R Lab #4 – Solutions

---

1. 6.5.1 Best Subset Selection

Use the regsubsets function in the leaps library to fit up to a 19-variable multiple linear regression model to the Hitters ISLR data using Salary as the response variable. The default method argument for the regsubsets function is set to "exhaustive" which performs an exhaustive search (called "best subset selection" in the book). Select the other two variables that are included in the best nine-variable model along with AtBat, Hits, Walks, CRuns, CRBI, DivisionW, and PutOuts.

a) CAtBat
b) Chits
c) CHmRun
d) CWalks
e) Assists

Solution: A, D

```
coef(regfit.full, 9)
```

2. 6.5.1 Best Subset Selection

Use the regsubsets function in the leaps library to fit up to a 19-variable multiple linear regression model to the Hitters data using Salary as the response variable. The default method argument for the regsubsets function is set to "exhaustive" which performs an exhaustive search (called "best subset selection" in the book). What are the values of adjusted R^2, C_p, and BIC for the best nine-variable model?

- Adjusted R^2 (round to 3 decimal places): _____
- C_p (round to 2 decimal places): _____
- BIC (round to nearest whole number): _____

Solution: 0.518, 6.16, -145

```
reg.summary$adjr2[9]
reg.summary$cp[9]
reg.summary$bic[9]
```

3. 6.5.2 Forward and Backward Stepwise Selection

Use the regsubsets function in the leaps library to perform forward stepwise and backward stepwise selection for the Hitters data using Salary as the response variable. You'll need to set the method argument appropriately in the regsubsets function. True or False? The best eight-variable models identified by forward stepwise selection and backward stepwise selection are identical?

Solution: True

```
coef(regfit.fwd, 8)
coef(regfit.bwd, 8)
```

4. 6.5.2 Forward and Backward Stepwise Selection

Use the regsubsets function in the leaps library to perform "sequential replacement" variable selection (a hybrid of forward and backward stepwise selection) for the Hitters data using Salary as the response variable. You'll need to set the method argument appropriately in the regsubsets function (type ?regsubsets to see what to set it to). Select the other two variables that are included in the resulting best seven-variable model along with Hits, Walks, CWalks, DivisionW, and PutOuts.

a) AtBat
b) CAtBat
c) Chits
d) CRuns
e) CRBI

Solution: A, D

```
regfit.seq <- regsubsets(Salary ~ ., data = Hitters, nvmax = 19, method = "seqrep")
summary(regfit.seq)
coef(regfit.seq, 7)
```

5. 6.5.3 Choosing Among Models Using the Validation Set Approach

Complete all the steps detailed in the Lab on pages 248 and 249 up to and including the definition of the predict.regsubsets function in the middle of page 249. We can use this function in place of the first two lines of the loop at the bottom of page 248/top of page 249 (the lines beginning coefi= and pred=). Which of the following lines of code is the correct single line of replacement code for these two lines in the loop? (The third line beginning val.errors[i]= should remain in the loop unchanged.) You can answer this by trying all the lines of code to see which doesn't return an error message and provides the same results for val.errors as in the Lab.

a) pred=predict(regfit.best, Hitters[train,],id)
b) pred=predict(regfit.best,Hitters[train,],i)
c) pred=predict(regfit.best,Hitters[test,],id)
d) pred=predict(regfit.best,Hitters[test,],i)
e) pred=predict(regfit.best,Hitters,i)

Solution: D

```
val.errors <- rep(NA, 19)
for(i in 1:19){
  pred <- predict(regfit.best, Hitters[test, ], i)
  val.errors[i] <- mean((Hitters$Salary[test] - pred)^2)
}
val.errors
coef(regfit.best, 3)
```

6. 6.5.3 Choosing Among Models Using the Validation Set Approach

Complete all the steps detailed in the Lab on page 250, which results in 10-fold cross-validation selecting an 11-variable model (since this has the lowest test MSE). Then re-run the exact same steps except with 5-fold cross-validation instead of 10-fold cross-validation. Use the same random seed of 1. How many variables are in the model that 5-fold cross-validation selects (your answer should be an integer)?

Solution: 10

```
k <- 5
set.seed(1)
folds <- sample(1:k, nrow(Hitters), replace = T)
cv.errors <- matrix(NA, k, 19, dimnames = list(NULL, paste(1:19)))

for (j in 1:k) {
  best.fit = regsubsets(Salary ~ ., data = Hitters[folds != j, ], nvmax = 19)
  for (i in 1:19) {
    pred = predict(best.fit, Hitters[folds == j, ], id = i)
    cv.errors[j, i] = mean((Hitters$Salary[folds == j] - pred)^2)
  }
}
mean.cv.errors <- apply(cv.errors, 2, mean)
which.min(mean.cv.errors)
```

7. 6.6.1 Ridge Regression

Follow the directions at the top of page 253 for splitting the dataset into training and test sets to estimate test error. MSE on the test set for $\lambda=4$ is 101037. What is MSE on the test set for $\lambda=50$ (round to the nearest whole number and don't include any commas in your answer)?

Solution: 97015

```
ridge.mod <- glmnet(x[train,], y[train], alpha = 0, lambda = grid, thresh = 1e-12)
ridge.pred <- predict(ridge.mod, s = 50, newx = x[test,])
mean((ridge.pred - y.test)^2) # Test MSE = 97015.36
```

8. 6.6.1 Ridge Regression

Follow the directions on page 251 for fitting a ridge regression model to the Hitters dataset. Once you've fit the ridge regression model enter plot(ridge.mod, xvar="lambda", label=T) to view the coefficient paths as log(λ) varies (the least squares estimates are on the left-hand side of the graph when $\lambda \approx 0$). The numbers on the graph label the paths according to the predictor. Which three predictor numbers have the largest (absolute) least squares coefficient estimates (and thus are shrunk the most as λ increases)? (Select all 3.)

a) 7
b) 14
c) 15
d) 18
e) 19

Solution: B, C, E

```
ridge.mod <- glmnet(x, y, alpha = 0, lambda = grid)
plot(ridge.mod, xvar = "lambda", label = TRUE)
```

9. 6.6.1 Ridge Regression

Follow the directions on page 254 for using cross-validation to choose λ. The value of λ that minimizes the cross-validation error is $\lambda = 212$, which results in a test MSE of 96016. The function cv.glmnet can also return the largest value of λ such that the cross-validation error is within 1 standard error of the minimum (type ?cv.glmnet to find out how). Sometimes (although not in this case) this can result in improved test prediction accuracy. What is this value of λ and what is the corresponding test MSE?

- Largest value of λ such that the cross-validation error is within 1 standard error of the minimum (round to the nearest who number and do not include any commas in your answer): _____
- Corresponding test MSE (round to the nearest who number and do not include any commas in your answer): _____

Solution: 7972, 149048

```
?cv.glmnet
ridge.mod <- glmnet(x[train,], y[train], alpha = 0, lambda = grid, thresh = 1e-12)
set.seed(1)
cv.out <- cv.glmnet(x[train,], y[train], alpha = 0)
largelam <- cv.out$lambda.1se
largelam # lambda = 7971.935 = 7972
ridge.mod <- glmnet(x[train,], y[train], alpha = 0, lambda = grid, thresh = 1e-12)
ridge.pred <- predict(ridge.mod, s = largelam, newx = x[test,])
```

```
mean((ridge.pred - y.test)^2) # test MSE = 149047.8 = 149048
```

10. 6.6.2 The Lasso

Follow the directions on page 255 for using cross-validation to choose λ. The value of λ that minimizes the cross-validation error is λ = 16.8, which results in a test MSE of 100743. True or False? This lasso model has worse test prediction accuracy than the ridge regression model that used a value of λ = 212, but this lasso model could be considered easier to interpret since it involves only a subset of the predictors.

Solution: True

```
set.seed(1)
cv.out <- cv.glmnet(x[train,], y[train], alpha = 1)
plot(cv.out)
bestlam <- cv.out$lambda.min
bestlam # 16.78016
lasso.mod <- glmnet(x[train,], y[train], alpha = 1, lambda = grid)
lasso.pred <- predict(lasso.mod, s = bestlam, newx = x[test,])
mean((lasso.pred - y.test)^2) # 100743.4
out <- glmnet(x, y, alpha = 1,lambda = grid)
lasso.coef <- predict(out, type = "coefficients", s = bestlam)[1:20,]
lasso.coef
lasso.coef[lasso.coef != 0]
```

11. 6.6.2 The Lasso

By extracting the appropriate output value from cv.out defined on page 255, you should find that the largest value of λ such that the cross-validation error is within 1 standard error of the minimum is λ = 129.9 with corresponding test MSE of 142495. You should also find that this model contains only five predictor variables. Select the other three variables that are included in this five variable model along with Hits and Walks.

a) CRuns
b) CRBI
c) LeagueN
d) DivisionW
e) PutOuts

Solution: A, B, E

```
set.seed(1)
cv.out <- cv.glmnet(x[train,], y[train], alpha = 1)
largelam <- cv.out$lambda.1se
```

```
largelam # 129.9227
lasso.mod <- glmnet(x[train,], y[train], alpha = 1, lambda = grid)
lasso.pred <- predict(lasso.mod, s = largelam, newx = x[test,])
mean((lasso.pred - y.test)^2) # test MSE = 142495.4
out <- glmnet(x, y, alpha = 1, lambda = grid)
lasso.coef <- predict(out, type = "coefficients", s = largelam)[1:20,]
lasso.coef
lasso.coef[lasso.coef != 0]
```

12. 6.7.1 Principal Components Regression

Apply PCR to the Hitters data by following the directions on page 256. You should find that the cross-validated root mean squared errors are 348.9 for M=1, 352.2 for M=2, etc. What are the cross-validated root mean squared errors for M=5 and M=6? (Report your answers to exactly one decimal place.)

- M=5: _____
- M=6: _____

Solution: 350.1, 349.1

```
set.seed(2)
pcr.fit <- pcr(Salary ~ ., data = Hitters, scale = TRUE, validation = "CV")
summary(pcr.fit)
```

13. 6.7.1 Principal Components Regression

After using the validationplot function at the bottom of page 256 you should see that the smallest CV mean squared error occurs when M=16 PCR components are used. To find the actual values of the CV error, type MSEP(pcr.fit). You should find that the CV error for M=16 is 120838. The CV error for M=1 isn't too much larger than this. What is it exactly (do not include any commas in your answer)?

Solution: 121735

```
MSEP(pcr.fit)
```

14. 6.7.1 Principal Components Regression

Next perform PCR on the training data by following the directions on page 257. Confirm that the lowest CV error occurs for M=7 and that the test MSE for M=7 is 96556. Is it true or false that the test MSE for M=8 is larger than this?

Solution: True

```
set.seed(1)
pcr.fit <- pcr(Salary ~ ., data = Hitters, subset = train, scale = TRUE, validation = "CV")
pcr.pred <- predict(pcr.fit, x[test,], ncomp = 8)
mean((pcr.pred - y.test)^2) # 102538.1
```

15. 6.7.1 Principal Components Regression

Fit PCR using M=7 on the full dataset by following the directions at the bottom of page 257. Then create plots of the response variable versus the predictions for M=1, 2, ..., 7 by entering plot(pcr.fit, ncomp=1:7). What is the best description of these plots for this application?

a) As M increases from 1 to 7 the linear association in the plots remains relatively weak and approximately the same (i.e., all the plots appear similarly scattered).
b) As M increases from 1 to 7 the linear association in the plots goes from relatively weak for M=1 to relatively strong for M=7.
c) As M increases from 1 to 7 the linear association in the plots goes from relatively strong for M=1 to relatively weak for M=7.
d) As M increases from 1 to 7 the linear association in the plots remains relatively strong and approximately the same (i.e., all the plots appear similarly strongly linear).

Solution: A

```
pcr.fit <- pcr(y ~ x, scale = T, ncomp = 7)
summary(pcr.fit)
plot(pcr.fit, ncomp = 1:7)
# the plots are all quite similar with very little evidence of any linear association.
```

16. 6.7.2 Partial Least Squares

Apply PLS to the training data by following the directions at the top of page 258. You should find that the cross-validated root mean squared errors are 394.2 for M=1, 391.5 for M=2, etc. What are the cross-validated root mean squared errors for M=5 and M=6? (Report your answer to exactly one decimal place.)

- M=5: _____
- M=6: _____

Solution: 415.0, 424.0

```
set.seed(1)
pls.fit <- plsr(Salary ~ ., data = Hitters, subset = train, scale = T, validation = "CV")
summary(pls.fit)
```

17. 6.7.2 Partial Least Squares

After using the validationplot function you should see that the smallest CV mean squared error occurs when M=2 PLS directions are used. To find the actual values of the CV error, type MSEP(pls.fit). You should find that the CV error for M=2 is 153298. The CV error for M=3 isn't too much larger than this. What is it exactly (do not include any commas in your answer)?

Solution:154496

```
set.seed(1)
pls.fit <- plsr(Salary ~ ., data = Hitters, subset = train, scale = T, validation = "CV")
MSEP(pls.fit)
```

18. 6.7.2 Partial Least Squares

Next compute the test MSE for M=2 by following the directions (you should find it is 101417). Is it true or false that the test MSE for M=3 is larger than this?

Solution: False

```
pls.pred <- predict(pls.fit, x[test,], ncomp = 3)
mean((pls.pred - y.test)^2) # 100861.5
```

19. 6.7.2 Partial Least Squares

Fit PLS using M=2 on the full dataset by following the directions at the bottom of page 258. Confirm that this model fit explains 46.40% of the variance in Salary, almost as much as that explained using the final seven-component model PCR fit, 46.69%.

True or false? The percentage of variance in Salary explained by a three-component PLS fit is even greater than 46.69%.

Solution: True

```
pls.fit <- plsr(Salary ~ ., data = Hitters, scale = T, ncomp = 3)
summary(pls.fit)
# TRAINING: % variance explained
#         1 comps  2 comps  3 comps
# X         38.08    51.03    65.98
# Salary    43.05    46.40    47.72
```