Lab 1

Attila Lazar

14.10.2020

Data

```
#install.packages("ISLR")
data(Hitters,package="ISLR")
data <- na.omit(Hitters)
str(data)
```

```
## 'data.frame':    263 obs. of  20 variables:
##  $ AtBat    : int  315 479 496 321 594 185 298 323 401 574 ...
##  $ Hits     : int  81 130 141 87 169 37 73 81 92 159 ...
##  $ HmRun    : int  7 18 20 10 4 1 0 6 17 21 ...
##  $ Runs     : int  24 66 65 39 74 23 24 26 49 107 ...
##  $ RBI      : int  38 72 78 42 51 8 24 32 66 75 ...
##  $ Walks    : int  39 76 37 30 35 21 7 8 65 59 ...
##  $ Years    : int  14 3 11 2 11 2 3 2 13 10 ...
##  $ CAtBat   : int  3449 1624 5628 396 4408 214 509 341 5206 4631 ...
##  $ CHits    : int  835 457 1575 101 1133 42 108 86 1332 1300 ...
##  $ CHmRun   : int  69 63 225 12 19 1 0 6 253 90 ...
##  $ CRuns    : int  321 224 828 48 501 30 41 32 784 702 ...
##  $ CRBI     : int  414 266 838 46 336 9 37 34 890 504 ...
##  $ CWalks   : int  375 263 354 33 194 24 12 8 866 488 ...
##  $ League   : Factor w/ 2 levels "A","N": 2 1 2 2 1 2 1 2 1 1 ...
##  $ Division : Factor w/ 2 levels "E","W": 2 2 1 1 2 1 2 2 1 1 ...
##  $ PutOuts  : int  632 880 200 805 282 76 121 143 0 238 ...
##  $ Assists  : int  43 82 11 40 421 127 283 290 0 445 ...
##  $ Errors   : int  10 14 3 4 25 7 9 19 0 22 ...
##  $ Salary   : num  475 480 500 91.5 750 ...
##  $ NewLeague: Factor w/ 2 levels "A","N": 2 1 2 2 1 1 1 2 1 1 ...
##  - attr(*, "na.action")=Class 'omit'  Named int [1:59] 1 16 19 23 31 33 37 39 40 42 ...
##   .. ..- attr(*, "names")= chr [1:59] "-Andy Allanson" "-Billy Beane" "-Bruce Bochte" "-Bob Boone" .
```

The data contains categorical variables with 2 levels. Since these are represented by numbers in R, we do not have to transfer them.

then we split the data in training and test sets

```
# separate data in train and test
set.seed(123)
train_i <- sample(seq_len(nrow(data)), size = floor(2/3 * nrow(data)))
train <- data[train_i, ]
test <- data[-train_i, ]
```

1 Full model

a)

We compute the LS estimator for the training set

```r
y <- train$Salary
intercept <- rep(1, nrow(train))
x <- data.matrix(train[, !names(train) %in% c("Salary")])
X <- cbind(intercept, x)
thetaH <- solve(t(X) %*% X) %*% t(X) %*% y
thetaH
```

```
##                   [,1]
## intercept  343.6102621
## AtBat       -1.3089304
## Hits         4.7088174
## HmRun        8.0741383
## Runs        -3.6999318
## RBI         -0.2789729
## Walks        5.1967844
## Years      -21.1672442
## CAtBat      -0.2672993
## CHits        0.6740308
## CHmRun      -0.9376378
## CRuns        1.6467722
## CRBI         0.4432463
## CWalks      -0.5422965
## League     116.8692469
## Division  -115.7803210
## PutOuts      0.3644510
## Assists      0.8029889
## Errors     -12.3160039
## NewLeague  -74.9226215
```

b)

We use *model.matrix()* to calculate the LS estimator

```r
X2 <- model.matrix(Salary~., data=train)
thetaH2 <- solve(t(X2) %*% X2) %*% t(X2) %*% y
thetaH2
```

```
##                     [,1]
## (Intercept)  269.7765665
## AtBat         -1.3089304
## Hits           4.7088174
## HmRun          8.0741383
## Runs          -3.6999318
## RBI           -0.2789729
## Walks          5.1967844
## Years        -21.1672442
## CAtBat        -0.2672993
## CHits          0.6740308
## CHmRun        -0.9376378
## CRuns          1.6467722
## CRBI           0.4432463
## CWalks        -0.5422965
## LeagueN      116.8692469
## DivisionW   -115.7803210
```

```
## PutOuts         0.3644510
## Assists         0.8029889
## Errors        -12.3160039
## NewLeagueN    -74.9226215
```

We get the same coeficients except for the intercept

c)

We calculate the estimator using *lm()*

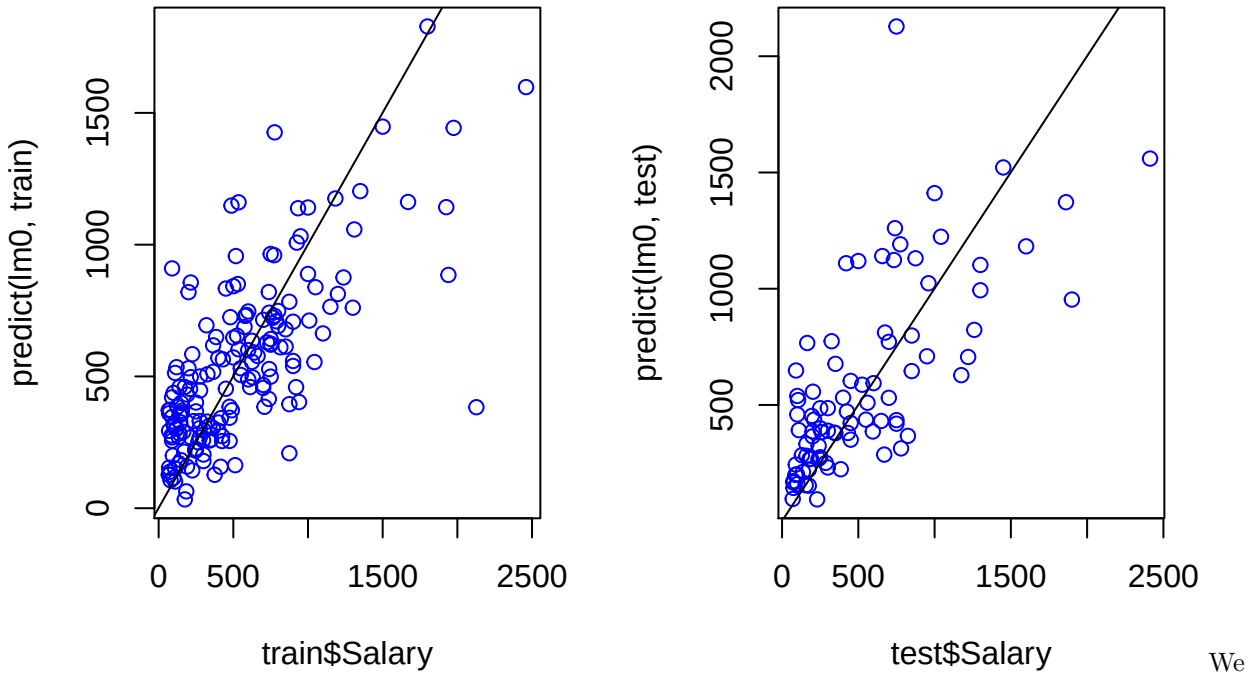
```
lm0 <- lm(Salary~., data=train)
summary(lm0)
```

```
##
## Call:
## lm(formula = Salary ~ ., data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -819.90 -184.10  -10.31  123.41 1744.18
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  269.7766   114.4115   2.358 0.019625 *
## AtBat         -1.3089     0.8505  -1.539 0.125856
## Hits           4.7088     3.3634   1.400 0.163508
## HmRun          8.0741     8.1638   0.989 0.324196
## Runs          -3.6999     3.8629  -0.958 0.339655
## RBI           -0.2790     3.6275  -0.077 0.938798
## Walks          5.1968     2.4762   2.099 0.037468 *
## Years        -21.1672    16.4114  -1.290 0.199046
## CAtBat        -0.2673     0.2012  -1.329 0.185912
## CHits          0.6740     1.0926   0.617 0.538188
## CHmRun        -0.9376     2.2543  -0.416 0.678028
## CRuns          1.6468     1.0187   1.617 0.108014
## CRBI           0.4432     1.0365   0.428 0.669500
## CWalks        -0.5423     0.4685  -1.157 0.248850
## LeagueN      116.8692   102.5820   1.139 0.256346
## DivisionW   -115.7803    51.9515  -2.229 0.027278 *
## PutOuts        0.3645     0.1005   3.628 0.000387 ***
## Assists        0.8030     0.3138   2.559 0.011458 *
## Errors       -12.3160     6.3718  -1.933 0.055073 .
## NewLeagueN   -74.9226   100.2536  -0.747 0.455996
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 326.2 on 155 degrees of freedom
## Multiple R-squared:   0.52,  Adjusted R-squared:  0.4612
## F-statistic: 8.839 on 19 and 155 DF,  p-value: < 2.2e-16
```

We select all variables with p-Value less than 0.1. For our reduced model we will select the variables “Walks”, “Division”, “PutOuts”, “Assists”, “Errors”

d)

```
par(mfrow=c(1,2))
plot(train$Salary, predict(lm0, train), col='blue')
abline(1,1)
plot(test$Salary, predict(lm0, test) ,col='blue')
abline(1,1)
```



We would expect the estimator to perform better on the training data but visually the results look vrey simular.

e)

MSE training data

```
mean((train$Salary - predict(lm0, train))^2)
```

```
## [1] 94222.08
```

MSE test data

```
mean((test$Salary - predict(lm0, test))^2)
```

```
## [1] 114941.3
```

As expected MSE is smaller (better) for the training data.

2 Reduced model

We compute the estimator with the training set

```
x <- data.matrix(train[, names(train) %in% c("Walks", "Division", "PutOuts", "Assists", "Errors")])
intercept <- rep(1, nrow(train))
X <- cbind(intercept, x)
```

```r
thetaH <- solve(t(X) %*% X) %*% t(X) %*% y
thetaH
```

```
##                   [,1]
## intercept  409.1036908
## Walks        6.8627023
## Division  -151.1972374
## PutOuts      0.4189772
## Assists      0.8320387
## Errors     -17.5567537
```

a)

```r
redmodel <- lm(Salary~Walks+Division+PutOuts+Assists+Errors+1, train)
summary(redmodel)
```

```
##
## Call:
## lm(formula = Salary ~ Walks + Division + PutOuts + Assists +
##     Errors + 1, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -862.37 -243.55  -42.93  155.19 1759.04
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  257.9065    77.2857   3.337 0.001041 **
## Walks          6.8627     1.4072   4.877 2.47e-06 ***
## DivisionW   -151.1972    57.1429  -2.646 0.008915 **
## PutOuts        0.4190     0.1091   3.842 0.000173 ***
## Assists        0.8320     0.3156   2.636 0.009164 **
## Errors       -17.5568     6.7537  -2.600 0.010160 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 377.3 on 169 degrees of freedom
## Multiple R-squared:  0.2998, Adjusted R-squared:  0.2791
## F-statistic: 14.47 on 5 and 169 DF,  p-value: 8.491e-12
```

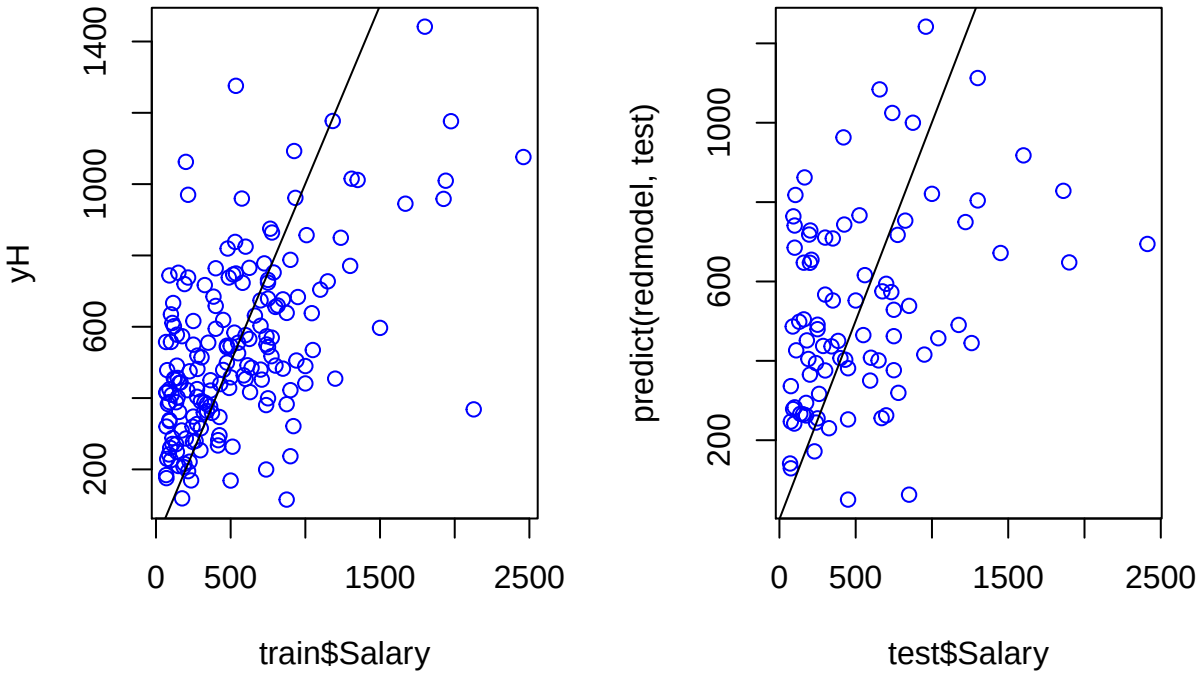
All variables seems to be significant in our reduced model.

b)

```r
par(mfrow=c(1,2))

yH <- X %*% thetaH
plot(train$Salary, yH, col='blue')
abline(1,1)

plot(test$Salary, predict(redmodel, test), col='blue')
abline(1,1)
```



c)

```
mean((train$Salary - yH)^2)
```

```
## [1] 137451.5
```

```
mean((test$Salary - predict(redmodel, test))^2)
```

```
## [1] 189249
```

We would expect the MSE from the reduced Model to be smaller but in fact is bigger then using the full model.

d

```
anova(redmodel, lm0)
```

```
## Analysis of Variance Table
##
## Model 1: Salary ~ Walks + Division + PutOuts + Assists + Errors + 1
## Model 2: Salary ~ AtBat + Hits + HmRun + Runs + RBI + Walks + Years +
##     CAtBat + CHits + CHmRun + CRuns + CRBI + CWalks + League +
##     Division + PutOuts + Assists + Errors + NewLeague
##   Res.Df      RSS Df Sum of Sq      F    Pr(>F)
## 1    169 24054016
## 2    155 16488864 14   7565152 5.0796 8.817e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As expected anove selects the full model and rejects the reduced model