> library(ISLR)

> College <- College

>

> set.seed(1)

> train <- sample(1:nrow(College), round(nrow(College))/2)

> train.set <- College[train,]

> test.set <- College[-train,]

> x.test <- test.set[,-2]

> y.test <- test.set[,2]

> x.train <- train.set[,-2]

> y.train <- train.set[,2]

>

> ### Subset Selection

> library(leaps)

> regfit.full <- regsubsets(Apps~., data=College)

> summary(regfit.full)

Subset selection object

Call: regsubsets.formula(Apps ~ ., data = College)

17 Variables (and intercept)

Forced in Forced out

PrivateYes FALSE FALSE

Accept FALSE FALSE

Enroll FALSE FALSE

Top10perc FALSE FALSE

Top25perc FALSE FALSE

F.Undergrad FALSE FALSE

P.Undergrad FALSE FALSE

Outstate FALSE FALSE

Room.Board FALSE FALSE

Books FALSE FALSE

Personal FALSE FALSE

PhD FALSE FALSE

Terminal FALSE FALSE

S.F.Ratio FALSE FALSE

perc.alumni FALSE FALSE

Expend FALSE FALSE

Grad.Rate FALSE FALSE

1 subsets of each size up to 8

Selection Algorithm: exhaustive

PrivateYes Accept Enroll Top10perc Top25perc F.Undergrad P.Undergrad Outstate Room.Board Books

1 ( 1 ) " " "\*" " " " " " " " " " " " " " " " "

2 ( 1 ) " " "\*" " " "\*" " " " " " " " " " " " "

3 ( 1 ) " " "\*" " " "\*" " " " " " " " " " " " "

4 ( 1 ) " " "\*" " " "\*" " " " " " " "\*" " " " "

5 ( 1 ) " " "\*" "\*" "\*" " " " " " " "\*" " " " "

6 ( 1 ) " " "\*" "\*" "\*" " " " " " " "\*" "\*" " "

7 ( 1 ) " " "\*" "\*" "\*" "\*" " " " " "\*" "\*" " "

8 ( 1 ) "\*" "\*" "\*" "\*" " " " " " " "\*" "\*" " "

Personal PhD Terminal S.F.Ratio perc.alumni Expend Grad.Rate

1 ( 1 ) " " " " " " " " " " " " " "

2 ( 1 ) " " " " " " " " " " " " " "

3 ( 1 ) " " " " " " " " " " "\*" " "

4 ( 1 ) " " " " " " " " " " "\*" " "

5 ( 1 ) " " " " " " " " " " "\*" " "

6 ( 1 ) " " " " " " " " " " "\*" " "

7 ( 1 ) " " " " " " " " " " "\*" " "

8 ( 1 ) " " "\*" " " " " " " "\*" " "

>

> regfit.full <- regsubsets(Apps~., data=College, nvmax=19)

> reg.summary.full <- summary(regfit.full)

> names(reg.summary.full)

[1] "which" "rsq" "rss" "adjr2" "cp" "bic" "outmat" "obj"

> reg.summary.full$rsq

[1] 0.8900990 0.9157839 0.9183356 0.9212640 0.9237599 0.9247464 0.9257649 0.9268725 0.9276780 0.9283103

[11] 0.9288011 0.9289945 0.9291223 0.9291632 0.9291878 0.9291885 0.9291887

> reg.summary.full$adjr2

[1] 0.8899572 0.9155663 0.9180186 0.9208560 0.9232655 0.9241600 0.9250892 0.9261108 0.9268294 0.9273744

[11] 0.9277773 0.9278792 0.9279147 0.9278617 0.9277921 0.9276978 0.9276027

>

> par(mfrow=c(2,2))

> plot(reg.summary.full$rsq, xlab="Number of regressors", ylab="R-square", type="l")

>

> plot(reg.summary.full$adjr2, xlab="Number of regressors", ylab="Adjusted R-square", type="l")

> a <- which.max(reg.summary.full$adjr2)

> points(a, reg.summary.full$adjr2[a], col="red", cex=2, pch=20)

>

> plot(reg.summary.full$cp, xlab="Number of regressors", ylab="Cp", type="l")

> a1 <- which.min(reg.summary.full$cp)

> points(a1, reg.summary.full$cp[a1], col="red", cex=2, pch=20)

>

> plot(reg.summary.full$bic, xlab="Number of regressors", ylab="BIC", type="l")

> a2 <- which.min(reg.summary.full$bic)

> points(a2, reg.summary.full$bic[a2], col="red", cex=2, pch=20)

>

> plot(regfit.full, scale="r2")

> plot(regfit.full, scale="adjr2")

> plot(regfit.full, scale="Cp")

> plot(regfit.full, scale="bic")

>

> coef(regfit.full, 12)

(Intercept) PrivateYes Accept Enroll Top10perc Top25perc F.Undergrad

-157.28685883 -511.78760196 1.58691470 -0.88265385 50.41131660 -14.74735373 0.05945481

P.Undergrad Outstate Room.Board PhD Expend Grad.Rate

0.04593068 -0.09017643 0.14776586 -10.70502848 0.07246655 8.63961002

>

> reg.fit <- lm(Apps~Private+Accept+Enroll+Top10perc+Top25perc+F.Undergrad+P.Undergrad+Outstate+Room.Board+PhD+Expend+Grad.Rate, data=College)

> reg.pred <- predict(reg.fit, x.test)

>

> sub.msep <- mean((reg.pred-y.test)^2)

>

>

> ### Shrinkage Method: Ridge

> library(glmnet)

> x.temp <- model.matrix(Apps~., College)

>

> head(x.temp)

(Intercept) PrivateYes Accept Enroll Top10perc Top25perc F.Undergrad

Abilene Christian University 1 1 1232 721 23 52 2885

Adelphi University 1 1 1924 512 16 29 2683

Adrian College 1 1 1097 336 22 50 1036

Agnes Scott College 1 1 349 137 60 89 510

Alaska Pacific University 1 1 146 55 16 44 249

Albertson College 1 1 479 158 38 62 678

P.Undergrad Outstate Room.Board Books Personal PhD Terminal S.F.Ratio

Abilene Christian University 537 7440 3300 450 2200 70 78 18.1

Adelphi University 1227 12280 6450 750 1500 29 30 12.2

Adrian College 99 11250 3750 400 1165 53 66 12.9

Agnes Scott College 63 12960 5450 450 875 92 97 7.7

Alaska Pacific University 869 7560 4120 800 1500 76 72 11.9

Albertson College 41 13500 3335 500 675 67 73 9.4

perc.alumni Expend Grad.Rate

Abilene Christian University 12 7041 60

Adelphi University 16 10527 56

Adrian College 30 8735 54

Agnes Scott College 37 19016 59

Alaska Pacific University 2 10922 15

Albertson College 11 9727 55

> x <- x.temp[,-2]

> y <- College$Apps

>

> grid <- 10^seq(10, -2, length=100)

> ridge.mod <- glmnet(x, y, alpha=0, lambda=grid)

>

> dim(coef(ridge.mod))

[1] 18 100

>

> # Cross validation to choose lambda

> train <- sample(1:nrow(x), round(nrow(x)/2))

> y.train1 <- y[train]

> x.train1 <- x[train,]

> y.test1 <- y[-train]

> x.test1 <- x[-train,]

>

> ridge.mod <- glmnet(x.train1, y.train1, alpha=0, lambda=grid, thresh=1e-12)

> ridge.pred <- predict(ridge.mod, s=4, newx=x.test1)

> ridge.msep <- mean((ridge.pred - y.test)^2)

>

> ridge.pred <- predict(ridge.mod, s=0, newx=x.test1, exact=TRUE, x=x.train1, y=y.train1)

> mean((ridge.pred - y.test1)^2)

[1] 1733471

>

> cv.out <- cv.glmnet(x.train1, y.train1, alpha=0)

> plot(cv.out)

> bestlam <- cv.out$lambda.min

> bestlam

[1] 380.8738

>

> ridge.pred <- predict(ridge.mod, s=bestlam, newx=x.test1)

> ridge.msep <- mean((ridge.pred - y.test1)^2)

>

>

>

> ### Dimensional Reduction

> library(pls)

> pcr.fit <- pcr(Apps~., data=College, scale=TRUE, validation="CV")

> summary(pcr.fit)

Data: X dimension: 777 17

Y dimension: 777 1

Fit method: svdpc

Number of components considered: 17

VALIDATION: RMSEP

Cross-validated using 10 random segments.

(Intercept) 1 comps 2 comps 3 comps 4 comps 5 comps 6 comps 7 comps 8 comps 9 comps

CV 3873 3840 2024 2036 1707 1583 1581 1569 1543 1496

adjCV 3873 3840 2022 2038 1623 1577 1578 1570 1539 1493

10 comps 11 comps 12 comps 13 comps 14 comps 15 comps 16 comps 17 comps

CV 1493 1496 1497 1503 1504 1443 1159 1125

adjCV 1490 1494 1494 1501 1501 1425 1153 1119

TRAINING: % variance explained

1 comps 2 comps 3 comps 4 comps 5 comps 6 comps 7 comps 8 comps 9 comps 10 comps 11 comps

X 31.670 57.30 64.30 69.90 75.39 80.38 83.99 87.40 90.50 92.91 95.01

Apps 2.316 73.06 73.07 82.08 84.08 84.11 84.32 85.18 85.88 86.06 86.06

12 comps 13 comps 14 comps 15 comps 16 comps 17 comps

X 96.81 97.9 98.75 99.36 99.84 100.00

Apps 86.10 86.1 86.13 90.32 92.52 92.92

> validationplot(pcr.fit, val.type="MSEP")

> pcr.fit <- pcr(Apps~., data=train.set, scale=TRUE, ncomp=5)

> pcr.pred <- predict(pcr.fit, x.test, ncomps=5)

> pcr.msep <- mean((pcr.pred-y.test)^2)

>

> pcr.fit <- pcr(Apps~., data=College, scale=TRUE)

> summary(pcr.fit)

Data: X dimension: 777 17

Y dimension: 777 1

Fit method: svdpc

Number of components considered: 17

TRAINING: % variance explained

1 comps 2 comps 3 comps 4 comps 5 comps 6 comps 7 comps 8 comps 9 comps 10 comps 11 comps

X 31.670 57.30 64.30 69.90 75.39 80.38 83.99 87.40 90.50 92.91 95.01

Apps 2.316 73.06 73.07 82.08 84.08 84.11 84.32 85.18 85.88 86.06 86.06

12 comps 13 comps 14 comps 15 comps 16 comps 17 comps

X 96.81 97.9 98.75 99.36 99.84 100.00

Apps 86.10 86.1 86.13 90.32 92.52 92.92

>

>

>

> pls.fit <- plsr(Apps~., data=College, scale=TRUE, validation="CV")

> summary(pls.fit)

Data: X dimension: 777 17

Y dimension: 777 1

Fit method: kernelpls

Number of components considered: 17

VALIDATION: RMSEP

Cross-validated using 10 random segments.

(Intercept) 1 comps 2 comps 3 comps 4 comps 5 comps 6 comps 7 comps 8 comps 9 comps

CV 3873 1838 1533 1421 1305 1154 1139 1133 1127 1127

adjCV 3873 1837 1531 1418 1295 1139 1132 1127 1122 1122

10 comps 11 comps 12 comps 13 comps 14 comps 15 comps 16 comps 17 comps

CV 1126 1126 1126 1125 1125 1125 1125 1125

adjCV 1121 1121 1121 1120 1119 1119 1119 1119

TRAINING: % variance explained

1 comps 2 comps 3 comps 4 comps 5 comps 6 comps 7 comps 8 comps 9 comps 10 comps 11 comps

X 25.76 40.33 62.59 64.97 66.87 71.33 75.39 79.37 82.36 85.04 87.92

Apps 78.01 85.14 87.67 90.73 92.63 92.72 92.77 92.82 92.87 92.89 92.90

12 comps 13 comps 14 comps 15 comps 16 comps 17 comps

X 90.65 92.69 95.50 96.87 98.65 100.00

Apps 92.91 92.92 92.92 92.92 92.92 92.92

> validationplot(pls.fit, val.type="MSEP")

>

> pls.fit <- plsr(Apps~., data=train.set, scale=TRUE, ncomp=5)

> pls.pred <- predict(pls.fit, x.test, ncomp=5)

>

> pls.msep <- mean((pls.pred-y.test)^2)

>

> pls.fit <- plsr(Apps~., data=College, scale=TRUE, ncomp=5)

> summary(pls.fit)

Data: X dimension: 777 17

Y dimension: 777 1

Fit method: kernelpls

Number of components considered: 5

TRAINING: % variance explained

1 comps 2 comps 3 comps 4 comps 5 comps

X 25.76 40.33 62.59 64.97 66.87

Apps 78.01 85.14 87.67 90.73 92.63

>

> msep <- list(sub.msep, ridge.msep, pls.msep, pcr.msep)

> best.method <- which.min(msep)

> best.method

[1] 1

> # 1 corresponds to best subset selection method