response", n.trees=300)
y_hat_max_p <- apply(y_hat,1,which.max)
y <- house_data$Occup
correct <- y == y_hat_max_p

#overall
pct_correct <- sum(correct)/length(correct)
1-pct_correct
```

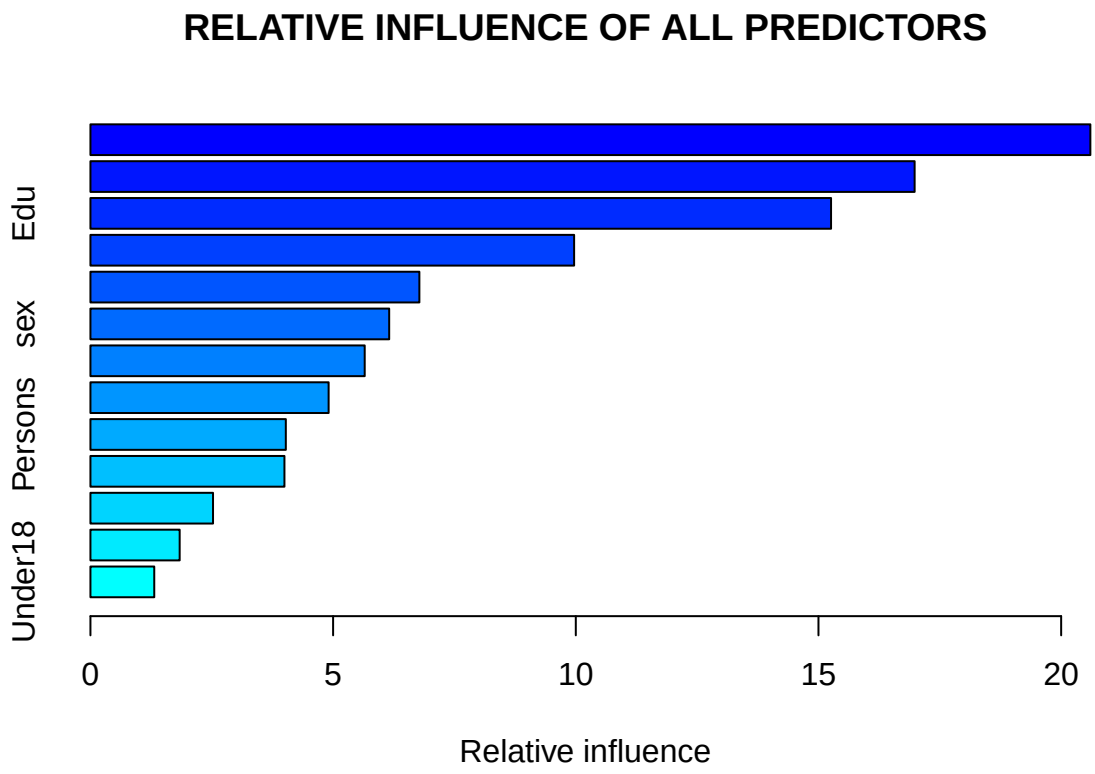
```
## [1] 0.3633284
```

```
for(occupation in levels(house_data$Occup)){
  occupation_i <- strtoi(occupation)
  print(occup_labels[occupation_i])
  correct_in_class <- correct[y==occupation_i]
  pct_correct_in_class <- sum(correct_in_class)/length(correct_in_class)
  print(1-pct_correct_in_class)
}
```

```
## [1] "Professional/Managerial"
## [1] 0.1719085
## [1] "Sales Worker"
## [1] 0.8651399
## [1] "Factory Worker/Laborer/Driver"
## [1] 0.5642384
## [1] "Clerical/Service Worker"
## [1] 0.6520496
## [1] "Homemaker"
## [1] 0.3323398
## [1] "Student, HS or College"
## [1] 0.1545763
## [1] "Military"
## [1] 0.5354331
## [1] "Retired"
## [1] 0.1618911
## [1] "Unemployed"
## [1] 0.6657682
```

(b) Identify the most important variables.

```
top4 <-summary(model,main="RELATIVE INFLUENCE OF ALL PREDICTORS")$var[1:4]
```




```
top4
```

```
## [1] age       Income    Edu       HouseStat
## 13 Levels: age DualInc Edu Ethnic HouseStat Income Lang LiveBA ... Under18
```