HW4 Tianchun Jiang

SUID ending in 0710

tcjiang108@gmail.com

Load required libraries

Problem 1

Chapter 8, Exercise 10 (p. 334).

a)

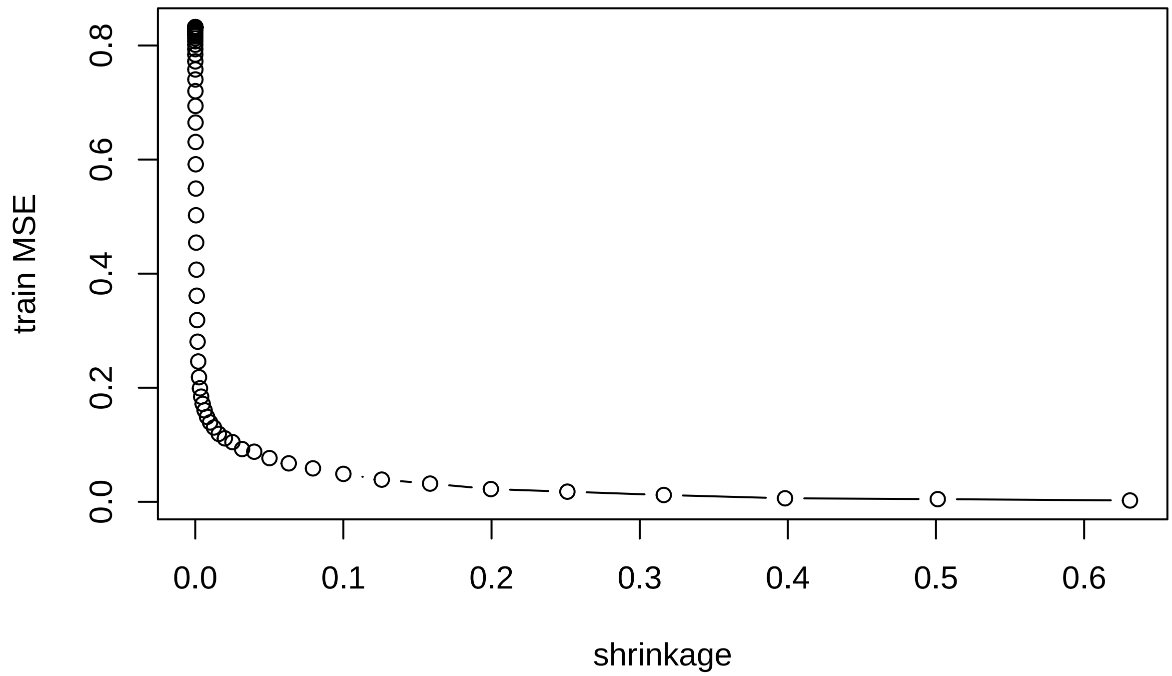
```
clean_Hitters <- Hitters[-which(is.na(Hitters$Salary)), ]
clean_Hitters$Salary_tr <- log(clean_Hitters$Salary)
#drop the non-transformed salary vector
clean_Hitters <- subset(clean_Hitters, select=-c(Salary))
```

b)

```
indices <- 1:200
train_set <- clean_Hitters[indices, ]
test_set <- clean_Hitters[-indices, ]
```

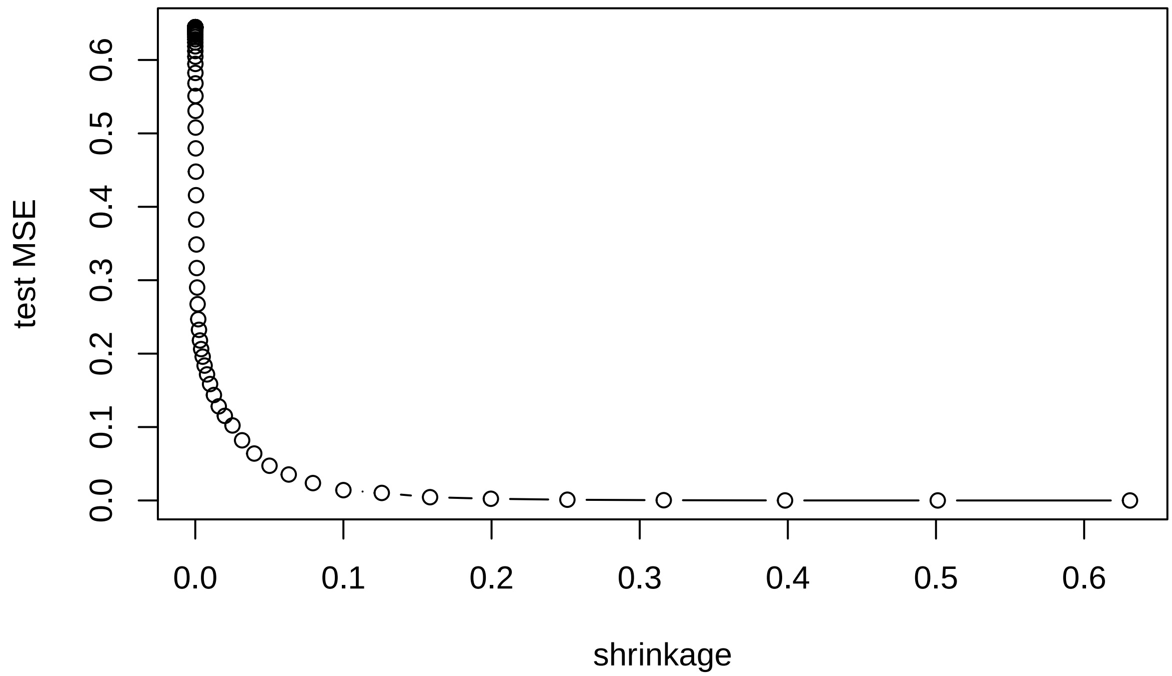
c)

```
set.seed(1)
pows <- seq(-10, -0.2, by=0.1)
lambdas <- 10^pows
train_err <- rep(NA, length(lambdas))
for (i in 1:length(lambdas)) {
    boost_hitters <- gbm(Salary_tr~., data=train_set, distribution="gaussian", n.trees=1000, shrinkage=l
    pred_train <- predict(boost_hitters, train_set, n.trees=1000)
    train_err[i] <- mean((pred_train - train_set$Salary_tr)^2)
}
plot(lambdas, train_err, type = "b", xlab = "shrinkage", ylab = "train MSE")
```



d)

```
set.seed(1)
test_err <- rep(NA, length(lambdas))
for (i in 1:length(lambdas)) {
    boost_hitters <- gbm(Salary_tr~., data=test_set, distribution="gaussian", n.trees=1000, shrinkage=la
    pred_test <- predict(boost_hitters, test_set, n.trees=1000)
    test_err[i] <- mean((pred_test - test_set$Salary_tr)^2)
}
plot(lambdas, test_err, type = "b", xlab = "shrinkage", ylab = "test MSE")
```



e)

```r
fit_slr <- lm(Salary_tr~., data = train_set)
pred <- predict(fit_slr, test_set)
sprintf('test MSE from linear regression: %0.3f', mean((pred - test_set$Salary_tr)^2))
```

```
## [1] "test MSE from linear regression: 0.492"
```

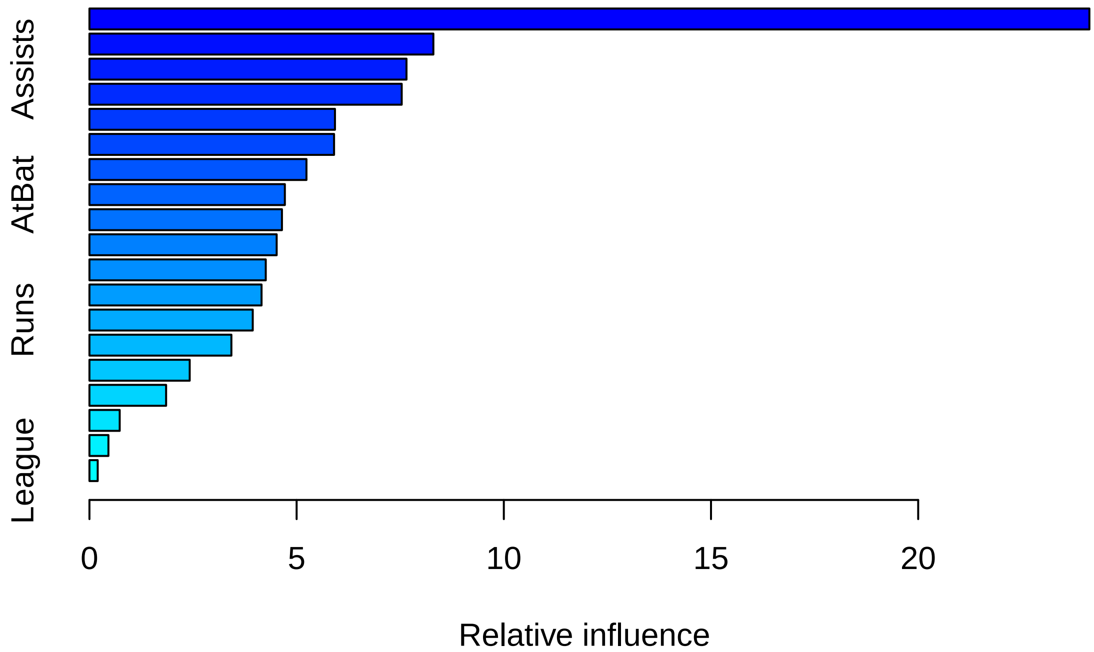
```r
x_train <- model.matrix(Salary_tr~., data = train_set)
x_test <- model.matrix(Salary_tr~., data = test_set)
y <- train_set$Salary_tr
fit <- glmnet(x_train, y, alpha = 0)
pred <- predict(fit, s = 0.01, newx = x_test)
sprintf('test MSE from regularized linear regression: %0.3f', mean((pred - test_set$Salary_tr)^2))
```

```
## [1] "test MSE from regularized linear regression: 0.457"
```

Boosting with shrinknage can dramatically outperform both linear regression and ridge regression in terms of test MSE

f)

```r
res <- gbm(Salary_tr~., data=train_set, distribution='gaussian', n.trees=1000, shrinkage=lambdas[which.r
summary(res)
```




```
##                 var    rel.inf
## CWalks       CWalks 24.1315727
## PutOuts     PutOuts  8.3002881
## Assists     Assists  7.6511486
## CRuns         CRuns  7.5357765
## Hits           Hits  5.9250881
## Walks         Walks  5.9033992
## Years         Years  5.2374697
## AtBat         AtBat  4.7162373
## CHmRun       CHmRun  4.6456138
## CRBI           CRBI  4.5184215
## CAtBat       CAtBat  4.2554463
## RBI             RBI  4.1536482
## Runs           Runs  3.9413883
## HmRun         HmRun  3.4257875
## Errors       Errors  2.4195882
## CHits         CHits  1.8504154
## Division   Division  0.7304689
## NewLeague NewLeague  0.4585250
## League       League  0.1997166
```

CAtBat is the most important predictor in the boosted model, followed by PutOuts, CHmRun and CRuns etc

g)

```r
set.seed(1)
rf_model <- randomForest(Salary_tr~., data=train_set, mtry=13)
pred <- predict(rf_model, test_set)
sprintf('test MSE using bagging: %0.3f', mean((pred - test_set$Salary_tr)^2))
```

```
## [1] "test MSE using bagging: 0.228"
```

Problem 2

Chapter 8, Exercise 11 (p. 335).

a)

```r
indices <- 1:1000
train_set <- Caravan[indices, ]
test_set <- Caravan[-indices, ]
```

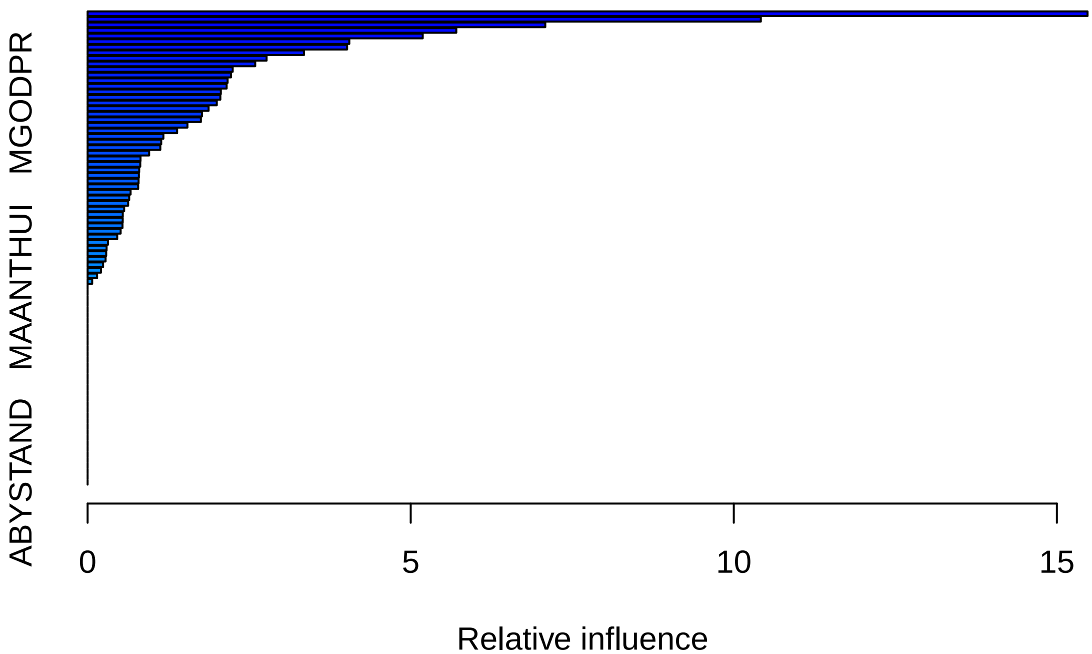
b)

```r
train_set$Purchase <- ifelse(train_set$Purchase == 'Yes', 1, 0)
test_set$Purchase <- ifelse(test_set$Purchase == 'Yes', 1, 0)
mod <- gbm(Purchase~., data=train_set, shrinkage=0.01, n.trees=1000, distribution='bernoulli')
```

```
## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 50: PVRAAUT has no variation.
```

```
## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 71: AVRAAUT has no variation.
```

```r
summary(mod)
```




```
##                var     rel.inf
## PPERSAUT PPERSAUT 15.47472835
## MKOOPKLA MKOOPKLA 10.41857837
## MOPLHOOG MOPLHOOG  7.08318936
## MBERMIDD MBERMIDD  5.70568040
## PBRAND      PBRAND  5.18643840
## ABRAND      ABRAND  4.05007854
## MGODGE      MGODGE  4.01641724
```

```
## MINK3045 MINK3045  3.34831941
## MOSTYPE   MOSTYPE  2.77004683
## MSKC         MSKC  2.59405800
## PWAPART   PWAPART  2.24554255
## MSKA         MSKA  2.21925284
## MAUT2       MAUT2  2.16678641
## MGODOV     MGODOV  2.15159869
## MBERARBG MBERARBG  2.06175867
## MAUT1       MAUT1  2.05431933
## MGODPR     MGODPR  1.99925534
## MBERHOOG MBERHOOG  1.87230335
## PBYSTAND PBYSTAND  1.76949749
## MFWEKIND MFWEKIND  1.75311002
## MRELGE     MRELGE  1.54494356
## MINKGEM   MINKGEM  1.38526632
## MINK4575 MINK4575  1.17307406
## MSKB1       MSKB1  1.13955322
## MGODRK     MGODRK  1.12552314
## MAUT0       MAUT0  0.95267786
## MHKOOP     MHKOOP  0.81875936
## MRELOV     MRELOV  0.81525497
## MFGEKIND MFGEKIND  0.79856580
## MOPLMIDD MOPLMIDD  0.79372187
## APERSAUT APERSAUT  0.78854949
## MBERARBO MBERARBO  0.78274301
## MBERBOER MBERBOER  0.66536577
## MGEMLEEF MGEMLEEF  0.64374651
## MINK7512 MINK7512  0.62783299
## PLEVEN     PLEVEN  0.56597462
## MINKM30   MINKM30  0.54444737
## MSKD         MSKD  0.54357285
## MHHUUR     MHHUUR  0.53962100
## MOSHOOFD MOSHOOFD  0.51124495
## MGEMOMV   MGEMOMV  0.45834621
## MRELSA     MRELSA  0.31480137
## MZFONDS   MZFONDS  0.29403378
## MINK123M MINK123M  0.28805552
## PMOTSCO   PMOTSCO  0.27808732
## MZPART     MZPART  0.23972903
## MFALLEEN MFALLEEN  0.20772270
## MOPLLAAG MOPLLAAG  0.14703706
## MSKB2       MSKB2  0.07078868
## MAANTHUI MAANTHUI  0.00000000
## MBERZELF MBERZELF  0.00000000
## PWABEDR   PWABEDR  0.00000000
## PWALAND   PWALAND  0.00000000
## PBESAUT   PBESAUT  0.00000000
## PVRAAUT   PVRAAUT  0.00000000
## PAANHANG PAANHANG  0.00000000
## PTRACTOR PTRACTOR  0.00000000
## PWERKT     PWERKT  0.00000000
## PBROM       PBROM  0.00000000
## PPERSONG PPERSONG  0.00000000
## PGEZONG   PGEZONG  0.00000000
```

```
## PWAOREG   PWAOREG  0.00000000
## PZEILPL   PZEILPL  0.00000000
## PPLEZIER PPLEZIER  0.00000000
## PFIETS     PFIETS  0.00000000
## PINBOED   PINBOED  0.00000000
## AWAPART   AWAPART  0.00000000
## AWABEDR   AWABEDR  0.00000000
## AWALAND   AWALAND  0.00000000
## ABESAUT   ABESAUT  0.00000000
## AMOTSCO   AMOTSCO  0.00000000
## AVRAAUT   AVRAAUT  0.00000000
## AAANHANG AAANHANG  0.00000000
## ATRACTOR ATRACTOR  0.00000000
## AWERKT     AWERKT  0.00000000
## ABROM       ABROM  0.00000000
## ALEVEN     ALEVEN  0.00000000
## APERSONG APERSONG  0.00000000
## AGEZONG   AGEZONG  0.00000000
## AWAOREG   AWAOREG  0.00000000
## AZEILPL   AZEILPL  0.00000000
## APLEZIER APLEZIER  0.00000000
## AFIETS     AFIETS  0.00000000
## AINBOED   AINBOED  0.00000000
## ABYSTAND ABYSTAND  0.00000000
```

PPERSAUT, MKOOPKLA and MOPLHOOG appear to be the most important predictors

c)

Problem 3

Chapter 9, Exercise 1 (p. 368).

Problem 4

Chapter 9, Exercise 8 (p. 371).